# Asymptotics of stochastic learning in structured networks

# Asymptotics of stochastic learning in structured networks

Asymptotics of stochastic learning in structured networks

A catalogue record is available from the Eindhoven University of Technology Library
ISBN: 978-90-386-5725-7

Cover design by: © Roman Bas Cerdà, 2023
Printed by: ProefschriftMaken || www.proefschriftmaken.nl

Printed in the Netherlands

# Asymptotics of stochastic learning in structured networks

PROEFSCHRIFT

ter verkijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr. S. K. Lenaerts, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op maandag 15 mei 2023 om 16:00 uur

door

Albert Senén Cerdà

geboren te Benicarló, Spanje

Dit proefschrift is goedgekeurd door de promotoren. De samenstelling van de promotiecommissie is als volgt:

| | |
|---|---|
| Voorzitter: | prof.dr. E. R. van den Heuvel |
| Promotor: | prof.dr.ir. S. C. Borst |
| Co-promotor: | dr.ir. J. Sanders |
| Leden: | dr.habil. C. de Campos |
| | prof.dr. N. V. Litvak |
| | prof.dr.ir. P. F. A. Van Mieghem (Technische Universiteit Delft) |
| | dr. V. Gupta (University of Chicago, USA) |
| | prof.dr. A. J. Schmidt–Hieber (Universiteit Twente) |

*Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de Technische Universiteit Eindhoven Gedragscode Wetenschapsbeoefening.*

# Acknowledgments

This dissertation that you now have in your hands or that you are reading on your screen is the result of a journey of discovery—perhaps also of self-discovery—that started some years ago. And like any eventful journey, at the end there are people involved that have helped with its progression and without whom this dissertation would not be possible. Now, at the end and looking back, I would like to acknowledge them.

First, I would like to thank my supervisor Jaron for his invaluable advice and guidance throughout my time as a doctoral student. He has helped me grow as a researcher, enabling me to finish this dissertation. I especially appreciate when he invited me to join TU/e together with him, and I am grateful for his support and flexibility, particularly during the pandemic. Besides his technical expertise, I have come to admire his ability to methodically drive projects to success while also being open to new ideas; both in the role of supervisor and co-author. I sincerely hope that we can collaborate in promising research in the future.

The research projects could not have been finished also without other brilliant co-authors: Mark Peletier, Jim Portegies, Oxana Manita, Alexander Van Werde, Gianluca Kosmella, Céline Comte and Matthieu Jonckheere. It has been a great pleasure and an enriching experience to work with all of you, and I hope the occasion to work together arises again. I particularly would like to thank Matthieu for hosting me in the *Laboratoire d'analyse et architecture des Systèmes* in Toulouse for three months that were very exciting, despite my initial personal circumstances.

I would also like to thank my promotor Sem for his excellent support, availability and expert advice in matters related to research, career and beyond; he has helped me at many critical moments. His sharp and useful comments and suggestions have always been of great value throughout my time at TU/e. From my time at TU Delft, I also thank Piet Van Mieghem for his support and direct but honest style.

I am fortunate of having been part of the SPOR group at TU/e which would not exist without great staff members, including Maria, Rui, Bert, Onno, Jacques, Pim, Stella, Marko and Remco among others. They have always been approachable and willing to share their experience and insight with younger people. I particularly found the conversations at lunchtime a source for the know-how of the applied math researcher.

My current and former colleagues have also been very helpful in creating a great environment to conduct research. Among them, I can thank Qiang, Bastian, Richard, Ellen, Marta, Kay, Youri, Mayank, Joost, Rik, Martijn, Rens, Tom, Mark, Alexander, Gianluca, Tim, Thomas, Sanne, Neeladri, Purva, Elene, Noela, Benoît, Wessel as well as others that I hope can forgive me for not including them here. Special thanks go to my office colleagues Dennis, Rowel, Ivo and Diego for the many coffees, insights, jokes,

and nuances that we have had together and shared. I am also grateful to Chantal, Ellen, Patty and Marianne for helping me with the planning of the defense and making event organization go smoothly, both as organizer and participant. I also would like to thank Miguel del Álamo for the many interesting discussions that we have had throughout the years and for his useful suggestions during the drafting of this thesis.

I am very grateful to the defense committee members, Johannes Schmidt–Hieber, Varun Gupta, Nelly Litvak, Piet Van Mieghem, and Cassio de Campos for taking the time to read my thesis and for participating in the defense ceremony.

Finally, I would like to thank my family, and especially my parents, Patri and Rosa, and brother, Víctor, for supporting me unconditionally and for helping me pursue my interests despite the uncertainty and my own doubts. I also thank Roman for helping me with the cover of the thesis. My grandparents would have enjoyed seeing me with this finished dissertation, and so I partly dedicate it to them. *És per a vosaltres.*

My last acknowledgment is for Olatz, my dear love, who these last years has made me realize that sometimes life can be quite like solving problems in mathematics. Life imitates art, or so they say.

*Albert Senén Cerdà,*
*April 2023.*

# Contents

# Chapter 1

# Introduction

Artificial intelligence (AI) technologies based on machine learning algorithms are viewed to have the potential to solve major global challenges. Recent estimates suggest that AI technologies can yield an increase of up to 14% of the global GDP, or around US $15 trillion, by 2030 [53, 45]. Applications in finance, medicine, education, industry as well as in governmental organizations are becoming more common and private investment in AI during 2022 alone totaled around US $90 billion [10]. The end-goal is to create systems which are sufficiently sophisticated to automate general decision-making.

Networks play directly or indirectly a critical role in large-scale machine learning algorithms. A *network* (or graph in this manuscript) consists of a set of *vertices*, and connecting these vertices there are *edges* that model pairwise binary relations. Vertices can represent, for example, different people in a social network, while edges can be friendships or professional connections. A graph is the simplest model to encode information about relationships between individual elements in a system, and there is a large field studying graphs as static objects. In applications, however, there is often an underlying process—usually stochastic—that is based on a network structure whose connections are also reciprocally altered by the process. In the social network example, a user will interact and share information with close contacts, and with this information, his/her contact network will also change when meeting new people. From their use, we can distinguish two essentially different types of networks in machine learning algorithms.

On the one hand, a network can be purposefully coupled to a stochastic process from real-world phenomena to capture and encode its features. In Neural Networks (NNs), for example, functions representable by layered graphs with weighted edges emulate the brain with neurons and axons as vertices and edges respectively. During training, NNs change their weights using samples in order to learn statistical relationships in the data. After training with a large number of samples, they become capable of predicting new datapoints. As it turns out, the predictive ability of such networks depends strongly on the network characteristics and the training rule used. Such structuring of the network to encode information about the process—data in this example—is usually referred to as "training of the network". This is usually the setting of *supervised* machine learning.

On the other hand, structured networks can simplify the learning process. A common example occurs when we have data in a high-dimensional space that actually has a simple lower-dimensional description. In community detection, for example, we aim to detect sets of points in the data that form communities and share some property. Here, we understand “property” as a feature that is mostly common among members of the same community. For example, in a social network, communities can be defined by hubs of people that form cliques or that share a similar number of contacts. A structured network is used to learn these communities by finding sets of datapoints that match a prescribed network structure. This is usually the setting of *unsupervised* machine learning.

The workhorse of machine learning is stochastic optimization. Its success relies upon the fact that a learning problem in a random environment can be translated into an optimization problem with stochastic elements. Learning algorithms based on stochastic optimization have immediate application in settings where randomness of data generates uncertainty. Perhaps the most famous example is translating the problem of learning a distribution with a finite number of samples into the problem of minimizing a function with random coefficients, which we will also explain in Section 1.3. Many learning problems can be expressed in this manner and examples can be found in both supervised, unsupervised, as well as reinforcement learning, where an agent acts upon an environment in order to learn how to maximize a received reward.

In this thesis we will study two specific algorithms that are used in the supervised and unsupervised settings, respectively. The first algorithm is *dropout* [117]. In the training of NNs, the network may learn just the samples provided for training without being able to predict new examples well. This phenomenon is called *overfitting*. *Dropout* is a technique to avoid overfitting during training of NNs. While training occurs, dropout changes the structure of the NN stochastically and there is a parameter determining the stochastic change of the network: the *dropout* probability. Dropout [117] was introduced by Krizhevsky, Sutskever and Hinton to train the winning Convolutional Neural Network (CNN)—a type of NN used to learn images—for the ImageNet–2012 (ILSVRC–2012) competition [118], which kick-started the interest in deep learning. Despite its ample use in data science, the understanding of the statistical and convergence properties of this algorithm has remained limited and the choice of its parameters, such as the dropout probability, is usually left to trial and error.

We will use stochastic optimization techniques to investigate the convergence and approximation properties of *dropout* and its variants during the training of NNs. Our contributions with respect to dropout in this thesis are as follows:

- ❖ We establish convergence guarantees (Chapter 2), and analyze how the convergence rate depends on the choice of parameters, like depth and width of the NN, as well as the *dropout* probability (Chapters 3 and 4). We provide explicit convergence rates for some models that depend on the NN architecture and the *dropout* probability. Moreover, we compare the theoretical results with simulations, and explain their consequences.

- ❖ We investigate the approximation properties of NNs with the randomness of dropout, and show that they can approximate functions arbitrarily well, even in the presence of this additional randomness (Chapter 4).

The second model we analyze is inspired by community detection. We study Block Markov Chains (BMCs), a class of relatively new models for clusters in sequential data. Clusters in a BMC are modeled within a random trajectory of states. The random transitions from one state to another in a BMC depend only on the clusters that these states belong to. Thus, in the random trajectory, there are *dependencies* between transitions through time that are cluster-dependent. The goal is then to infer the underlying cluster structure just from the random trajectory of states. This is possible by knowing that the transition dynamics are of low rank and only depend on the clusters.

There is an algorithm with theoretical guarantees for exact recovery of the clusters in BMCs [27]. For the provable guarantees of the algorithm, a precise description of the estimation error of the dynamics is required in one of its steps. The observed trajectory is recorded with a random matrix whose size is the number of different states, and which encodes the transition dynamics. This matrix is then used to recover the clusters, and as it turns out, the order of the spectrum of this matrix determines part of the estimation error that the cluster algorithm makes. Our contributions with respect to BMCs in this thesis are:

- ❖ We use random matrix theory to study the spectral error of BMCs (Chapter 5). This involves spectral concentration of certain matrices with dependent entries coming from a BMC. We obtain order-sharp bounds for the spectral error of BMCs when the observed trajectory is long, and short (sparse regime). Our results improve the estimates of [27] and characterize the order of the spectral error in BMCs.

- ❖ We test the BMC clustering algorithm in real-world sequential data from finance, genetics, and geography (Chapter 6). Concretely, we evaluate the spectral error from the clusters recovered from these datasets and numerically assess the robustness of the BMC model by using a perturbative approach. Finally, we determine if the model is appropriate to describe the observed transition dynamics in the data by conducting model selection.

Dropout and clustering in BMCs pertain to fundamentally different problems. One is related to stochastic optimization in NNs and the other to community detection. Nonetheless, both algorithms share the underlying idea of using structured networks to represent and constrain the learning process.

This thesis has two parts. The first part contains the contributions to the research on dropout and consists of Chapters 2, 3 and 4. The second part of the thesis covers the research on the BMC model and consists of Chapters 5 and 6.

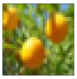
The remainder of this introduction has the following structure. We will first introduce *dropout* within the context of stochastic learning algorithms for NNs in Sections 1.3 and 1.4. In Sections 1.5 to 1.7 we will describe the BMC model together with its applications. The results of this thesis for these two parts are summarized in Sections 1.4 and 1.7, respectively. Finally, in Section 1.8 the reader can find additional literature and positioning of the results of this thesis.

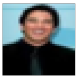
# Part I: Dropout in NNs

In this first part we consider dropout from the stochastic optimization perspective and aim to understand its convergence properties. We investigate models for NNs for which precise theoretical results can be derived, which we later complement with simulations. Furthermore, we analyze the approximation capability of random NNs that have dropout as the source of randomness.

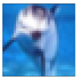
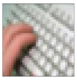
## 1.1 Background of Part I

NNs have found ample use in present-day big data applications. They are commonly used for supervised learning, that is, they learn first from samples that contain input and output pairs and are later tasked with predicting an output if a new input is provided. For example, in classification tasks, the input space $\mathcal{X}$ can be the space of images and the output space $\mathcal{Y}$ can be the space of labels. In this example, we are given samples $(x,y) \in \mathcal{X} \times \mathcal{Y}$ of the type[1]

$\Bigg($  , $\texttt{orange}\Bigg)$, $\Bigg($ , $\texttt{dolphin}\Bigg)$, $\Bigg($ , $\texttt{keyboard}\Bigg)$,

$\Bigg($ , $\texttt{orange}\Bigg)$, $\Bigg($  , $\texttt{man}\Bigg)$, $\Bigg($ , $\texttt{shark}\Bigg), \cdots$ (1.1)

and the task of supervised learning is to find a map $\theta : \mathcal{X} \to \mathcal{Y}$ using the previous labeled images such that $\theta$ can also predict the labels of *unseen* images. You may think of these as a list of pairs such as

$\Bigg($ $, ?\Bigg)$, $\Bigg($ $, ?\Bigg)$, $\Bigg($  $, ?\Bigg)$, $\Bigg($  $, ?\Bigg)$, $\Bigg($ $, ?\Bigg)$. (1.2)

When $\theta$ can correctly predict the labels of new unlabeled images, such as those in (1.2), we say that $\theta$ has good *generalization* properties. On the contrary, if $\theta$ can only correctly predict labels of the previous labeled images in (1.1), we say that $\theta$ *overfits* and does not generalize well.

In deep learning with NNs, the map $\theta$ is usually encoded by a NN. A typical NN with $L$ layers has a structure as shown in Figure 1.1, where an output vector of layer $i-1$ is fed into the next layer $i$. First, it is multiplied by a weight matrix $W_i$ and added to a vector or bias $b_i$, and second, a componentwise nonlinear function $\sigma : \mathbb{R} \to \mathbb{R}$ is applied to the resulting vector. While there are many variants of NNs, we will consider only feed-forward NNs as depicted in Figure 1.1 for the sake of simplicity.

The problem in supervised learning consists of finding the right weights $W = (W_i, b_i)_{i=1}^{L}$ for the NN by using available training data so that the prediction error becomes small. In our example with pictures from (1.1) and (1.2), we may consider e.g., the $0-1$ loss for labels, that is, for two labels $a, b$,

$$l(a,b) = \begin{cases} 0 \text{ if } a = b, \\ 1 \text{ if } a \neq b. \end{cases} \tag{1.3}$$

If we denote a function that uses the NN in (1.1) with its weights $W$ by $\theta_W$, then the empirical error with the first three images in (1.1) will be

$f(W) = l\Bigg(\theta_W\Bigg($ $\Bigg), \texttt{orange}\Bigg) + l\Bigg(\theta_W\Bigg($ $\Bigg), \texttt{dolphin}\Bigg) + l\Bigg(\theta_W\Bigg($ $\Bigg), \texttt{keyboard}\Bigg)$.

[1] These images belong to the Canadian Institute For Advanced Research (CIFAR)-100 dataset [128], which we will also use in Chapters 2 and 3.

$$
\begin{aligned}
a_0 &= x \in \mathcal{X}, \\
&\cdots \\
a_i &= \sigma(W_i a_{i-1} + b_i), \\
&\cdots \\
a_L &= y \in \mathcal{Y}.
\end{aligned}
$$

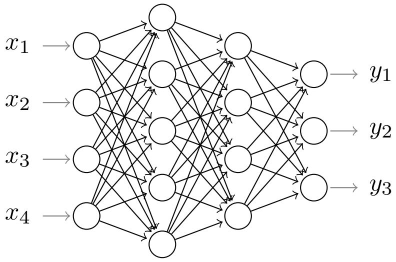


*Figure 1.1: A NN-graph on the right and its computational analog on the left. Each edge $(i,j)$ in this graph represents a weight $W_{i,j}$ which multiplies the output of a node or neuron. After each node sums the incoming contributions, a nonlinear function $\sigma$ is applied. In this example, the input is a vector $x \in \mathbb{R}^4$, and the output a vector $y \in \mathbb{R}^3$. After each layer $i$, the weight multiplication is equivalent to a multiplication by a weight matrix $W_i$ as shown on the left. A vector of biases $b_i$ can also be added after weight multiplication. The number of layers in this case is $L = 3$.*

The common way to determine the weights of a NN is by using gradient descent or its stochastic version, Stochastic Gradient Descent (SGD), to minimize this empirical error. In gradient descent, minimization of a function is accomplished by initializing the parameters—the weights $W$ of a network in our case—and updating them iteratively in the direction that the function decreases locally, namely in the (negative) gradient direction if $f$ is differentiable. Thus, at step $t$, we update recursively a vector $w^{[t]}$ containing the weights with the gradient of $f$ at $w^{[t]}$, denoted by $\nabla f(w^{[t]})$, with some step size $\alpha^{\{t\}} > 0$ controlling the update speed, that is

$$
w^{[t+1]} = w^{[t]} - \alpha^{\{t\}} \nabla f(w^{[t]}). \tag{1.4}
$$

In our previous example on a classification task with images and labels from (1.1), the hope is that gradient descent will converge to weights $w^*$ that also make the prediction or test error of the labels in (1.2) small.

One of the early suggested issues when training NNs was that weights of individual neurons could be very correlated with other neurons nearby. Predictions using these neurons would thus become sensitive to out-of-sample data and make a NN overfit. A proposed solution was to 'drop' neurons at random during training. In this way, the training would spread correlations across the NN. This motivation to avoid overfitting by dropping nodes of NNs was the origin of *dropout* [117]. It was successfully used for classification of images in *Alex–Net*, the NN of the winning submission in the ImageNet–2012 competition [118], which encouraged the use of deep NNs for label prediction and sparked the interest in deep learning. An empirical example of the effect of dropout training can be seen in Figure 1.2.

The original dropout algorithm works by dropping nodes of the NN independently at random. At each step of SGD, nodes are selected randomly with probability $1-p$ and temporarily dropped from the network (meaning that their associated weights are set to zero for that step). Here, we denote by $p \in [0,1]$ the probability of a node to remain. Then, a gradient update is computed using the remaining subnetwork in the usual manner. The

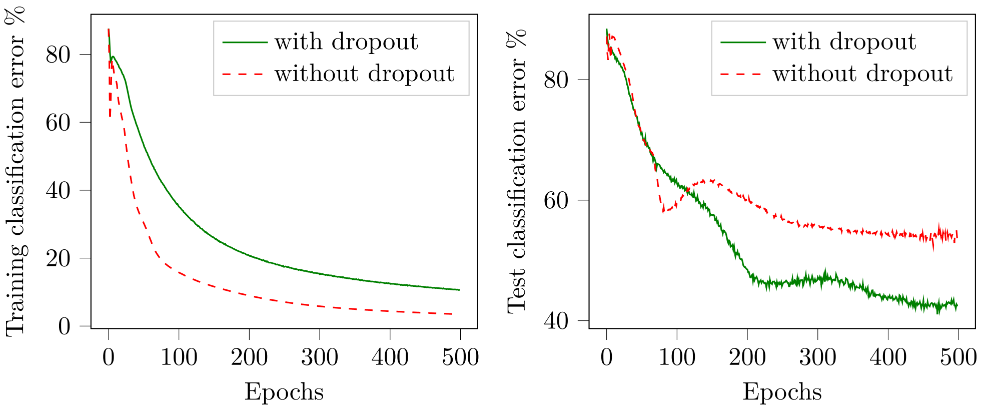
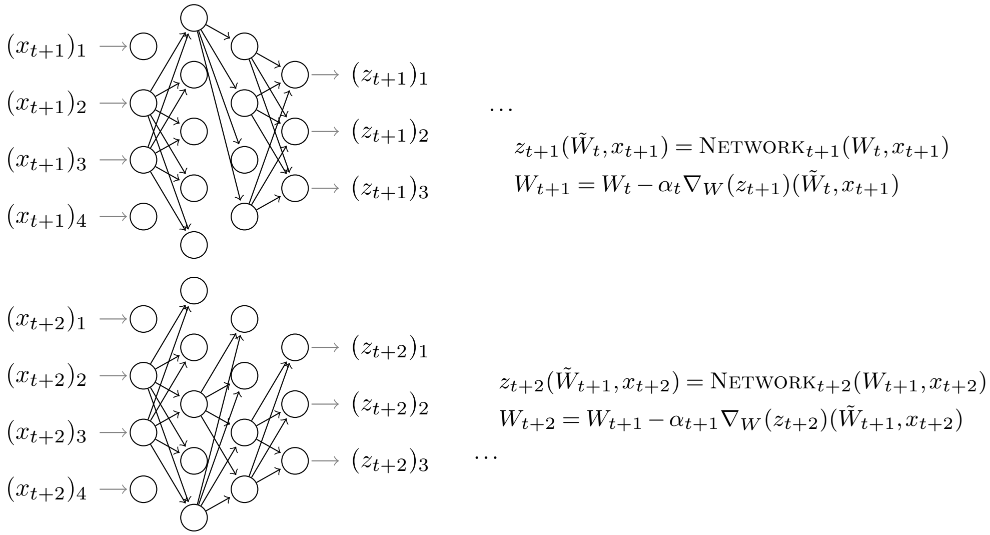


*Figure 1.2: Example of the effect of dropout in the classification error of images depending on the number of epochs of training with SGD. Note that with dropout, while the error in the training set is larger and more time to converge is needed, in the test set the error is actually smaller.*

update direction is the same then *as if* the network structure had none of the dropped nodes. Figure 1.3 shows a schematic depiction of the procedure.



$$z_{t+1}(\tilde{W}_t, x_{t+1}) = \textsc{Network}_{t+1}(W_t, x_{t+1})$$
$$W_{t+1} = W_t - \alpha_t \nabla_W (z_{t+1})(\tilde{W}_t, x_{t+1})$$

$$z_{t+2}(\tilde{W}_{t+1}, x_{t+2}) = \textsc{Network}_{t+2}(W_{t+1}, x_{t+2})$$
$$W_{t+2} = W_{t+1} - \alpha_{t+1} \nabla_W (z_{t+2})(\tilde{W}_{t+1}, x_{t+2})$$

*Figure 1.3: Example of using dropout SGD on a NN. For each step $t$, the gradient is computed assuming that only the weights present in the network—denoted by $\tilde{W}_t$—are variables at that step $t$. Note that only the weights $\tilde{W}_t$ are also updated since the gradient for the dropped edges is zero.*

Intuitively, by training a different network at each step, dropout adds redundancy in the weights of the NN. One can expect in this case that the network learns only the general features of the training data instead of specific images. Thus, it may be able to generalize better. One can also anticipate, however, that the additional stochasticity introduced by dropout during training, namely the fact that we choose a different subnetwork to update every step, will come at the cost of a lower convergence rate.

From a practical point of view, we would like to know what the cost of training NNs with dropout is, how it depends on the parameters of the network like width and depth, and most importantly how it depends on the dropout probability. We remark that the dropout probability should also tune the penalization on overfitting. We can therefore ask the following fundamental questions:

(1) Is training of NNs with dropout or its variants guaranteed to converge?

(2) What do the weights converge to when training with dropout?

(3) At what rate does a training algorithm using dropout converge?

Regarding the study of dropout as a stochastic optimization algorithm, there are surprisingly few results in the literature that tackle questions (1)–(2) and especially (3). For this latter question in particular, apart from the results presented in this thesis, we can only mention [25], which we review in Section 1.8 at the end of this chapter.

Looking at dropout in a more abstract setting, we can similarly consider a class of random networks that use dropout as a source of randomness. The study of random NNs is motivated by the need to make training less complex. In particular, in NNs with two layers, if only one layer needs to be trained while the other is random and fixed, the optimization problem can become convex [130]. In this case, more efficient optimization methods for training than SGD exist. Hence, random networks may yield an alternative pathway to training NNs in a cost-efficient manner. Determining if random NNs can still approximate functions is a first step in this direction. If we consider networks with dropout as a class of random NNs, we may ask if they still can approximate the same functions as a NN, despite the randomness in the network structure.

(4) Do NNs with dropout still have a universal approximation property?

The focus in the first part of the thesis is on questions (1)–(4) which are partially answered in Chapters 2, 3, and 4 respectively.

We will briefly introduce basic concepts from NNs and learning in Sections 1.2 and 1.3 respectively before giving an overview of the results in Section 1.4.

## 1.2 Neural networks

NNs as well as an efficient implementation for computing gradients of their functions with *backpropagation* have been known since the 1980s. The large-scale data availability and computational capacity have allowed NNs with up to billions of parameters to be trained and used in the last decade. They are useful in tasks ranging from computer vision [8, 48], scheduling [24], natural language processing, and generative models for text and images [19, 6] to reinforcement learning [94]. With enough parameters and the right architecture, NNs can approximate arbitrary functions and are therefore said to satisfy a universal approximation property [69, 156]. Many different activation functions have been used in NNs; to name a few, ReLU, Leaky ReLU, sigmoid, soft-max, GeLU, etc. Similarly, many tailor-made NN architectures have appeared that try to use the network architecture to better encode data for the target task. An example of a well-known CNN, which is commonly used for image classification, is depicted in Figure 1.4. We refer to Section 1.8 at the end of this chapter for futher references on other types of NNs.

Dropout, which we will study in the first part, exploits the stochastic properties of the training algorithm for NNs with SGD, which we describe in the next section in the context of learning.

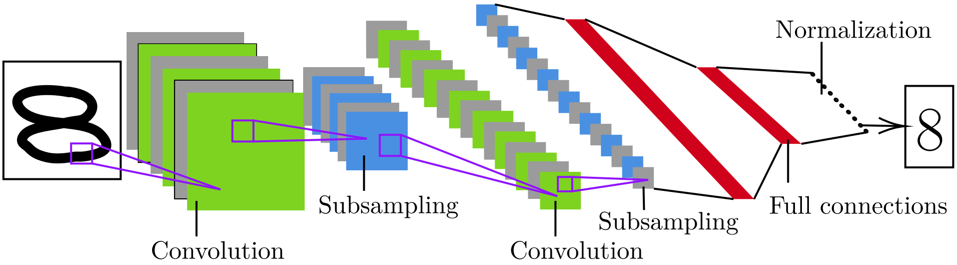


*Figure 1.4:* LeNet-5 *architecture of a CNN used for image classification of digits [145]. This NN architecture has several convolutional layers as well as fully-connected ones and can be trained to classify text as well as images. This network is the base architecture used for simulations in Chapters 2 and 3 that examine the convergence rate of SGD with dropout in deep and shallow networks.*

## 1.3 Stochastic gradient descent and learning

We now formalize the learning problem that we have introduced in Section 1.1. Suppose that we have a distribution $\mu$ with values on $\mathcal{X}\times\mathcal{Y}=\mathbb{R}^d\times\mathbb{R}$ for $d>1$ and suppose that we have $n$ i.i.d. samples $(x_1,y_1),\dots,(x_n,y_n)$ of $\mu$, which we will also refer to as datapoints. In function approximation, we would like to find a map $\theta:\mathcal{X}\to\mathcal{Y}$ such that $\theta(x)$ *approximates* the value of $y\in\mathcal{Y}$ for a given $x\in\mathcal{X}$ sampled from $\mu$.

In order to define what constitutes a good approximation, we define first a loss function $l:\mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$, for example, $l(a,b)=(a-b)^2$ or the 0–1 loss for labels that we have defined in (1.3). Then the loss for a datapoint $(x_i,y_i)$ is defined by $l(\theta(x_i),y_i)$ and we define the *empirical risk* of $\theta$ as

$$\hat{\mathcal{R}}_n(\theta)=\frac{1}{n}\sum_{i=1}^{n}l(\theta(x_i),y_i). \tag{1.5}$$

In learning we want to find a map $\theta$ that approximately captures the general features of the data ($\mu$ in this case). To do so we minimize (1.5) with $n$ datapoints available. However, we actually would like to find a minimizer of the true or test risk

$$\mathcal{R}(\theta)=\mathbb{E}_{(x,y)\sim\mu}(l(\theta(x),y)). \tag{1.6}$$

A common way to approach the learning problem is to choose a function class $\mathcal{F}$ for $\theta$ where maps are expressive enough to approximate the pairs $(x_i,y_i)_{i=1}^n$ and simple enough to not be able to exactly fit the data and thus overfit. In most cases, the function class is parametrized by some vector $w \in \Theta \subseteq \mathbb{R}^N$ and so $\mathcal{F} = \{\theta_w : w \in \Theta \subseteq \mathbb{R}^N\}$. In particular, the minimization problem in (1.5) becomes a problem of minimizing the function $f(w) = \hat{\mathcal{R}}_n(\theta_w)$.

A way to minimize a function $f(w)$, or in general to minimize other differentiable functions, is to use a descent scheme as described in (1.4). In common applications, however, the amount of available datapoints $n$ or the dimension $d$ of the input space $\mathcal{X}$ are large. Thus, computing $\nabla f(w^{[t]})$ exactly for each iteration is computationally costly. Finding a sufficiently good update at low computational cost is then beneficial in the trade-off between accuracy and complexity. SGD addresses this issue by using an unbiased estimator $g_t(w^{\{t\}})$ of $\nabla f(w^{\{t\}})$, which is enough to decrease the objective function in expectation. In particular, the update rule becomes

$$\begin{aligned} w^{[t+1]} &= w^{[t]} - \alpha^{\{t\}} g_{t+1}(w^{[t]}) \\ \nabla f(w^{[t]}) &= \mathbb{E}[g_{t+1}(w^{[t]}) | \mathcal{G}_t], \end{aligned} \tag{1.7}$$

where the expectation is taken conditional on all previous samples $g_i$ for $i = 1, \ldots, t$ and $w^{[0]}$. The most common choice for $g_{t+1}$ is to take uniformly at random a datapoint or a batch $n_b$ of them, where $n_b \ll n$. In the case of a single datapoint,

$$g_{t+1}(w^{[t]}) = \nabla_w l(\theta_{w^{[t]}}(x^{[t]}), y^{[t]}), \tag{1.8}$$

where the pairs $(x^{[t]}, y^{[t]})$ are sampled uniformly at random from the training dataset at each step $t$.

Under mild assumptions on the step sizes $\alpha^{\{t\}}$ and the variance and regularity of the gradient estimators $(g_t)_t$, the iterates $(w^{[t]})_t$ converge to a stationary point of the target function $f$ [171, 140, 127, 50, 132].

Abusing notation by denoting $\hat{R}_n(\theta_w) = \hat{R}_n(w)$, SGD is not guaranteed to converge to a neighborhood of a global minimum $w_n^* \in \text{argmin}_w(\hat{R}_n(w))$. This occurs, for example, in nonconvex functions that possess many local minima. If $w_{\text{SGD}}$ is the output of SGD, there can be a positive gap of the empirical risk minimizer compared to the optimum:

$$\hat{\mathcal{R}}_n(w_{\text{SGD}}) - \text{argmin}_w \hat{\mathcal{R}}_n(w) \geq 0. \tag{1.9}$$

Even in the case that the global optimum $w_n^*$ of $\hat{\mathcal{R}}_n$ is found, there is still a so-called generalization gap

$$\hat{\mathcal{R}}_n(w_n^*) - \mathcal{R}(w_n^*), \tag{1.10}$$

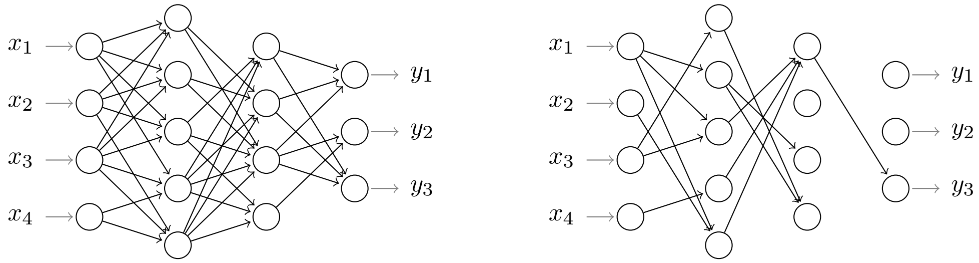
which will depend on the class $\mathcal{F}$, the distribution $\mu$ of data and the number of samples $n$. There is extensive work on studying the properties of the generalization gap in (1.10) from the statistical point of view for many function approximators, including NNs [28].

When the dimension of the parameter space $\dim(\Theta) = N$ is much larger than that of the number of available samples $n \ll N$, such as with NNs, overfitting of data can occur since the problem is usually highly underdetermined. A common way to avoid this issue and improve generalization is to try to penalize the complexity of the functions $\theta_w$ in the optimization problem. Adding a penalization is commonly referred to as *regularizing*. Dropout is an indirect technique that uses stochasticity in the NNs to regularize the type of weights that SGD finds. For more information about other regularization techniques, we refer to Section 1.8 at the end of this chapter.

## 1.4 Dropout and summary of results of Part I

The class of dropout algorithms that we will study consists in practice of multiplying weight matrices of the NN in each iteration componentwise by independently drawn random matrices with $\{0,1\}$-valued entries. The elements of these random matrices indicate during a training step whether each individual edge or node is filtered (0) or is not filtered (1). The resulting weight matrices are then used in the backpropagation algorithm for computing the gradient of a NN. While in the original *Dropout* algorithm in [117] only nodes from the network were dropped, several stochastic training algorithms that avoid overfitting in NNs have appeared since then (for example, *Dropconnect* [115], *Cutout* [64]). Figure 1.5 depicts a NN where we use *Dropconnect* and drop individual edges instead of nodes.

Mathematically, dropout turns the backpropagation algorithm into a step of an SGD algorithm in which the primary source of randomness is the NN's configuration. Under mild independence assumptions, dropout actually optimizes a risk function averaged over all possible NN configurations [110].



*(a) Case* $p = 0.7$. *(b) Case* $p = 0.25$.

*Figure 1.5:* (a,b) *Training step of dropconnect [115] on a NN with* $L = 3$ *layers. In this algorithm, every iteration, a random NN is first generated by removing each edge independently of all other edges with probability* $1-p \in (0,1]$. *The output of this random NN is then used to update all weights using backpropagation.*

Most theoretical focus has been on the regularization effects of dropout algorithms [117, 110, 114, 105, 100, 51, 60, 42, 26, 31]. For example, it is known that in NNs with linear activations—also equivalent to a matrix factorization problem—dropout induces a regularization on the nuclear norm of the weight matrices [63]. Thus, using dropout reduces the complexity of the class of estimators in matrix completion problems [12].

In this thesis we look instead at dropout from the stochastic optimization point of view and investigate the convergence properties of SGD with dropout. Dropout will yield an approximate minimizer $w_{\mathrm{SGD}}$ for a different empirical risk $\tilde{\mathcal{R}}_n$ than $\hat{\mathcal{R}}_n$ for which a similar gap to (1.9) can be expected. In this case, we expect that convergence guarantees and the convergence rate will depend on the characteristics of the NN and the dropout algorithm used. On a related note, we will also study the approximation properties of dropout and explore if there is still a universal approximation property for random NNs with the randomness of dropout. We present in the following three subsections the summary of results of this part.

### 1.4.1 Convergence guarantee

In Chapter 2, we will consider dropout from the stochastic approximation point of view and prove that dropout converges. In particular, we use a set of activation functions that can be bounded by polynomials and consider feed-forward NNs. We will prove—under some assumptions on the loss, moments of the data, and step sizes—that if we project the iterates of SGD with dropout onto a compact set, then the iterates converge to a stationary point of the Ordinary Differential Equation (ODE)

$$\frac{dW}{dt} = -\nabla \mathcal{D}(W), \tag{1.11}$$

where $\mathcal{D}$ is an expectation of the original risk over the dropout filters. The results hold not just for *dropout*, but for dropout-like filters that are not necessarily independent and can also depend on the data. This convergence guarantee shows that dropout converges and to which points it implicitly converges. Our first result in Chapter 2 will thus answer questions (1) and (2) listed before.

### 1.4.2 Convergence rate

The convergence guarantee result in Section 1.4.1 is a first result to understand dropout. In Chapters 2 and 3, we go further and study the convergence rate of dropout and its dependence on the distribution of the filter variables.

In Chapter 2, we examine first the convergence rate to reach $\epsilon$-stationarity on a generic nonconvex function with SGD when we use dropout filters in the stochastic update. Specifically, we obtain a decreasing bound for $\min_{t\in[T]} \mathbb{E}[\|\nabla \mathcal{D}(W^{[t]})\|_2^2]$ in the number of iterations $T$ of SGD, which also depends on the dropout probability $1-p$. With this result, we discuss how dropout changes the difficulty of finding stationary points.

In Chapter 2 we go one step further and study the convergence rate in arborescences—a toy model for deep NNs; see Figure 2.1c in Chapter 2—with linear activations, where the effects of dropout can already be observed. In particular, we will determine that in these networks, the convergence rate can decrease up to an exponential factor of $p^L$, where $L$ is the number of layers where dropout is used and $1-p$ is the dropout probability. Hence, dropout affects very strongly the convergence rate in this model. In Chapter 2 we also conduct experiments with realistic NNs and real datasets. In contrast, we do not experimentally observe an exponential decrease in wide NNs with two or three layers of dropout. We will also explain heuristically why this is the case.

The results in Chapter 2 refer to the convergence rate of dropout depending on the depth of the dropout layers of the NN. A related question concerns what the dependence on the width of the NN is. In Chapter 3, we will consider a simplified model of a NN: a shallow linear NN. We will use the ODE method from stochastic approximation to relate the iterates of SGD with dropout with the gradient flow of an ODE as in (1.11). We will prove that the expected convergence rate exponent $\omega$ of this gradient flow close to convergence to a minimum behaves as

$$\omega \approx \frac{p(1-p)}{fp+(1-p)}, \tag{1.12}$$

where $f$ is the width of the network. This result will also shed light onto the difficulty of finding a minimum with dropout.

Numerical simulation will show, moreover, that the rate in (1.12) qualitatively agrees with simulations. Interestingly, the inverse dependence on the width $f$ in (1.12) occurs only close to convergence. Far away the opposite occurs: overparametrization with large $f$ seems to favor convergence to a minimum in the first stages of Gradient Descent (GD) but not up-close. Together with the simulations these results may be translated to heuristics of what to expect when training with dropout. Combined, the results from Chapters 2 and 3 address question (3) with two different approaches.

### 1.4.3 Approximation properties

In Chapter 4, we will analyze dropout in NNs more abstractly and investigate the approximation properties of random NNs where the randomness comes from dropout.

Denote a feed-forward NN, $h : \mathbb{R}^d \to \mathbb{R}$, with random dropout filters $m$ by $h(w \odot m)$ for any $w \in \mathbb{R}^d$, where $\odot$ is the componentwise product. With $m$ as a random filter, $h$ will be a random function which we will call for now a dropout NN. In Chapter 4 we will prove that a class of dropout NNs, denoted by $\mathsf{DDNN}$, satisfies a universal approximation theorem. In particular, for any $g \in C([0,1])$ and any $\epsilon > 0$, we have that there exists $h \in \mathsf{DDNN}$ such that

$$\mathbb{E}_m[\|g(w) - h(w \odot m)\|_\infty] < \epsilon. \tag{1.13}$$

In Chapter 4 we will prove (1.13) under a variety of assumptions, like the choice of normed space, the filter distributions, and if we consider dropout also on the input layer, which requires special attention. In practice, the weights of the NN after training with dropout are usually scaled by the expectation of $m$ for prediction. We go one step further and prove furthermore that $\mathsf{DDNN}$ satisfies a universal approximation theorem also in this sense. Namely, in the notation of the previous example, there also exists $h \in \mathsf{DDNN}$ such that

$$\mathbb{E}_m[\|g(w) - h(w \odot m)\|_\infty] < \epsilon \quad \text{and} \quad \|g(w) - h(w \odot \mathbb{E}[m])\|_\infty < \epsilon. \tag{1.14}$$

The proof of such results is constructive and we provide concrete examples in the case of *dropout*.

# Part II: BMCs

In the second part of this thesis, we discuss Block Markov Chains (BMCs), a class of models for sequential data with clusters. Firstly, we investigate the spectral properties of BMCs, and secondly, we use real-world examples of sequential data to evaluate the clustering algorithm for BMCs.

## 1.5 Background of Part II

Discovering low-dimensional structures in data can improve the understanding of how the underlying data-generating process behaves. This is especially interesting for sequential data, which are ubiquitous. We introduce this topic with a concrete example from natural language processing used in machine learning.

*Example:* We have several texts extracted from newspapers and our task is to classify the texts by topic. By looking at keywords, one may be able to classify the texts well, but with many topics this may not be possible since there are too many words. We can try instead to find a lower-dimensional representation of a text by checking if it contains sets of words with similar meanings and by how they follow each other as strings of words. These sets of words—which we refer also as clusters or communities[2]—can have similar contexts as depicted in Figure 1.6. Once we have identified them, texts can be more easily classified using this low-dimensional latent structure [143]. The problem now is apparent: *Can we find these clusters of words by just observing a sufficiently long text?*

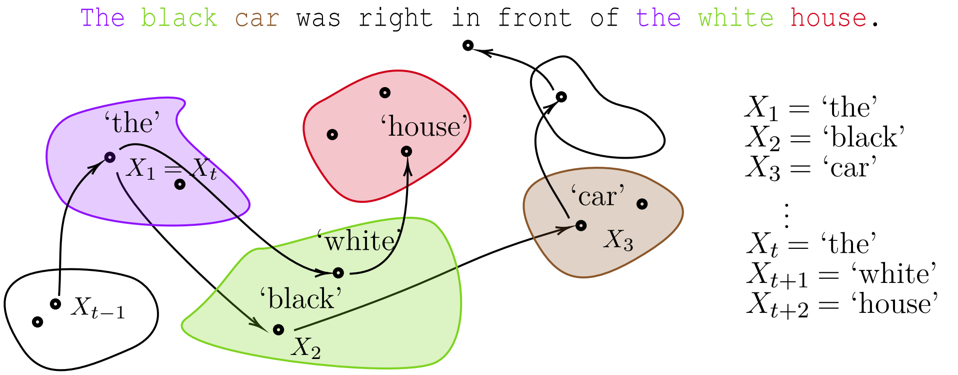


*Figure 1.6: Text can be modeled as a sequence of words or states (circles in the picture) in a chain $X_1, \dots, X_t, \dots$. Each state $X_t$ belongs to a cluster, like the green cluster which contains* 'colors'*, but which we do not know a priori.*

As hinted by the example in Figure 1.6, there are many potential applications of community detection in the sequential setting. For example, in streaming services of movies or music, future recommendations for a user are generated by looking at his/her past sequence of consumed media genres or categories. Especially interesting is its potential use in model-based reinforcement learning, where an agent tries to learn the state transition dynamics of its environment with sequences of states and rewards [4, 30]. In Chapter 6 we will see other examples. These previous applications rely on the same idea: *Can we detect and recover communities from a trajectory in sequential data?*

To proceed, we introduce modeling assumptions in order to formally describe communities and make the problem tractable. In our setting, there are $n$ states or nodes with labels in $[n] = \{1, \dots, n\}$ and we observe a trajectory of length $T_n$ of connected nodes $X_1, \cdots, X_t, \cdots, X_{T_n}$. We will first assume that the probability of seeing a node $X_{t+1}$ at time $t+1$ depends only on the immediate previous node $X_t$ in the trajectory, that is, the process $\{X_t\}_{t \geq 0}$ satisfies the Markov property

$$\mathbb{P}[X_{t+1} = j \mid X_t = i, X_{t-1} = i_{t-1}, \dots X_0 = i_0] = \mathbb{P}[X_{t+1} = j \mid X_t = i], \tag{1.15}$$

for all nodes $j, i, i_{t-1}, \dots, i_0 \in [n]$.

To define the communities, we will first assume that there are $K \ll n$ communities and each node $j \in [n]$ has a label in $[K]$ which determines the community it belongs to. Hence, we assume that there exists a true cluster assignment map $\nu_n : [n] \to [K]$ which

[2] In this thesis we will make no difference between communities and clusters.

encodes the clusters. All we observe is a trajectory of the process $\{X_t\}_{t\geq 0}$, which is driven by dynamics that are based on the community structure. The first dynamical assumption about the process is that if $X_t$ is a state in a cluster $k \in [K]$, then the process transitions to a cluster $l \in [K]$ with some probability $q_{k,l}$. The probabilities $q_{k,l} \in [0,1]$, which we call *cluster transition probabilities*, thus satisfy $\sum_{l=1}^{K} q_{k,l} = 1$ for $k \in [K]$. In the example with music streaming recommendations, if a user has listened to rock, then this assumption is equivalent to assuming that there are certain probabilities of choosing a song in a different genre, like pop or classical music, without specifying the song in that genre first. The second dynamical assumption is that if at time $t+1$ a transition to cluster $l \in [K]$ occurs, the trajectory will transition to a state $X_{t+1}$ chosen uniformly at random in this cluster $l$.

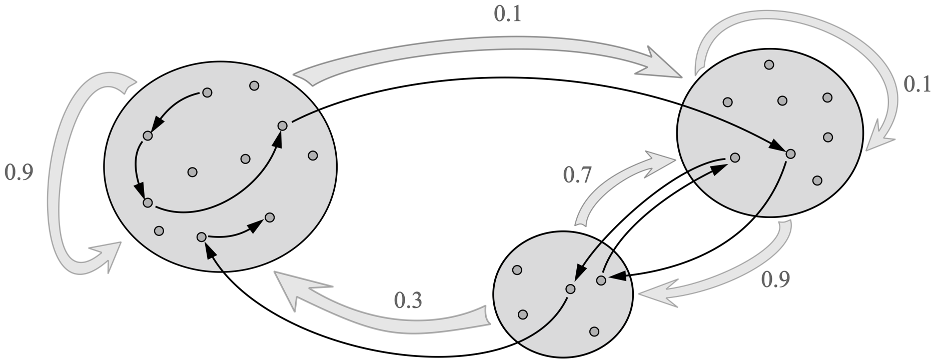


*Figure 1.7: A visualization of a BMC with $K = 3$ clusters and $q = [[0.9,0.1,0],[0,0.1,0.9],[0.3,0.7,0]]$. The thick arrows visualize the cluster transition probabilities $q_{k,l}$, while the thin arrows visualize the transitions of a sample path $\{X_t\}_{t\geq 0}$. Figure courtesy of [2].*

These two dynamical assumptions are the basis of the BMC model for communities in sequential data. An example of a BMC with $K = 3$ is depicted in Figure 1.7. The two dynamical assumptions imply that the transition from a node $i \in [n]$ to a node $j \in [n]$ occurs with the following *state transition probability*

$$P_{i,j} := \mathbb{P}[X_{t+1} = j \mid X_t = i] = \frac{q_{\nu_n(i),\nu_n(j)}}{\text{Size of cluster } \nu_n(j)}. \tag{1.16}$$

We consider disjoint clusters in the model, which we denote by $\mathcal{V}_l$ for $l \in [K]$. Thus, we will partition the state space $[n]$ of a Markov chain. The process $\{\nu_n(X_t)\}_{t\geq 0}$ given by the sequence of clusters of each state together with the size of the clusters fully capture the dynamics of the trajectory $\{X_t\}_{t\geq 0}$.

## Spectral error in BMCs

It was shown by Sanders, Proutière and Yun in [27] that community detection in BMCs and recovery of clusters is possible under some assumptions on the cluster transition probabilities, the size of the clusters $|\mathcal{V}_l|_{l\in[K]}$ and the length $T_n$ of the observed trajectory. The clustering algorithm consists of two steps. The first one yields a guess $\hat{\nu}_n$ for the true cluster assignment map $\nu_n$. The second step exploits the structure of the BMC to iteratively improve the cluster assignment based on the observed trajectory.

The guess $\hat{\nu}_n$ for the true assignment map $\nu_n$ in the first step is based on spectral clustering, where a matrix encoding the transitions is expected to have low rank. In the case of a BMC we can define an empirical counting matrix

$$\hat{N}_{i,j} = \sum_{t=1}^{T_n} \mathbb{1}[X_{t+1} = j, X_t = i], \tag{1.17}$$

which will encode the number of transitions seen in a trajectory of length $T_n$.

The idea of using spectral clustering is that under the BMC assumption $N = \mathbb{E}[\hat{N}]$ has rank $K$ due to the $K$-cluster structure. Since we do not have access to $N$, we can use the empirical approximation $\hat{N}$ and examine the rank of the matrix $\hat{N}$ which we expect to be approximately $K$. This notion of approximation is characterized by the spectrum of $\hat{N}$ having a certain size and more importantly the spectrum of the difference $\hat{N} - N$. A measure of this difference is the spectral norm or *spectral error* $\|\hat{N} - N\|$, defined as

$$\|\hat{N} - N\| = \sup_{\|x\|_2 = 1} \|(\hat{N} - N)x\|_2. \tag{1.18}$$

The spectral error thus characterizes part of the approximation error that we make when using $\hat{N}$ to estimate the clusters that are encoded in $N$.

In [27] it is proven that if $T_n = \omega(n)$, then there exist $c > 0$ and $\hat{N}_\Gamma$, a regularized matrix of $\hat{N}$ such that asymptotically in $n$, with high probability,

$$\|\hat{N}_\Gamma - N\| < c\sqrt{\frac{T_n}{n}\log\left(\frac{T_n}{n}\right)}. \tag{1.19}$$

The bound in (1.19) is sufficient to prove that detection and recovery are possible in BMCs *asymptotically.* However, when comparing to other models from graph theory, this bound was suspected not to be sharp in terms of the ratio $T_n/n$.

Knowing the correct order of the spectral norm in BMCs will determine the limits of the spectral clustering and shed light on the effect of dependencies in the BMC model. Hence we ask the following questions:

(5) Is the upper bound for the spectral error of BMCs in (1.19) sharp?

(6) Is there a matching lower bound to the spectral error in BMCs?

From the technical perspective, questions (5)–(6) can also be asked from the study of random matrices (namely, the random matrix $\hat{N}$) with dependencies coming from a Markov chain. In particular, methods that may work for matrices with independent entries are not readily applicable in the setting of BMCs.

## Experimental evaluation of BMCs

While BMCs provide an appealing model for communities in sequential data, as shown with the example of Figure 1.6, it is also interesting to examine whether they can be used to capture hidden structures in real-life data sets. This question had not yet received attention in the literature.

One may think that, perhaps due to the simple assumptions of the BMC model, finding interesting clusters with the clustering algorithm for BMCs in real data is unlikely. As it turns out, despite these assumptions, meaningful clusters can still be found. For example, clusters like the '*colors*' cluster in Figure 1.6 can be obtained with the clustering algorithm for BMCs with texts [9]. For a given dataset, however, assessing the performance of the clustering algorithm may not be so obvious as with the words example. One of the concerns is that other simpler and more interpretable models than BMCs could still be suitable for obtaining meaningful clusters. Thus, a clear motivation for understanding the scope of the model is to evaluate if BMCs are appropriate for a particular dataset. A question we may ask is the following:

(7) How well do the BMC model and the clusters obtained with its clustering algorithm describe dynamics in real-world sequential data?

A parallel motivation to the suitability of the model is also to ascertain the limitations of the model, which could provide bounds on its usefulness. On the one hand, a way to approach the limitations is to compare previously known theoretical properties of BMCs with those obtained from real data. Specifically, we can analyze for example if the asymptotic bounds for the spectral error of a BMC that we have seen in the previous paragraphs match those inferred from examples. On the other hand, we may also probe the limitations of the model in a more controlled environment. Specifically, we may consider the performance of the algorithm under different models for clusters in sequential data that are not exactly BMCs. A broad question encompassing these two examples is then:

(8) How robust is the clustering algorithm for BMCs?

In the second part of this thesis we will address questions (5)–(8). In Chapter 5, we bound the spectral norm for generic BMCs. In Chapter 6 we focus on the applicability of BMCs to real sequential data from a practical point of view. As for the remainder of this part, we first introduce spectral clustering for BMCs in Section 1.6, which will provide context for the summary of results later in this part of the thesis.

## 1.6 Spectral clustering

The clustering algorithm in [27] that reaches the detection and recovery thresholds for BMCs consists of two steps. Firstly, an initial guess for the underlying cluster assignment $\hat{\nu}_n$ is found by using the random matrix $\hat{N}$ defined in (1.17) associated with the sample path. If we assume that the number of clusters $K$ is given, then the first $K$ singular vectors of $\hat{N}$ are used to construct a low-rank approximation of the random matrix $\hat{N}$, after which a distance-based clustering algorithm like k-means is applied to find the first guess $\hat{\nu}_n$ of the cluster assignment. This method is also called *spectral clustering*, which is one of the most popular algorithms for community detection due to its efficiency [93]. Secondly, the initial guess for the cluster structure $\hat{\nu}_n$ is refined by means of an *improvement algorithm* that reconsiders the sequence of observations and performs a greedy, local maximization of a log-likelihood function of the BMC model. In order to provably obtain a first guess $\hat{\nu}_n$ that is close to the true assignment $\nu_n$, a sufficiently small bound on the spectral error of (1.19) is required.

In BMCs, the spectral clustering is inspired from similar spectral clustering algorithms found in random graphs. In a graph, the adjacency matrix $A_n$ encodes the graph structure by setting the entries of a matrix with dimension $n$ to 1 or 0 depending on whether there is an edge or not respectively. Formally, $(A_n)_{i,j} = \mathbb{1}[(i,j) \in \mathcal{E}]$, where $\mathcal{E}$ is the edge set of the graph. When the graph is random and contains clusters, we have a random adjacency matrix $\hat{A}_n$. Then $\hat{A}_n - \mathbb{E}[\hat{A}_n]$ is a centered random matrix with independent entries and a certain block structure, namely, the one corresponding to the true cluster assignment $\nu_n$. A block structure contributes significantly to the spectrum of $\hat{A}_n$, which usually encodes global properties of the graph. For the recovery of clusters, this fact suggests that examining the largest singular values of $\hat{A}_n$ may provide information about the clusters. In our asymptotic setting, this translates to expecting that the largest $K$ singular values of $\hat{A}_n$ will be approximately those of $\mathbb{E}[\hat{A}_n]$ and will be increasing in $n$. The difference $\hat{A}_n - \mathbb{E}[\hat{A}_n]$ will thus have asymptotically lower order than the main singular values of $\hat{A}_n$ and so the contributions of the block structure to the spectrum can be asymptotically separated from the stochastic noise. This is the fundamental property that justifies the use of spectral methods for clustering and will also be the case in BMCs with $\hat{N}$.

A term that appears in the theoretical analysis of spectral clustering and in the recovery thresholds for BMCs, and generally in random graphs, is the average degree, that is, the average number of edges connected to a node in the network. An example from random graphs is the Erdös–Rényi random graph (ERRG), where there is only one community of (all) $n$ vertices and an edge is present independently of others with probability $p_n \in (0,1)$. Hence, the average degree in ERRGs is $np_n$. In BMCs, we can similarly define the average degree as the average number of transitions per state, that is, $T_n/n$.

Depending on the average degree, there are fundamentally different regimes for a graph, where global properties can be different and can impact the performance of the spectral clustering. These regimes are the dense and sparse regimes. We say that a graph of $n$ vertices is *dense* if the average degree is asymptotically larger than or equal in order to $\log(n)$ and *sparse* if the average degree has asymptotically an order less than $\log(n)$. For example, an ERRG becomes asymptotically disconnected with high probability only if the graph is sparse [167, 166]. Similar thresholds appear also in the generalization of the ERRG with more communities, which is the well-known Stochastic Block Model (SBM) [169]. In the sparse regime, the spectrum for random graphs can be dominated by a small amount of vertices with large connectivity compared to the remaining vertices [133, 139]. Global properties of the graph like communities can be consequently hidden in the spectrum by the contribution of these large-degree vertices, which negatively impacts the performance of the spectral clustering. We expect similar phenomena to occur in BMCs, and indeed, in Chapter 5 we verify that a different treatment to $\hat{N}$ is required in the sparse regime.

From the previous analogies of the BMC model and random graphs, we can try to compare the spectrum of a BMC to that of a random graph in a fair setting. In Table 1.1, the spectral norm of BMCs is compared to the spectral norm $\|\hat{A}_n - \mathbb{E}[\hat{A}_n]\|$ of an ERRG with a similar order of the average degree. In both dense and sparse regimes, the comparison suggests that the spectral norm of BMCs is of order $\sqrt{T_n/n}$, which is not the order of the bound in (1.19). Indeed, as we show in Chapter 5 this intuition is correct.

| Model | Average degree | Spectral norm | Reference |
|---|---|---|---|
| ERRG | $np_n = T_n/n$ | $O_{\mathbb{P}}(\sqrt{T_n/n})$ | [133] |
| BMC | $T_n/n$ | $O_{\mathbb{P}}(\sqrt{(T_n/n)\log(T_n/n)})$ | [27] |
| BMC | $T_n/n$ | $\Theta_{\mathbb{P}}(\sqrt{T_n/n})$ | this thesis |

*Table 1.1: Comparison of previously known spectral bounds for an ERRG and a BMC, together with the new bound in this thesis. We assume that the average degree is $T_n/n$ in all cases, which in the ERRG is equivalent to an edge probability of $p_n = T_n/n^2$. We show in this thesis that the previous bound for BMCs can be improved to match the expected order. The notation of $O_{\mathbb{P}}$ and $\Theta_{\mathbb{P}}$ denotes that $O$ and $\Theta$ hold with high probability. See Chapter 5 for a formal definition.*

## 1.7 Summary of results of Part II

The focus of the second part of the thesis is on questions (5)–(7). As mentioned in Section 1.6, the spectral error plays an important role in the clustering algorithm for BMCs, and knowing sharp bounds will determine the limits of the spectral clustering. This question is investigated in Chapter 5. In Chapter 6 we evaluate the clustering algorithm for BMCs in real datasets. The contributions of Chapters 5 and 6 are summarized in the following two subsections respectively.

### 1.7.1 Spectral norm in BMCs

The previously known bound in (1.19) is sufficient to prove that detection and recovery of clusters are possible in BMCs. However, if we compare the bounds in the BMC model to similar bounds in random graphs fairly, the bound in Equation (1.19) is suspected not to be sharp. In both dense and sparse regimes, we expect the spectral norm in BMCs to have order $\sqrt{T_n/n}$.

In Chapter 5 we examine the spectral norm of $\hat{N} - N$ and show first a sharp upper bound in the dense regime. Namely, we establish that

$$\|\hat{N} - N\| = O_{\mathbb{P}}(\sqrt{T_n/n}). \tag{1.20}$$

Futhermore, we also establish the bound (1.20) in a sparse regime, for which a different analysis is required. To avoid the effect of large-degree nodes in the spectrum, we have to use a special regularization $\hat{N}_\Gamma$ of the matrix $\hat{N}$. Namely, a proportion of the largest entries of $\hat{N}$ will be set to zero. This type of regularization is necessary in BMCs and random graphs to avoid nodes with comparably large degree [133]. See Chapter 5 for further details.

We show the bound in (1.20) with an approach used in [27, 133, 93] for bounding the spectrum of random graphs, which we will adapt to the case of BMCs. For this purpose, we will leverage concentration inequalities for Markov chains [98] to deal with the characteristic dependencies in BMCs.

Finally, in Chapter 5, we show that the order in (1.20) is actually sharp. Namely, asymptotically with high probability there is a lower bound $\|\hat{N} - N\| > C\sqrt{T_n/n}$ for $C > 0$. Thus, the order of the singular values of the spectral error will be fully characterized. These two results answer questions (5)–(6) positively.

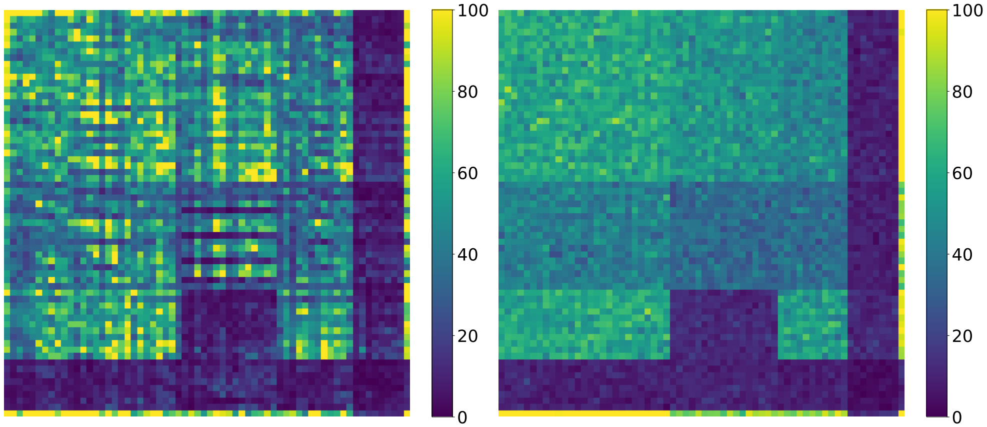


*Figure 1.8:* (Left) *The frequency matrix* $\hat{N}$ *of the DNA dataset where states have been ordered with the clusters.* (Right) *The frequency matrix* $\hat{N}$ *of a sampled BMC model with the inferred parameters from the DNA dataset with states ordered with the clusters, similarly to (Left). The BMC model captures some cluster information, despite the inhomogeneity in the dataset.*

### 1.7.2 Experimentally testing the BMC model with sequential data

From an application point of view, a BMC could be a useful model for dimensionality reduction in a time series as hinted in Section 1.5. Examples of time series for which Markov chains have historically been used include plant / human DNA (microbiology) [162, 134], speech / text (natural language processing) [158], and GPS location data (geography / ethology) [39, 47].

In Chapter 6 we consider BMCs from a practical point of view and use the cluster recovery algorithm for BMCs in real-world sequential data. We consider three datasets: stock market data, Deoxyribonucleic Acid (DNA) sequences and the Global Positioning System (GPS) coordinates of bisons (see Chapter 6 for further details on the datasets and their sources). We use the clusters and BMC parameters recovered for these three datasets in order to evaluate different properties of the clustering algorithm. For comparison, in Figure 1.8 the count matrix $\hat{N}$ from a BMC-generated model versus one from DNA is shown. Chapter 6 is based on [9], where additionally, a comprehensive analysis and evaluation of the BMC model and the recovered clusters in the aforementioned datasets are conducted.

First, we will examine the spectral gap for BMCs in the data in a qualitative manner and compare the expected orders for the singular values. The analysis will show that the spectral error, while enough to provide guarantees in a true BMC, appears not to be very robust in real data and is sensitive to model violations. We will also provide a heuristic argument for why this is the case.

In order to further assess the clustering algorithm, a robustness analysis is conducted. To do so, BMC models with additional noise are considered. The cluster comparison in these perturbed BMC models will show numerically that the algorithm is robust under small to moderate perturbations of different noise models and that it can still approximately recover parameters of models that are not BMCs.

In the last part of Chapter 6, we will evaluate the merit of the BMC model. A particular motivation is to decide if the clusters and the transition dynamics of the data could also be explained by a simpler model instead of a Markov chain, namely, by a model without time dependencies. For these datasets, we will use model selection tools and decide if the BMC model is a good representation of the dynamics of the data compared to other less and more complex models. In the full state space, conducting model selection will not be possible due to the large number of free parameters. Thus, we will exploit the state space reduction and perform model selection in the low-dimensional latent space of clusters. The analysis will provide evidence that the cluster transitions of the different datasets seem to follow a model with Markovian dependence, except for the stock market.

## 1.8 Related literature and positioning

In this last section of the introduction we provide references and position the results of this thesis in the literature.

### Part I: Dropout

**NNs.** In the last years, many NN architectures have appeared that are tailor-made for the target task, e.g., image classification like those in (1.2) or for reinforcement learning, where sequences of observations and rewards are the input that an agent receives when acting upon the environment. We can mention among many others CNNs [54], Graph NNs [131], VGG16 [104], Variational Autoencoders [40] and policy-value networks for reinforcement learning and its variants [94, 43].

Despite its ample use and success, the understanding of NNs for tasks in supervised learning is still incomplete. It is not yet fully understood, for example, why SGD succeeds at finding good local minima of the nonconvex risk function in NNs [73], that is, the gap in (1.9) is often found to be small. While there are convergence results for overparameterized NNs [34]—the number of free parameters $N$ is larger than the number of samples $n$—it is not fully clear why NNs generalize well in spite of the potential for overfitting [66]. From the statistical point of view, a major issue is the difficulty of estimating parameters in a high-dimensional space from comparably small amounts of data, which is typically the case in NNs. This is the so-called *curse of dimensionality* in statistics. Nonetheless, it is found experimentally that NNs can have a small gap in (1.10), and there are results in the literature [28] that provide some understanding of this dichotomy.

**Variants and applications of dropout.** The first version of a dropout algorithm was introduced in [117] which was used for the ImageNet competition in [118]. Variants of the algorithm have appeared ever since, including versions in which edges are dropped [115]; groups of edges are dropped from the input layer [64]; the distribution of the filters are Gaussian [92, 68]; the dropout probabilities change adaptively [109, 82]; and that are suitable for recurrent NNs [108, 84], which are NNs that can be used to model evolution of differential equations. The performance of the original algorithm has been investigated on common datasets [117, 105], and dropout algorithms have found application in e.g. image classification [118], handwriting recognition [103], heart sound classification [79], and drug discovery in cancer research [62].

**Regularization and dropout.** Theoretical studies of dropout algorithms have mainly focused on their regularization effect. The effect was first noted in [117, 105], and subsequently investigated in-depth for both linear as well as nonlinear NNs by [110, 114, 100, 31]. Within the context of matrix factorization, it has been shown that *Dropout*'s regularization induces a shrinkage and a thresholding of the singular values of the optimal matrices [51]. Characterizations of *Dropout*'s risk function and *Dropout*'s regularizer for (usually linear) NNs can be found in [60, 42, 26]. In the context of matrix factorizaton, we will see in Chapter 3 that the risk function of dropout, after rescaling by $p$, takes the form

$$\mathcal{D}(W) = \|Y - W_2 W_1\|_{\mathrm{F}}^2 + R(W), \tag{1.21}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm but $R(W)$ is not a norm. The regularization term $R(W^*)$ at a minimum $W^*$ of $\mathcal{D}(W)$ plays a role in bounding the generalization gap [12] in NNs.

Other regularization techiques for NNs exist besides dropout. For example, [90] considers normalizing the output of the layers. This is also called batch normalization. Early stopping of SGD is also a common regularization technique. Other more classical approaches include adding an explicit regularization term. For example, a $L^1$ norm [122] or $L^2$ norm [160] of the weights may be added as an explicit penalization term in the minimization problem.

**Convergence of dropout.** Detailed theoretical investigations into the convergence of dropout algorithms are relatively scarce. Independently of the results presented in this thesis, a parallel result by [25] appeared that gives insight into the convergence rate of *Dropout* in ReLU shallow NNs for a classification task. There, a sample complexity bound to reach $\epsilon$-suboptimality for the test risk is provided. In particular, $\mathrm{O}(1/\epsilon)$ iterations of SGD are required to reach $\epsilon$-suboptimality. The main result in [25] considers a 'lazy regime' analysis for overparametrized NNs, for which weights do not change much with the updates of SGD and can be controlled. The result relies on an assumption that the data distribution is separable with a margin in a particular Reproducing Kernel Hilbert Space (RKHS) that is coupled to the distribution of the filters. However, for dropout the convergence rate derived there is independent of the dropout probability $1-p$, which as explained in [25] is because of the separability assumption. Compared to our generic convergence result of Proposition 2 in Chapter 2, we do not assume structure of the function or data and look instead at the iterations required to reach $\epsilon$-stationarity of the dropout risk in nonconvex smooth functions using dropout-like SGD. Despite this generic assumption, an explicit dependence on the dropout probability is obtained.

We can also compare the result in [25] to the convergence of dropout in shallow linear NNs of Theorem 10 in Chapter 3. While local in nature, the latter convergence result gives also an explicit convergence rate for any width of the network. This comes, however, at the cost of assuming that the stochasticity can be controlled by using the gradient flow of an ODE. Moreover, we use completely different techniques than those in [25].

Finally, it must be noted that convergence properties have been thoroughly studied within the context of NNs trained without dropout algorithms, see e.g. [35, 44, 33, 14] and references therein.

**Stochastic approximation.** Dropout algorithms can, by construction, be understood as forms of stochastic approximation algorithms. Stochastic approximation techniques in optimization were introduced by [171, 170], and have been the subject to a wide literature due to their ubiquity. For overviews and their application to NNs, we refer to books by [140, 127, 149]. We use results from [140] to prove our first result in Chapter 2.

**Universal approximation in NNs.** The first universal approximation theorem for NNs with a sigmoidal activation function can be found in [156], and this canonical work led to much follow-up research. Several years later [154] showed that the universal approximation property relies more on a NN's architecture than on the specific use of sigmoid activation functions. Moreover, [153] established that deep, feed-forward NNs require a nonpolynomial activation function in order for a universal approximation theorem to hold. With a different approach, [148, 146] used the so-called probabilistic method to prove the existence of a deterministic function that suitably approximates a target function in deterministic NNs.

**Approximation by random NNs.** Parallel to the study of the approximation properties of NNs, the study of random NNs started. [159] is one of the first works where universal approximation is mentioned (but not proved) side by side with a NN algorithm in which random hidden nodes are placed.

A class of networks with random weights and biases, called Random Vector Functional-Link Nets, was introduced in 1994 by [152]. [150] proved a universal approximation property of these networks, by showing that the span of the node functions is almost surely asymptotically dense in the many-node limit. This result does not apply to dropout schemes since in the dropout setup the randomness is applied after choosing coefficients.

[141, 142] introduced a class of NNs that relies on a fixed NN topology on top of which neurons forward positive and negative signals (spikes) at random points in time based on their own "potential". Specifically, a constructive proof of the universal approximation theorem for such stochastic NNs in steady state was given. This class of networks also does not cover the usual dropout—when we drop nodes—or dropconnect due to the different dynamics assumed; moreover a dropout NN is trained randomly, but typically operated deterministically.

[129] investigated uniform approximation of functions with random bases. This is a particular case of a so-called random feature method, in which parameters are split in two groups: parameters in one group are taken randomly (and not tuned), and the other part is trained to achieve the best approximation. Therefore these results also do not cover dropout or dropconnect since for the latter algorithms all parameters are trained.

Another commonly used class of NNs is the *mixture of experts* model. The idea is that for different input regions different, typically simpler, networks (learners) are used for prediction. The choice is performed by the *gating network*; training of the model consists then of training individual learners together with training the gating network. [83] proved a universal approximation theorem for a mixture-of-experts model, and [69] subsequently generalized their findings to allow for so-called Gaussian gating.

[37] considered a network architecture that can handle probability measures as input and output. A universal approximation in Wasserstein metric was proven for continuous maps from the space of measures into itself. Our results that will be described in Chapter 4 are more specific, and not covered by this result, since we study a different (more restricted) approximation scheme.

Finally, we refer to the following surveys on results regarding the approximation properties by random NNs. A survey of approximation-theoretic problems was written by [144]; a recent survey by [13] contains a comparison of approximation properties for finite-width and finite-depth networks. Several uniform approximation results for random NNs can be found in [126, Section 5.4]; see also [46].

## Part II: BMCs

**Community detection.** Regardless of the model for communities, randomness adds difficulties to detecting and recovering a hidden cluster structure. Detection can become difficult if the clusters are too similar or if the clusters are too loosely defined and only a low number of connections exist. This latter problem occurs, for example, with sparse networks, where clusters, if statistically different, cannot be detected due to lack of data. Similarly, with a detection, that is, a statistical guarantee that clusters exist, recovering such clusters may not be possible even in the limit that the number of vertices grows large. In particular, the recovery algorithm may be computationally intractable. As hinted by these trade-offs, there is an information-theoretical limit to community detection and recovery, which has been studied for certain models.

For example, investigation of community detection problems within the context of SBMs—the generalization of an ERRG with clusters—has seen great progress. In the sparse regime, necessary and sufficient conditions for extraction of clusters that are positively correlated with the true clusters have been obtained [121, 102, 96]. In the dense regime, conditions have been established under which the proportion of misclassified vertices can tend to zero as well as under which asymptotic exact recovery occurs [86, 87, 91, 95, 107, 106, 85, 75, 72]. We refer to [65] for an overview of clustering algorithms.

**BMCs and Markov chains.** Community detection problems for BMCs have thus far received less attention. In the sparse regime, an information-theoretical lower bound on the detection error rate satisfied under any clustering algorithm was derived in [27] together with a two-stage clustering algorithm that can accurately recover the cluster structure.

In a BMC, there is an information quantity $I(\alpha,q)$ depending on the cluster transition probabilities $q$ and cluster ratios $\alpha$ such that exact recovery is possible if

$$T_n - n\log(n)/I(\alpha,q) = \omega(1), \tag{1.22}$$

that is, the true cluster assignment $\nu_n$ can be asymptotically recovered up to a *finite* set of states of $[n]$ with high probability if (1.22) holds. In particular, exact recovery is possible if $T_n = O(n\log(n))$. On the other hand it can be proven that detection of clusters can be conducted when $T_n = \omega(n)$ [27]. As we will see, these scalings will also appear in Chapter 5 when estimating the spectral error in (1.18).

Besides BMCs, there are other models for Markov chains with low-dimensional latent structures. In the dense regime, learning of low-rank structures in Markov chains from trajectories is studied in [47], where spectral methods are used to recover a low-rank approximation of the Markov chain's transition matrix; [18], where a maximum likelihood estimation method was used; and [39], where an algorithm is analyzed that relies on a spectral decomposition followed by an approximation of the convex hull of singular vectors. Noteworthy too is [38], which describes a method for recovering a latent transition model from observations of a dynamical system switched by a Markov chain with low-rank structure; and [32, §5], where the problem is related to estimating a low rank 'tensor-train' decomposition from noisy high-order tensor observations.

**Spectrum of random matrices and Markov chains.** As mentioned in Section 1.5, random graphs can be analyzed by considering the spectrum of an associated random matrix. Such spectral properties have been extensively studied. Most results hold for random matrices with independent or weakly dependent entries, see e.g. [168, 119, 99, 76, 58, 57]. Results on the spectra of adjacency matrices of e.g. SBMs or ERRGs also make use of independence assumptions [133, 88]. There are further intriguing results on the spectra of random Markov chains [123, 120, 116]. Compared to these models, in Chapter 5 we will examine instead singular values coming from the random frequency matrix in (1.17). A single sample path of a (nonrandom) Markov chain with an underlying block structure is used as the source of randomness and the sample path can also be short compared to the size of the system. For the dense regime, a recent article characterizes the limiting distribution of the singular values of BMCs when $T_n = \Omega(n^2)$ [2].

In Chapter 5 we sharpen the spectral norm bounds of [27] and quantify an asymptotic gap between the largest and smallest singular values. The proof method builds on the techniques in [133, 124, 93] by incorporating concentration inequalities for Markov chains [98] and relying on a perturbative argument using Weyl's inequality [151].

**Regularization in random graphs.** As proposed in [27], our analysis in Chapter 5 also requires regularization of the random frequency matrix $\hat{N}$ in the sparse regime. We zero out the entries of the frequency matrix that correspond to a fixed-size subset of most-visited states. There exist also other regularization techniques for random graphs. Namely, for an average degree $d_a$, one may discard vertices with degree higher than a threshold $(1+\epsilon)d_a$ [133] for some $\epsilon > 0$ independent of $n$. Scaling down the weights of edges incident to vertices of high degree is also possible [67, 59] and has some advantages in the sparse regime compared to the previous methods that discard vertices.

# Part I

# Dropout

# Chapter 2

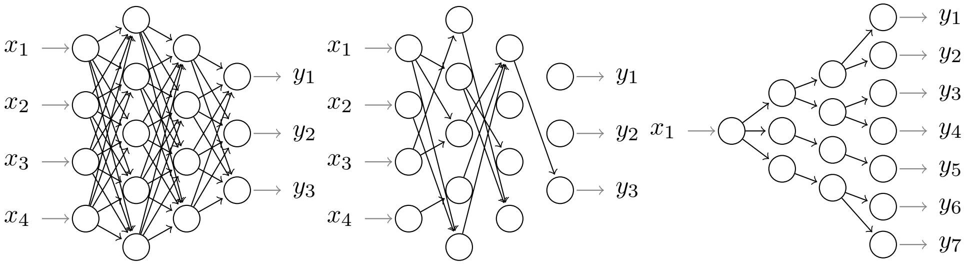
# Almost sure convergence of dropout algorithms for neural networks

Based on [29]:
*"Almost sure convergence of dropout algorithms for neural networks"*
by A. Senen–Cerda, and J. Sanders

In Chapter 1 we have motivated the analysis of dropout from the stochastic optimization perspective. In this chapter, we introduce the first results concerning the convergence guarantees of dropout as well as its convergence rate.

## 2.1 Introduction

While not explicitly mentioned in Chapter 1, an interesting aspect of dropout algorithms is that they lie at the intersection of stochastic optimization and *percolation theory*, which investigates properties related to connectedness of random graphs and deterministic (possibly infinite) graphs in which vertices and edges are deleted at random. In the case of dropout, the output of the filtered Neural Network (NN) with temporarily deleted edges is used to update the weights. If dropout filters too many weights we expect that little information can pass through the network, which will consequently also yield a gradient update for that step that contains little information. As an example, we may consider the networks in Figures 2.1(a)–(b) when we use *Dropconnect*, that is, we filter each edge with probability $1-p$ independently of all other edges. In an $L$-layer NN with no biases, a path from the input layer to the output goes through $L$ weights that have filters. Then, the probability that a path from input to output stays unfiltered and contributes to a weight update is $p^L$. If we now fix one edge in the path, then the probability of updating its corresponding weight through that path in particular is also $p^L$. There are, however, many other paths in a NN passing through a single edge. The probability that one of those paths is not filtered will be large and may compensate the exponential factor $p^L$. Considering the connection to bond percolation, one may therefore expect at first glance that dropout algorithms may perform worse than a routine implementation of the backpropagation algorithm. However, dropout algorithms usually perform well due to their regularization properties [117, 105]. From the point of view of bond percolation however, this should still come at the cost of slower convergence of dropout algorithms, and conceivably by as much as a factor $p^L$.



*(a) Case $p = 1$.* *(b) Case $p = 0.25$.* *(c) An arborescence.*

*Figure 2.1:* (a,b) *Examples of* Dropconnect*'s training step [115] in a NN with $L = 3$ layers. We can easily observe that the number of paths $\chi$ in (b) that fully transverse the network ($\chi = 5$) is much smaller compared to those of (a) ($\chi = 240$).* (c) *An example arborescence of depth $L = 3$.*

In this chapter we establish first a convergence guarantee for dropout algorithms. Furthermore, we analyze the sample complexity of Stochastic Gradient Descent (SGD) with dropout and the convergence rate of dropout depending on the depth of the NN with a toy model. At the end of the chapter, we investigate numerically and with heuristics what convergence rate to expect in realistic NNs.

## Summary of results

When using dropout with SGD, we use a stochastic estimate $\Delta^{[t+1]}$ of the gradient of the empirical risk. Denote by $F^{[t+1]}$, $X^{[t+1]}, Y^{[t+1]}$ the dropout filters and the samples provided to the SGD algorithm at time $t$ respectively. We define $B_W(X,Y)$ to be the gradient at $W$ of the risk such as (1.5) if we have input and output pairs $(X,Y)$ (denoted by $g_{t+1}$ in Section 1.3). Also depicted in Figure 1.3, dropout defines the estimate of the gradient update as

$$\Delta^{[t+1]} \triangleq F^{[t+1]} \odot \mathrm{B}_{F^{[t+1]} \odot W^{[t]}}(X^{[t+1]}, Y^{[t+1]}), \tag{2.1}$$

where $\odot$ denotes the componentwise product.

Note that the filters appear twice in (2.1). Firstly, they filter the weights $W^{[t]}$ when the gradient is computed depending only on the subnetwork provided by dropping some edges or nodes. Secondly, they filter the updates in $\Delta^{[t+1]}$ since only the remaining weights will be updated. Moreover, other distributions for the filters than those for dropout and dropconnect are allowed (see also 2.2.3).

Our first result is a formal probability theoretical proof that for any (fully connected) NN topology and with differentiable polynomially bounded activation functions (see Definition 2 for a formal definition), the iterates of projected SGD with dropout-like filters converge. In particular, a step of projected SGD with dropout is given by

$$W^{[t+1]} = \mathrm{P}_{\mathcal{H}}(W^{[t]} - \alpha^{\{t+1\}}\Delta^{[t+1]}) \quad \text{for} \quad t \in \mathbb{N}_0, \tag{2.2}$$

where $\Delta^{[t+1]}$ is the estimate of the gradient with dropout in (2.1) and $\mathrm{P}_{\mathcal{H}}$ is an operator that projects the iterates onto a compact convex set $\mathcal{H}$ [61]. In order to state our first result, we define *dropout algorithm's risk function* as

$$\mathcal{D}(W) \triangleq \int l(\Psi_{f\odot W}(x), y)\,\mathrm{d}\mathbb{P}[(F,X,Y) = (f,x,y)], \tag{2.3}$$

and we will consider $l(a,b) = |a-b|^2$ to be the $\ell_2$-loss.

The first result is stated informally in the next proposition (see Proposition 1 below for the exact set of assumptions (N1)–(N6)).

**Proposition** (informal)**.** *Under sufficient regularity of the activation functions, bounded moments and independence of random variables and some assumptions on the boundary $\mathcal{H}$, with update* (2.2)*, the weights $(W^{[t]})_t$ converge to a unique stationary set of a projected system of Ordinary Differential Equations (ODEs)*

$$\frac{\mathrm{d}W}{\mathrm{d}t} = -\nabla_W \mathcal{D}|_{\mathcal{H}}(W) + \pi(W), \tag{2.4}$$

*where $\pi(W)$ is a* constraint term*, which describes the minimum force required to keep the gradient flow of $\nabla\mathcal{D}$ in $\mathcal{H}$.*

This result gives us the formal guarantee that dropout algorithms are well-behaved for a wide range of NNs and activation functions, and will at least asymptotically (meaning after sufficiently many iterations) not suffer from problems that could have arisen from the relation to bond percolation. Moreover, the function $\mathcal{D}(W)$ is the expectation of the risk over the dropout's filters distribution, which in our result is not restricted to dropping nodes and can even be coupled to the data. This result also shows that SGD with dropout converges to the stationary points of $\mathcal{D}(W)$. Note that while not explicit in the definition of $\mathcal{D}(W)$ in (2.3), in practical scenarios where we have only a set of datapoints, $\mathcal{D}(W)$ will be defined analogously to the empirical risk $\hat{\mathcal{R}}_n$ in (1.5) in Chapter 1. In online settings, however, this may not be the case.

While a guarantee is necessary, a convergence rate would yield more insight into the trade-offs of the algorithm. In our second result of this chapter, we compute a bound for the sample complexity of the convergence of dropout to an $\epsilon$-stationary point of a generic smooth nonconvex function $\mathsf{D}(W)$. We say $W \in \mathcal{W}$ is an $\epsilon$-stationary point of $\mathsf{D}$ if $\|\nabla\mathsf{D}(W)\|_2 \leq \epsilon$ holds. Note that stationary points are not necessarily minima, but the sample complexity, understood as the number of iterations $T$ required to reach $\epsilon$-stationarity, is usually associated with the complexity of the function to be optimized.

For a generic smooth nonconvex function $\mathsf{D}(W)$, we consider dropout to be SGD with the update in (2.1), where filters $F$ are chosen independently at each step and are $\{0,1\}$-valued for each parameter. In our result we assume boundedness and Lipschitzness conditions on $\mathsf{D}(W)$. Moreover, under some additional assumptions on the loss function, examples of NNs with sigmoid activation functions $\sigma(t) = 1/(1+\exp(-t))$ are also covered by our result. In this particular case, $\mathsf{D}(W) = \mathcal{D}(W)$ holds with the definition in (2.3). For the general case we prove the following (see Proposition 2 below for the full description of the assumptions (Q1)–(Q5)):

**Proposition** (informal). *Assume that* $\mathsf{D}(W)$ *has enough regularity and satisfies some boundedness and Lipschitzness assumptions. Let* $W^{\{t\}}$ *be iterates of* (2.2). *For any* $T \in \mathbb{N}$ *there exist* $c > 0$ *and* $c_1, c_2 > 0$ *and* $\alpha^{\{t\}} = \eta$ *constant such that if* $p > c/T$, *then*

$$\min_{t\in[T]} \mathbb{E}\Big[\|\nabla \mathsf{D}(W^{\{t\}})\|_2^2\Big] = O\left(\sqrt{\frac{p(c_1+(1-p)c_2)}{T}}\right). \tag{2.5}$$

Hence, at least $T$ iterations of dropout-like SGD algorithms are required to reach an $O((p(c_1+(1-p)c_2)/T)^{1/4})$-stationary point of nonconvex smooth functions in expectation. Here, $c_1, c_2$ are constants depending on the data and function respectively. Compared to the theoretical optimum rate of $O(T^{-1/4})$ for SGD on nonconvex smooth functions [20], this result shows that dropout changes the optimization landscape and approximate stationary points are easier to find depending on the dropout probability. To investigate the convergence rate assuming a NN structure, we further examine theoretically and numerically the convergence rates in specific types of networks.

Our third result in this chapter is an explicit upper bound for the rate of convergence of regular Gradient Descent (GD) on the limiting ODEs of dropout algorithms for arborescences (a class of trees, see Figure 2.1c for an example), of arbitrary depth with linear activation functions $\sigma(t) = t$. In particular, we will consider the update rule

$$W^{\{t+1\}} = W^{\{t\}} - \alpha \nabla \mathcal{D}(W^{\{t\}}). \tag{2.6}$$

Analyzing the convergence of training algorithms on simplified NNs structures with linear activation functions is commonly used to gain insight into more complex models, see e.g. [35, 44, 49]. Even without a dropout algorithm present, this task already provides a substantial theoretical challenge as the optimization landscape is nonconvex. Our choice to restrict the analysis to arborescences allows us to quantitatively tie our upper bound for the convergence rate to the depth and the number of paths within the arborescence. We prove the following (see Proposition 3 below for the full statement):

**Proposition** (informal). *Assume that the base graph* $G$ *of the NN is an arborescence of depth* $L$ *with* $|\mathcal{L}(G)|$ *leaves and the filters* $F$ *follow the distribution prescribed by* Dropconnect *or* Dropout *with dropout probability* $1-p$. *Then there exist* $\alpha > 0$ *and* $1 > \eta > 0$ *depending on the initialization such that the iterates of* (2.6) *satisfy*

$$\mathcal{D}(W^{\{t\}}) - \min_W \mathcal{D}(W) \leq \big(\mathcal{D}(W^{\{0\}}) - \min_W \mathcal{D}(W)\big)\exp(-\omega t/2), \tag{2.7}$$

*with*

$$\omega = \mathrm{O}\Big(\frac{p^L}{L|\mathcal{L}(G)|^2}\eta^{2L}\Big). \tag{2.8}$$

One important consequence of this result is that the convergence rate exponent indeed deteriorates by a factor $p^L$ in these NNs. Finally, we complement this result with numerical experiments. We target the dependency of the convergence on $p$ for more realistic wider and nonlinear networks on commonly used datasets. Perhaps surprisingly, we do not observe an exponential decrease of the convergence rate exponent due to dropout in these simulations. We will offer some heuristic explanation for this result by looking at the update rate of a generic weight. To our knowledge, the contents of this chapter present the first experimental study of the convergence rate of SGD with dropout for deep NNs with the dropout probability and depth as hyperparameters—parameters that are chosen before training.

Our results lead to the following consequences. First, whenever the iterates of a dropout algorithm with $\ell_2$-loss are bounded, they are guaranteed to converge to a stationary point of the risk function $\mathcal{D}(W)$ induced by the dropout algorithm. Secondly, we prove rigorously that the convergence rate when training with e.g. *Dropout* or *Dropconnect* can change the convergence rate on the empirical risk depending on $p$ and in arborescences can decrease by as much as a factor $p^L$. For more realistic wider networks, however, we conduct numerical experiments that suggest that the convergence rate is not necessarily affected by depth as much across different dropout rates $1-p$ in deep neural networks. Our findings motivate the theoretical study of the convergence rate of dropout for wide networks. We suspect that there is a transition regime of the convergence rate. Such transition would affect the dependence on $p$ and would be observed when going from networks with many layers of dropout with small width, where dependence on the rate may be exponential in $p$, to networks with a few layers of dropout but very wide, where dependence is not exponential anymore.

## Notation

In this chapter we index deterministic sequences with curly brackets: $\alpha^{\{1\}}, \beta^{\{1\}}$, etc. This distinguishes them from sequences of random variables, which we index using square brackets, e.g. $X^{[1]}, Y^{[1]}$, etc. This is the same notation used in Sections 1.1 to 1.4 of Chapter 1.

Deterministic vectors are written in lower case like $x \in \mathbb{R}^d$, but an exception is made for random variables (which are always capitalized). Matrices are also always capitalized. For a function $\sigma : \mathbb{R} \to \mathbb{R}$ and a matrix $A \in \mathbb{R}^{a\times b}$, $a, b \geq 1$, we denote by $\sigma(A)$ the matrix with $\sigma$ applied componentwise to $A$. Subscripts will be used to denote the entries of any tensor, e.g. $x_i$, $A_{i,j}$, or $T_{i,j,l}$. For any vector $x \in \mathbb{R}^d$, the $\ell_2$-norm is defined as $\|x\|_2 \triangleq (\sum_{i=1}^d |x_i|^2)^{1/2}$. For any matrix $A \in \mathbb{R}^{a\times b}$, the Frobenius norm is defined as $\|A\|_{\mathrm{F}} \triangleq (\sum_{i=1}^a \sum_{j=1}^b |A_{i,j}|^2)^{1/2}$. For two matrices $A, B$, the Hadamard (componentwise) product is denoted by $A \odot B$.

Let $\mathbb{N}_+$ be the strictly positive integers and $\mathbb{N}_0 \triangleq \mathbb{N}_+ \cup \{0\}$. For $l \in \mathbb{N}_+$, we denote $[l] = \{1, \ldots, l\}$. For a function $f \in C^2(\mathbb{R}^n)$, we denote the gradient and Hessian of $f$ with respect to the Euclidean norm $\|\cdot\|_2$ in $\mathbb{R}^n$ by $\nabla f$ and $\nabla^2 f$, respectively.

## 2.2 Model

We now formally define NNs, which we had depicted in Figure 1.1 of Chapter 1, as well as the class of activation functions that we will use for the convergence guarantee below. We also formally define SGD with dropout in the context of NNs and the risk function that dropout induces, which will be also used in the next chapter, albeit in different context

### 2.2.1 Neural networks and their structure

Let $L$ denote the number of layers in the NN, and $d_l \in \mathbb{N}_+$ the output dimension of layer $l = 1,\dots,L$. Let $W_{l+1} \in \mathbb{R}^{d_{l+1}\times d_l}$ denote the matrix of weights in between layers $l$ and $l+1$ for $l = 0,1,\dots,L-1$. Denote $W = (W_L,\dots,W_1) \in \mathcal{W}$ with $\mathcal{W} \triangleq \mathbb{R}^{d_L\times d_{L-1}} \times \cdots \times \mathbb{R}^{d_1\times d_0}$ the set of all possible weights. In this chapter, we consider NNs without biases. Later in Chapter 3, we will consider again a more general definition of NNs with biases in the context of random NNs.

**Definition 1.** *Let $\sigma$ be an activation function $\sigma : \mathbb{R} \to \mathbb{R}$. A* Neural Network (NN) *with* $L$ layers *is given by the class of functions $\Psi_W : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ defined recursively by*

$$A_0 = x, \quad A_i = \sigma(W_i A_{i-1}) \quad \forall i \in \{1,\dots,L-2\}, \quad \Psi_W(x) = W_L A_{L-1} = A_L. \tag{2.9}$$

Commonly used activation functions include the Rectified Linear Unit (ReLU) function $\sigma(t) = \max\{0,t\}$, the sigmoid function $\sigma(t) = 1/(1+\exp(-t))$, and the linear function $\sigma(t) = t$. In Sections 2.2 and 2.3 we restrict to the case that $\sigma$ belongs to a class of polynomially bounded differentiable functions.

**Definition 2.** *For $\sigma : \mathbb{R} \to \mathbb{R}$ differentiable, denote the $l$th derivative of $\sigma$ by $\sigma^{(l)}$. The set of polynomially bounded maps with continuous derivatives up to order $r \in \mathbb{N}_0$ is given by*

$$C^r_{\mathrm{PB}}(\mathbb{R}) = \Big\{\sigma \in C^r(\mathbb{R}) \Big| \forall l = 0,\dots,r \ \exists k_l > 0 : \sup_{x\in\mathbb{R}} |\sigma^{(l)}(x)(1+x^2)^{-k_l}| < \infty \Big\}.$$

Note that both the linear and the sigmoid activation function belong to $C^r_{\mathrm{PB}}(\mathbb{R})$ for any $r \in \mathbb{N}_0$. Also, any polynomial activation function $P(x) \in \mathbb{R}[x]$ belongs to $C^{\deg(P)}_{\mathrm{PB}}(\mathbb{R})$. The ReLU activation function is not in $C^r_{\mathrm{PB}}(\mathbb{R})$ for any $r \in \mathbb{N}_0$. However, because the class $C^r_{\mathrm{PB}}(\mathbb{R})$ contains polynomials of any degree, we can approximate cases such as ReLU by using, e.g., the softplus activation function $\sigma_t(x) = \log(1+\exp(tx))/t$, which satisfies that $\lim_{t\to\infty} \sigma_t(x) = \mathrm{ReLU}(x)$ for every $x \in \mathbb{R}$. Note that the softplus activation function belongs to $C^2_{\mathrm{PB}}(\mathbb{R})$.

### 2.2.2 Backpropagation and SGD

In the same setting as in Section 1.3, we let $(X,Y) : \Omega \to \mathbb{R}^{d_0} \times \mathbb{R}^{d_L}$ be a random variable on the probability space $(\Omega,\mathcal{F},\mathbb{P})$ with distribution $\mu$ over input–output pairs. For a NN $\Psi_W : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$, we can define a risk analogous to that of (1.5),

$$\mathcal{U}(W) \triangleq \int l(\Psi_W(x),y)\,\mathrm{d}\mathbb{P}[(X,Y) = (x,y)]. \tag{2.10}$$

Here, $l : \mathbb{R}^{d_L} \to [0,\infty)$ is a convex loss function of one's choice. Throughout this chapter, we will specify the Euclidean $\ell_2$-norm $l(x,y) \triangleq \|x-y\|_2^2$ as our loss function of interest without loss of generality.[1] Furthermore, we make no distinction between an oracle risk

[1] The argument can be extended to other smooth loss functions $l(x,y)$ whose partial derivatives can be bounded by polynomials of finite degree.

function or empirical risk function. Both situations are namely covered by (2.10), which can be seen by choosing the distribution $\mu$ appropriately. What we assume is that one has the ability to repeatedly draw independent and identically distributed samples from $\mu$. Thus, the results cover the empirical risk case (where $\mu$ has finite support) as well as the online learning case.

As explained in Section 1.3, we want to find weights $W$ that are close to or exactly in the set $\arg\min_W \mathcal{U}(W)$. In an attempt to find a critical point in the set $\arg\min_W \mathcal{U}(W)$, SGD is commonly used in NNs training in order to approximate a gradient descent step with low-iteration complexity. Let $\{(Y^{[t]},X^{[t]})\}_{t\in\mathbb{N}_+}$ be a sequence of independent copies of $(X,Y)$, let $W^{[0]}\in\mathcal{W}$ be an arbitrary nonrandom initialization of the weights. For $i=1,\ldots,L$, $r=1,\ldots,d_{i+1}$, $l=1,\ldots,d_i$, the weights are iteratively updated according to

$$W_{i,r,l}^{[t+1]} = W_{i,r,l}^{[t]} - \alpha^{\{t+1\}}\big(\mathrm{B}_{W^{[t]}}(X^{[t+1]},Y^{[t+1]})\big)_{i,r,l} \tag{2.11}$$

for $t=0,1,2,\ldots$. Here $\{\alpha^{\{t\}}\}_{t\in\mathbb{N}_+}$ denotes a positive, deterministic step size sequence, and the estimate of the gradient $\mathrm{B}_W(\cdot,\cdot)=\nabla_W l(\Psi_W(\cdot),\cdot)$ in NNs is computed using the backpropagation algorithm, which is given in Definition 4 in Appendix 2.A. The stochastic gradient is an unbiased estimate of the gradient of $\mathcal{U}(W)$. In particular, we have

$$\mathbb{E}[\big(\mathrm{B}_W(X,Y)\big)_{i,r,l}] = \mathbb{E}\Big[\frac{\partial l(\Psi_W(x),y)}{\partial W_{i,r,l}}\Big] = \frac{\partial \mathcal{U}(W)}{\partial W_{i,r,l}} = (\nabla\mathcal{U})_{i,r,l}. \tag{2.12}$$

### 2.2.3 Dropout algorithms and their risk functions

We examine a class of dropout algorithms that work by applying $\{0,1\}$-valued random matrices as filters of the weights during the backpropagation step. Let $(F,X,Y):\Omega\to\{0,1\}^{d_L\times d_{L-1}}\times\ldots\times\{0,1\}^{d_1\times d_0}\times\mathbb{R}^{d_0}\times\mathbb{R}^{d_L}$ be a random variable on the probability space $(\Omega,\mathcal{F},\mathbb{P})$. Here, we write $F=(F_L,\ldots,F_1)$ and $F_{i+1}\in\{0,1\}^{d_{i+1}\times d_i}$ for $i=0,\ldots,L-1$, similar to how we denote weight matrices. Let $\{(F^{[t]},X^{[t]},Y^{[t]})\}_{t\in\mathbb{N}_+}$ be a sequence of independent copies of $(F,X,Y)$. In tensor notation, the weights are updated iteratively by setting

$$W^{[t+1]} = W^{[t]} - \alpha^{\{t+1\}}\Delta^{[t+1]} \tag{2.13}$$

where $\Delta^{[t+1]}$ is given by (2.1) for $t=0,1,2,\ldots$. Note that in (2.1), if $F_{i,r,l}^{[t+1]}=0$ for some $i,r,l$, then $\Delta_{i,r,l}^{[t]}=0$. In other words, filtered variables are not updated in these dropout algorithms.

The update rule (2.13) together with (2.1) describes different dropout algorithms. In canonical *Dropout* [117], $F_{i,r,l'}=F_{i,r,l}\sim\mathrm{Bernoulli}(p)$ for any $l,l'\in[d_i]$ with $p=1/2$. In *Dropconnect* [115], $F_{i,r,l}\sim\mathrm{Bernoulli}(p)$ for all $i,r,l$ with $p=1/2$. In *Cutout* [64], $F_{1,r,l}=0$ whenever $|r-S_1|<c$, $c\in\mathbb{N}_+$ and $|l-S_2|<c$ with $(S_1,S_2)\sim\mathrm{Uniform}([d_1]\times[d_0])$. In fact, the class of dropout algorithms we consider is quite large. For example, $F^{[t]}$ can depend on $(X^{[t]},Y^{[t]})$, and $F_i^{[t]}$ does not need to have the same distribution as $F_j^{[t]}$ for $i\neq j$.

If $F^{[t]}$ is independent of $(X^{[t]},Y^{[t]})$ for each $t\in\mathbb{N}_0$ and $\Omega$ countable, then the dropout algorithm's risk function of (2.3) simplifies to

$$\mathcal{D}(W) = \sum_f \mathbb{P}[F=f]\sum_{x,y} l(\Psi_{f\odot W}(x),y)\mathbb{P}[(X,Y)=(x,y)]. \tag{2.14}$$

Here the sums are over all possible outcomes of the random variables $F$ and $(X,Y)$, respectively.

## 2.3 Convergence of projected dropout algorithms

Our first result pertains to the convergence of dropout algorithms for a wide range of activation functions and dropout filters. While convergence is expected in practice, we prove such convergence rigorously. In order to control the iterates of the stochastic algorithm, we project the iterates into a compact set. The projection assumption is common when investigating the convergence of stochastic algorithms [140, 127, 149, 61]; it essentially bounds the weights. For example, for $V^{[t]} \in \mathbb{R}$ and an update function $f : \mathbb{R} \to \mathbb{R}$, $f(V^{[t]})$ is projected onto an interval $[a,b]$ is by clipping and setting $V^{[t+1]} = \min\{\max\{f(V^{[t]}),a\},b\}$. There are also results involving bounds on the generalization gap defined in (1.10) for NNs where bounded weights play a role in avoiding overfitting [97].

### 2.3.1 Almost sure convergence

We first consider the notation and assumptions regarding the projection step of SGD. Let $\mathcal{H} \subseteq \mathcal{W}$ be a convex compact nonempty set and let $\mathrm{P}_{\mathcal{H}} : \mathcal{W} \to \mathcal{H}$ be the projection onto $\mathcal{H}$. By compactness and convexity of $\mathcal{H}$, the projection is unique. In a projected dropout algorithm, the weight update in (2.13) is replaced by (2.2). Because of the projection, our analysis will tie the limiting behavior of (2.2) to a *projected* ODE. To state such type of ODE, we need to define a *constraint term* $\pi(W)$, which is defined as the minimum vector required to keep the solution of the gradient flow

$$\frac{\mathrm{d}W}{\mathrm{d}t} = -\nabla_W \mathcal{D}|_{\mathcal{H}}(W) + \pi(W) \tag{2.15}$$

in $\mathcal{H}$. Appendix 2.C defines the projection term carefully for the case that $\mathcal{H}$'s boundary is piecewise smooth. Finally, define the set of stationary points

$$S_{\mathcal{H}} \triangleq \{W \in \mathcal{H} :\ -\nabla_W \mathcal{D}|_{\mathcal{H}}(W) + \pi(W) = 0\}. \tag{2.16}$$

The set $S_{\mathcal{H}}$ can be divided into a countable number of disjoint compact and connected subsets $S_1, S_2, \cdots$, say. We choose the following set of assumptions:

(N1) $\sigma \in C^2_{\mathrm{PB}}(\mathbb{R})$.

(N2) $\mathbb{E}[\|Y\|_2^m \|X\|_2^n] < \infty \forall m \in \{0,1,2\}, n \in \mathbb{N}_0$.

(N3) The random variables $(F^{[t]}; X^{[t]}; Y^{[t]})_{t\in\mathbb{N}}$ are independent copies of $(F, X, Y)$.

(N4) The step sizes $\alpha^{\{t\}}$ satisfy

$$\sum_{t=1}^{\infty} \alpha^{\{t\}} = \infty, \quad \sum_{t=1}^{\infty} (\alpha^{\{t\}})^2 < \infty. \tag{2.17}$$

(N5) $\sigma \in C^r_{\mathrm{PB}}(\mathbb{R})$, with $\dim(\mathcal{W}) \leq r$.

(N6) $-\nabla_W \mathcal{D}|_{\mathcal{H}}(W) + \pi(W) \neq 0$ whenever $\nabla_W \mathcal{D}|_{\mathcal{H}}(W) \neq 0$.

We are now in position to state our first result:

**Proposition 1.** *Let* $\{W^{[t]}\}_{t\in\mathbb{N}_0}$ *be the sequence of random variables generated by* (2.2) *with* (2.1) *on a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$. *Under assumptions (N1)–(N4) , there is a set* $N \subset \Omega$ *of probability zero such that for* $\omega \notin N$, $\{W^{[t]}(\omega)\}$ *converges to a limit set of the projected ODE in* (2.15). *If moreover (N5)–(N6) hold, then for almost all* $\omega \in \Omega$, $\{W^{[t]}(\omega)\}_{t\in\mathbb{N}}$ *converges to a unique point in* $\{W \in \mathcal{H} | \nabla \mathcal{D}|_{\mathcal{H}}(W) = 0\}$.

Theoretically, Proposition 1 guarantees that projected dropout algorithms converge for regression with the $\ell_2$-norm almost surely. Proposition 1 implies that if one is using a regular *nonprojected* dropout algorithm and one sees that the iterates $\{W^{[t]}\}_{t>0}$ are bounded, then these iterates are in fact converging to a stationary point of (2.3). Assumptions (N5)–(N6) are technical but are expected to hold in many cases. In particular, (N5) holds for the uniformly convergent approximation to a ReLU activation function given by softplus $\sigma_t(x) = \log(1+\exp(tx))/t$, and holds for many smooth activation functions. Also (N6) is expected to hold when $\mathcal{H}$ is generic polytope for which the gradient $\nabla\mathcal{D}$ is not exactly orthogonal to the normal to the surface.

Observe also that Proposition 1 holds remarkably generally. For example, the dependence structure of $(F,X,Y)$ as random variables is not restricted; it covers commonly used dropout algorithms such as *Dropout*, *Dropconnect*, and *Cutout*; and it holds for differentiable activation functions. Proposition 1 includes also online and offline learning, depending on the distribution $\mu$ from which we sample.

Our proof of Proposition 1 is in Appendix 2.D and relies on the framework of stochastic approximation in [140, Theorem 2.1, p. 127]. In the background the stochastic process $\{W^{[t]}\}_{t>0}$ is being scaled in both parameter space and time so that the resulting sample paths provably converge to the gradient flow in (2.15). Examining the proof, we expect that Proposition 1 can be extended to cases where the filters as random variables have finite moments, for example, when they are Gaussian distributed [68]. Concretely, the proofs of Lemmas 4 and 5 in Appendix 2.D rely only on the assumption that $F$ has finite moments, and may therefore be extended.

## 2.3.2 Generic sample complexity for dropout SGD

Examining Proposition 1, we note that it does not give insight into the convergence rate or the precise stationary point of $\mathcal{D}(W)$ to which the iterates $\{W^{[t]}\}$ converge. A related goal in stochastic optimization is to ask for the number of iterations of (2.13) required to achieve a point close to stationarity in expectation, also referred to the sample complexity of the algorithm. We say $W \in \mathcal{W}$ is an $\epsilon$-stationary point of a differentiable function $\mathsf{D}$ if $\|\nabla\mathsf{D}(W)\|_2 \leq \epsilon$ holds. For nonconvex functions $\mathsf{D}$ with a Lipschitz continuous gradient $\nabla\mathsf{D}$, SGD convergence to an $\epsilon$-stationary point in expectation can be achieved in $O(\epsilon^{-4})$ iterations; see [50, 20].

We will look at nonconvex functions with a Lipschitz continuous gradient and assume that the filters $F$ and the data $Z = (X,Y)$ are independent. We will also assume that the distribution of $Z$ is well-behaved so as to guarantee that we also have the following relations for the functions $\mathsf{D}, \mathsf{U}$ and $r$:

$$\begin{aligned} \mathsf{D}(W) &= \mathbb{E}_F[\mathsf{U}(F \odot W)] = \mathbb{E}_{F,Z}[r(F \odot W, Z)], \quad \text{and} \\ \nabla\mathsf{D}(W) &= \mathbb{E}_F[F \odot \nabla\mathsf{U}(F \odot W)] = \mathbb{E}_{F,Z}[F \odot \nabla r(F \odot W, Z)]. \end{aligned} \tag{2.18}$$

Note that the function $r$ in this setting includes the loss function formulation from (2.10) with

$$r(W,Z) = l(\Psi_W(X), Y), \qquad \text{and} \qquad Z = (X,Y). \tag{2.19}$$

In the case of *dropout*, for example, we expect that the sample complexity of finding an $\epsilon$-stationary point for the empirical risk will change depending on the dropout probability $1-p$. In particular, if $p$ is very small and $\|\nabla\mathsf{U}(W)\|_\infty < C$ holds for any $W \in \mathcal{W}$, then

$\nabla\mathsf{D}(W) = \mathbb{E}_F(F \odot \nabla\mathsf{U}(F \odot W)) = O(pC)$ as $p \to 0$. On the other hand if $p \simeq 1$, then the variance of $F \odot \nabla\mathsf{U}(F \odot W)$, will also be small. We make these intuitions rigorous in the next proposition. We denote $\mathcal{W} = \mathbb{R}^N$ for some $N \in \mathbb{N}$ to be the parameter space and $z \in \mathcal{Z} \subseteq \mathbb{R}^d$ a Lebesgue measurable set. We assume the following:

(Q1) $r \in C^1(\mathcal{W}, \mathcal{Z})$ and $\sup_{W \in \mathcal{W}, Z \in \mathcal{Z}} |r(W,Z)| < M$.
(Q2) $\sup_{W \in \mathcal{W}, Z \in \mathcal{Z}} \|\nabla r(W,Z)\|_2 < S$.
(Q3) $\nabla\mathsf{U}(W)$ is Lipschitz with Lipschitz constant $\ell$ (also referred to as $\mathsf{U}$ being $\ell$-smooth).
(Q4) The random variable $F : \Omega \to \{0,1\}^N$ satisfies $\mathbb{E}[F] = p(1,\dots,1) \in \mathcal{W}$ for $p \in (0,1]$.
(Q5) The iterates $(W^{[t]})_t$ of (2.13) are bounded, that is, $\sup_t \|W^{[t]}\|_2 < R$ almost surely.

Except for (Q4) and (Q5), all other assumptions are routinely used in sample complexity analysis. While the assumptions of Proposition 2 below hold for general nonconvex smooth functions $\mathsf{D}$, in the case of NNs and the setting in (2.19) we remark that there are examples that satisfy these assumptions such as the following one:

**Example 3.** *In a binary classification setting, we assume that the set $\mathcal{Z}$ is compact, that is, the data pairs $(x,y) \in \mathcal{Z}$ take values in a compact set where $y \in \{0,1\}$ are labels for the two classes. A NN, denoted by $\tilde{\Psi}_W(\cdot)$, uses sigmoid activation functions $\sigma(t) = 1/1+\exp(-t)$ with output in $\mathbb{R}$. The output of $\tilde{\Psi}_W$ is then used for binary classification with a logistic map, that is, the predicted probability of belonging to one of the classes is given by $\Psi_W(x) = 1/(1+\exp(-\tilde{\Psi}_W(x))$. In this setting, assumptions (Q1)–(Q3) will hold if the loss l is also smooth (such as the $\ell_2$-loss). In this case, in the notation of Section 2.2, we have $\mathcal{D}(W) = \mathsf{D}(W)$.*

Regarding (Q4), note that it allows for dependencies between filters. We also assume (Q5) for the sake of simplicity: we could instead use projected SGD with updates from (2.2) instead of (Q5), but using projected SGD would leave the scalings in $p$ and $T$ invariant.[2] Recall that $\mathsf{D}(W) = \mathbb{E}_F[\mathsf{U}(F \odot W)]$.

**Proposition 2.** *Let $(F^{[t]})_{t\in\mathbb{N}}$ be a sequence of independent random variables with distribution $F$. Let $W^{[t]}$ be iterates of (2.13). Assume (Q1)–(Q5). Define $J = S^2 + \frac{3}{2}N^2(\ell^2R^2 + 2\ell R)$.*
(a) *Let $T \in \mathbb{N}_+$. If $p > 2M\ell/(NS^2T)$, then there exists a constant stepsize $\alpha^{\{t\}} = \eta > 0$ such that for all $t \in [T]$,*

$$\min_{t\in[T]} \mathbb{E}\Big[\|\nabla\mathsf{D}(W^{[t]})\|_2^2\Big] \le 4\sqrt{p(S^2 + (1-p)J)}\sqrt{\frac{M\ell N}{T}}. \tag{2.20}$$

(b) *Let $T \geq 4$. There exists a sequence of decreasing stepsizes satisfying $\alpha^{\{t\}} = 1/(\ell\sqrt{t})$ for all $t \in [T]$ such that*

$$\min_{t\in[T]} \mathbb{E}\Big[\|\nabla\mathsf{D}(W^{[t]})\|_2^2\Big] \le \frac{4M\ell^2 + 4Np(S^2 + (1-p)J)\log(T)}{\sqrt{T}}. \tag{2.21}$$

[2] With projected SGD, we would moreover have to use the expression $\nabla\mathsf{U}^p(w) = (w - \mathrm{P}_{\mathcal{H}}(W^{[t]} - \alpha^{\{t+1\}}\Delta^{[t+1]}))/\alpha^{\{t+1\}}$, which makes the analysis more tedious. Note that $\nabla\mathsf{U}^p(w) = \nabla\mathsf{U}(w)$ whenever $w \in \mathrm{int}(\mathcal{H})$. See [89] for an example of such analysis.

In Proposition 2, we observe that finding approximate stationary points with dropout is easier with a larger dropout probability $1-p$ for a wide range of filter distributions like *dropout* and *dropconnect*, as guaranteed by (Q4). In Proposition 2(a) we see a dependence of the convergence rate on $\sqrt{p(S^2+(1-p)J}$. The term $pS^2$ corresponds to the variance of the gradient due the distribution of data in $\mathcal{Z}$ and decreases with $p$; while the term $p(1-p)J$ stems from the variance due to dropout. Note that the sum achieves a maximum for $p \in (0,1)$. We note that Proposition 2 does not suggest that the convergence to minima, a subset of the stationary points, is faster for smaller $p$. Indeed, as seen in the numerical experiments in Section 2.5.1, the next chapter, or in similar work in [25], the NN structure and data distribution changes the convergence rate dependence on the dropout probability considerably. In particular, we will see in the next chapter that for shallow linear NNs the local convergence rate of dropout close to a minimum seems to decrease as $p \uparrow 1$. Similarly, smaller $p$ does not necessarily improve generalization. In particular, if the dropout probability $1-p$ is large, the optimization landscape will be flat with many approximate stationary points. In this case, SGD with dropout with a limited sample complexity of $T$ iterations will not explore the landscape as much as when using a smaller dropout probability. With a flatter landscape in mind, it may be better in the complexity trade-off to use a larger $p$ for finding an approximate minimum and generalize better instead of finding a stationary point.

In Section 2.3 we discussed a generic convergence rate for the empirical risk of dropout. In Section 2.4 we will next compute the convergence rate explicitly for NNs that are shaped like arborescences.

## 2.4 Convergence rate of gradient descent for arborescences with linear activations

In the previous section we have obtained a convergence guarantee as well as a bound for the sample complexity of dropout. In this section, we focus on the convergence rate of dropout in functions that have the structure of NNs. In particular, we will derive an explicit convergence rate for dropout algorithms in the case that we have linear activations $\sigma(z)=z$ and that the NN is structured as an arborescence: see Figure 2.1c. Specifically, we will study the following regular GD algorithm on dropout's risk function:

$$W^{\{t+1\}} = W^{\{t\}} - \alpha \nabla \mathcal{D}(W^{\{t\}}) \quad \text{for} \quad t \in \mathbb{N}_0. \tag{2.22}$$

Here, we keep the step size $\alpha > 0$ fixed. Note that this algorithm generates a deterministic sequence $\{W^{\{t\}}\}_{t\in\mathbb{N}_0}$ as opposed to a sequence of random variables $\{W^{[t]}\}_{t\in\mathbb{N}_0}$ as generated by (2.13),or (2.1). We will use a linear activation function $\sigma(t)=t$, which combined with the arborescence structure will allow us to obtain an explicit convergence rate. While the iterates of (2.22) are not stochastic, analogous to Proposition 1, the stochastic iterates will converge to a gradient flow of an ODE, whose discretization is given in (2.22). Analyzing ODEs related to NNs is common in literature [17, 56]. For more discussion on the relationship between the iterates of (2.22) and dropout we refer to Appendix 2.B.

Our main convergence result in Proposition 4 below holds for general distribution functions. However we show here the cases of *Dropout* and *Dropconnect*, which are most insightful. We use the following notation adapted from graph theory. Consider a fixed, directed *base graph* $G=(\mathcal{E},\mathcal{V})$ without cycles in which all paths have length $L$, which describes a NN's structure as follows. Each vertex $v\in\mathcal{V}$ represents a neuron of the NN, and each directed edge $e=(u,v)\in\mathcal{E}$ indicates that neuron $u$'s output is input to neuron $v$. Note that with each edge $e\in\mathcal{E}$ in the NN, a weight $W_e\in\mathbb{R}$ and a filter variable $F_e\in\{0,1\}$ are associated. We will write $\mathcal{W}=\mathbb{R}^{|\mathcal{E}|}$ for simplicity. For an arborescence $G$, we denote by $\mathcal{L}(G)$ the edge set of leaves. Let $M>2\delta>0$ be real numbers and suppose that we initialize the weights $\{W_e\}_{e\in\mathcal{E}}$ as follows:

$$\begin{aligned} &M > W_e^{\{0\}} > \sqrt{2}\delta \text{ for } e\in\mathcal{E}\backslash\mathcal{L}(G)\\ &|W_l| \leq \delta/\sqrt{|\mathcal{L}(G)|} \text{ for } l\in\mathcal{L}(G). \end{aligned} \tag{2.23}$$

The proof of Proposition 3 below is deferred to Appendix 2.I, which is a consequence of our more general result in Proposition 4.

**Proposition 3.** *Assume that the base graph $G$ is an arborescence of depth $L$ with $|\mathcal{L}(G)|$ leaves, the activation function $\sigma(t)=t$ is linear, $F$ is independent of $(X,Y)$, and $\{W_e^{\{0\}}\}_{e\in\mathcal{E}}$ is initialized according to* (2.23)*. If the $\{F_e\}_{e\in\mathcal{E}}$ follow the distribution prescribed by* Dropconnect *or* Dropout*, then there exists $\alpha>0$ such that the iterates of* (2.22) *satisfy*

$$\mathcal{D}(W^{\{t\}})-\mathcal{D}(W^{\text{opt}}) \leq \big(\mathcal{D}(W^{\{0\}})-\mathcal{D}(W^{\text{opt}})\big)\exp(-\omega t/2). \tag{2.24}$$

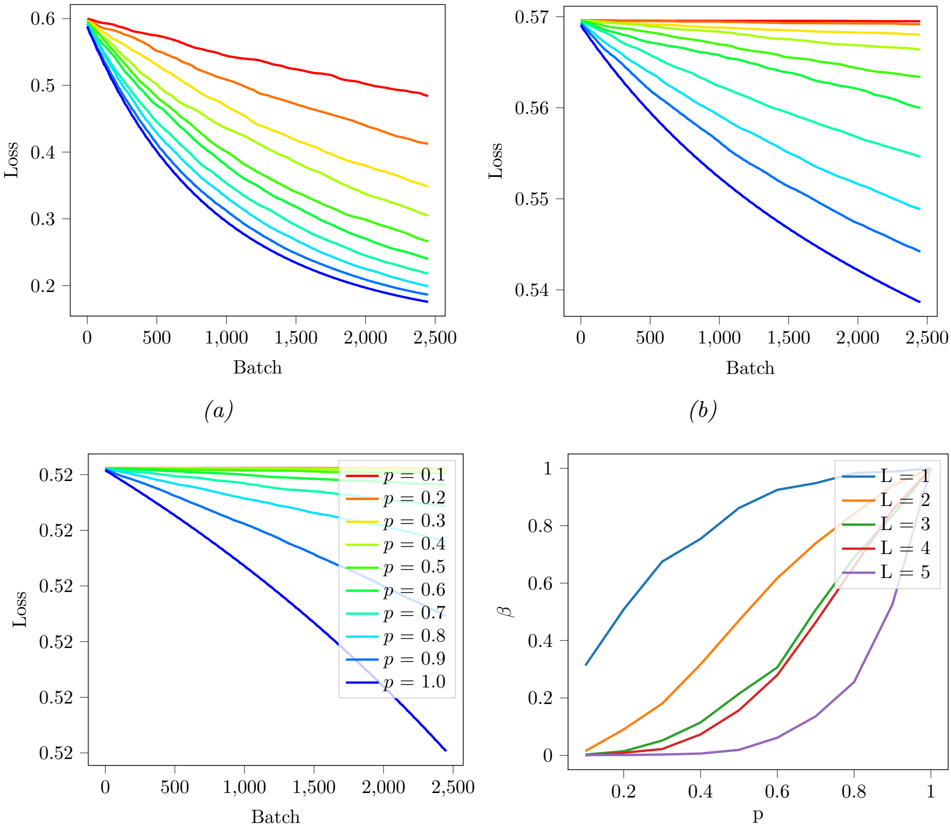
*with*

$$\omega = \mathrm{O}\Big(\frac{p^L}{L|\mathcal{L}(G)|^2}\Big(\frac{2\delta^2}{M^2}\Big)^{2L}\Big). \tag{2.25}$$

### 2.4.1 Discussion

In Proposition 3 we consider the cases of *Dropout* and *Dropconnect*, in which nodes or edges are dropped with probability $1-p$, respectively. Observe that the convergence rate exponent depends on $p^L$ and $(2\delta^2/M^2)^{2L}$ where $2\delta^2/M^2<1$; see (2.23). The first term in particular indicates that as the NN becomes deeper, the convergence rate exponent of GD with *Dropout* or *Dropconnect* will decrease by a factor $p^L$. The second term $(2\delta^2/M^2)^{2L}$ shows the increased difficulty of training deeper NNs and has been observed e.g., by [44, 35]. The exponential dependence in $L$ is moreover tight when using GD and is intrinsic to the method [44]. Hence, dropout adds another exponential dependence to the convergence rate in arborescences, which is due to the stochastic nature of the algorithm. In Figure 2.2 an experiment confirming this intuition on the convergence rate of dropout on a single path for different depths can be seen.

Finally, our proofs of Proposition 3 and the related more general result in Proposition 4 below can be found in Appendix 2.H. The proof strategy is to show that a Polyak–Łojasiewicz (PL) inequality holds, which allows one to obtain convergence rates for GD on nonconvex functions [77]. The new part of the argument is that we use conserved quantities and a double induction to identify a compact set in which the iterates remain and simultaneously a PL inequality holds. The method that we develop depends intricately on the arborescence structure and cannot be readily applied to other cases. We provide a sketch of the proof in the next section.



*(a)* *(b)*

*(c)* *(d)*

*Figure 2.2: The average loss depending on the number of steps of SGD with dropout of the function $f(w) = (y - \prod_{i=1}^{L} w_i x)^2$ and its average convergence slope.* (a) *The average loss for $L = 1$.* (b) *The average loss for $L = 3$.* (c) *The average loss for $L = 5$.* (d) *The slope $\beta$ of the fit of $y = -\beta x + \gamma$ for the curves in (a), (b) and (c). The slopes $\beta$ for a given $l$ have been normalized at $p = 1$ for comparison across depths $L$. Note that for larger $L$, the effect of $p$ becomes also more pronounced. This is in agreement with the conclusion in Section 2.4, where we expect a convergence rate depending on $p^L$. In this case, other effects of depth are also observed, such as a dependence on the initialization.*

### 2.4.2 Sketch of the proof

Besides the previous notation, we need to introduce the notation corresponding to subgraphs and paths. Let $\mathcal{G}$ be the set of all subgraphs of the base layered directed graph $G$ with $d$ vertices, and let $\mathcal{E}(g)$ be the set of edges of a subgraph $g \in \mathcal{G}$. Let $\Gamma_i^j(g;e)$ be defined as the set of all paths in the directed graph $g$ that start at vertex $i$, traverse edge $e$, and end at vertex $j$. If the origin or end vertices are in the input or output layer, the subscript or superscript is dropped from the notation, respectively. For every path $\gamma \triangleq (\gamma_1,\ldots,\gamma_L) \in \Gamma(g)$, we write $P_\gamma \triangleq \prod_{e\in\gamma} W_e$ and $F_\gamma \triangleq \prod_{e\in\gamma} F_e$ for notational convenience. Finally, let $G_F \triangleq (\mathcal{E}_F,\mathcal{V})$ be the random subgraph of base graph $G$ that has edge set $\mathcal{E}_F \triangleq \{e \in \mathcal{E} | F_e = 1\}$. We denote $\mu_g \triangleq \mathbb{P}[G_F = g]$, and $\eta_\gamma \triangleq \sum_{\{g\in\mathcal{G}|\gamma\in\Gamma(g)\}} \mu_g$. We first provide an explicit characterization of dropout's risk function in (2.3) in terms of paths in the graph that describes the structure of the NN. This is possible since we assume linear activation functions. The following lemma now holds, and is proved in Appendix 2.F.

**Lemma 1.** *Assume that the base graph $G$ is a fixed, directed graph without cycles in which all paths have length $L$ and there are $d_L$ output nodes (N6'), that $\sigma(t) = t$ (N7), and that $F$ is independent of $(X,Y)$ (N8). Then*

$$\mathcal{D}(W) = \sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{s=1}^{d_L}\big(Y_s - \sum_{\gamma\in\Gamma^s(g)} P_\gamma X_{\gamma_0}\big)^2\Big]. \tag{2.26}$$

*Moreover $\mathcal{D}(W) = \mathcal{J}(W) + R(W)$, where*

$$\mathcal{J}(W) = \sum_{\gamma\in\Gamma(G)} \eta_\gamma \mathbb{E}[(Y_{\gamma_L} - P_\gamma X_{\gamma_0})^2], \tag{2.27}$$

$$R(W) = -\sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{s=1}^{d_L}\sum_{\gamma\in\Gamma^s(g)}\Big(\Big(1-\frac{1}{|\Gamma^s(g)|}\Big)Y_s^2 - P_\gamma X_{\gamma_0}\sum_{\delta\in\Gamma^s(g)\setminus\{\gamma\}} P_\delta X_{\delta_0}\Big)\Big]. \tag{2.28}$$

*Here, the constants $\eta_\gamma, \mu_\gamma$ depend explicitly on $F$'s distribution and the NN's architecture.*

Note that Lemma 1 essentially changes variables to rewrite the dropout risk function as a sum over paths instead of a sum over graphs. This representation allows us to clearly identify the regularization term $R(W)$. For example in the case of *Dropconnect* [115], where the filter variables $\{F_e\}_{e\in\mathcal{E}}$ are independent random variables with distribution Bernoulli($p$), Lemma 1 holds with $\mu_g = p^{|\mathcal{E}(g)|}(1-p)^{|\mathcal{E}(G)|-|\mathcal{E}(g)|}$. Also note that if for all subgraphs $g \in \mathcal{G}$ and vertices $i \in [d]$ the number of paths that end at $i$ satisfies $|\Gamma^i(g)| = 1$ , such as when $G$ is an arborescence, then for all subgraphs $g \in \mathcal{G}$ and paths $\gamma \in \Gamma(g)$ there is only one path ending at a leave node $\gamma_L$, that is, $\Gamma^{\gamma_L}(g) = \{\gamma\}$.

We now focus on a base graph that is an arborescence of arbitrary depth; see Figure 2.1c. Hence we now replace (N6') in Lemma 1 that assumes a generic graph by assumption (N6), where $G$ is specifically an arborescence. The following specification of Lemma 1 is also proven in Appendix 2.F.

**Corollary 1.** *Assume that the base graph $G$ is an arborescence of depth $L$ (N6), and*

*(N7)–(N8) from Lemma 1. Then* $\mathcal{D}(W) = \mathcal{I}(W) + \mathcal{D}(W^{\mathrm{opt}})$*, where*

$$\mathcal{I}(W) \triangleq \sum_{\gamma\in\Gamma(G)} \nu_\gamma (z_\gamma - P_\gamma)^2,$$

$$\mathcal{D}(W^{\mathrm{opt}}) = \sum_{\gamma\in\Gamma(G)} \eta_\gamma (\mathbb{E}[Y_{\gamma_L}^2] - \mathbb{E}[Y_{\gamma_L} X_{\gamma_0}]^2 / \mathbb{E}[X_{\gamma_0}^2]), \tag{2.29}$$

*and* $\nu_\gamma \triangleq \eta_\gamma \mathbb{E}[X_{\gamma_0}^2]$*,* $z_\gamma \triangleq \mathbb{E}[Y_{\gamma_L} X_{\gamma_0}]/\mathbb{E}[X_{\gamma_0}^2]$ *for* $\gamma \in \Gamma(G)$*. Consequently,* $R(W) = 0$ *for an arborescence.*

The convergence result we are about to show uses the fact that for the system of ODEs $\mathrm{d}W/\mathrm{d}t = -\nabla_W \mathcal{D}(W)$ there are conserved quantities. Within the proof, these conserved quantities have the crucial role of guaranteeing compactness for the iterates. Specifically, let $\mathcal{L}(g; f)$ denote the leaves of the subtree of $g \in \mathcal{G}$ rooted at a vertex $f \in \mathcal{E}(g)$, and define the set of leaves of $G$ as $\mathcal{L}(G) \triangleq \cup_{f\in\mathcal{E}} \mathcal{L}(G; f)$. We remark that in the previous notation $d_L = |\mathcal{L}(G)|$. For $W \in \mathcal{W}$ and each leaf $f \in \mathcal{E}\backslash\mathcal{L}(G)$, define the quantity

$$C_f = C_f(W) \triangleq W_f^2 - \sum_{l\in\mathcal{L}(G;f)} W_l^2. \tag{2.30}$$

Define $C_{\min} \triangleq \min_{e\in\mathcal{E}\backslash\mathcal{L}(G)} C_e$ and $C_e^{\{t\}} = C_e(W^{\{t\}})$ for $t \in \mathbb{N}_+$ also, both of which we require later. Lemma 2 now proves that the function $C_f$ in (2.30) is a conserved quantity; the proof is in Appendix 2.G.

**Lemma 2.** *Assume (N2) from Proposition 1, (N6) from Corollary 1 , (N7), (N8) from Lemma 1. Then under the negative gradient flow* $\mathrm{d}W/\mathrm{d}t = -\nabla\mathcal{D}(W)$*,*

$$\frac{\mathrm{d}C_f}{\mathrm{d}t} = 0 \tag{2.31}$$

*for all* $f \in \mathcal{E}\backslash\mathcal{L}(G)$*.*

We are almost in position to state our second result, but need to introduce still some notation. We define the following constants

$$\|\nu\|_1 \triangleq \sum_{\gamma\in\Gamma(G)} \nu_\gamma, \quad \nu_{\min} \triangleq \min_{\gamma\in\Gamma(G)} \nu_\gamma, \quad \nu_{\max} \triangleq \max_{\gamma\in\Gamma(G)} \nu_\gamma \tag{2.32}$$

for notational convenience. Also, for $0 < \delta < M$, we define

$$\mathcal{S} \triangleq \{W \in \mathcal{W} : M > \left|W_f\right| > \delta > 0 \ \forall f \in \mathcal{E}(G)\backslash\mathcal{L}(G); M > \left|W_f\right| \ \forall f \in \mathcal{L}(G)\}, \tag{2.33}$$

a bounded set of parameters where if the weight is associated with a leaf, they are furthermore bounded away from zero. Let finally

$$B(\epsilon, I) \triangleq \left\{W \in \mathcal{W} : \mathcal{I}(W) \le \epsilon, W_f^2 - \sum_{l\in\mathcal{L}(G;f)} W_l^2 \in I_f \text{ for } f \in \mathcal{E}\backslash\mathcal{L}(G)\right\} \tag{2.34}$$

denote the set of all weight parameters that are $\varepsilon$-close to a critical point and for which the conserved quantities in (2.30) deviate by no more than $O(C_f^{\{0\}})$ from their initial value $C_f^{\{0\}}$. These deviations are made explicit by the intervals

$$I_f \triangleq [C_f^{\{0\}}/2, 3C_f^{\{0\}}/2] \text{ for } f \in \mathcal{E}\backslash\mathcal{L}(G), \text{ and the set } I \triangleq \times_{f\in\mathcal{E}\backslash\mathcal{L}(G)} I_f \subseteq \mathbb{R}^{|\mathcal{E}|-|\mathcal{L}(G)|}. \tag{2.35}$$

Our proof shows that the iterates $\{W^{\{t\}}\}_{t\geq 0}$ stay in the intersection $\mathcal{S}\cap B(\varepsilon, I)$, and this implies that the weights (including those associated with the leaves) remain bounded. The following now holds, and its proof can be found in Appendix 2.H.

**Proposition 4.** *Assume (N2) from Proposition 1, (N6) from Corollary 1, (N7)–(N8) from Lemma 1, that* $W^{\{0\}} \in \mathcal{S}\cap B(\epsilon, I)$ *and* $M^L \geq |z_\gamma|$ *for all* $\gamma \in \Gamma(G)$ *(N9), that* $\frac{1}{2}C_{\min}(W^{\{0\}}) > \delta^2$ *(N10). If*

$$\alpha \leq \min\Big(\nu_{\min}\frac{\mathrm{e}^{1/2}(C_{\min}^{\{0\}})^L}{16\,\|\nu\|_1\,LM^{2(L-1)}\mathcal{I}(W^{\{0\}})}, \frac{1}{12\nu_{\max}\,|\mathcal{E}|\,|\Gamma(G)|\,M^{2(L-1)}}, \frac{1}{2\nu_{\min}(C_{\min}^{\{0\}})^{L-1}}\Big), \tag{2.36}$$

*then the iterates of* (2.22) *satisfy*

$$\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\text{opt}}) \leq \big(\mathcal{D}(W^{\{0\}}) - \mathcal{D}(W^{\text{opt}})\big)\exp(-\tfrac{\alpha\tau}{2}t). \tag{2.37}$$

*where* $\tau = 4\nu_{\min}\exp(-1/2)(C_{\min}^{\{0\}})^{L-1}$.

Proposition 4 identifies explicitly how the convergence rate of GD on a dropout's risk function depends on the dropout algorithm and the structure of the arborescence: parameters such as $p, |\mathcal{L}(G)|, L$ are implicitly present in the constants $\nu_{\min}$ and $\|\nu\|_1$ in $\alpha, \tau$.

Note that Assumptions (N9)–(N10) are relatively benign. These assumptions are for example satisfied when initializing $M > W_e^{\{0\}} > \sqrt{2}\delta$ for $e \in \mathcal{E}\backslash\mathcal{L}(G)$ and setting $|W_l| \leq \delta/\sqrt{|\mathcal{L}(G)|}$ for all $l \in \mathcal{L}(G)$ and $\epsilon = \mathcal{I}(W^{\{0\}})$, which we assume in Proposition 3. In other words, this initialization sets the weights that are associated with leaves small compared to all other weights.

## 2.5 Effect of dropout on the convergence rate in wider networks

In Proposition 4, we have proven that the convergence rate depends on $p^L$ for NNs shaped like arborescences. Let $G_{\text{tree}}$ be a tree and $e \in \mathcal{E}(G_{\text{tree}})$ be an edge. Denote by $\Gamma^{[t]}(e)$ the set of paths passing through $e$ that are not filtered by dropout at time $t$. We observe that at any given time $t$ of dropout SGD,

$$\mathbb{P}[w_e^{[t]} \text{ is updated}] = \mathbb{P}[\Gamma^{[t]}(e) \neq \emptyset] = p^L. \tag{2.38}$$

If we denote by $t_{\text{update}}(G_{\text{tree}}) = 1/p^L$ the average update time for a weight in $G_{\text{tree}}$, then we need $1/p^L$ more time on average for a given edge to be updated than when we do not use dropout. For wider networks $G$, however, edges can be updated simultaneously and repeatedly via different available paths. By the previous intuition we might still expect that, if the updates are sufficiently independent, the convergence rate depends approximately on $1/t_{\text{update}}$. In order to verify this intuition we will determine $t_{\text{update}}$ for NNs that are much wider than deep, and later simulate their convergence rates also in realistic settings.

Suppose now that $G$ is a graph of a fully-connected NN with $L$ dropout layers each of which has width $D$. For each of the vertices $u \in G$ in a dropout layer, there is an associated dropout filter variable $F_u \sim_{\text{i.i.d.}} \text{Ber}(p)$ where $p > 0$ is fixed. That is, we use *dropout.* Note that any other additional input or output layer without filters only changes the number of paths by a multiplicative factor. Hence, we will restrict to the case that all nodes in the layers have filter variables. In this case, we may consider a path $\gamma = (u_1, \dots, u_L)$ as a set of $L$ vertices—one for each dropout layer—instead of edges. For two paths $\gamma$ and $\delta$, we consider their intersection $\gamma \cap \delta$ as the subset of vertices belonging to both paths. Hence, $|\gamma \cap \delta| = l$ implies that the intersection has $l$ vertices, not necessarily forming a path.

We remark that we can restrict to the case $L > 2$. In the case of one dropout layer $L = 1$, an edge $e = (u,v)$ conected to a dropout node $u$ is updated if and only if the filter $F_u = 1$, where $u \in G$ is the adjacent vertex to $e$ with a dropout filter, so that in this case $\mathbb{P}[w_e^{[t]}$ is updated$] = 1 - p$. For $L = 2$, an edge $e = (u,v)$ is updated if and only if $F_u = F_v = 1$, so that $\mathbb{P}[w_e^{[t]}$ is updated$] = 1 - p^2$. Recall that we denote by $\Gamma(e)$ the set of paths $\gamma$ of $G$ passing through $e$. For a path $\gamma \in \Gamma(e)$, in the following, we let $F_\gamma = \prod_{u \in \gamma} F_u$ be the indicator of a path being filtered. Thus, $F_\gamma$ is 1 is $\gamma$ is not filtered and 0 otherwise. We will use Greek letters for paths and Latin letters for vertices when referring to filters $F_\gamma$ and $F_u$ respectively.

**Lemma 3.** *Let $G$ be a graph of a fully-connected NN with $L > 2$ dropout layers, each with the same width $D$ and with dropout filters $F_u$ for $u \in G$. For an edge $e \in \mathcal{E}(G)$, let $F_{\Gamma(e)} = \sum_{\gamma \in \Gamma(e)} F_\gamma$ denote the random variable that counts the number of nonfiltered traversing paths through $e$. If $L,p$ are fixed, then as $D \to \infty$,*

$$\mathbb{P}[F_{\Gamma(e)} = 0] = 1 - p^2 + O\Big(\frac{pL}{D}\Big). \tag{2.39}$$

*Proof.* We will use the Paley–Zygmund inequality. For a nonnegative random variable $Z$ with finite second moment, for any $\theta \in (0,1)$,

$$\mathbb{P}[Z > \theta \mathbb{E}[Z]] \geq (1-\theta)^2 \frac{\mathbb{E}[Z]^2}{\mathbb{E}[Z^2]}. \tag{2.40}$$

We will use (2.40) with the random variable $F_{\Gamma(e)}$. The idea is that if $D$ is much larger than $L$, the average number of paths passing through $e$ is also large. We are using dropout, so the filter variable corresponding to an edge $e = (u,v)$ will depend on only the vertex $u$, that is, $F_e = F_u$. For counting paths we also need to take into account that the filter $F_v$ will occurring in all paths passing through $e$. Since only the two vertices $u$ and $v$ of $e$ are fixed we can compute

$$\mathbb{E}[F_{\Gamma(e)}] = \sum_{\gamma \in \Gamma(e)} \mathbb{E}[F_\gamma] = p^L |\Gamma(e)| = p^L D^{L-2}. \tag{2.41}$$

We define the set of broken paths in $\Gamma(e)$ as

$$\Gamma_b(e) = \{\gamma = (u_{i_1}, \dots, u_{i_k}) \in G^k : \exists \eta, \delta \in \Gamma(e), \gamma = \eta \cap \delta\}, \tag{2.42}$$

that is, $\gamma \in \Gamma_b(e)$ if and only if there exist $\eta, \delta \in \Gamma(e)$ such that $\gamma = \eta \cap \delta$. In particular,

$\Gamma_b(e)$ contains paths and unions of vertices of paths that pass through $e$. Then we have:

$$\mathbb{E}[F_{\Gamma,e}^2] = \sum_{\gamma\in\Gamma(e)}\sum_{\delta\in\Gamma(e)}\mathbb{E}[F_\gamma F_\delta] \overset{\text{(i)}}{=} \sum_{\gamma\in\Gamma(e)}\sum_{l=2}^{L}\sum_{\substack{\delta\in\Gamma(e)\\|\gamma\cap\delta|=l}}\mathbb{P}[F_\gamma=1,F_\delta=1] \tag{2.43}$$

$$\overset{\text{(ii)}}{=} \sum_{\gamma\in\Gamma(e)}\sum_{l=2}^{L}\sum_{\substack{\delta\in\Gamma(e)\\|\gamma\cap\delta|=l}} p^l p^{2L-2l} \overset{\text{(iii)}}{=} \sum_{l=2}^{L}\sum_{\substack{\eta\in\Gamma_b(e)\\|\eta|=l}}\sum_{\substack{\gamma,\delta\in\Gamma(e)\\ \eta\subseteq\delta,\gamma\\ \gamma\cap\delta=\eta}} p^l p^{2L-2l} \tag{2.44}$$

$$\overset{\text{(iv)}}{=} \sum_{l=2}^{L}\sum_{\substack{\eta\in\Gamma_b(e)\\|\eta|=l}} (D(D-1))^{L-l}p^l p^{2L-2l} \tag{2.45}$$

$$\overset{\text{(v)}}{=} \sum_{l=2}^{L}\binom{L-2}{l-2}D^{l-2}(D(D-1))^{L-l}p^l p^{2L-2l} \tag{2.46}$$

$$= p^{2L-2}D^{2L-4} + O(Lp^{2L-3}D^{2L-5}), \tag{2.47}$$

where (i) we have first used that $F_\gamma$ are indicators for occurring $\gamma\in\Gamma(e)$ and that at least $l\geq 2$ since vertices $u$ and $v$ are shared among all paths in $\Gamma(e)$; secondly, that we have separated the sum over paths into a path $\gamma$ and all other paths $\delta$ that coincide in $l$ vertices. In (ii) we have computed the probability by noting that for $\gamma$ and $\delta$ such that $|\gamma\cap\delta| = l \geq 2$, $\mathbb{E}[F_\gamma F_\delta] = p^l p^{2L-2l}$, where the term $p^l$ accounts for the $l$ shared filters corresponding to $l$ shared vertices and $p^{2L-2l}$ for the remaining products of filters. Note that we have used the independence assumption for filters here. (iii) We have used here that $\eta = \delta\cap\gamma\in\Gamma_b(e)$, so that we can separate the previous sum into first, fixing the $l$ vertices where two paths intersect—including $e$—with $\eta\in\Gamma_b(e)$ such that $|\eta| = l$, and then looking for all possible $\delta,\gamma\in\Gamma(e)$ such that $\gamma\cap\delta=\eta$. For (iv) we fix $l$ vertices where $\gamma$ and $\delta$ coincide, then there are still $(D(D-1))^{L-l}$ possible ordered vertex pairs to choose from all the other vertices where $\gamma$ and $\delta$ do not coincide. (v) For the remaining sum, for each $l$ fixed locations—including the vertices of $e$, which are fixed—we can still choose $D^{l-2}$ remaining possible vertices. Additionally, there are for each $l$, $\binom{L-2}{l-2}$ distinct $l-2$ locations for these vertices. Hence, plugging (2.47) and (2.41) into (2.40) yields

$$\mathbb{P}[F_{\Gamma(e)} > \theta p^L D^{L-2}] \geq (1-\theta)^2\frac{p^{2L}D^{2L-4}}{p^{2L-2}D^{2L-4}+O(Lp^{2L-3}D^{2L-5}))} \tag{2.48}$$

$$= (1-\theta)^2\frac{p^2}{1+O(L/(Dp))} \tag{2.49}$$

$$= (1-\theta)^2(p^2+O(pL/D)). \tag{2.50}$$

In particular, setting $\theta^{-1} = 2p^L D^{L-2}$ and computing the higher order noting that $L>2$, we obtain that

$$\mathbb{P}[F_{\Gamma(e)} > 1/2] \geq p^2 + O(pL/D), \tag{2.51}$$

or alternatively noting that $\{F_{\Gamma(e)}\leq 1/2\} = \{F_{\Gamma(e)} = 0\}$, since $F_{\Gamma(e)}\in\mathbb{N}$ we obtain

$$\mathbb{P}[F_{\Gamma(e)} = 0] \leq 1 - p^2 + O(pL/D). \tag{2.52}$$

Finally note that $1-p^2 \leq \mathbb{P}[F_{\Gamma(e)}=0]$ since the edge $e$ can be present in a path only if the filters at both vertices of $e$ have value 1, which occurs with probability $p^2$, so that $\mathbb{P}[F_{\Gamma(e)}>0]<p^2$. □

Note that in the proof of Lemma 3 we can recover the scaling $p^L$ that we have seen in Proposition 4 by setting $D=1$ in (2.47) and in (2.45).

From Lemma 3 we expect that for a wide network with $L$ layers where $D \gg L$ and an edge $e \in \mathcal{E}(G)$, we have that

$$\mathbb{P}[w_e^{[t]} \text{ is updated}] = p^2 + O(pL/D). \tag{2.53}$$

If the convergence rate is related to the update rule, then we would expect that for a wide network the rate would be independent of $L$ which is different from the path network considered in Proposition 4. In the next section we will verify this intuition on real datasets. Note, however, that we do not expect to see the dependence on $p$ as shown in (2.53): this heuristic argument provides only the rate at which a weight is updated, and stochastic averaging is not solely driving the convergence rate. In particular, in the next chapter, we show that close to a critical point on a dropout ODE for wide shallow linear networks, the dependence scales with a factor $p(1-p)$ instead of $p$. This is due to the fact that for larger $p$, there are regions of the landscape close to minima that become flat, as also hinted by Proposition 2. Indeed, when $p \uparrow 1$ the term $(1-p)J \downarrow 0$ in the convergence rate of Proposition 2 lowers the complexity of finding an $\epsilon$-stationary point. Hence, there are landscape regimes and initialization issues that also account for the convergence rate in NNs.

### 2.5.1 Numerical Experiments

In this section we conduct the dropout stochastic gradient descent algorithm numerically,[3] for different datasets and network architectures. We measure the convergence rate for different widths $D$, depths $L$, and dropout probabilities $1-p$. We then compare these measurements to the bounds on the convergence rates obtained in Section 2.4. We use *Tensorflow*[4] for the implementation.

#### Setup

*Datasets.* We will consider three commonly used data sets of images: the MNIST[5] [125], CIFAR-100-fine[6], and CIFAR-100-coarse datasets [128].

*NN Architecture.* We use as a base architecture a LeNet with 11 layers where the two dense layers have been substituted with $L$ fully-connected ReLU layers of width $D$. Each of these layers have dropout with dropout probability $1-p$. While larger networks are commonly used in practice, a LeNet architecture is sufficient to test the effect of dropout on the convergence rate as we verify with the simulations.

[3] The source code of our implementation is available at https://gitlab.tue.nl/20194488/almost-sure-convegence-of-dropout-algorithms-for-neural-networks.

[4] https://www.tensorflow.org/

[5] Modified National Institute of Standards and Technology (MNIST)

[6] Canadian Institute For Advanced Research (CIFAR)

*Loss.* We use the cross-entropy loss, which is commonly used for classification. For two distributions $p$ and $q$ with support on $[n]$ labels, the cross-entropy loss is defined as

$$l(p,q) = -\sum_{i=1}^{n} q_i \log(p_i). \tag{2.54}$$

*Stopping criteria.* In all experiments, we stop after 40 epochs.

*Initialization.* In order to see the convergence rate close to a minimum. We use first a *Gaussian initialization*, that is, we set every weight on the dense layers to $W_{ijk} \sim \mathrm{Normal}(0, 1/\sqrt{D})$ in an independent manner, where $D$ is the width of the layer. While this initialization is standard, we note that we cannot expect to compare convergence rates for different numbers of layers $L \in \{1,2,3\}$ and for different dropout probabilities $1-p$, since the loss functions are also different. In the course of our experiments, we found that there are also many saddle points where SGD remains stuck, which complicated the estimation of the convergence rate. In order to start approximately at the same neighborhood where the iterates stay and continuously track minima across different choices of $p$, for each $L \in \{1,2,3\}$ we have used a two-step approach in order to avoid areas of the landscape with saddle points. We first run ADAM[7] for 2 epochs with $p = 0.1$ and store the weights. Secondly, for each $p \in P$ we then perform dropout SGD with initialization given by the stored weights. In this manner, we expect that we are approximately "tracking" the same local region across the optimization landscape when we change $p$. Optimization with ADAM is less prone to remain in flat areas of the landscape since it uses a dynamic step size. Hence, if after the dynamic step the iterates remain in a part of the landscape with no saddle points that smoothly changes with $p$, we also expect in this case to obtain comparable convergence rates for SGD for each fixed $L$.

*Step size and batch size.* In each experiment, the step size is given by $\eta = 10^{-5}$ and the batch size is $b = 1024$.

*Fitting procedure.* We fix a set of probabilities $P \subset [0,1]$ and depths $L = \{1,2,3\}$ and for each pair $(p,l) \in P \times L$ we run the algorithm above. From the value of the loss from all $T$ iterations of SGD $\mathcal{L} = (l_t)_{t=0}^{T}$ in one run, we compute a moving average $a(\mathcal{L})_{t=0}^{T}$, where we average the loss across a window with size given by the number of batches $n_b$ required to complete one epoch. In this manner we obtain an average convergence rate and diminish the stochasticity from the dataset. We then fit the averaged loss of the iterates $a(\mathcal{L})_{t=0}^{T}$ for each $p$ and $l$ to the function

$$f(\alpha_{p,l}, \beta_{p,l}, \gamma_{p,l}) = \alpha_{p,l} \exp(-\beta_{p,l} t) + \gamma_{p,l}. \tag{2.55}$$

We run the experiment $R = 10$ times for each $(p,l)$ and obtain an average convergence exponent $(\tilde{\beta}_{p,l})_{(p,l) \in P \times L}$.

### Results

In Figure 2.3 we can see the plots of $\tilde{\beta}_{p,l}$. As suspected from the heuristic argument, we do not see an increasingly large dependence on $p$ for $L = 1,2$ or $3$ when $D \in \{50,100\}$. For the MNIST dataset some dependence on the depth is appreciated, but this may be due to other factors that affect the convergence rate, like initialization issues. For

[7] Adaptative Moment Estimation (See [101]).

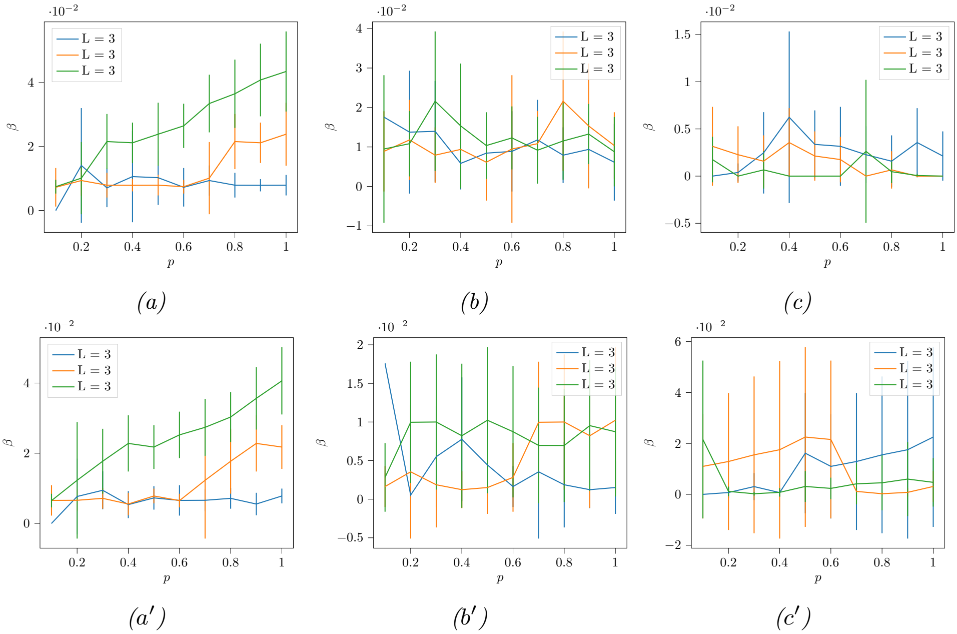


*Figure 2.3: The fit $\tilde{\beta}_{p,l}$ for $p \in \{i \times 10^{-1} : i \in [10]\}$ and $l \in \{1,2,3\}$ for LeNet with different widths $D$ and different datasets. Here* (a) *MNIST with $D = 50$; ($a'$) MNIST with $D = 100$;* (b) *CIFAR-100-fine labels with $D = 50$;* ($b'$) *CIFAR-100-fine labels with $D = 100$;* (c) *CIFAR-100-coarse labels with $D = 50$; ($c'$) CIFAR-100-coarse labels with $D = 100$. While for the MNIST dataset there seems to be an increasing dependence of dropout on the convergence rate with the depth L, for CIFAR no such dependence is observed. We remark, however, that in the CIFAR datasets encountering saddle points was more common. For those areas the loss profile is flat and so we expect the fits to be biased towards the origin in some cases.*

the CIFAR datasets, convergence is greatly affected by saddlepoints despite the use of dropout. This is, however, common when using SGD with small constant stepsizes. In particular, in practical scenarios other schemes that adjust the stepsize, like e.g. ADAM, may be more appropiate when dealing with deep networks with dropout in different layers. From the experiments it is concluded that despite the stochasticity provided by dropout, the convergence rate is not affected much by a varying dropout probability $1-p$ in wide networks with a few dropout layers.

## 2.6 Conclusion

In this chapter we have seen a probability theoretical proof that a large class of dropout algorithms for neural networks, converge almost surely to a unique stationary set of a projected system of ODEs. The result gives a formal guarantee that these dropout algorithms are well-behaved for a wide range of NNs and activation functions, and will at least asymptotically not suffer from issues because of the connection to bond percolation. We leave the extension of this result for nonsmooth activation functions such as ReLU for future work. Additionally, we established a bound for the rate of convergence of dropout to a stationary point of a generic nonconvex function. An upper bound the rate of convergence of GD on the limiting ODE of dropout algorithms was established as well for arborescences of arbitrary depth with linear activation functions. While GD on the limiting ODE is not strictly a dropout algorithm, the result is a necessary step towards analyzing the convergence rate of the actual stochastic implementations of dropout algorithms. Finally, Proposition 3 specifically implies that *Dropout* and *Dropconnect* can impair the convergence rate by as much as an exponential factor in the number of layers of very thin but deep networks. We have theoretically and experimentally verified this claim in experiments with a path network. This fact is in contrast to wide networks with a few dropout layers where a strong dependence on the dropout probability $p$ is not experimentally observed. These two observations together imply that there is a change of regime in the convergence rate from networks that are wide with a few dropout layers to thin networks with many dropout layers.

# Appendix

## 2.A Backpropagation Algorithm

We define the backpropagation algorithm used in Section 2.2 to compute the estimate of the gradient.

**Definition 4.** *Assume $\sigma \in C^1(\mathbb{R})$. Given weights $W \in \mathcal{W}$ and input–output pair $(x,y) \in \mathbb{R}^{d_0} \times \mathbb{R}^{d_L}$, the tensor $\mathrm{B}_W(x,y) \in \mathbb{R}^{d_L \times d_{L-1}} \times \cdots \times \mathbb{R}^{d_1 \times d_0}$ is calculated iteratively by:*

1. *Computing $A_1, \ldots, A_L$ using Definition 1.*
2. *Calculating for $i = L-1, \ldots, 1$,*

$$\begin{aligned} R_L &= A_L = (y - W_L A_{L-1}) \in \mathbb{R}^{d_L}, \\ R_i &= (W_{i+1}^{\mathrm{T}} R_{i+1}) \odot (\sigma'(W_i A_{i-1})) \in \mathbb{R}^{d_i}. \end{aligned} \tag{2.56}$$

3. *Setting for $i \in [L]$, $\big(\mathrm{B}_W(x,y)\big)_i = -2R_i A_{i-1}^{\mathrm{T}}$.*

Definition 4 is essentially a computationally efficient manner of calculating the gradient $\nabla l(\Psi_W(x),y)$ in (2.10) by leveraging the NN's layered structure together with the chain rule of differentation in order to come to a recursive computation of the partial derivatives.

## 2.B ODE method

Regarding our second result in Proposition 4, observe that GD on a limiting ODE is not exactly a dropout algorithm. Analyzing GD's convergence rate however is an important stepping stone towards analyzing the convergence rate of dropout algorithms. To see the mathematical relation, consider that any dropout algorithm updates the weights

$$W^{[n+1]} = W^{[n]} + \alpha^{\{n\}} \Delta^{[n+1]} \tag{2.57}$$

randomly for $n = 0, 1, 2, \cdots$. Here, the $\alpha^{\{n\}}$ denote the step sizes of the algorithm, and the $\Delta^{[n+1]}$ represent the random directions that result from the act of dropping weights. As we will show in this chapter under assumptions of independence, these random directions satisfy

$$\mathbb{E}[\Delta^{[n+1]} \mid W^{[0]},\dots,W^{[n]}] = -\nabla\mathcal{D}(W^{[n]}) \tag{2.58}$$

for some continuous, differentiable function $\mathcal{D}(W)$. Observe that the algorithm in (2.57) satisfies $W^{[n+1]} = W^{[n]} + \alpha^{\{n\}}(-\nabla\mathcal{D}(W^{[n]}) + M^{[n+1]})$ where $M^{[n+1]}$ here is $M^{[n+1]} = \mathbb{E}[\Delta^{[n+1]} \mid W^{[0]},\dots,W^{[n]}] - \Delta^{[n+1]}$ and describes a *martingale difference* sequence. This martingale difference sequence's expectation with respect to the past $W^{[0]},\dots,W^{[n]}$ is zero.

For diminishing step sizes $\alpha^{\{n\}}$, we can consequently view dropout algorithms as in (2.57) as being noisy discretizations of the ordinary differential equation

$$\frac{\mathrm{d}W}{\mathrm{d}t} = -\nabla\mathcal{D}(W(t)). \tag{2.59}$$

In fact, we employ the so-called *ordinary differential equation method* [140, 127], which formally establishes that the random iterates in (2.57) follow the trajectories of the gradient flow in (3.2). Hence, after sufficiently many iterations $n$ and for a sufficiently small step size $\alpha$, the convergence rate of the deterministic GD algorithm

$$W^{\{n+1\}} = W^{\{n\}} - \alpha\nabla\mathcal{D}(W^{\{n\}}) \tag{2.60}$$

gives insight into the convergence rate of the stochastic dropout algorithm in (2.57).

## 2.C Projection operator

We define here the projection operator $\pi$ used in Section 2.3. Say that $\mathcal{H}$ is defined by $l$ smooth constraints $q_i : \mathcal{W} \to \mathbb{R}$, $i = 1,\dots,l$ satisfying $q_1(W) \leq 0,\dots,q_l(W) \leq 0$, i.e., $\mathcal{H} = \{W \in \mathcal{W} : q_i(W) \leq 0 \ \forall i \in [l]\}$. Denote by $\nabla\mathcal{D}|_{\mathcal{H}}(W)$ the gradient of $\mathcal{D}(W)$ restricted to $\mathcal{H}$ and let $\mathrm{T_W}\mathcal{W}$ be the tangent space of $\mathcal{W}$ at $W$. Suppose that $\nabla q_i(W) \neq 0$ whenever $q_i(W) = 0$, and that these are linearly independent. At any point $W \in \partial\mathcal{H}$, we define the outer normal cone

$$C(W) \triangleq \{v \in \mathrm{T_W}\mathcal{W} \ : \ \nabla q_i(W)v^T \geq 0 \text{ for } i \in [l] \text{ s.t. } q_i(W) = 0\}. \tag{2.61}$$

We also assume that $C(W)$ is upper semicontinuous, i.e., if $\tilde{W} \in B_{\mathcal{H}}(W,\delta)$, where $B_{\mathcal{H}}(W,\delta)$ is the ball of radius $\delta > 0$ centered at $W$ and intersected with $\mathcal{H}$, then $C(W) = \cap_{\delta>0}\big(\cup_{\tilde{W}\in B_{\mathcal{H}}(W,\delta)}C(\tilde{W})\big)$. Let $\pi(W) \triangleq -t\mathbb{1}[W \in \partial\mathcal{H}]$ with $t \in C(W)$ minimal to resolve the violated constraints of $\mathcal{D}|_{\mathcal{H}}(W)$ at $W \in \partial\mathcal{H}$ so that $\mathcal{D}|_{\mathcal{H}}(W) + \pi(W)$ points inside $\mathcal{H}$. In particular, we have

$$\pi(W) = -\sum_{i=1}^{l}\lambda_i(W)\nabla q_i(W) \in -C(W) \tag{2.62}$$

where $\{\lambda_i(W) \geq 0\}_{i=1}^{l}$ are functions such that $\lambda_i(W) = 0$ if $q_i(W) < 0$.

## 2.D Proof of Proposition 1

The proof of Proposition 1 relies on the framework of stochastic approximation in [140]. Specifically, Proposition 1 follows from Theorem 2.1 on p. 127 if we can show that its conditions (A2.1)–(A2.6) on p. 126 are satisfied. In the notation of Sections 2.2, 2.3, these conditions read:

(A2.1) $\sup_t \mathbb{E}[\|\Delta^{[t+1]}\|_{\mathrm{F}}] < \infty$;

(A2.2) there is a measurable function $\bar{g}(\cdot)$ of $W$ and there are random variables $\beta^{[t+1]}$ such that

$$\mathbb{E}[\Delta^{[t+1]} \mid \mathcal{F}_t] = \bar{g}(W^{[t]}) + \beta^{[t+1]}, \tag{2.63}$$

where $\mathcal{F}_t$ denotes the smallest $\sigma$-algebra generated by $\cup_{s\leq t}\{W^{[0]},(F^{[s]},X^{[s]},Y^{[s]})\}$;

(A2.3) $\bar{g}(\cdot)$ is continuous;

(A2.4) the step sizes satisfy

$$\sum_{t=1}^{\infty}\alpha^{\{t\}} = \infty, \alpha^{\{n\}} \geq 0, \alpha^{\{n\}} \to 0 \text{ for } n \geq 0 \text{ and } \alpha^{\{n\}} = 0 \text{ for } n < 0; \tag{2.64}$$

$$\sum_{t=1}^{\infty}(\alpha^{\{t\}})^2 < \infty; \tag{2.65}$$

(A2.5) $\sum_t \alpha^{\{t\}} \|\beta^{[t]}\|_{\mathrm{F}} < \infty$ w.p. one;

(A2.6) $\bar{g}(\cdot) = -\nabla\mathcal{D}(\cdot)$ for a continuously differentiable real-valued $\mathcal{D}(\cdot)$ and $\mathcal{D}(\cdot)$ is constant on each stationary set $S_i$.

We next also state for convenience Theorem 2.1 from [140] in the notation of this chapter. Their result does require some notation, as it characterizes the limiting behavior of the iterates of

$$W^{[n+1]} = \mathcal{P}_{\mathcal{H}}\big(W^{[n]} - \alpha\Delta^{[n+1]}\big) \triangleq W^{[n]} - \alpha\Delta^{[n+1]} + Z^{[n+1]}. \tag{2.66}$$

For any sequence of step sizes $\alpha^{\{n\}}$ satisfying (A2.4), define $t_0 = 0$ and $t_n = \sum_{i=0}^{n-1}\alpha^{\{i\}}$. Define the continuous-time interpolation

$$W_0(t) = \begin{cases} W^{[n]} & \text{for} \quad t_n \leq t < t_{n+1}, \\ W^{[0]} & \text{for} \quad t \leq 0, \end{cases} \tag{2.67}$$

as well as for $m \in \mathbb{N}_0$, the shifted processes $W_m(t) = W_0(t_m + t)$ for $t \in (-\infty,\infty)$. Let furthermore $o(t) = \inf\{n \in \mathbb{N}_0 : t_n \leq t < t_{n+1}\}$ for $t \in [0,\infty)$, and $o(t) = 0$ for $t \in (-\infty,\infty)$, and define

$$Z_0(t) = \begin{cases} \sum_{i=0}^{o(t)-1}\alpha^{\{i\}}Z_i & \text{for} \quad t \in [0,\infty), \\ 0 & \text{for} \quad t \in (-\infty,\infty), \end{cases} \tag{2.68}$$

as well as for $m \in \mathbb{N}_0$, the shifted processes $Z_m(t) = \sum_{i=m}^{o(t_m+t)-1}$ for $t \in [0,\infty)$ and $Z_m(t) = -\sum_{i=o(t_m+t)}^{m-1}\alpha^{\{i\}}Z_i$ for $t \in (-\infty,0)$. The following now holds:

**Theorem 5** (A part of Theorem 2.1 from [140])**.** *Let conditions (A2.1)–(A2.5) hold for algorithm* (2.66)*, with the projection onto $\mathcal{H}$ being as described in Appendix 2.C. Let* $(\Omega,\mathcal{F},\mathbb{P})$ *be the probability space of the processes. Then there is a set $N$ of probability zero such that for $\omega \notin N$, the set of functions $\{W_m(\omega,\cdot),Z_m(\omega,\cdot),m < \infty\}$ is equicontinuous.*

*Let* $(W(\omega,\cdot),Z(\omega,\cdot))$ *denote the limit of some convergent subsequence. Then this pair satisfies the projected ODE* (2.15)*, and* $\{W^{[n]}(\omega)\}$ *converges to some limit set of the ODE in* $\mathcal{H}$*. Suppose that (A2.6) holds. Then, for almost all* $\omega$*,* $\{W^{[n]}(\omega)\}$ *converges to a unique* $S_i$*.*

In order to apply Theorem 5 and arrive at Proposition 1, we verify conditions (A2.1)–(A2.6) through Lemmas 4–6 shown next in Appendix 2.D.1.

## 2.D.1 Verification of conditions (A2.1)–(A2.6)

First we assume conditions (N1)–(N3) and we prove that the variance of the random update direction in (2.1) is finite. This verifies condition (A2.1). The proof can be found below.

**Lemma 4.** *Assume (N1)–(N3) from Proposition 1. Then* $\sup_{t\in\mathbb{N}}\mathbb{E}[\|\Delta_i^{[t+1]}\|_{\mathrm{F}}^2]<\infty$ *for* $i=0,1,\ldots,L$*.*

We prove next that if $\sigma\in C^r_{PB}(\mathbb{R})$ , then the random update direction in (2.1), conditional on all prior updates, has conditional expectation $\nabla\mathcal{D}(W^{[t]})$. Lemma 5 verifies conditions (A2.2), (A2.3), and (A2.5) (in particular, here $\beta^{[t]}=0$). The proof can be found also below.

**Lemma 5.** *Assume (N2)–(N4) from Proposition 1. Then* $\mathbb{E}[\Delta^{[t+1]}|\mathcal{F}_t]=\nabla\mathcal{D}(W^{[t]})$*. Furthermore,* $\nabla\mathcal{D}:\mathcal{W}\to\mathcal{W}$ *is* $r-1$ *times continuously differentiable.*

From these conditions the first part of Proposition 1 follows. To prove the second part of Proposition 1, we have to prove that the set of stationary points $S_{\mathcal{H}}$ is well-behaved in the sense that $\mathcal{D}|_{S_i}(W)$ is constant. If an objective function is sufficiently differentiable, this is guaranteed by the Morse–Sard Theorem [173, 172]. In the present case however we must take into account the possibility of an intersection of the set of stationary points with the boundary $\partial\mathcal{H}$. Assuming (N4) and (N5) provides sufficient conditions. The proof of Lemma 6 can be found in Appendix 2.D.1:

**Lemma 6.** *If (N2)–(N5) hold, then* $\mathcal{D}(W)$ *is constant on each* $S_i$*.*

Since Conditions (A2.1)–(A2.6) of Theorem 2.1 on p. 127 in [140] are proven satisfied, the proof of Proposition 1 is now completed. □

### Boundedness of $\Delta^{[t+1]}$ in expectation – Proof of Lemma 4

We need to carefully track all sequences of random variables created by a dropout algorithm throughout this proof, which we state here first explicitly.

**Definition 6** (Dropout iterates)**.** *During its* $(t+1)$*-st* feedforward step*, the algorithm iteratively calculates*

$$A_0^{[t+1]}=X^{[t+1]},\quad A_i^{[t+1]}=\sigma((W_i^{[t]}\odot F_i^{[t+1]})A_{i-1}^{[t+1]}) \tag{2.69}$$

*for* $i=1,2,\ldots,L-1$*, to output*

$$\Psi_{F^{[t+1]}\odot W^{[t]}}(X^{[t+1]})=(W_L^{[t]}\odot F_L^{[t+1]})A_{L-1}^{[t+1]}=A_L^{[t+1]}. \tag{2.70}$$

*Subsequently for its* $(t+1)$*-st* backpropagation step *the algorithm calculates*

$$
\begin{aligned}
R_L^{[t+1]} &= (Y^{[t+1]} - (W_L^{[t]} \odot F_L^{[t+1]}) A_{L-1}^{[t+1]}) \in \mathbb{R}^{d_L}, \\
R_j^{[t+1]} &= ((W_{j+1}^{[t]} \odot F_{j+1}^{[t+1]})^T R_{j+1}^{[t+1]}) \odot (\sigma'((W_j^{[t]} \odot F_j^{[t+1]}) A_{j-1}^{[t+1]})) \in \mathbb{R}^{d_i},
\end{aligned} \tag{2.71}
$$

*iteratively for* $j = L-1, \dots, 1$*. The algorithm then calculates*

$$
\Delta_i^{[t+1]} = -2F_i^{[t+1]} \odot (R_i^{[t+1]} (A_{i-1}^{[t+1]})^{\mathrm{T}}) \tag{2.72}
$$

*for* $i = 1, \dots, L$*, and finally updates all weights according to* (2.11).

The idea of the proof of Lemma 4 is to expand the terms in $\Delta_i^{[t+1]}$ defined in Definition 6 recursively, and identify a polynomial in variables $\{\|Y\|_2^n \|X\|_2^m\}_{m\in\mathbb{N}_0}$ and $n = 0, 1, 2$. We will use several bounds that pertain to the Frobenius norm, described in Lemma 14 in Appendix 2.J. We use these below repeatedly.

First, we will prove two bounds on the activation function applied to an arbitrary matrix $A$. Recall that $\sigma \in C_{PB}^2(\mathbb{R})$ by assumption (N1). There thus (i) exist some $C_0, k_0 > 0$ such that $|\sigma(z)| \leq C_0(1+z^2)^{k_0}$ for all $z \in \mathbb{R}$, and there exist some $C_1, k_1 > 0$ such that $|\sigma'(z)| \leq C_1(1+z^2)^{k_1}$ for all $z \in \mathbb{R}$. Let $k = \max\{1, k_0, k_1\}$. Then

$$
\|\sigma(A)\|_{\mathrm{F}}^2 = \sum_{i,j} |\sigma(A_{ij})|^2 \overset{\text{(i)}}{\leq} C_0 \sum_{i,j} (1 + A_{ij}^2)^k \overset{\text{(Lemma 14)}}{\leq} C_2 (1 + \|A\|_{\mathrm{F}})^{2k} \tag{2.73}
$$

for some constant $C_2 > 0$. Similarly there exists some $C_3 > 0$ such that $\|\sigma'(A)\|_F \leq C_3(1+\|A\|_{\mathrm{F}})^k$. Note furthermore that (ii) for all $l \geq 0$, by submultiplicativity of the Frobenius norm,

$$
\begin{aligned}
(1 + \|A\sigma(B)\|_{\mathrm{F}})^l &\overset{\text{(ii)}}{\leq} (1 + \|A\|_{\mathrm{F}} \|\sigma(B)\|_{\mathrm{F}})^l \\
&\overset{(2.73)}{\leq} \big(1 + C_2^{1/2} \|A\|_{\mathrm{F}} (1 + \|B\|_{\mathrm{F}})^k\big)^l \leq C_4 (1 + \|A\|_{\mathrm{F}})^l (1 + \|B\|_{\mathrm{F}})^{kl}
\end{aligned} \tag{2.74}
$$

for $C_4 = \max\{1, C_2^{l/2}\} > 0$. Again, a similar bound holds for $\sigma'$.

Next, note that we have by (i) submultiplicativity and Lemma 14 that

$$
\|\Delta_i^{[t+1]}\|_{\mathrm{F}} = \|F_i^{[t+1]} \odot (R_i^{[t+1]} (A_{i-1}^{[t+1]})^{\mathrm{T}})\|_{\mathrm{F}} \overset{\text{(i)}}{\leq} \|F_i^{[t+1]}\|_{\mathrm{F}} \|R_i^{[t+1]}\|_{\mathrm{F}} \|A_{i-1}^{[t+1]}\|_{\mathrm{F}}. \tag{2.75}
$$

The first term is bounded with probability one: $F_{i,r,l}^{[t]} \in \{0,1\}$ for all $i, r, l, t$. For the second term, consider the following bound:

$$
\begin{aligned}
\|R_i^{[t+1]}\|_{\mathrm{F}} &\overset{(2.71)}{=} \|(W_{i+1}^{[t]} \odot F_{i+1}^{[t+1]})^{\mathrm{T}} R_{i+1}^{[t+1]} \odot \sigma'\big((W_i^{[t]} \odot F_i^{[t+1]}) A_{i-1}^{[t+1]}\big)\|_{\mathrm{F}} \\
&\overset{\text{(Lemma 14)}}{\leq} \|W_{i+1}^{[t]} \odot F_{i+1}^{[t+1]}\|_{\mathrm{F}} \|\sigma'\big((W_i^{[t]} \odot F_i^{[t+1]}) A_{i-1}^{[t+1]}\big)\|_{\mathrm{F}} \|R_{i+1}^{[t+1]}\|_{\mathrm{F}}
\end{aligned} \tag{2.76}
$$

for $1 \leq i \leq L$, where we have also used the submultiplicative property. For the third term, consider the next bound: (i) recursing (2.74) with $A = I$ and $B = (W_j^{[t]} \odot F_j^{[t+1]}) A_{j-1}^{[t+1]}$

etc, we obtain that there exists some $C_5 > 0$, say, so that

$$
\begin{aligned}
\|A_j^{[t+1]}\|_{\mathrm{F}} &\overset{(2.69)}{=} \|\sigma((W_j^{[t]}\odot F_j^{[t+1]})A_{j-1}^{[t+1]})\|_{\mathrm{F}} \overset{(2.73)}{\le} C_2(1+\|(W_j^{[t]}\odot F_j^{[t+1]})A_{j-1}^{[t+1]}\|_{\mathrm{F}})^k \\
&\overset{\text{(Lemma 14)}}{\le} C_2(1+\|W_j^{[t]}\odot F_j^{[t+1]}\|_{\mathrm{F}})^k(1+\|A_{j-1}^{[t+1]}\|_{\mathrm{F}})^k \\
&\overset{\text{(i)}}{\le} C_5\big(1+\|X^{[t+1]}\|_2\big)^{k^j}\prod_{l=1}^{j-1}\big(1+\|W_l^{[t]}\odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{j-l}}
\end{aligned}
\tag{2.77}
$$

for $j = 1,2,\ldots,L-1$. Similar by the derivation in (2.77), we obtain instead with $\sigma'$ that there exists some $C_6 > 0$ such that

$$
\|\sigma'((W_j^{[t]}\odot F_j^{[t+1]})A_{j-1}^{[t+1]})\|_{\mathrm{F}} \le C_6\big(1+\|X^{[t+1]}\|_2\big)^{k^j}\prod_{l=1}^{j-1}\big(1+\|W_l^{[t]}\odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{j-l}}. \tag{2.78}
$$

Recall that $\|\Delta_i^{[t+1]}\|_{\mathrm{F}} \le \|F_i^{[t+1]}\|_{\mathrm{F}}\|R_i^{[t+1]}\|_{\mathrm{F}}\|A_{i-1}^{[t+1]}\|_{\mathrm{F}}$. This, together with using (2.76) repeatedly for $j = i,\ldots,L-1$, and (2.77), (2.78), yields the following inequality

$$
\begin{aligned}
\|\Delta_i^{[t+1]}\|_{\mathrm{F}} &\overset{(2.76)}{\le} \|F_i^{[t+1]}\|_{\mathrm{F}}\|R_L^{[t+1]}\|_{\mathrm{F}}\|A_i^{[t+1]}\|_{\mathrm{F}}\times \\
&\qquad\qquad \prod_{j=i}^{L-1}\|W_{j+1}^{[t]}\odot F_{j+1}^{[t+1]}\|_{\mathrm{F}}\|\sigma'\big((W_j^{[t]}\odot F_j^{[t+1]})A_{j-1}^{[t+1]}\big)\|_{\mathrm{F}} \\
&\overset{(2.77)}{\le} C_5\|F_i^{[t+1]}\|_{\mathrm{F}}\big(1+\|X^{[t+1]}\|_2\big)^{k^i}\prod_{l=1}^{i-1}\big(1+\|W_l^{[t]}\odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{i-l}} \\
&\quad\times\|R_L^{[t+1]}\|_{\mathrm{F}}\prod_{j=i}^{L-1}\|W_{j+1}^{[t]}\odot F_{j+1}^{[t+1]}\|_{\mathrm{F}}\|\sigma'\big((W_j^{[t]}\odot F_j^{[t+1]})A_{j-1}^{[t+1]}\big)\|_{\mathrm{F}} \\
&\overset{(2.78)}{\le} C_7\|F_i^{[t+1]}\|_{\mathrm{F}}\big(1+\|X^{[t+1]}\|_2\big)^{k^i}\prod_{l=1}^{i-1}\big(1+\|W_l^{[t]}\odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{i-l}} \\
&\quad\times\|R_L^{[t+1]}\|_{\mathrm{F}}\prod_{j=i}^{L-1}\|W_{j+1}^{[t]}\odot F_{j+1}^{[t+1]}\|_{\mathrm{F}}\big(1+\|X^{[t+1]}\|_2\big)^{k^j}\prod_{l=1}^{j}\big(1+\|W_l^{[t]}\odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{j-l}} \\
&\le C_7\|F_i^{[t+1]}\|_{\mathrm{F}}\|R_L^{[t+1]}\|_{\mathrm{F}}\Big(\prod_{j=i}^{L-1}\|W_{i+1}^{[t]}\odot F_{i+1}^{[t+1]}\|_{\mathrm{F}}\Big) \\
&\quad\times\Big(\prod_{j=i}^{L-1}\big(1+\|X^{[t+1]}\|_2\big)^{2k^j}\prod_{l=1}^{j}\big(1+\|W_l^{[t]}\odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{2k^{j-l}}\Big) \\
&= C_7\|F_i^{[t+1]}\|_{\mathrm{F}}\|R_L^{[t+1]}\|_{\mathrm{F}}\Big(\prod_{j=i}^{L-1}\|W_{i+1}^{[t]}\odot F_{i+1}^{[t+1]}\|_{\mathrm{F}}\Big) \\
&\quad\times\big(1+\|X^{[t+1]}\|_2\big)^{\sum_{j=i}^{L-1}2k^j}\Big(\prod_{j=i}^{L-1}\prod_{l=1}^{j}\big(1+\|W_l^{[t]}\odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{2k^{j-l}}\Big).
\end{aligned}
\tag{2.79}
$$

Lastly, we bound $\|R_L^{[t+1]}\|_{\mathrm{F}}$. By applying (i) subadditivity of the norm $\|A+B\|_F \leq \|A\|_F + \|B\|_F$ and then using the elementary bound $(a+b)^2 \leq 2(a^2+b^2)$ as well as submultiplicativity, we obtain

$$\begin{aligned}
&\|R_L^{[t+1]}\|_{\mathrm{F}} \overset{(2.71)}{=} \|Y^{[t+1]} - (W_L^{[t]} \odot F_L^{[t+1]})A_{L-1}^{[t+1]}\|_{\mathrm{F}} \qquad (2.80)\\
&\overset{(\mathrm{i})}{\leq} \|Y^{[t+1]}\|_2^2 + \|W_L^{[t]} \odot F_L^{[t+1]}\|_{\mathrm{F}} \|A_{L-1}^{[t+1]}\|_{\mathrm{F}}\\
&\overset{(2.77)}{\leq} \|Y^{[t+1]}\|_2 + \|W_L^{[t]} \odot F_L^{[t+1]}\|_{\mathrm{F}} \big(1 + \|X^{[t+1]}\|_2\big)^{k^{L-1}} \prod_{l=1}^{L-1} \big(1 + 2\|W_l^{[t]} \odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{L-l}}.
\end{aligned}$$

By combining inequalities (2.79), (2.80), and upper bounding the exponent $k^{L-1}$ of the term $1+\|X^{[t+1]}\|_{\mathrm{F}}$ in (2.80) by $2\sum_{j=1}^{L-1} k^j$, we conclude that

$$\begin{aligned}
&\|\Delta_i^{[t+1]}\|_{\mathrm{F}}\\
&\leq C_8 \|Y^{[t+1]}\|_2 \big(1 + \|X^{[t+1]}\|_2\big)^{2\sum_{j=1}^{L-1} k^j} \times \qquad (2.81)\\
&\qquad\qquad \|F_i^{[t+1]}\|_{\mathrm{F}} P_1\big(\|W_1^{[t]} \odot F_1^{[t+1]}\|_{\mathrm{F}}, \dots, \|W_L^{[t]} \odot F_L^{[t+1]}\|_{\mathrm{F}}\big)\\
&\quad + C_9 \big(1 + \|X^{[t+1]}\|_2\big)^{2\sum_{j=1}^{L} k^j} \|F_i^{[t+1]}\|_{\mathrm{F}} P_2(\|W_1^{[t]} \odot F_1^{[t+1]}\|_{\mathrm{F}}, \dots, \|W_L^{[t]} \odot F_L^{[t+1]}\|_{\mathrm{F}})
\end{aligned}$$

for $i = 1, \dots, L$ and some constants $C_8, C_9$ and polynomials $P_1(z_1, \dots, z_L), P_2(z_1, \dots, z_L)$, say, the latter both in $L$ variables. Because of the projection and by definition of $\mathcal{H}$, there exists a constant $M$ such that $\|W_i^{[t]}\|_{\mathrm{F}} \leq M$ with probability one for all $i = 1, \dots, L$, $t \in \mathbb{N}_+$. Furthermore, $\|F_i^{[t]}\|_{\mathrm{F}} \leq \max_{i=0,\dots,L-1} \sqrt{d_i d_{i+1}}$ with probability one for all $i = 1, \dots, L$, $t \in \mathbb{N}_+$. These two bounds, together with (2.81) and the fact that $P_1, P_2$ are polynomials, as well as the hypothesis that $\mathbb{E}[\|Y\|_2^m \|X\|_2^n] < \infty \, \forall m \in \{0,1,2\}, n \in \mathbb{N}_0$, implies the result. □

### Conditional expectation of $\Delta^{[t+1]}$ – Proof of Lemma 5

Let $i \in \{1, \dots, L\}$, $r \in \{1, \dots, d_{i+1}\}$ and $l \in \{1, \dots, d_i\}$. Recall that $\mathcal{F}_t$ is the smallest $\sigma$-algebra generated by $\{W^{[0]}, (F^{[s]}, X^{[s]}, Y^{[s]})\}_{s \leq t}$, and note that $W^{[t]}$ is $\mathcal{F}_t$-measurable. The (i) $\mathcal{F}_t$-measurability of $W^{[t]}$ together with the (ii) hypothesis that the sequence of random variables $\{(F^{[s]}, X^{[s]}, Y^{[s]})\}_{s \in \mathbb{N}_+}$ is i.i.d. , implies that

$$\begin{aligned}
\mathbb{E}[\Delta_{i,r,l}^{[t]} | \mathcal{F}_t] &\overset{(2.1)}{=} \mathbb{E}\Big[\big(F_{i,r,l}^{[t+1]} \mathrm{B}_{F^{[t+1]} \odot W^{[t]}}(X^{[t+1]}, Y^{[t+1]})\big)_{i,r,l} \Big| \mathcal{F}_t\Big]\\
&\overset{(\mathrm{i,ii})}{=} \int F_{i,r,l} \mathrm{B}_{F \odot W^{[t]}}(X,Y)_{i,r,l} \, \mathrm{d}\mathbb{P}[F^{[t+1]} = F, X^{[t+1]} = X, Y^{[t+1]} = Y]\\
&\overset{(2.12)}{=} \int \Big(F_{i,r,l} \frac{\partial l(\Psi_{F \odot V^{[t]}}(X), Y)}{\partial (F_{i,r,l} V_{i,r,l})}\Big)(W^{[t]}) \, \mathrm{d}\mathbb{P}[F^{[t+1]} = F, X^{[t+1]} = X, Y^{[t+1]} = Y]\\
&= \int \frac{\partial l(\Psi_{F \odot V^{[t]}}(X), Y)}{\partial V_{i,r,l}}(W^{[t]}) \, \mathrm{d}\mathbb{P}[F^{[t+1]} = F, X^{[t+1]} = X, Y^{[t+1]} = Y]. \qquad (2.82)
\end{aligned}$$

Next, we need to check that we can exchange the derivative and expectation. Note that we have the same assumptions $\mathbb{E}[\|Y\|_2^m \|X\|_2^n] < \infty \, \forall m \in \{0,1,2\}, n \in \mathbb{N}_+$ as for Lemma 4. as well as that $\sigma \in C_{\mathrm{PB}}^r(\mathbb{R})$. Therefore, by (2.81) in Lemma 4 we have that $|\Delta_{i,r,l}^{[t+1]}|$ is upper bounded and moreover $\mathbb{E}[\Delta_{i,r,l}^{[t+1]}] \leq C_{\mathcal{H}}$ for some $C_{\mathcal{H}} \leq \infty$ only dependent on $\mathcal{H}$. The interchange is then warranted by the dominated convergence theorem. Hence continuing from (2.82), we obtain

$$\begin{aligned}\mathbb{E}[\Delta_{i,r,l}^{[t]}|\mathcal{F}_t] &= \frac{\partial}{\partial W_{i,r,l}} \int l(\Psi_{F\odot W^{[t]}}(X),Y)\,\mathrm{d}\mathbb{P}[F^{[t+1]}=F, X^{[t+1]}=X, Y^{[t+1]}=Y] \\ &\overset{(2.3)}{=} \frac{\partial \mathcal{D}(W^{[t]})}{\partial W_{i,r,l}}.\end{aligned}$$

If $\sigma \in C^r_{\mathrm{PB}}(\mathbb{R})$, then for any multi-index $s$ on the set of weights, a bound similar to (2.81) holds by the chain rule:

$$\begin{aligned}|\partial^s l(Y,\Psi_{W\odot F}(X))| \leq \|Y\|_{\mathrm{F}} P_{1,s}(\|W_1\|_{\mathrm{F}},\ldots,\|W_L\|_{\mathrm{F}},\{\|X\|_2^j\}_{j=1}^{n_{s,1}}) \\ + P_{2,s}(\|W_1\|_{\mathrm{F}},\ldots,\|W_L\|_{\mathrm{F}},\{\|X\|_2^j\}_{j=1}^{n_{s,2}})\end{aligned} \tag{2.83}$$

where $P_{1,s}, P_{2,s}$ are polynomials and $n_{s,1}, n_{s,2}$ are the top exponents in the expansion in $\|X\|_{\mathrm{F}}$. Hence, using the assumption $\mathbb{E}[\|Y\|_2^m \|X\|_2^n] < \infty$ for all $m \in \{0,1,2\}, n \in \mathbb{N}_+$, we can find a compact set $\mathcal{K} \subset \mathcal{W}$ and a constant $C_{\mathcal{K}} > 0$ such that for any $W \in \mathcal{K}$ we have $\mathbb{E}[|\partial^s l(Y,\Psi_{W\odot F}(X))|] \leq C_{\mathcal{K}}$ . In particular we can apply the dominated convergence theorem and conclude $\mathcal{D}(W) \in C^{r-1}(\mathcal{W})$ with $\partial^s \mathcal{D}(W) = \mathbb{E}[\partial^s l(Y,\Psi_{W\odot F}(X))]$. □

**Constant $\mathcal{D}(W)$ on a critical set – Proof of Lemma 6**

We use Sard's theorem [172] to prove Lemma 6, which gives sufficient conditions for condition (A2.6):

**Proposition 5.** *[172] Let $f: M \to N$ be a $f \in C^r$ map between manifolds with* $\dim(M) = m$*,* $\dim(N) = n$*. Let* $\mathrm{Crit}(f) = \{x \in M : \nabla f(x) = 0\}$ *be the set of critical points of $f$. If $r > m/n - 1$, then $f(\mathrm{Crit}(f))$ has measure zero.*

*Proof of Lemma 6.* By Lemma 5, we have $\mathcal{D}(W) \in C^r(\mathcal{W})$. By assumption (N5) we have that if $W \in \partial\mathcal{H}$ and $\mathcal{D}(W) + \pi(W) = 0$, then $\mathcal{D}(W) = 0$. Furthermore $W \in S_j$ for some $j$, i.e., the critical points of $\mathcal{D}(W) + \pi(W)$ are $\{W \in \mathcal{W} \mid \nabla\mathcal{D}(W) = 0\} \cap \mathcal{H}$. We apply Sard's theorem (Proposition 5) to $\mathcal{D}(W)$. We have that if $r \geq \dim(\mathcal{W})$, then $\mathcal{D}(S_i) \subseteq \mathbb{R}$ has measure zero. Since $S_i$ is connected there is a continuous path $z_{a,b}: [0,1] \to S_i$ joining any two points $a, b \in S_i$. By continuity of $\mathcal{D}(W)$ we must have then $\mathcal{D}(a) = \mathcal{D}(b)$, since otherwise we would have $[\mathcal{D}(a), \mathcal{D}(b)] \subseteq \mathcal{D}(S_i)$ which has positive measure in $\mathbb{R}$. Therefore $\mathcal{D}(S_i)$ must be a constant. □

Remark that in Lemma 6 the condition $r \geq \dim(\mathcal{W})$ cannot immediately be eliminated. When $r < \dim(\mathcal{W})$, there are examples of functions which are not constant on their connected critical sets, see e.g. [138].

## 2.E Proof of Proposition 2

We use standard tools for proving convergence to an $\epsilon$-stationary point (for a reference, see [50]). We require first the following bounds on the variance induced by dropout.

**Lemma 7.** *Assume that $F$ is a random variable satisfying (Q4). If $f$ is a vector of random variables with distribution $F$, then* *(i)* $\mathbb{E}\Big[\|f-\mathbb{E}[f]\|_1\Big]=2Np(1-p)$.

*(ii)* $\mathbb{E}\Big[\|f-\mathbb{E}[f]\|_1^2\Big]=2N^2p(1-p)$.

*Proof.* We prove first (i). If we denote by $f_i$ the $i$th entry of $f$, then note that from (Q4) $\mathbb{P}[f_i=1]=p$ and so $\mathbb{E}[|f_i-\mathbb{E}[f_i]|]=\mathbb{E}[|f_i-p|]=2p(1-p)$. From linearity (i) follows. For (ii), we have

$$\begin{aligned}\mathbb{E}\Big[\|f-\mathbb{E}[f]\|_1^2\Big]&=\sum_i\mathbb{E}\Big[|f_i-p|^2\Big]+\sum_{i\neq j}\mathbb{E}\Big[|f_i-p||f_j-p|\Big]\\&\leq 2Np(1-p)+2N(N-1)p(1-p)=2N^2p(1-p),\end{aligned}\tag{2.84}$$

where in the last inequality we have used the Cauchy–Schwartz inequality. □

**Lemma 8.** *Assume (Q3) and (Q4), that is,* $\nabla\mathsf{U}$ *is* $\ell$*-Lipschitz and the distribution of the filters is* $\{0,1\}$*-valued. Then,* $\nabla\mathsf{D}$ *is also* $\ell$*-Lipschitz.*

*Proof.* Using (i) Jensen's inequality with the norm, we have for a fixed $w,s\in\mathcal{W}$ that

$$\begin{aligned}\|\nabla\mathsf{D}(w)-\nabla\mathsf{D}(s)\|_2&=\|\mathbb{E}_f[f\odot\nabla\mathsf{U}(w\odot f)-f\odot\nabla\mathsf{U}(s\odot f)]\|_2\\&\overset{\text{(i)}}{\leq}\mathbb{E}_f\Big[\|f\odot\nabla\mathsf{U}(w\odot f)-f\odot\nabla\mathsf{U}(s\odot f)\|_2\Big]\\&\overset{\text{(ii)}}{\leq}\mathbb{E}_f\Big[\|\nabla\mathsf{U}(w\odot f)-\nabla\mathsf{U}(s\odot f)\|_2\Big]\\&\overset{\text{(iii)}}{\leq}\ell\mathbb{E}_f\Big[\|w\odot f-s\odot f\|_2\Big]\\&\overset{\text{(ii)}}{\leq}\ell\mathbb{E}_f\Big[\|w-s\|_2\Big]=\ell\|w-s\|_2\end{aligned}\tag{2.85}$$

where we have also used (ii) the fact that for a vector $u$ and $\{0,1\}$-valued vector $f$ we have $\|f\odot u\|_2\leq\|u\|_2$, (iii) $\nabla\mathsf{U}$ is $\ell$-Lipschitz. □

The proof of the following lemma can be found in Appendix 2.E.1.

**Lemma 9.** *Assume (Q1)–(Q4), then for any* $w\in\mathcal{W}$ *with* $\|w\|_2<R$*, we have*

$$\mathbb{E}\Big[\|\nabla\mathsf{D}(w)-f\odot\nabla\mathsf{U}(w\odot f)\|_2^2\Big]\leq Np(1-p)\big(4S^2+6N^2(\ell^2R^2+2\ell R)\big).\tag{2.86}$$

We obtain in the next lemma a simple bound for the variance of the gradient that depends on the data.

**Lemma 10.** *Assume (Q1)–(Q4), then for any* $w\in\mathcal{W}$*, we have*

$$\mathbb{E}_{z,f}\Big[\|f\odot\nabla\mathsf{U}(w\odot f)-f\odot\nabla r(w\odot f,z)\|_2^2\Big]\leq 4pNS^2.\tag{2.87}$$

*Proof.* We use first the definition of $\mathsf{U}$ as an expectation. We have

$$\begin{aligned}
&\mathbb{E}_{z,f}\left[\|f\odot\nabla\mathsf{U}(w\odot f)-f\odot\nabla r(w\odot f,z)\|_2^2\right]\\
&\qquad=\mathbb{E}_{z,f}\left[\|\mathbb{E}_{z_1}[f\odot\nabla r(w\odot f,z_1)]-f\odot\nabla r(w\odot f,z)\|_2^2\right]\\
&\qquad\leq\mathbb{E}_{z,f}\left[\mathbb{E}_{z_1}\left[\|f\odot(\nabla r(w\odot f,z_1)-\nabla r(w\odot f,z))\|_2\right]^2\right]\\
&\qquad\overset{(i)}{\leq}\mathbb{E}_{z,f}\left[\mathbb{E}_{z_1}\left[2S\|f\|_2\right]^2\right]\\
&\qquad\overset{(ii)}{\leq}\mathbb{E}_{z,f}\left[4S^2\|f\|_2^2\right]=4pNS^2,
\end{aligned}\tag{2.88}$$

where in (i) we have used the upper bound for $\|\nabla r(w\odot f,z)\|_2$ from (Q2) and in (ii) that since $f_i\in\{0,1\}$ for all $i\in[N]$, we have $\|f\|_2^2=\|f\|_1$ so using linearity with (Q4) the bound follows. □

By (Q3)–(Q4) and Lemma 8, $\nabla\mathsf{D}$ is $\ell$-Lipschitz. In this case, we can then use the following common argument: if $\nabla\mathsf{D}$ is $\ell$-Lipschitz then we have the inequality

$$\mathsf{D}(W^{[t+1]})\leq\mathsf{D}(W^{[t]})+\langle\nabla\mathsf{D}(W^{[t]}),W^{[t+1]}-W^{[t]}\rangle+\frac{\ell}{2}\|W^{[t+1]}-W^{[t]}\|_2^2.\tag{2.89}$$

We can then use the definition of $W^{[t+1]}$ to write

$$\begin{aligned}
\mathsf{D}(W^{[t+1]})&\leq\mathsf{D}(W^{[t]})-\alpha^{\{t\}}\langle\nabla\mathsf{D}(W^{[t]}),F^{[t+1]}\odot\nabla r(W^{[t]}\odot F^{[t+1]},Z^{[t+1]})\rangle\\
&\qquad+\frac{\ell(\alpha^{\{t\}})^2}{2}\|F^{[t+1]}\odot\nabla r(W^{[t]}\odot F^{[t+1]},Z^{[t+1]})\|_2^2.
\end{aligned}\tag{2.90}$$

Let $\mathcal{F}_t$ be the $\sigma$-algebra of $(W^{[0]},F^{[1]},Z^{[1]},\dots,W^{[t]},F^{[t]},Z^{[t]})$. Conditional on $\mathcal{F}_t$, $F^{[t+1]}\odot\nabla r(W^{[t]}\odot F^{[t+1]},Z^{[t+1]})$ is an unbiased estimator of $\nabla\mathsf{D}(W^{[t]})$ so that by linearity

$$\mathbb{E}\left[\left\langle\nabla\mathsf{D}(W^{[t]}),F^{[t+1]}\odot\nabla r(W^{[t]}\odot F^{[t+1]},Z^{[t+1]})\right\rangle|\mathcal{F}_t\right]=\|\nabla\mathsf{D}(W^{[t]})\|_2^2.\tag{2.91}$$

Similarly to (2.91), we can decompose

$$\begin{aligned}
&\mathbb{E}\left[\|F^{[t+1]}\odot\nabla r(W^{[t]}\odot F^{[t+1]},Z^{[t+1]})\|_2^2\middle|\mathcal{F}_t\right]\\
=&\mathbb{E}\Big[\|F^{[t+1]}\odot\nabla r(W^{[t]}\odot F^{[t+1]},Z^{[t+1]})-F^{[t+1]}\odot\nabla\mathsf{U}(W^{[t]}\odot F^{[t+1]})\\
&\qquad\qquad+F^{[t+1]}\odot\nabla\mathsf{U}(W^{[t]}\odot F^{[t+1]})\|_2^2\Big|\mathcal{F}_t\Big]\\
=&\,\mathbb{E}\left[\|F^{[t+1]}\odot\nabla\mathsf{U}(W^{[t]}\odot F^{[t+1]})\|_2^2\middle|\mathcal{F}_t\right]\\
&+\mathbb{E}\left[\|F^{[t+1]}\odot\nabla\mathsf{U}(W^{[t]}\odot F^{[t+1]})-F^{[t+1]}\odot\nabla r(W^{[t]}\odot F^{[t+1]},Z^{[t+1]})\|_2^2\middle|\mathcal{F}_t\right]\\
&+2\mathbb{E}\Big[\Big\langle F^{[t+1]}\odot\nabla\mathsf{U}(W^{[t]}\odot F^{[t+1]})-F^{[t+1]}\odot\nabla r(W^{[t]}\odot F^{[t+1]},Z^{[t+1]}),\\
&\qquad\qquad F^{[t+1]}\odot\nabla\mathsf{U}(W^{[t]}\odot F^{[t+1]})\Big\rangle\Big|\mathcal{F}_t\Big]\\
=&\,\mathbb{E}\left[\|F^{[t+1]}\odot\nabla\mathsf{U}(W^{[t]}\odot F^{[t+1]})\|_2^2\middle|\mathcal{F}_t\right]\\
&+\mathbb{E}\left[\|F^{[t+1]}\odot\nabla\mathsf{U}(W^{[t]}\odot F^{[t+1]})-F^{[t+1]}\odot\nabla r(W^{[t]}\odot F^{[t+1]},Z^{[t+1]})\|_2^2\middle|\mathcal{F}_t\right],
\end{aligned}\tag{2.92}$$

where in the last step the cross-term vanishes since, by using the independence assumption of $Z^{[t+1]}$ and $F^{[t]}$, if we take the expectation first with respect to $Z^{[t+1]}$ we have

$$\mathbb{E}_{Z^{[t+1]}}[F^{[t+1]} \odot \nabla r(W^{[t]} \odot F^{[t+1]}, Z^{[t+1]}) | \mathcal{F}_t] = F^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot F^{[t+1]}). \tag{2.93}$$

Similarly, we can add and substract $\nabla \mathsf{D}(W^{[t]})$ in the first term and repeat the argument with the definitions of $\nabla \mathsf{U}$ and $\nabla \mathsf{D}$ where we take the expectation of (2.92) with respect to $F^{[t+1]}$ instead. A similar cross-term vanishes. We then obtain

$$\begin{aligned}\mathbb{E}\Big[\|F^{[t+1]} \odot \nabla r(W^{[t]} \odot F^{[t+1]}, Z^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] &\leq \|\nabla \mathsf{D}(W^{[t]})\|_2^2 \\ &+ \mathbb{E}\Big[\|\nabla \mathsf{D}(W^{[t]}) - F^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot F^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] \\ &+ \mathbb{E}\Big[\|F^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot F^{[t+1]}) - F^{[t+1]} \odot \nabla r(W^{[t]} \odot F^{[t+1]}, Z^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big].\end{aligned} \tag{2.94}$$

Define the constant $J = S^2 + \frac{3}{2}N^2(\ell^2 R^2 + 2\ell R)$. Using the bounds of Lemma 9 together with assumption (Q5) and Lemma 10 in (2.94) we obtain

$$\mathbb{E}\Big[\|F^{[t+1]} \odot \nabla r(W^{[t]} \odot F^{[t+1]}, Z^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] \leq \|\nabla \mathsf{D}(W^{[t]})\|_2^2 + 4pNS^2 + 4Np(1-p)J. \tag{2.95}$$

Substitute now (2.91) and (2.95) in (2.90). After taking the expectation, we can use a telescopic sum in (2.90) with the previous bounds, which yields

$$\begin{aligned}\sum_{t=1}^{T} \alpha^{\{t\}}\Big(1 - \frac{\ell \alpha^{\{t\}}}{2}\Big)\mathbb{E}\Big[\|\nabla \mathsf{D}(W^{[t]})\|_2^2\Big] &\leq \mathbb{E}[\mathsf{D}(W^{[0]})] - \mathbb{E}[\mathsf{D}(W^{[T]})] \\ &\quad + 2\ell Np(S^2 + (1-p)J)\sum_{t=1}^{T}(\alpha^{\{t\}})^2.\end{aligned} \tag{2.96}$$

By (Q1) we have $\mathbb{E}[\mathsf{D}(W^{[0]})] - \mathbb{E}[\mathsf{D}(W^{[T]})] \leq 2M$. Assuming that $\alpha^{\{t\}} < \frac{1}{\ell}$ for all $t \in [T]$, we then have

$$\min_{t \in [T]} \mathbb{E}\Big[\|\nabla \mathsf{D}(W^{[t]})\|_2^2\Big] \leq \frac{4M + 4\ell Np(S^2 + (1-p)J)\sum_{t=1}^{T}(\alpha^{\{t\}})^2}{\sum_{t=1}^{T}\alpha^{\{t\}}}. \tag{2.97}$$

*Proof of (a):* Setting $\alpha^{\{t\}} = \eta$ a constant we find

$$\min_{t \in [T]} \mathbb{E}\Big[\|\nabla \mathsf{D}(W^{[t]})\|_2^2\Big] \leq \frac{4M + 4\eta^2 \ell Np(S^2 + (1-p)J)}{T\eta}. \tag{2.98}$$

Minimizing the bound over $\eta$ yields that the minimum occurs at $\eta^2 = M/(\ell Np(S^2 + (1-p)J)T)$. For this $\eta$, the bound reads

$$\min_{t \in [T]} \mathbb{E}\Big[\|\nabla \mathsf{D}(W^{[t]})\|_2^2\Big] \leq 4\sqrt{p(S^2 + (1-p)J)}\sqrt{\frac{M\ell N}{T}}. \tag{2.99}$$

Finally note that the condition $\eta < 1/\ell$ is satisfied if $p > 2M\ell/(NS^2 T)$.

*Proof of (b):* We can set $\alpha^{\{t\}} = 1/(\ell\sqrt{t})$. It is easily verified that for $T \geq 4$:

$$\sum_{t=1}^{T} \alpha^{\{t\}} > \frac{\sqrt{T}}{\ell}$$
$$\sum_{t=1}^{T} (\alpha^{\{t\}})^2 < \frac{\log(T)}{\ell^2} \tag{2.100}$$

Substituting these bounds in (2.97) yields the result. □

### 2.E.1 Proof of Lemma 9

Noting that $\nabla \mathsf{D}(w) = \mathbb{E}[f \odot \nabla \mathsf{U}(w \odot f)]$ we can write

$$\begin{aligned}
\mathbb{E}\Big[\|\nabla \mathsf{D}(w) - f \odot \nabla \mathsf{U}(w \odot f)\|_2^2\Big] &= \mathbb{E}_{f_1}\Big[\|\mathbb{E}_{f_2}[f_2 \odot \nabla \mathsf{U}(w \odot f_2) - f_1 \odot \nabla \mathsf{U}(w \odot f_1)]\|_2^2\Big] \\
= \mathbb{E}_{f_1}\Big[\|\mathbb{E}_{f_2}[f_2 \odot \nabla \mathsf{U}(w \odot f_2) - f_2 \odot \nabla \mathsf{U}(w \odot f_1) + & \qquad (2.101) \\
&+ f_2 \odot \nabla \mathsf{U}(w \odot f_1) - f_1 \odot \nabla \mathsf{U}(w \odot f_1)]\|_2^2\Big] \\
\overset{\text{(i)}}{\leq} \mathbb{E}_{f_1}\Big[\mathbb{E}_{f_2}\Big[\|f_2 \odot (\nabla \mathsf{U}(w \odot f_2) - \nabla \mathsf{U}(w \odot f_1)) + (f_2 - f_1) \odot \nabla \mathsf{U}(w \odot f_1)\|_2\Big]^2\Big] \\
\overset{\text{(ii)}}{\leq} \mathbb{E}_{f_1}\Big[\mathbb{E}_{f_2}\Big[\|f_2 \odot (\nabla \mathsf{U}(w \odot f_2) - \nabla \mathsf{U}(w \odot f_1))\|_2 + \|(f_2 - f_1) \odot \nabla \mathsf{U}(w \odot f_1)\|_2\Big]^2\Big],
\end{aligned}$$

where (i) we have used Jensen's inequality for a vector-valued random variable $v$, namely $\|\mathbb{E}[v]\|_2 \leq \mathbb{E}[\|v\|_2]$, and (ii) the subadditivity of the norm $\|a+b\|_2 \leq \|a\|_2 + \|b\|_2$ for any $a, b \in \mathbb{R}^N$. We now note that

$$\begin{aligned}
\|f_2 \odot (\nabla \mathsf{U}(w \odot f_2) - \nabla \mathsf{U}(w \odot f_1))\|_2^2 &= \sum_i f_2^i |\nabla_i \mathsf{U}(w \odot f_2) - \nabla_i \mathsf{U}(w \odot f_1)|^2 \\
&\overset{\text{(i)}}{\leq} \sum_i f_2^i \ell^2 \|w \odot f_2 - w \odot f_1\|_2^2 \\
&\leq \sum_i f_2^i \ell^2 \|f_2 - f_1\|_2^2 \|w\|_2^2 \\
&\overset{\text{(ii)}}{\leq} \|f_2\|_1 \|f_2 - f_1\|_1 \ell^2 R^2,
\end{aligned} \tag{2.102}$$

where (i) we have used the Lipschitzness assumption from (Q3) and (ii) used that $\|w\|_2^2 < R^2$ and the fact that $\|f_2\|_2^2 = \|f_2\|_1$ since for any vector $f$ with entries $\{-1, 0, 1\}$ we have $\|f\|_2^2 = \|f\|_1$. We can reason similarly with $f_1 - f_2$.

Using (Q2) we can also bound

$$\|(f_2 - f_1) \odot \nabla \mathsf{U}(w \odot f_1)\|_2^2 \leq \|f_2 - f_1\|_1 S^2. \tag{2.103}$$

Hence we have in (2.101) that

$$\mathbb{E}_{f_1}\Big[\mathbb{E}_{f_2}\Big[\|f_2 \odot (\nabla \mathsf{U}(w \odot f_2) - \nabla \mathsf{U}(w \odot f_1))\|_2 + \|(f_2 - f_1) \odot \nabla \mathsf{U}(w \odot f_1)\|_2\Big]^2\Big]$$

$$
\begin{aligned}
&\leq \mathbb{E}_{f_1}\Big[\mathbb{E}_{f_2}\Big[\|f_2\|_1^{1/2}\|f_2-f_1\|_1^{1/2}\ell R+\|f_2-f_1\|_1^{1/2}S\Big]^2\Big]\\
&\overset{\text{(i)}}{\leq} \mathbb{E}_{f_1}\Big[\mathbb{E}_{f_2}\Big[\|f_2-f_1\|_1(\|f_2\|_1^{1/2}\ell R+S)^2\Big]\Big]\\
&\leq \mathbb{E}_{f_1,f_2}\Big[\|f_2-f_1\|_1(\|f_2\|_1\ell^2R^2+\|f_2\|_1^{1/2}2S\ell R+S^2)\Big]\\
&\overset{\text{(ii)}}{\leq} \mathbb{E}_{f_1,f_2}\Big[\|f_2-f_1\|_1(\|f_2\|_1(\ell^2R^2+2\ell R)+S^2)\Big], \qquad (2.104)
\end{aligned}
$$

where (i) for a random variable $v$ we have $\mathbb{E}[v]^2 \leq \mathbb{E}[v^2]$ and (ii) $\|f_2\|_1^{1/2} \leq \|f_2\|_1$ since either $\|f_2\|_1 = 0$ or $\|f_2\|_1 \geq 1$. We can now add an expectation term in the norm $\|f_2 - f_1\|_1 \leq \|f_2-\mathbb{E}[f_2]\|_1+\|f_1-\mathbb{E}[f_1]\|_1$ and $\|f_2\|_1 \leq \|f_2-\mathbb{E}[f_2]\|_1+\|\mathbb{E}[f_2]\|_1$. Here, $\|\mathbb{E}[f_2]\|_1 = \|\mathbb{E}[f_1]\|_1 = pN$ by (Q4). Hence, from (2.104) onward we can write

$$
\begin{aligned}
&\mathbb{E}_{f_1,f_2}\Big[\|f_2-f_1\|_1(\|f_2\|_1(\ell^2R^2+2\ell R)+S^2)\Big]\\
&\leq \mathbb{E}_{f_1,f_2}\Big[\Big(\|f_2-\mathbb{E}[f_2]\|_1+\|f_1-\mathbb{E}[f_1]\|_1\Big)\Big(\|f_2-\mathbb{E}[f_2]\|_1(\ell^2R^2+2\ell R)(1+pN)+S^2\Big)\Big]\\
&= \mathbb{E}_{f_1,f_2}\Big[\Big(\|f_2-\mathbb{E}[f_2]\|_1^2+\|f_1-\mathbb{E}[f_1]\|_1\|f_2-\mathbb{E}[f_2]\|_1\Big)(\ell^2R^2+2\ell R)(1+pN)\Big]\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad +2S^2\mathbb{E}_{f_2}\Big[\|f_2-\mathbb{E}[f_2]\|_1\Big]\\
&\overset{\text{(i)}}{\leq} \mathbb{E}_{f_1,f_2}\Big[\Big(\|f_2-\mathbb{E}[f_2]\|_1^2+\|f_1-\mathbb{E}[f_1]\|_1\|f_2-\mathbb{E}[f_2]\|_1\Big)(\ell^2R^2+2\ell R)(1+pN)\Big]\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad +4S^2Np(1-p)\\
&\overset{\text{(ii)}}{\leq} \mathbb{E}_{f_1,f_2}\Big[\Big(\|f_2-\mathbb{E}[f_2]\|_1^2+4N^2p^2(1-p)^2\Big)(\ell^2R^2+2\ell R)(1+pN))\Big]+4S^2Np(1-p)\\
&\overset{\text{(iii)}}{\leq} \Big(2N^2p(1-p)+4N^2p^2(1-p)^2\Big)\big(\ell^2R^2+2\ell R\big)\big(1+pN\big)+4S^2Np(1-p)\\
&= Np(1-p)\Big(\big(2N+4Np(1-p)\big)(1+pN)(\ell^2R^2+2\ell R)+4S^2\Big)\\
&\overset{\text{(iv)}}{\leq} Np(1-p)(4S^2+6N^2(\ell^2R^2+2\ell R)), \qquad (2.105)
\end{aligned}
$$

where (i) we have used Lemma 7(i), (ii) we have used independence of $f_1$ from $f_2$ and Lemma 7(i) again, (iii) we have used Lemma 7(ii), and (iv) we have bounded $1+pN < 2N$ and $p(1-p) \leq 1/4$. □

## 2.F Path representation of $\mathcal{D}(W)$ – Proofs of Lemma 1 and Corollary 1

*Proof of* (2.26). Recall that $G_F = (\mathcal{E}_F, \mathcal{V})$ is a random subgraph of $G = (\mathcal{E}, \mathcal{V})$ with edge set $\mathcal{E}_F = \{e \in \mathcal{E} : F_e = 1\}$. By (i) the law of total expectation, and by (ii) independence of $F$ and $(X,Y)$,

$$
\mathcal{D}(W) = \mathbb{E}\Big[\sum_{i=1}^{d_L}\big(Y_f - \sum_{\gamma\in\Gamma^i(G)} P_\gamma F_\gamma X_{\gamma_0}\big)^2\Big]
$$

$$
\begin{aligned}
&\overset{\text{(i)}}{=} \sum_{g\in\mathcal{G}} \mathbb{E}\Big[\sum_{f=1}^{d_L}\big(Y_f - \sum_{\gamma\in\Gamma^f(G_F)} P_\gamma X_{\gamma_0}\big)^2 \Big| \{G_F = g\}\Big]\mathbb{P}[G_F = g] \\
&\overset{\text{(ii)}}{=} \sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{f=1}^{d_L}\big(Y_f - \sum_{\gamma\in\Gamma^f(g)} P_\gamma X_{\gamma_0}\big)^2\Big]. \qquad (2.106)
\end{aligned}
$$

*Proof of* (2.27). Expand (2.106) to find

$$
\mathcal{D}(W) = \sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{f=1}^{d_L}\Big(Y_f^2 - 2Y_f \sum_{\gamma\in\Gamma^f(g)} P_\gamma X_{\gamma_0} + \sum_{\gamma\in\Gamma^f(g)}\sum_{\delta\in\Gamma^f(g)} P_\gamma X_{\gamma_0} P_\delta X_{\delta_0}\Big)\Big]. \qquad (2.107)
$$

Setting $\eta_\gamma = \sum_{\{g\in\mathcal{G} | \gamma\in\Gamma(g)\}} \mu_g$, we obtain

$$
\mathcal{D}(W) = \sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\Big(\sum_{f=1}^{d_L}\sum_{\gamma\in\Gamma^f(g)}\Big(\frac{Y_f^2}{|\Gamma^f(g)|} - 2Y_f P_\gamma X_{\gamma_0}\Big) + \sum_{\gamma\in\Gamma(g)}\sum_{\delta\in\Gamma^{\gamma_L}(g)} P_\gamma X_{\gamma_0} P_\delta X_{\delta_0}\Big)\Big] \qquad (2.108)
$$

$$
\begin{aligned}
&= \sum_{\gamma\in\Gamma(G)} \eta_\gamma \mathbb{E}\Big[\big(Y_{\gamma_L} - P_\gamma X_{\gamma_0}\big)^2\Big] \\
&- \sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{f=1}^{d_L}\sum_{\gamma\in\Gamma^f(g)}\Big(\Big(1 - \frac{1}{|\Gamma^f(g)|}\Big)Y_f^2 - P_\gamma X_{\gamma_0}\sum_{\delta\in\Gamma^f(g)\setminus\{\gamma\}} P_\delta X_{\delta_0}\Big)\Big]
\end{aligned}
$$

after rearranging terms. This completes the proof of Lemma 1 after identifying $\mathcal{J}(W)$ and $R(W)$ here as the left and right sum, respectively.

To prove Corollary 1, consider that since for an arborescence $R(W) = 0$, we can write

$$
\begin{aligned}
&\sum_{\gamma\in\Gamma(G)} \eta_\gamma \mathbb{E}\Big[\big(Y_{\gamma_L} - P_\gamma X_{\gamma_0}\big)^2\Big] \qquad (2.109) \\
&= \sum_{\gamma\in\Gamma(G)} \eta_\gamma \mathbb{E}[X_{\gamma_0}^2]\Big(\frac{\mathbb{E}[Y_{\gamma_L} X_{\gamma_0}]}{\mathbb{E}[X_{\gamma_0}^2]} - P_\gamma\Big)^2 + \sum_{\gamma\in\Gamma(G)} \eta_\gamma\Big(\mathbb{E}[Y_{\gamma_L}^2] - \frac{\mathbb{E}[Y_{\gamma_L} X_{\gamma_0}]^2}{\mathbb{E}[X_{\gamma_0}^2]}\Big) \\
&\overset{\text{(iii)}}{=} \mathcal{I}(W) + \mathcal{D}(W^{\text{opt}}).
\end{aligned}
$$

Here, (iii) follows because since $\mathcal{I}(W) \geq 0$ and $\mathcal{I}(W) = 0$ at $z_\gamma = P_\gamma$, what remains must be the optimum. This completes the proofs of Lemma 1 and Corollary 1. □

## 2.G Conserved quantities – Proof of Lemma 2

For any edge $f \in \mathcal{E}$,

$$
\begin{aligned}
W_f \frac{\partial\mathcal{D}}{\partial W_f} &\overset{(2.26)}{=} \sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{e=1}^{d} 2\big(Y_e - \sum_{\gamma\in\Gamma^e(g)} P_\gamma X_{\gamma_0}\big)\big(\sum_{\delta\in\Gamma^e(g;f)} P_\delta X_{\delta_0}\big)\Big] \\
&= \sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{\delta\in\Gamma(g;f)} 2\big(Y_{\delta_L} - \sum_{\gamma\in\Gamma^{\delta_L}(g)} P_\gamma X_{\gamma_0}\big) P_\delta X_{\delta_0}\Big]. \qquad (2.110)
\end{aligned}
$$

Note that $\Gamma(g;l)=\Gamma^l(g)$ for any leaf $l\in\mathcal{L}(G)$ and $g\in\mathcal{G}$, and therefore in particular

$$W_l\frac{\partial\mathcal{D}}{\partial W_l}=\sum_{g\in\mathcal{G}}\mu_g\sum_{\delta\in\Gamma^l(g)}\mathbb{E}\Big[2\big(Y_{\delta_L}-\sum_{\gamma\in\Gamma^{\delta_L}(g)}P_\gamma X_{\gamma_0}\big)P_\delta X_{\delta_L}\Big]. \tag{2.111}$$

Recall that $\mathcal{L}(G;f)$ is the set of leaves of the subtree of the base graph $G$ rooted at $f\in\mathcal{E}$. By the fact that $\{\Gamma^l(g;f)\}_{l\in\mathcal{L}(G;f)}$ partitions $\Gamma(g;f)$ for any $g\in\mathcal{G}$, viz.,

$$\Gamma(g;f)=\cup_{l\in\mathcal{L}(G;f)}\Gamma^l(g;f),\quad \Gamma^{l_1}(g;f)\cap\Gamma^{l_2}(g;f)=\emptyset\text{ for all }l_1\neq l_2,g\in\mathcal{G}, \tag{2.112}$$

it follows that

$$\sum_{l\in\mathcal{L}(G;f)}W_l\frac{\partial\mathcal{D}}{\partial W_l}=W_f\frac{\partial\mathcal{D}}{\partial W_f}. \tag{2.113}$$

Note in fact that this proof works for *any* base graph $G$ that has no cycles and only length-$L$ paths, so not just an arborescence. This is why we make Assumption (N6') as opposed to the stronger Assumption (N6) in Corollary 1. □

## 2.H Proof of Proposition 4

The proof of Proposition 4 is by double induction on the statements $A(t)\equiv\{\mathcal{I}(W^{\{s\}})\leq\mathcal{I}(W^{\{s-1\}})\exp(-2\nu_{\min}\kappa\alpha),\forall s\in[t]\}$ and $B(t)\equiv\{W^{\{s\}}\in K,\forall s\in[t]\}$ where $\kappa>0$ is a free parameter and $K$ is a compact set which we will define. Specifically, we prove that there exist $\alpha$ and $\kappa$ such that $A(t)\cap B(t)\Rightarrow B(t+1)$ and $A(t)\cap B(t+1)\Rightarrow A(t+1)$. Section 2.H.4 describes in detail how the upcoming Lemmas 11–13 provide sufficient conditions for the induction step. There we also maximize the upper bound for the convergence rate over $\kappa$, which gives the rate in (2.36).

We start by giving Lemmas 11–13. Recall first the definition of the set $B(\epsilon,I)$ in (2.34). Here, with a minor abuse of notation, we define also

$$B(\epsilon,\{C_f\}_{f\in\mathcal{E}\backslash\mathcal{L}(G)})\triangleq\Big\{W\in\mathcal{W}\big|\mathcal{I}(W)\leq\epsilon,W_f^2-\sum_{l\in\mathcal{L}(G;f)}W_{\gamma^l}^2=C_f\Big\} \tag{2.114}$$

where $\{\gamma^l\}\triangleq\Gamma^l(G)$ for $l\in\mathcal{L}(G)$ if $G$ is an arborescence.

**Lemma 11.** *Assume (N2) from Proposition 1 and (N6) from Corollary 1. Then:*
*(i) If $\epsilon>0$ and $|C_f|<\infty$ for $f\in\mathcal{E}\backslash\mathcal{L}(G)$, then the set $B(\epsilon,\{C_f\}_{f\in\mathcal{E}\backslash\mathcal{L}})$ is compact.*
*(ii) If $\max_{\gamma\in\Gamma(G)}|z_\gamma|\leq M^L$, then the function $\mathcal{I}(W)$ is $\beta$-smooth in $\mathcal{S}$ with $\beta=6\nu_{\max}\cdot|\mathcal{E}(G)||\Gamma(G)|M^{2(L-1)}$.*

Lemma 11 implies that $B(\epsilon,I)$ is compact and that $\mathcal{D}(W)$ is $\beta$-smooth on the compact set $K=\mathcal{S}\cap B(\epsilon,I)$, i.e.,

$$\mathcal{D}(W')-\mathcal{D}(W)\leq\nabla\mathcal{D}(W)^{\mathrm{T}}(W'-W)+\beta\|W'-W\|_2^2 \tag{2.115}$$

for $W,W'\in K$. Its proof is deferred to Section 2.H.1.

Next, Lemma 12 gives a lower bound for the curvature of $\mathcal{D}(W)$ on $K$ in the direction of $\nabla\mathcal{D}(W)$, in the form of a PL-inequality [77]. Its proof is in Section 2.H.2.

**Lemma 12.** *Assume (N2) from Proposition 1 and (N6) from Corollary 1. If* $W^{\{t\}} \in \mathcal{S} \cap B(\epsilon, I)$*, then*

$$\|\nabla \mathcal{D}(W^{\{t\}})\|_2^2 \geq 4\nu_{\min}(C_{\min}^{\{t\}})^{(L-1)}\big(\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\text{opt}})\big). \tag{2.116}$$

Lemma 13 proves that the conserved quantities of the gradient flow remain bounded under the GD algorithm in (2.22). This lemma allows us to keep track of the iterates in the compact set $K = \mathcal{S} \cap B(\epsilon, I)$ by relating them to conserved quantities and exploiting the fact that under GD $|C_f^{\{t+1\}} - C_f^{\{t\}}|$ has order $O(\alpha^2)$. Section 2.H.2 contains its proof.

**Lemma 13.** *Assume (N2) from Proposition 1 and (N6) from Corollary 1. If* $W^{\{t\}} \in \mathcal{S}$*, and* $C_f^{\{t\}} > 0$ *for all* $f \in \mathcal{E}\backslash\mathcal{L}(G)$*, then* $4\alpha^2\|\nu\|_1 M^{2(L-1)}\big(\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\text{opt}})\big) \geq |C_f^{\{t+1\}} - C_f^{\{t\}}|$.

*A note on the exchange of derivative and expectation in this section.* Whenever we make both Assumption (N2) in Proposition 1 and (N7) in Lemma 1, the exchange of derivative and expectation is warranted. This occurs several times throughout this section. We refer to the proof of Lemma 5 for the details.

## 2.H.1 Compactness, and smoothness – Proof of Lemma 11

In the proof of Lemma 11, we will upper bound the operator norm of the Hessian. Recall that for a symmetric bilinear matrix $A$, $\|A\|_{\text{op}} \triangleq \sup_{\|v\|_2=1}|v^T Av|$.

*Proof of (i).* By continuity of the conditions in (2.34), the set $B(\epsilon, \{C_f\}_{f\in\mathcal{E}\backslash\mathcal{L}})$ is closed. We need to prove boundedness. Let $W \in B(\epsilon, \{C_f\}_{f\in\mathcal{E}\backslash\mathcal{L}})$, and suppose w.l.o.g. that for some $f^* \in \mathcal{E}\backslash\mathcal{L}$ we have $|W_{f^*}| > Q$, where $Q > \max_{j\in\mathcal{E}\backslash\mathcal{L},\gamma\in\Gamma(G)}\{|C_j|, |z_\gamma|\}$. We want to find a path $\gamma \in \Gamma(G)$ such that $P_\gamma$ is large for a contradiction with the assumption that $\mathcal{I}(W) \leq \epsilon$. By (2.30), we have the inequality $\sum_{l\in\mathcal{L}(G;f^*)} W_l^2 > Q^2 - |C_{f^*}|$ so that for some $l^* \in \mathcal{L}(G;f^*)$ we must have $W_{l^*}^2 > (Q - |C_{f^*}|)/|\mathcal{L}(G;f^*)|$. Consequently, we have by (2.30) that $|W_e|^2 > (Q^2 - |C_{f^*}|)/|\mathcal{L}(G;f^*)| - |C_e|$ for any edge $e \in \gamma$ in any path $\gamma \in \Gamma^{l^*}(G)$ except for the edge $f^*$ where we have $|W_{f^*}| > Q$ by assumption. In particular, we have the bound $|W_e| > O(Q)$ for any edge $e \in \gamma$ for any path $\gamma \in \Gamma(G;f^*)$. Therefore if we pick $\gamma \in \Gamma(G;f^*)$ we have

$$\epsilon \overset{(2.34)}{\geq} \mathcal{I}(W) \geq \nu_\gamma(z_\gamma - P_\gamma)^2 \geq \nu_\gamma(|P_\gamma| - |z_\gamma|)^2 > O(Q^{2L}) \tag{2.117}$$

for sufficiently large $Q$, which is a contradiction. We must thus have $|W_{f^*}| \leq Q$ for some $Q < \infty$. If on the other hand $|W_l| > Q$ for some $l \in \mathcal{L}(G;f^*)$, by (2.30) we must also have $(W_{f^*})^2 > Q^2 + C_{f^*} > O(Q^2)$ for sufficiently large $Q$. This case is, thus, the same as before.
*Proof of (ii).* Using a regular upper bound for the entries of $\nabla^2\mathcal{I}(W)$ when $W \in \mathcal{S}$ will suffice. Element-wise, we have

$$(\nabla^2\mathcal{I}(W))_{i,j} \tag{2.118}$$
$$= \begin{cases} 2\sum_{\delta\in\Gamma(G;i)\cap\Gamma(G;j)}\nu_\delta\Big(\frac{P_\delta}{W_i}\frac{P_\delta}{W_j} - \frac{P_\delta}{W_iW_j}(z_\gamma - P_\gamma)\Big), & \text{if } i \neq j, \Gamma(G;i)\cap\Gamma(G;j) \neq \emptyset, \\ 2\sum_{\gamma\in\Gamma(G;i)}\nu_\gamma(\frac{P_\gamma}{W_i})^2 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, noting that since we have $\left|W_f\right| \leq M$ for all $f \in \mathcal{E}$ on $\mathcal{S}$, we can bound $\left|P_\gamma / W_f\right| \leq M^{L-1}$, $|z_\gamma| \leq M^L$ and the other terms similarly. We upper bound the number of terms in the sum over $\Gamma(G;i)$ and $\Gamma(G;i) \cap \Gamma(G;j)$ by $|\Gamma(G)|$ and $\nu_\gamma \leq \nu_{\max}$. Adding all terms, we obtain that $6\nu_{\max}|\Gamma(G)|\, M^{2(L-1)}$ is an upper bound for each of the entries of $\nabla^2 \mathcal{I}(W)$. This gives an upper bound $\|\nabla^2 \mathcal{I}(W)\|_{\mathrm{op}} \leq 6|\mathcal{E}|\nu_{\max}|\Gamma(G)|\, M^{2(L-1)}$ in $\mathcal{S}$. □

### 2.H.2 PL-inequality on a compact set – Proof of Lemma 12

Recall the definition of a PL-inequality:

**Definition 7.** *Let* $u \in C^2(K,\mathbb{R})$ *where* $K \subset \mathbb{R}^n$ *is compact and* $K \backslash \partial K \neq \emptyset$. *Denote by* $u^* = \min_{x \in K} u(x)$ *and suppose that* $u^* \in K \backslash \partial K$. *We say that* $u$ *satisfies a* Polyak–Łojasiewicz (PL) *inequality if there exist a* $\tau_K > 0$ *depending only on* $K$ *such that*

$$\|\nabla u(x)\|_2^2 \geq \tau_K (u(x) - u^*) \quad \textit{for all} \quad x \in K. \tag{2.119}$$

A PL-inequality together with $\beta$-smoothness on a compact set will imply that $\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\mathrm{opt}})$ decreases. To see this, note that by (i) $\beta$-smoothness, and (ii) the update rule

$$\begin{aligned} \mathcal{D}(W^{\{t+1\}}) - \mathcal{D}(W^{\{t\}}) &\overset{\text{(i)}}{\leq} \nabla \mathcal{D}(W^{\{t\}})^{\mathrm{T}} (W^{\{t+1\}} - W^{\{t\}}) + \beta \|W^{\{t+1\}} - W^{\{t\}}\|_2^2 \\ &\overset{\text{(ii)}}{=} \alpha\big(\beta\alpha - 1\big) \|\nabla \mathcal{D}(W^{\{t\}})\|_2^2 \end{aligned} \tag{2.120}$$

If furthermore $\alpha \leq 1/(2\beta)$, then also $\beta\alpha - 1 \leq -1/2$. Together with (2.119), and after rearranging terms, one finds that

$$\mathcal{D}(W^{\{t+1\}}) - \mathcal{D}(W^{\{t\}}) \leq \frac{\alpha \tau_K}{2} (\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\mathrm{opt}})) \quad \text{for all} \quad W \in K. \tag{2.121}$$

By (iii) $1 + x \leq \exp(x)$ for all $x \in \mathbb{R}$, we obtain (2.37). The strategy will now be to prove that there is a PL-inequality in some compact set, that the iterates remain in that compact set, and that the function is $\beta$-smooth.

*Proof of Lemma 12.* First note that if $l \in \mathcal{L}(G)$ and $\gamma \in \Gamma(G;l)$, the indexes of the weights in the product $|P_\gamma^{\{t\}} / W_l^{\{t\}}|$ belong to the index set $\mathcal{E} \backslash \mathcal{L}(G)$. The proof follows (i) by restricting the sum, and (ii) from the fact that for every path $\gamma \in \Gamma(G)$ in an arborescence $G$, there is exactly one leaf $l \in \mathcal{L}(G)$ such that $\gamma^l = \gamma$. Thus

$$\begin{aligned} \sum_{e \in \mathcal{E}} \Big| \frac{\partial}{\partial W_e} \mathcal{I}(W^{\{t\}}) \Big|^2 &= 4 \sum_{e \in \mathcal{E}} \Big| \sum_{\gamma \in \Gamma(G;e)} \nu_\gamma \frac{P_\gamma^{\{t\}}}{W_e^{\{t\}}} (z_\gamma - P_\gamma^{\{t\}}) \Big|^2 \\ &\overset{\text{(i)}}{\geq} 4 \sum_{l \in \mathcal{L}(G)} \Big| \nu_{\gamma^l} \frac{P_{\gamma^l}^{\{t\}}}{W_l^{\{t\}}} (z_{\gamma^l} - P_{\gamma^l}^{\{t\}}) \Big|^2 \\ &\overset{\text{(ii)}}{=} 4 \sum_{\gamma \in \Gamma(G)} \nu_\gamma^2 \Big| \frac{P_\gamma^{\{t\}}}{W_{\gamma_L}^{\{t\}}} (z_\gamma - P_\gamma^{\{t\}}) \Big|^2 \\ &\overset{\text{(iii)}}{\geq} 4\nu_{\min} \big( \min_{f \in \mathcal{E} \backslash \mathcal{L}(G)} |W_f^{\{t\}}|^2 \big)^{L-1} \mathcal{I}(W^{\{t\}}), \end{aligned} \tag{2.122}$$

where in (iii) we have used the bound $|W_i^{\{t\}}| \geq \min_{e \in \mathcal{E} \backslash \mathcal{L}(G)} |W_e^{\{t\}}|$ for all $i \in \mathcal{E} \backslash \mathcal{L}(G)$ and similarly with $\nu_\gamma \geq \nu_{\min}$ for $\gamma \in \Gamma(G)$. Finally, by (2.30), we have $\min_{e \in \mathcal{E} \backslash \mathcal{L}(G)} |W_e^{\{t\}}|^2 \geq C_{\min}^{\{t\}}$. This completes the proof. □

### 2.H.3 Conserved quantities remain bounded throughout GD – Proof of Lemma 13

*Proof.* Pick $f \in \mathcal{E} \backslash \mathcal{L}(G)$. By (i) Corollary 1, and (ii) Lemma 2, we have

$$\begin{aligned}
C_f^{\{t+1\}} &= (W_f^{\{t+1\}})^2 - \sum_{l \in \mathcal{L}(G;i)} (W_l^{\{t+1\}})^2 \\
&\overset{(2.22)}{=} \Big(W_f^{\{t\}} - \alpha \frac{\partial}{\partial W_f} \mathcal{D}(W^{\{t\}})\Big)^2 - \sum_{l \in \mathcal{L}(G;f)} \Big(W_l^{\{t\}} - \alpha \frac{\partial}{\partial W_l} \mathcal{D}(W^{\{t\}})\Big)^2 \\
&\overset{\text{(i)}}{=} \Big(W_f^{\{t\}} - \alpha \frac{\partial}{\partial W_f} \mathcal{I}(W^{\{t\}})\Big)^2 - \sum_{l \in \mathcal{L}(G;f)} \Big(W_l^{\{t\}} - \alpha \frac{\partial}{\partial W_l} \mathcal{I}(W^{\{t\}})\Big)^2 \\
&\overset{\text{(ii)}}{=} C_f^{\{t\}} + \alpha^2 \Big( \Big(\frac{\partial}{\partial W_f} \mathcal{I}(W^{\{t\}})\Big)^2 - \sum_{l \in \mathcal{L}(G;f)} \Big(\frac{\partial}{\partial W_l} \mathcal{I}(W^{\{t\}})\Big)^2 \Big)
\end{aligned}$$

$$= C_i^{\{t\}} + 4\alpha^2 \Big( \Big( \sum_{\gamma \in \Gamma(G;f)} \nu_\gamma \frac{P_\gamma^{\{t\}}}{W_f^{\{t\}}} (z_\gamma - P_\gamma^{\{t\}}) \Big)^2 - \sum_{l \in \mathcal{L}(G;f)} \nu_{\gamma^l}^2 \Big( \frac{P_{\gamma^l}^{\{t\}}}{W_l^{\{t\}}} \Big)^2 (z_{\gamma^l} - P_{\gamma^l}^{\{t\}})^2 \Big) \tag{2.123}$$

$$\geq C_f^{\{t\}} - 4\alpha^2 \Big( \sum_{l \in \mathcal{L}(G;f)} \nu_{\gamma^l}^2 \Big( \frac{P_{\gamma^l}^{\{t\}}}{W_l^{\{t\}}} \Big)^2 (z_{\gamma^l} - P_{\gamma^l}^{\{t\}})^2 \Big). \tag{2.124}$$

By Cauchy–Schwartz we also have

$$\Big( \sum_{\gamma \in \Gamma(G;f)} \nu_\gamma \frac{P_\gamma^{\{t\}}}{W_f^{\{t\}}} (z_\gamma - P_\gamma^{\{t\}}) \Big)^2 \leq \Big( \sum_{\gamma \in \Gamma(G;f)} \nu_\gamma \Big) \sum_{\gamma \in \Gamma(G;f)} \nu_\gamma \Big( \frac{P_\gamma^{\{t\}}}{W_l^{\{t\}}} \Big)^2 (z_\gamma - P_\gamma^{\{t\}})^2. \tag{2.125}$$

If we have $C_f^{\{t\}} > 0$, then $(W_f^{\{t\}})^2 > (W_{\gamma_L}^{\{t\}})^2$ for any $\gamma \in \Gamma(G;f)$. Thus, combining the estimate (2.123) with (2.125) we obtain

$$C_f^{\{t+1\}} \leq C_f^{\{t\}} + 4 \Big( \sum_{\gamma \in \Gamma(G;f)} \nu_\gamma \Big) \alpha^2 \Big( \sum_{l \in \mathcal{L}(G;f)} \nu_{\gamma^l} \Big( \frac{P_{\gamma^l}^{\{t\}}}{W_l^{\{t\}}} \Big)^2 (z_{\gamma^l} - P_{\gamma^l}^{\{t\}})^2 \Big). \tag{2.126}$$

Extending the sums in (2.126) from $\Gamma(G;f)$ to $\Gamma(G)$ and from $\mathcal{L}(G;f)$ to $\mathcal{L}(G)$, respectively, yields

$$C_f^{\{t+1\}} - C_f^{\{t\}} \leq 4 \|\nu\|_1 \alpha^2 \big( \max_{e \in \mathcal{E} \backslash \mathcal{L}(G)} |W_e^{\{t\}}|^2 \big)^{L-1} \mathcal{I}(W^{\{t\}}), \tag{2.127}$$

where we have used the bound $|W_f| \leq \max_{e \in \mathcal{E} \backslash \mathcal{L}(G)} |W_e|$ for all $f \in \mathcal{E} \backslash \mathcal{L}(G)$. Similarly, using (2.124) and the trivial bound $\nu_\gamma \leq \|\nu\|_1$ for any $\gamma \in \Gamma$, and by absorbing one $\nu_\gamma$-term into $\mathcal{I}(W)$'s expression, we obtain

$$C_f^{\{t+1\}} \geq C_f^{\{t\}} - 4 \|\nu\|_1 \alpha^2 \big( \max_{e \in \mathcal{E} \backslash \mathcal{L}(G)} |W_e^{\{t\}}|^2 \big)^{L-1} \mathcal{I}(W^{\{t\}}) \tag{2.128}$$

for the lower bound. Because $W^{\{t\}} \in \mathcal{S}$ by assumption, $\max_{e \in \mathcal{E} \backslash \mathcal{L}(G)} |W_e^{\{t\}}|^2 \leq M^2$. This completes the proof. □

### 2.H.4 Double induction

We now use Lemmas 11–13 together in a double induction to finally prove Proposition 4. Let $\kappa > 0$ and denote the statements:

$$A(t) \equiv \{\mathcal{I}(W^{\{s\}}) \leq \mathcal{I}(W^{\{s-1\}})\exp(-2\nu_{\min}\kappa\alpha), \forall s \in [t]\}, \tag{2.129}$$

$$B(t) \equiv \{W^{\{s\}} \in B(\epsilon, I) \cap \mathcal{S} \forall s \in [t]\}. \tag{2.130}$$

We will prove that there exists a $\kappa > 0$ such that when choosing $\alpha$ appropriately, firstly

$$A(t) \cap B(t) \Rightarrow B(t+1), \tag{2.131}$$

and secondly,

$$A(t) \cap B(t+1) \Rightarrow A(t+1). \tag{2.132}$$

*Step 1: $A(t) \cap B(t) \Rightarrow B(t+1)$.* We need to prove that $W^{\{t+1\}} \in B(\epsilon, I) \cap \mathcal{S}$ assuming (2.129) and (2.130). Using (2.127) from the proof of Lemma 13 repeatedly with the bound $\max_{e\in\mathcal{E}} |W_e^{\{t\}}| \leq M$, we obtain

$$C_f^{\{t+1\}} \leq C_f^{\{0\}} + 4\|\nu\|_1 M^{2(L-1)}\alpha^2 \sum_{s=0}^{t} \mathcal{I}(W^{\{s\}}). \tag{2.133}$$

By (2.129), we can upper bound

$$\sum_{s=0}^{t} \mathcal{I}(W^{\{s\}}) \overset{(2.129)}{\leq} \sum_{s=0}^{t} \mathcal{I}(W^{\{0\}})\exp(-2\nu_{\min}\kappa\alpha s) \leq \mathcal{I}(W^{\{0\}})\frac{1}{1-\exp(-2\nu_{\min}\kappa\alpha)}. \tag{2.134}$$

If furthermore (C1) $0 < 2\nu_{\min}\kappa\alpha < 1$, then (i) the inequality $1/(1-\exp(-2\nu_{\min}\kappa\alpha)) < 1/(\nu_{\min}\kappa\alpha)$ holds, so that

$$C_{\min}^{\{t+1\}} \overset{(2.133)}{\leq} C_{\min}^{\{0\}} + 4\|\nu\|_1 M^{2(L-1)}\alpha^2 \sum_{s=0}^{t} \mathcal{I}(W^{\{s\}}) \overset{(i)}{\leq} C_{\min}^{\{0\}} + 4\frac{\|\nu\|_1}{\nu_{\min}} M^{L-1}\alpha\kappa^{-1}\mathcal{I}(W^{\{0\}}). \tag{2.135}$$

In the same manner, we can also prove (2.135) for $C_f^{\{0\}}$ instead of $C_{\min}^{\{0\}}$. This yields

$$C_f^{\{t+1\}} \leq C_f^{\{0\}} + 4\frac{\|\nu\|_1}{\nu_{\min}\kappa} M^{2(L-1)}\alpha\mathcal{I}(W^{\{0\}}), \tag{2.136}$$

for any $f \in \mathcal{E}\backslash\mathcal{L}(G)$. Similarly, for a lower bound, we can use (2.128) repeatedly together with the bound (2.134) and condition (C1) yielding

$$C_f^{\{t+1\}} \geq C_f^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}\kappa} M^{2(L-1)}\alpha\mathcal{I}(W^{\{0\}}). \tag{2.137}$$

for any $f \in \mathcal{E}\backslash\mathcal{L}(G)$. Now, suppose (D1) $C_{\min}^{\{0\}} - \kappa^{1/(L-1)} > 0$ and let (C2) the step size satisfy

$$\alpha \leq \nu_{\min}\kappa\frac{C_{\min}^{\{0\}} - \kappa^{1/(L-1)}}{8\|\nu\|_1 M^{2(L-1)}\mathcal{I}(W^{\{0\}})}. \tag{2.138}$$

We have (i) by (2.136) and (2.137) that

$$\begin{aligned}
C_f^{\{t+1\}} &\overset{\text{(i)}}{\in} [C_f^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}} M^{2(L-1)}\alpha\kappa^{-1}\mathcal{I}(W^{\{0\}}), C_f^{\{0\}} + 4\frac{\|\nu\|_1}{\nu_{\min}} M^{2(L-1)}\alpha\kappa^{-1}\mathcal{I}(W^{\{0\}})] \\
&\overset{(2.138)}{\subseteq} [C_f^{\{0\}} - \frac{1}{2}(C_{\min}^{\{0\}} - \kappa^{1/(L-1)}), C_f^{\{0\}} + \frac{1}{2}(C_{\min}^{\{0\}} - \kappa^{1/(L-1)})] \\
&\overset{(D1)}{\subseteq} [C_f^{\{0\}} - C_f^{\{0\}}/2, C_f^{\{0\}} + C_f^{\{0\}}/2] \subseteq [C_f^{\{0\}}/2, 3C_f^{\{0\}}/2] = I_f.
\end{aligned} \tag{2.139}$$

Then $W^{\{t+1\}} \in B(\epsilon, I)$ by (2.34). Hence, $M > W_f^{\{t+1\}} \overset{(2.30)}{>} (C_f^{\{0\}}/2)^{1/2} \geq (C_{\min}^{\{0\}}/2)^{1/2} > \delta$ for any $f \in \mathcal{E}\backslash\mathcal{L}(G)$. Since moreover $C_e^{\{t+1\}} > 0$ for all $e \in \mathcal{E}\backslash\mathcal{L}(G)$, we have that if $f \in \mathcal{L}(G)$, then $M^2 > (W_j^{\{t+1\}})^2 > (W_f^{\{t+1\}})^2$ for some $j \in \mathcal{E}\backslash\mathcal{L}(G)$. Consequently $M \geq |W_f^{\{t+1\}}|$ and $W^{\{t+1\}} \in \mathcal{S}$.

*Step 2:* $A(t) \cap B(t+1) \Rightarrow A(t+1)$. Suppose that $W^{\{s\}} \in B(\epsilon, I) \cap \mathcal{S}$ for $s = 0, 1, \ldots, t+1$. Using the bound in (2.136) which requires the induction hypothesis $A(t)$ and (C1) for $C_{\min}^{\{t\}}$, we obtain

$$C_{\min}^{\{t\}} \geq C_{\min}^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}\kappa} M^{2(L-1)}\alpha\mathcal{I}(W^{\{0\}}). \tag{2.140}$$

Suppose now for a moment that (C2) the right-hand side of (2.140) is positive for some sufficiently small $\alpha$. We could then use the PL inequality from Lemma 12 together with $\min_{e\in\mathcal{E}\backslash\mathcal{L}(G)} |W_e^{\{t\}}|^{2(L-1)} \geq (C_{\min}^{\{t\}})^{L-1}$, that is,

$$\|\nabla\mathcal{I}(W^{\{t\}})\|_2^2 \geq 4\nu_{\min}(C_{\min}^{\{t\}})^{L-1}\mathcal{I}(W^{\{t\}}). \tag{2.141}$$

To see how, note that the argumentation around (2.121) together with (2.141) and (i) the induction hypothesis $B(t+1)$ we have $W^{\{t\}}, W^{\{t+1\}} \in B(\epsilon, I) \cap \mathcal{S}$ and (ii) the clause (L1) $\alpha \leq 1/(2\beta)$, implies

$$\begin{aligned}
\mathcal{I}(W^{\{t+1\}}) &\overset{\text{(i,ii, 2.141)}}{\leq} \mathcal{I}(W^{\{t\}})\exp\big(-2\nu_{\min}\alpha(C_{\min}^{\{t\}})^{L-1}\big) \\
&\overset{(2.140)}{\leq} \mathcal{I}(W^{\{t\}})\exp\Big(-2\nu_{\min}\alpha\big(C_{\min}^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}\kappa}M^{2(L-1)}\alpha\mathcal{I}(W^{\{0\}})\big)\Big) \qquad (2.142) \\
&\overset{\text{(iii)}}{\leq} \mathcal{I}(W^{\{0\}})\exp\Big(-2\nu_{\min}\alpha\big(C_{\min}^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}\kappa}M^{2(L-1)}\alpha\mathcal{I}(W^{\{0\}})\big)^{L-1} - 2\nu_{\min}\alpha\kappa t\Big),
\end{aligned}$$

where we have also used (iii) the induction hypothesis $A(t)$, i.e., that $\mathcal{I}(W^{\{t\}}) \leq \mathcal{I}(W^{\{0\}}) \cdot \exp(-2\nu_{\min}\kappa\alpha t)$.

We now investigate the exponent in (2.142) for a moment. Assuming (C2) and if (C3) the right-hand side of (2.142) is furthermore smaller than $\mathcal{I}(W^{\{0\}})\exp(-2\nu_{\min}\kappa\alpha(t+1))$, then the induction step would be complete. Note finally that both conditions (C2) and (C3) are satisfied when choosing

$$\kappa \leq \big(C_{\min}^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}}M^{2(L-1)}\alpha\kappa^{-1}\mathcal{I}(W^{\{0\}})\big)^{L-1}, \tag{2.143}$$

or equivalently

$$\alpha \leq \nu_{\min}\kappa\frac{C_{\min}^{\{0\}} - \kappa^{1/(L-1)}}{4\|\nu\|_1 M^{2(L-1)}\mathcal{I}(W^{\{0\}})}. \tag{2.144}$$

To also satisfy condition (C1), we thus require that

$$\alpha \leq \min\Big(\frac{1}{2\nu_{\min}\kappa}, \nu_{\min}\kappa\frac{C_{\min}^{\{0\}} - \kappa^{1/(L-1)}}{4\left\|\nu\right\|_1 M^{2(L-1)}\mathcal{I}(W^{\{0\}})}\Big). \tag{2.145}$$

*Step 3.* Let us summarize. Convergence occurs at rate at most $2\nu_{\min}\kappa\alpha$ if conditions (L1), (D1), (C1)–(C3) hold. Hence we have to choose $\kappa > 0$ such that $C_{\min}^{\{0\}} - \kappa^{L-1} > 0$ and

$$\alpha \leq \min\Big(\nu_{\min}\kappa\frac{C_{\min}^{\{0\}} - \kappa^{1/(L-1)}}{8\left\|\nu\right\|_1 M^{2(L-1)}\mathcal{I}(W^{\{0\}})}, \frac{1}{2\beta}, \frac{1}{2\nu_{\min}\kappa}\Big). \tag{2.146}$$

Note that we can maximize the convergence rate $2\nu_{\min}\alpha\kappa$ by maximizing $\kappa^2(C_{\min}^{\{0\}} - \kappa^{1/(L-1)})$, which occurs whenever

$$\kappa = (C_{\min}^{\{0\}})^{L-1}(1+1/(2(L-1)))^{-(L-1)} \geq \exp(-1/2)(C_{\min}^{\{0\}})^{L-1}. \tag{2.147}$$

Substituting this in (2.146) we require a step size

$$\alpha \leq \min\Big(\nu_{\min}\frac{\exp(1/2)(C_{\min}^{\{0\}})^{L}}{8\left\|\nu\right\|_1 (2L-1)M^{2(L-1)}\mathcal{I}(W^{\{0\}})}, \frac{1}{2\beta}, \frac{1}{2\nu_{\min}(C_{\min}^{\{0\}})^{L-1}}\Big). \tag{2.148}$$

Finally, we have the bound $\beta \leq 6\nu_{\max}\left|\mathcal{E}(G)\right|\left|\Gamma(G)\right| M^{2(L-1)}$ from Lemma 11 in $\mathcal{S}$, so that

$$\begin{aligned}\alpha \leq \min\Big(\nu_{\min}\frac{\exp(1/2)(C_{\min}^{\{0\}})^{L}}{16\left\|\nu\right\|_1 LM^{2(L-1)}\mathcal{I}(W^{\{0\}})}, \qquad\qquad & \qquad (2.149)\\ \frac{1}{12\nu_{\max}\left|\mathcal{E}(G)\right|\left|\Gamma(G)\right| M^{2(L-1)}}, \frac{1}{2\nu_{\min}(C_{\min}^{\{0\}})^{L-1}}\Big).\end{aligned}$$

This completes the proof of Proposition 4. □

## 2.I Convergence rate in the case of *Dropout* and *Dropconnect* – Proof of Corollary 3

We consider first the case of *Dropconnect*. We have that $\{F_e\}_{e\in\mathcal{E}}$ are independent Bernoulli($p$) random variables. Suppose that the base graph $G$ has no cycles and every path is of length $L$. Then by definition in Lemma 1, we have

$$\begin{aligned}\eta_\gamma &= \sum_{\{g\in\mathcal{G}|\gamma\in\Gamma(g)\}} \mathbb{P}[G_F = g] = \sum_{g\in\mathcal{G}} \mathbb{1}[\gamma\in\Gamma(g)]\mathbb{P}[G_F = g]\\ &= \sum_{g\in\mathcal{G}} \mathbb{P}[\gamma\in\Gamma(g)|G_F = g]\mathbb{P}[G_F = g] = \mathbb{P}[\gamma\in\Gamma(G_F)] \overset{\text{(i)}}{=} p^L,\end{aligned} \tag{2.150}$$

where (i) we have used *Dropconnect*'s distribution on $F$.

Now suppose that additionally we make the stronger assumption that $G$ is an arborescence. Then by definition in Corollary 1 $\nu_\gamma = \mathbb{E}[X^2]\eta_\gamma$, and subsequently we can calculate $\left\|\nu\right\|_1 = \mathbb{E}[X^2]\sum_{\gamma\in\Gamma(G)}\nu_\gamma = \mathbb{E}[X^2]\left|\Gamma(G)\right|p^L = \mathbb{E}[X^2]d_L p^L = \mathbb{E}[X^2]|\mathcal{L}(G)|p^L$.

Now, since by assumption $\max_\gamma |z_\gamma| \le M^L$ and $\left|W_f\right| \le M$ for all $f \in \mathcal{E}$, then $\mathcal{I}(W^{\{0\}}) \le O(|\Gamma(G)|\, M^{2L})$ so that substitution of in the definition of $\alpha$ in Proposition 4 yields

$$\alpha = O\Big(\frac{(C_{\min}^{\{0\}})^L}{LM^{4L}}\Big), \tag{2.151}$$

where we have used that $C_{\min} \le M^2$. Finally multiplying by $\tau$ gives the rate

$$\alpha\tau = O\Big(\frac{p^L (C_{\min}^{\{0\}})^2 L}{L|\mathcal{L}(G)|^2 M^{4L}}\Big). \tag{2.152}$$

Substituting these results in the rate $\tau\alpha$ in Proposition 4 yields the result for *Dropconnect*.

Finally we note that for the case of *Dropout*, filtering all nodes independently in an arborescence is equivalent to filtering all edges independently except the edge at the root. In particular, in (2.150), we have $\mathbb{P}[\gamma \in \Gamma(G_F)] = p^{L-1}$. The remaining steps of the proof are then the same as for *Dropconnect* and comparing $p^L$ with $p^{L-1}$ we can absorb the missing $p$ factor into the $O$ notation, which does not change the order in $L$. □

## 2.J Inequalities pertaining to the Frobenius norm

**Lemma 14.** *For any matrix $A \in \mathbb{R}^{m\times n}$ and $1 \le k < \infty$, it holds that $\sum_{i,j}(1+A_{ij}^2)^k \le nm(1+\|A\|_{\mathrm{F}})^{2k}$. For any two matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$ and $0 \le k < \infty$, it holds that $(1+\|AB\|_{\mathrm{F}})^k \le (1+\|A\|_{\mathrm{F}})^k(1+\|B\|_{\mathrm{F}})^k$. For any two matrices $A, B \in \mathbb{R}^{n\times m}$, it holds that $\|A \odot B\|_{\mathrm{F}} \le \|A\|_{\mathrm{F}}\,\|B\|_{\mathrm{F}}$.*

*Proof.* Recall Minkowski's inequality for sequences which states that for $1 \le k < \infty$, the inequality $\big(\sum_i |x_i + y_i|^k\big)^{1/k} \le \big(\sum_i |x_i|^k\big)^{1/k} + \big(\sum_i |y_i|^k\big)^{1/k}$ holds. It (i) implies that for any matrix $A \in \mathbb{R}^{n\times m}$ and $1 \le k < \infty$, that

$$\sum_{i,j}(1+A_{ij}^2)^k \overset{\text{(i)}}{\le} \Big((nm)^{1/k} + \big(\sum_{i,j}|A_{i,j}^2|^k\big)^{1/k}\Big)^k \overset{\text{(ii)}}{\le} nm\Big(1 + \big(\sum_{i,j}|A_{i,j}^2|^k\big)^{1/k}\Big)^k \tag{2.153}$$

where (ii) we have used that the function $z^k$ is nondecreasing in $z \ge 0$ whenever $k \ge 0$. Because (iii) for the $\ell_k$-norm for sequences it holds that $\|x\|_{2k}^2 \le \|x\|_2^2$ whenever $1 \le k < \infty$, we obtain

$$\sum_{i,j}(1+A_{ij}^2)^k \overset{\text{(iii)}}{\le} nm(1+\|A\|_{\mathrm{F}}^2)^k \overset{\text{(iv)}}{\le} nm(1+\|A\|_{\mathrm{F}})^{2k} \tag{2.154}$$

where (iv) we have used that the function $(1+z^2)^k \le (1+z)^{2k}$ for all $z \ge 0$ whenever $k \ge 0$. This proves the first inequality.

The second inequality is an immediate consequence of the submultiplicativity property of the Frobenius norm and its positivity, i.e.,

$$1 + \|AB\|_{\mathrm{F}} \le 1 + \|A\|_{\mathrm{F}}\|B\|_{\mathrm{F}} \le 1 + \|A\|_{\mathrm{F}} + \|B\|_{\mathrm{F}} + \|A\|_{\mathrm{F}}\|B\|_{\mathrm{F}}. \tag{2.155}$$

Raising to the $k$-th power left and right finishes its proof.

The third inequality follows from strict positivity of the summands:

$$\|A \odot B\|_{\mathrm{F}}^2 = \sum_{i,j} A_{ij}^2 B_{ij}^2 \le \Big(\sum_{i,j} A_{ij}^2\Big)\Big(\sum_{i,j} B_{ij}^2\Big) = \|A\|_{\mathrm{F}}^2 \|B\|_{\mathrm{F}}^2. \tag{2.156}$$

Each of the inequalities has now been shown. □

# Chapter 3

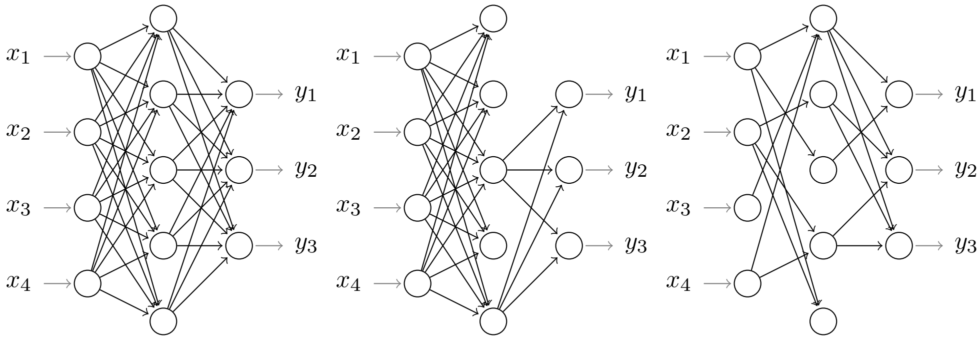
# Asymptotic convergence rate of dropout for shallow linear neural networks

Based on [7]:
*"Asymptotic convergence rate of dropout in shallow linear neural networks"*
by A. Senen–Cerda, and J. Sanders

## 3.1 Introduction

The results in Chapter 2 yield theoretical insight into the convergence guarantees of dropout and how the convergence rate depends on the depth of the Neural Network (NN). However, the explicit dependencies of the convergence rate on properties of the data or width of the NN were not investigated. In this chapter we tackle this problem and shed light on the explicit dependence of the convergence rate of *Dropout* and *Dropconnect* on properties of the data, the dropout probability $1-p$, and the structure of the NN. To do so, we investigate in detail the convergence rate of the gradient flow on an objective function induced by *Dropout* and *Dropconnect* on shallow linear NNs. To our knowledge, this problem had not received attention in the dropout literature before. The fact that there are relatively few convergence results regarding NNs with dropout algorithms to build on [29, 25], combined with the additional challenges of a stochastic algorithm, means that at least for now, linear NNs give the right balance of complexity and feasibility in order to obtain sharp rates of convergence. Our results in this chapter show that these NNs are in fact sufficiently rich for a precise description of the dependencies of the convergence rate on the data, the dropout probability, and the structure of the NN, while still presenting a technical challenge in their analysis.

In this chapter we investigate the convergence rate of dropout in shallow linear NNs. We will assume that we have $n$ input–output data points $(x_1,\dots,x_n)\in\mathbb{R}^{e\times n}$ and $(y_1,\dots,y_n)$



*(a) Full NN.* *(b)* Dropout *($p = 0.5$).* *(c)* Dropconnect *($p = 0.5$).*

*Figure 3.1.1:* (a) *The base graph of a shallow NN with $h = 4, f = 5, e = 3$ input, hidden, and output nodes, respectively.* (b) *A random subgraph being trained by* Dropout*. When applying canonical* Dropout*, one drops every hidden node of the graph with probability $1-p$ in an independent fashion.* (c) *A random subgraph being trained by* Dropconnect*. When applying* Dropconnect*, one drops edges with probability $1-p$ in an independent fashion.*

$\in \mathbb{R}^{h\times n}$. By using linearity, we can encode the information of these datapoints in a matrix $Y \in \mathbb{R}^{e\times h}$—this process is also called data whitening. We will study the gradient flow of an ordinary differential equation that approximates the behavior of training a shallow linear NNs with *Dropout* and *Dropconnect*. In a shallow NN, we have weight matrices $W = (W_2, W_1) \in \mathbb{R}^{e\times f} \times \mathbb{R}^{f\times h}$, where $f$ denotes the width of the network. With whitened data, the update direction $\Delta^{[n+1]}$ of gradient descent at time $n+1$ in shallow linear NNs with dropout satisfies [60, 29]

$$\begin{aligned}
&\mathbb{E}[\Delta^{[n+1]} \mid W^{[0]},\ldots,W^{[n]}] = \nabla\mathcal{J}(W^{[n]}), \quad \text{where} \\
&\mathcal{J}(W) = \|Y - aW_2W_1\|_{\mathrm{F}}^2 + b\mathrm{Tr}[\mathrm{Diag}(W_1W_1^T)\mathrm{Diag}(W_2W_2^T)]. \qquad (3.1)
\end{aligned}$$

Here, the Frobenius norm for a matrix $A$ is defined as $\|A\|_{\mathrm{F}}^2 = \mathrm{Tr}[A^TA]$ and the expectation is conditional on the previous iterates $W^{[0]},\ldots,W^{[n]}$. The constants $a, b$ have a closed-form expression in terms of the probability $1-p$ of dropping nodes when using *Dropout*, and another closed-form expression in terms of the probability $1-p$ of dropping edges when using *Dropconnect* (see Section 3.2.3 for their expressions).

For diminishing step sizes $\alpha^{\{n\}}$, we can view both *Dropout* and *Dropconnect* schemes as being noisy discretizations[1] of the ordinary differential equation

$$\frac{\mathrm{d}W}{\mathrm{d}t} = -\nabla\mathcal{J}(W(t)). \qquad (3.2)$$

To formally establish that the random iterates $\{W^{[n]}\}$ indeed follow the trajectories of the gradient flow in (3.2), one may employ the so-called *ordinary differential equation method*

[1] Observe that a step of Stochastic Gradient Descent (SGD) satisfies $W^{[n+1]} = W^{[n]} + \alpha^{\{n\}}(-\nabla\mathcal{J}(W^{[n]}) + M^{[n+1]})$ where $M^{[n+1]} = \mathbb{E}[\Delta^{[n+1]} \mid W^{[0]},\ldots,W^{[n]}] - \Delta^{[n+1]}$ describes a *martingale difference sequence*. This martingale difference sequence's expectation with respect to the past $W^{[0]},\ldots,W^{[n]}$ is zero.

[147, 140, 127, 29]. In this chapter, however, we take the relation for granted and the problem is about estimating the convergence rate of the gradient flow in (3.2).

For shallow linear NNs, the main result of this chapter in Theorem 10 gives a lower bound $\underline{\omega}$ to the exponent $\omega$ of the convergence rate for the gradient flow of $\mathcal{J}(W)$ when starting close to a minimizer. Let $M$ denote the closed set of global minimizers of $\mathcal{J}(W)$, and $d(x,M) = \inf_{y\in M}\{|x-y|\}$ denote the Euclidean distance between the point $x$ and the set $M$. Informally stated, we prove the following

**Proposition** (informal)**.** *Let $W \in \mathcal{P} = \mathbb{R}^{e\times f} \times \mathbb{R}^{f\times h}$. If $Y \in \mathbb{R}^{e\times h}$ has distinct positive singular values, for almost all $W \in M$, there exists neighborhood $V \subset U$ of $W$ such that for any gradient flow $\theta : [0,\infty) \to \mathcal{P}$ satisfying*

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = -\nabla\mathcal{J}(\theta_t) \quad \text{and} \quad \theta_0 \in V, \tag{3.3}$$

*there exists an explicitely computable constant $\underline{\omega}(p,f,Y,W) > 0$ such that for any $\omega \geq \underline{\omega}$ we have*

$$d(\theta_t, U\cap M) \leq \exp(-\omega t) d(\theta_0, U\cap M) \quad \text{for all} \quad t \in (0,\infty). \tag{3.4}$$

This result guarantees local convergence and shows that dropout converges in shallow linear NNs. While it may look similar to the convergence guarantee of Proposition 1 in Chapter 2, this is actually not the case. In comparison, Theorem 10 gives an implicit characterization of $\underline{\omega}$: if $W(t)$ converges to a so-called *balanced* minimizer (see Section 3.2.5 for the exact definition), the dependencies of $\underline{\omega}$ can be analytically computed. These depend namely on the probability of dropping nodes or edges $1-p$, the width $f$, and the nonzero singular values $\sigma_1,\dots,\sigma_r$ of the data matrix $Y \in \mathbb{R}^{e\times h}$, where here, $r \leq \min(e,h)$.

We give a bound for $\underline{\omega}$ that holds whenever $W(0)$ is close enough to $M$ for arbitrary input–output dimensions, and that is computable and partially explicit. For the case of a one-dimensional output ($e = 1$) a closed-form expression for $\underline{\omega}$ is obtained. Informally stated (see Proposition 6 in Section 3.3), we establish the following:

**Corollary** (informal)**.** *If the output is one-dimensional ($e = 1$), then the lower bound $\underline{\omega}$ for $\omega$ satisfies*

$$\omega \geq \underline{\omega} \approx \frac{2p^2(1-p^2)}{p^2 f + 1 - p^2}\sigma_1 \quad \text{for } \textit{Dropconnect}\text{, and} \quad \omega \geq \underline{\omega} \approx \frac{2p(1-p)}{pf + 1 - p}\sigma_1 \quad \text{for } \textit{Dropout.} \tag{3.5}$$

Some consequences from this corollary can be readily derived. For example, in an overparametrized regime ($f \gg e = 1$), the bound for the convergence rate $\underline{\omega}$ in (3.5) for $p$ fixed decays as $2\sigma_1/f$. Furthermore, for every $f$, there is a choice of $p^*$ that maximizes $\underline{\omega}$. This maximizer satisfies $p^* = 1/(1+f^{1/2})^{1/2}$ for *Dropconnect* and $p^* = 1/(1+f^{1/2})$ for *Dropout.* Lastly, it must be remarked that these results and insights also pertain to certain matrix factorization problems. Indeed, the minimization of (3.1) is in fact a matrix factorization problem with a regularization term induced by dropout. This was originally observed in [51, 60, 42].

In order to prove the results of this chapter we use a result in [21] on the local convergence of gradient flow for nonconvex objective functions. A condition in this convergence analysis is that a nondegeneracy condition of the set of minima needs to be satisfied. We combine this approach with a careful analysis of the set of minimizers of the dropout loss function, and of its Hessian. As a set, the set of minimizers $M$ has been characterized in [51, 60, 42] for *Dropout* and *Dropconnect* and we build on their result. A related but different loss landscape analysis within the context of NNs can be found in [70].

Formally, our lower bounds $\underline{\omega}$ for $\omega$ in (3.4) and (3.5) hold only close to $M$. Nonetheless, we expect that the iterates of a gradient descent counterpart should exhibit a similar decay with an exponent similar to our lower bound with enough iterations. To substantiate this claim, we show simulation results in Section 3.5 that compare numerically measured convergence rates to the rate in (3.5). We compare convergence rates across datasets with different characteristics. For example, each has different input and output dimensions. We also look at the convergence rate far away from minima, which goes beyond the scope of the local convergence statements of the results. These latter simulations yield insight in the objective function of dropout in the overparametrized regime.

The simulations show for different initializations that, indeed, the convergence rate of the gradient descent counterpart exhibits similar qualitative dependencies as the bound in (3.5). Moreover, when starting sufficiently close to a minimizer, the dependency of the numerically measured convergence rates on the width $f$ matches the decay in $f$ provided by the bound $\underline{\omega}$. This indicates that the bound in (3.5) is sharp in its dependence of $f$, i.e., $\omega(f) \approx \Theta(\underline{\omega}(f))$. Also, this observation is found to be distinct from when the iterands are far away from minima, in which case overparametrization—large width $f$—is seen to improve the convergence rate instead. This finding can be understood with the following intuition: dropout regularization makes the objective function less symmetric in the sense that as the dropout probability $1-p$ changes, the symmetry in the types of minima in the matrix factorization problem when $p=1$ is lost. Therefore, deep valleys in the optimization landscape exist in which it will take a long time to converge to the minimum as $p \uparrow 1$, while the value of the objective function at these points will already be close to its minimum. Note that this is in agreement with the convergence to $\epsilon$-stationarity of (2) in Chapter 2 where as $p \uparrow 1$, up to the term $c_1$ which depends only on the variance of the data, the other term scales as $(1-p)$.

Finally, we discuss a few simulations that extend beyond the model assumptions by considering stochastic *Dropout*, and also investigate the qualitative limitations of our results.

Our results shed light on the dependency of the convergence rate on properties of the data, the dropout probability, and the structure of the NN. These results add to the relatively scarce literature on the convergence properties of dropout algorithms and imply that the convergence rate of the stochastic *Dropout* and *Dropconnect* algorithms will intricately depend on the data, the dropout probability, and the structure of the NN. By extension, we expect that this conclusion must also hold for nonlinear shallow NNs, though quantitatively establishing such fact requires more research.

## 3.2 Preliminaries

In this chapter, we specialize the analysis to shallow NNs with linear activations, which allow us to view the optimization problem as an optimization over products of matrices.

### 3.2.1 Shallow neural networks

In shallow NNs we have three parameters that characterize their structure. Let $e, f, h \in \mathbb{N}_+$ denote the dimensions of the input, hidden, and output layer, respectively; see Figure 3.1.1 for a depiction. A shallow NN with parameters $W = (W_2, W_1) \in \mathbb{R}^{e\times f} \times \mathbb{R}^{f\times h} \triangleq \mathcal{P}$ is given by the function

$$\Psi_W(x) = W_2\vartheta(W_1x), \quad \text{where} \quad \vartheta : \mathbb{R} \to \mathbb{R} \quad \text{is applied componentwise.} \tag{3.6}$$

Here, the weights of the first and second layer are collected in the matrices $W_1$ and $W_2$, respectively. Common choices for $\vartheta$ include $\mathrm{ReLU}(t) = \max\{0,t\}$ and the sigmoid activation function $\vartheta(t) = 1/(1+\exp(-t))$.

As we have seen in Section 1.3 in Chapter 1, if we have $n$ pairs of input–output data points $\{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^h \times \mathbb{R}^e$, we aim to find weights $W$ that minimize the empirical risk, that is, weights in

$$\underset{W\in\mathcal{P}}{\operatorname{argmin}} \quad \hat{\mathcal{R}}_n(W) \triangleq \sum_{i=1}^n \|y_i - \Psi_W(x_i)\|_2^2. \tag{3.7}$$

We will assume that the $n$ data points are fixed, and we will make no reference to their underlying distribution.

In this chapter we focus on shallow linear NNs, that is, the family of functions parameterized by $\Psi_W(x) = W_2W_1x$ so $\vartheta(t) = t$. For these NNs, the optimization problem in (3.7) is already quite challenging, as the empirical risk turns out to be nonconvex and has multiple minima and saddle points. The analysis of linear NNs can be expected to also give some insight into the optimization of nonlinear NNs, and is in fact common (see [35, 3, 49] for examples).

### 3.2.2 Data whitening

Data whitening is a preprocessing step that rescales the data points such that their empirical covariance matrix equals the identity. Let $\mathcal{X} = (x_1, \ldots, x_n) \in \mathbb{R}^{h\times n}$ and $\mathcal{Y} = (y_1, \ldots, y_n) \in \mathbb{R}^{e\times n}$ be matrices containing the input and output data points, respectively. In order to be able to whiten the data, one must assume $\mathcal{X}\mathcal{X}^T \in \mathbb{R}^{h\times h}$ to be nonsingular.

Under said assumption, define now the matrix $Y = \mathcal{Y}\mathcal{X}^T(\mathcal{X}\mathcal{X}^T)^{-1/2} \in \mathbb{R}^{e\times h}$, where $(\mathcal{X}\mathcal{X}^T)^{-1/2}$ is the inverse of the unique positive definite square root of $\mathcal{X}\mathcal{X}^T$. In Appendix 3.B.1, we derive that

$$\hat{\mathcal{R}}_n(W) = \|\mathcal{Y} - W_2W_1\mathcal{X}\|_\mathrm{F}^2 = \|Y - W_2W_1(\mathcal{X}\mathcal{X}^T)^{1/2}\|_\mathrm{F}^2 + c, \tag{3.8}$$

for some constant $c \in \mathbb{R}$ independent of $W$. Consequently, after data whitening, which applies a transformation $(W_2, W_1) \to (W_2, W_1(\mathcal{X}\mathcal{X}^T)^{-1/2})$, we may focus on

$$\text{minimizing} \quad R(W) \triangleq \|Y - W_2W_1\|_\mathrm{F}^2 \quad \text{over} \quad W \in \mathcal{P}, \tag{3.9}$$

instead of minimizing the risk of (3.8). The reader can find the details in Appendix 3.B.1.[2]

[2] Note that if the input data points $(x_i)_{i=1}^n$ are sampled from an affine subspace, then we should define $Y$ using the pseudoinverse of $\mathcal{X}\mathcal{X}^\mathrm{T}$ instead.

### 3.2.3 *Dropout* and *Dropconnect* on shallow, linear NNs with whitened data

If the data samples are whitened first and kept fixed, then we can view a dropout algorithm on a shallow linear NN as a stochastic algorithm that finds a stationary point of the objective function [29]

$$\mathcal{J}(W) \triangleq \mathbb{E}[R(F\odot W)] = \mathbb{E}[\|Y-(W_2\odot F_2)(W_1\odot F_1)\|_{\mathrm{F}}^2]. \tag{3.10}$$

Here, $F\odot W$ denotes the componentwise product of each of the elements of the weight matrices $W=(W_2,W_1)$ by the elements of the two random matrices $F=(F_2,F_1)\in\{0,1\}^{e\times f}\times\{0,1\}^{f\times h}$. The expectation here is with respect to the distribution of $F$. Lemma 15 contains explicit expressions for (3.10) for the cases of *Dropout* and *Dropconnect.* The proof of Lemma 15 can be found in Appendix 3.B.2.

**Lemma 15.** *When using* Dropout*,*

$$\mathcal{J}(W) = \|Y-pW_2W_1\|_{\mathrm{F}}^2 + (p-p^2)\mathrm{Tr}[\mathrm{Diag}(W_1W_1^{\mathrm{T}})\mathrm{Diag}(W_2W_2^{\mathrm{T}})]. \tag{3.11}$$

*When using* Dropconnect*,*

$$\mathcal{J}(W) = \left\|Y-p^2W_2W_1\right\|_{\mathrm{F}}^2 + (p^2-p^4)\mathrm{Tr}[\mathrm{Diag}(W_1W_1^{\mathrm{T}})\mathrm{Diag}(W_2W_2^{\mathrm{T}})]. \tag{3.12}$$

For convenience and without loss of generality, we now choose to scale both weight matrices $W_2,W_1$ by $1/\sqrt{p}$ in the case of *Dropout*, and by a factor $1/p$ in the case of *Dropconnect.* More generally, this means that we will study the *scaled risk function*

$$\mathcal{I}(W) \triangleq \|Y-W_2W_1\|_{\mathrm{F}}^2 + \lambda\mathrm{Tr}[\mathrm{Diag}(W_2^{\mathrm{T}}W_2)\mathrm{Diag}(W_1W_1^{\mathrm{T}})], \tag{3.13}$$

with $\lambda=(1-p)/p$ in the case of *Dropout*, and $\lambda=(1-p^2)/p^2$ in the case of *Dropconnect.* The parameter $\lambda$ relates to the relative strength of the regularization term in either dropout algorithm and becomes large whenever the dropout probability $1-p$ increases.

We remark now that (3.13) has saddle points, and this prevents us from obtaining a convergence rate that holds globally *and* is sharp. We therefore conduct a local analysis, with the aim of obtaining a sharp convergence rate. Finally, note that with a few extra conditions the function in (3.13) will satisfy the strict saddle property [51, 60]. A randomly initialized gradient descent algorithm would therefore converge to a local minima with probability one [80].

### 3.2.4 Characterization of the set of global minimizers

The set of global minimizers of (3.13) have been characterized implicitly in [60, 42]. We build on one of their results, which we repeat here for convenience. Concretely, let

$$M = \{W\in\mathcal{P} : \mathcal{I}(W) = \inf_{s\in\mathcal{P}}\mathcal{I}(s)\} \tag{3.14}$$

be the set of global minimizers. Let the nonzero singular values of $Y$ be denoted by $\sigma_1\geq\cdots\geq\sigma_r$ with $r\leq\min(e,h)$; and let the compact Singular Value Decomposition (SVD) of $Y$ be $U_{\mathrm{c}}\Sigma_YV_{\mathrm{c}}$ where thus $\Sigma_Y=\mathrm{Diag}(\sigma_1,\ldots,\sigma_r)$ and $U_{\mathrm{c}}^TU_{\mathrm{c}}=V_{\mathrm{c}}V_{\mathrm{c}}^T=\mathrm{I}_r$. Define

$$\kappa_j = \frac{1}{j}\sum_{i=1}^{j}\sigma_i, \quad \rho=\max\left\{j\in[f] \,:\, \sigma_j>\frac{j\lambda\kappa_j}{f+j\lambda}\right\}, \quad \text{and} \quad \alpha=\frac{\rho\lambda\kappa_\rho}{f+j\lambda}. \tag{3.15}$$

The *shrinkage thresholding operator with threshold* $\alpha$ applied to $Y$ is defined as

$$\mathcal{S}_\alpha(Y) = U_\mathrm{c}(\Sigma_Y - \alpha \mathrm{I}_r)_+ V_\mathrm{c}, \text{where} \quad ((\Sigma_Y - \alpha \mathrm{I}_r)_+)_{ii} = \max(0, \sigma_i - \alpha). \tag{3.16}$$

By [60, Theorem 3.4, Theorem 3.6]: *if* $W^* = (W_2^*, W_1^*) \in M$ *and* $\rho < f$*, then*

$$\mathcal{W}^* = W_2^* W_1^* = \mathcal{S}_\alpha[Y] \quad \text{and} \quad \mathrm{Diag}((W_2^*)^\mathrm{T} W_2^*) \mathrm{Diag}(W_1^*(W_1^*)^\mathrm{T}) = \frac{\|\mathcal{W}^*\|_1^2}{f^2} \mathrm{I}_f. \tag{3.17}$$

If $f = \rho$, then in (3.17) the conclusion on $\mathcal{W}^*$ must be replaced by the fact that $\mathcal{W}^*$ equals the rank-$f$ approximation of $\mathcal{S}_\alpha[Y]$.[3]

### 3.2.5 Balanced and diagonally balanced minimizers

The notion of *(approximately) balanced* weights has been found to be a sufficient condition for gradient descent on the objective function of deep linear NNs to converge to their minima [35, 3]. This has also been observed experimentally in *Dropout* for shallow linear NNs [60]. We too will use the notion of balanced weights in our convergence proof.

**Definition 8.** *Weights* $(W_2, W_1) \in \mathcal{P}$ *are* balanced *if* $W_2^\mathrm{T} W_2 = W_1 W_1^\mathrm{T}$*. Weights* $(W_2, W_1) \in \mathcal{P}$ *are* diagonally balanced *if* $\mathrm{Diag}(W_2^\mathrm{T} W_2) = \mathrm{Diag}(W_1 W_1^\mathrm{T})$*. Let*

$$M_b = \left\{ (W_2, W_1) \in M : W_2^\mathrm{T} W_2 = W_1 W_1^\mathrm{T} \right\}, \tag{3.18}$$

*and*

$$M_{db} = \left\{ (W_2, W_1) \in M : \mathrm{Diag}(W_2^\mathrm{T} W_2) = \mathrm{Diag}(W_1 W_1^\mathrm{T}) \right\} \tag{3.19}$$

*be the sets of* balanced minimizers*, and* diagonally balanced minimizers*, respectively.*

We will characterize the sets $M_{db} \subset M_b$ explicitly as part of our proof. To that end, Lemma 16 contains the key observation that both sets are actually equal. Lemma 16 is proven in Appendix 3.B.3 and allows us, via $M_{db}$, to relate singular values of the critical points at a minimum through $M_b$ to explicit constraints of the loss function and its Hessian.

**Lemma 16.** *The set* $M_b = M_{db}$*, and is an invariant set for the gradient flow of* (3.13)*.*

## 3.3 Results

### 3.3.1 Assumptions

We assume a generic degree of symmetry of the set of global minimizers to be able to characterize $M_b$ explicitly. Specifically, we rely on the following assumption on the multiplicity of the positive singular values:

**Assumption 9.** *Let* $r = \mathrm{rk}(Y) \leq \min\{e, h\}$ *and let the positive singular values* $\{\sigma_i\}_{i=1}^r$ *of* $Y$ *satisfy* $\sigma_1 > \cdots > \sigma_r > 0$*.*

Assumption 9 is mild and not unusual in the literature [78]. Assumption 9 is typically satisfied when working with real data, since noise is typically breaks the possible exact symmetries of the data. For further discussion on Assumption 9, we refer to Appendix 3.A.

[3] This is perhaps not immediately clear in [60, Theorem 3.6] for the case $\rho = f \leq r$. The fact that the rank-$f$ approximation must be used instead follows from the second-to-last step in the proof of [60, Theorem 3.6].

### 3.3.2 Convergence rate of the gradient flow of the risk functions of *Dropout* and *Dropconnect*

We are now in a position to state our main result. Here, for $W \in M$ and an open set $U_W$ of $W$ which we will specify later we define the set

$$V_{R/2,\delta}(W) = \{x \in M \cap U_W : d(x, M \cap U_W) = d(x, \bar{B}_{R/2}(W) \cap M \cap U_W) < \delta\}, \tag{3.20}$$

and $\bar{B}_{R/2}(W) = \{x \in \mathcal{P} : \|x - W\| \leq R\}$.

**Theorem 10.** *Under Assumption 9, for a generic*[4] $W \in M$*, there exist a neighborhood* $U_W \subseteq \mathcal{P}$ *of* $W$*,* $\delta_0 > 0$*, and* $R_0 > 0$ *such that: for all* $\delta \in (0, \delta_0]$*,* $R \in (0, R_0]$ *and* $\theta : \mathcal{P} \times [0, \infty) \to \mathcal{P}$ *satisfying*

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = -\nabla \mathcal{I}(\theta_t) \quad \text{and} \quad \theta_0 \in V_{R/2,\delta}(W), \tag{3.21}$$

*there exists a* $\omega_U > 0$ *such that*

$$d(\theta_t, U_W \cap M) \leq \exp(-\omega_U t) d(\theta_0, U_W \cap M) \quad \text{for all} \quad t \in (0, \infty). \tag{3.22}$$

*If moreover* $W \in M_b$*, then there exists an* $\epsilon_U \geq 0$ *such that* $\omega_U \in [\omega_W - \epsilon_U, \omega_W + \epsilon_U]$*, where*

$$\omega_W = \begin{cases} \min\left\{2\frac{\lambda \kappa_\rho \rho}{f + \lambda \rho} - 2\sigma_{\rho+1}, \zeta_W\right\} & \text{if } \rho < f, \\ \min\left\{2(\sigma_\rho - \sigma_{\rho+1}), \zeta_W\right\} & \text{if } \rho = f. \end{cases} \tag{3.23}$$

*Here,* $\sigma_{\rho+1} = 0$ *if* $r = \rho$*, and* $\zeta_W > 0$ *depends implicitly on* $W$*,* $p$ *and* $\sigma_1, \ldots, \sigma_r$ *(see* (3.210) *in Appendix 3.D.6).*

Note that Theorem 10 gives an approximate lower bound for the exponent of the convergence rate of the gradient flow of $\mathcal{I}(W)$ as long as we start close enough to the set of minima. Moreover, near balanced minimizers $(W_2^*, W_1^*) \in M_b$ an implicitely computable bound is given in (3.23). Note that to obtain a lower bound of the rate for the gradient flow on $\mathcal{J}(W)$ in (3.10), we need to multiply $\omega_U$ by $p$ for *Dropout* or $p^2$ for *Dropconnect.*

The constant $\zeta_W$ is the solution to the constrained quadratic program (3.210) in Appendix 3.D.6. Because the objective function of this constrained quadratic program depends on $W$ (recall that the convergence analysis here is local), a bound for $\zeta_W$ that is simultaneously sharp and independent of $W$ cannot be given in general. However, if the output has dimension one ($e = 1$), then we can prove the following special case of Theorem 10:

**Proposition 6.** *If the output dimension is one* ($e = 1$)*, then we can replace* (3.23) *in Theorem 10 by*

$$\omega_U \in [\omega_W - \epsilon_U, \omega_W + \epsilon_U] \quad \text{where} \quad \omega_W = 2\frac{\lambda}{f + \lambda}\sigma_1. \tag{3.24}$$

[4] Generic here refers to an ‘almost everywhere’ sense, whenever $M$ has a Lebesgue measure. $M$ is an algebraic variety defined as the zero locus of a set of polynomials from (3.17). A point $W \in M$ is smooth in $M$ whenever the rank of a Jacobian is maximal. Only at the points where the rank is not maximal we do not have generic points. This occurs only in an algebraic set of strictly lower dimension than that of $M$. Formally, a generic set of the algebraic variety $M$ consists of all $W \in M$ up to a proper Zariski closed set in $M$. See [136] for reference.

While Theorem 10 already hints at dependencies on the hyperparameters, Proposition 6 provides a lower bound for $\omega$ that explicitly depends on the singular value $\sigma_1$ of the data $Y$, the probability $1-p$ of dropping nodes (or edges) encoded in $\lambda$, and the number of nodes in the hidden layer $f$. In particular, we obtain from Proposition 6 the rates

$$\omega_W^{\mathrm{DC}} = \frac{2(1-p^2)}{p^2 f + 1 - p^2}\sigma_1, \quad \omega_W^{\mathrm{DO}} = \frac{2(1-p)}{pf+1-p}\sigma_1, \tag{3.25}$$

also shown in (3.5), after multiplying by the scaling $p$ and $p^2$ for the cases of *Dropconnect*, *Dropout*, respectively.

### 3.3.3 Discussion

Theorem 10 yields a convergence rate that depends on the singular values of the data matrix $Y$, the dropout probability $1-p$, and the structure parameters $e, f, h$ of the NN. Observe that the rate $\omega_W$ in (3.23) is the minimum of two terms. The first term $\zeta_W$ depends on the point $W$ as well as $p$ and gives a local perspective on the dependence of the convergence rate on the initialization and the dropout probability (see Appendix 3.D.6 for its exact dependency)—for our purposes, the fact that it is strictly positive suffices. The second term is namely independent of $W \in M_b$ and provides a more global perspective on the convergence rate's dependency on the data matrix, the dropout probability, and the structure parameters. Notably, the dependency on $\zeta_W$ disappears in the case $e = 1$, as evidenced from Proposition 6. The simulations in Section 3.5 will additionally suggest that the term $\zeta_W$ in (3.23) does not appear to numerically dominate the convergence rate when $f > \min\{e, h\}$.

We obtain the rates in (3.5) from Proposition 6 through a multiplication using the scalings discussed above in (3.13). We observe then that *Dropout* and *Dropconnect* have an impaired convergence rate: the convergence rate in (3.5) is reduced by a factor $p$ in the case of *Dropout*, and $p^2$ in case of *Dropconnect*. This is in agreement with the results in [29]. Observe furthermore from (3.5) that as $p \uparrow 1$, i.e., a regime without dropout, $\omega \downarrow 0$. This tells us that for small dropout rates, convergence can be apparently slow for some trajectories of the gradient flow problem. This is explained by the fact that for $p \approx 1$, points $W$ satisfying $Y = W_2 W_1$ are almost minimizers of $\mathcal{J}(W)$. Finding an exact minimizer becomes then less important since there is almost no regularization.

Note also that the rates in (3.5) tell us that in the overparametrized regime $f \gg e = 1$, for every $f$ there is a dropout probability $1 - p^*$ that maximizes the rate $\omega$. Solving $\mathrm{d}\omega / \mathrm{d}p = 0$ shows that,

$$p^* = \frac{1}{\sqrt{1+\sqrt{f}}} \sim \frac{1}{f^{1/4}} \quad \text{for } \textit{Dropconnect}\text{, and} \quad p^* = \frac{1}{1+\sqrt{f}} \sim \frac{1}{f^{1/2}} \quad \text{for } \textit{Dropout}. \tag{3.26}$$

Setting $p^*$ as in (3.26) still implies that the maximizing convergence rate is $\omega^* \sim 2\sigma/f$. Hence, the maximizing dropout probability $1-p^*$ will still have limited influence on the convergence rate in this regime. To see this more explicitly, consider that for *Dropout* the best rate $\omega^* = \omega(p^*)$ compared to the rate when choosing a generic dropout probability $1 - p \in (\delta, 1-\delta)$ satisfies $\omega(p)/\omega^* \gtrsim \delta$.

Lastly, let us also consider the matrix factorization problem in which $f \ll e, h$. It follows from Theorem 10 that when conducting matrix factorization with *Dropout* regularization, degeneracies at the minimum are avoided when $\mathrm{rk}(Y) < \min\{e,h\}$, i.e., when the data is of low rank. In the objective function (3.13), with $\lambda > 0$, the set of minima $M$ around a point $W \in M_b$, only becomes degenerate when $p \uparrow 1$. Indeed, we have $2\lambda\kappa_\rho \rho/(f+\lambda\rho) \downarrow 0$ as $p \uparrow 1$ (recall that $\zeta_W > 0$ for any $p \in (0,1)$). For matrix factorization, we will usually have $f \ll \min\{e,h\}$ and $\rho = f = r$. In this case, up to the term $\zeta_W$, there is no dependence of the convergence rate in (3.23) on the dropout probability $1-p$. Dependence starts appearing when the smallest positive singular value satisfies $\sigma_f \simeq 2\lambda\kappa_\rho\rho/(f+\lambda\rho)$, which can occur as we increase the dropout probability. As we will see in the simulations in Section 3.5, the dependence of the convergence rate on $f$ also seems to display a different behavior when $f \leqslant \min\{e,h\}$.

## 3.4 Proofs

The proofs of Theorem 10 and Proposition 6 are based on two ideas. The first idea is that the trajectories of a gradient flow, when starting close to a minimizer $W^* \in M$, should depend to leading order only on the Hessian $\nabla^2\mathcal{I}(W^*)$. However, when $M$ is a connected set (or a manifold in this case), this may not be true. Additionally, we need the point $W^*$ to be *nondegenerate*, in the sense that directions tangent to the manifold $M$ are included in the kernel of the Hessian and other directions must be orthogonal to $M$ and not in the kernel. A gradient flow ending in $M$ can then be locally bounded using the eigenvalues of $\nabla^2\mathcal{I}$. As it will turn out, 'almost every' point in $M$ is nondegenerate. The second idea is that we can give an explicit lower bound for the eigenvalues of the Hessian by restricting to directions orthogonal to $M$. This requires careful computations and is the most involved part of the proof.

### 3.4.1 Overview

Here is an overview of the steps that will prove Theorem 10 and Proposition 6:

*Step 1.* We formalize the first idea by relating a lower bound for the Hessian to the convergence rate of gradient flow by using a recent result on nonconvex optimization [21]. This result holds whenever the gradient flow is started close to a minimizer in $M$, and requires the minimizer to be nondegenerate. Therefore, to prove Theorem 10, we next need to explicitly compute a lower bound for the Hessian on directions orthogonal to $M$ and verify the nondegeneracy condition.

*Steps 2, 3.* We reduce the set of minimizers $M$ to the set of balanced minimizers $M_b$ using a group action. The set of balanced minimizers is namely easier to handle: we can prove that $M_b$ is, up to a set of lower dimension than that of $M_b$, a manifold, i.e., $M_b$ is *generic*. We compute the tangent space $\mathrm{T}_W M_b$ explicitly at a generic point $W$.[5] Using the group action again, we can then also obtain the tangent space $\mathrm{T}_W M$ by extending the results from the set of balanced minimizers to the set of minimizers.

[5] By this, we mean 'for any $W \in M_b$ up to an algebraic set of lower dimension than $M$' (formally, 'up to a proper closed Zariski set in the algebraic variety $M_b$').

*Steps 4, 5.* We compute the Hessian $\nabla^2\mathcal{I}$ and calculate a lower bound when $W \in M_b \subset M$. This also implies immediately that $W$ is nondegenerate in $M$. Leveraging the group action again, we can then show that all generic points in $M$ are nondegenerate.

*Step 6.* Finally, we combine the result of *Step 1* with the bound and nondegeneracy property in *Steps 4, 5* to prove Theorem 10 and Proposition 6.

### 3.4.2 Key steps

We now prove Theorem 10 and Proposition 6 step by step as listed previously. The detailed proofs of each proposition presented here can be found in the Appendix.

**Step 1.** We use a recent result on the convergence rate of gradient descent methods for general objective functions [21], in which the local convergence in a neighborhood $U_W$ of $W \in M$ is guaranteed by the local nondegeneracy of the Hessian:

**Definition 11.** *A set $M \subset \mathcal{P}$ of minimizers of $\mathcal{I}(W)$ is* locally nondegenerate at $W$ *if there exists a neighborhood $U \subseteq \mathcal{P}$ of $W$, such that:*

*(i) $M \cap U$ is a submanifold of $\mathcal{P}$, and*

*(ii) for any $p \in M \cap U$, $\dim \mathrm{T}_p(M \cap U)) = \dim \ker \nabla^2\mathcal{I}(p)$.*

*We also say that the set $M \cap U$ is* nondegenerate *if it is locally nondegenerate at any $W \in M \cap U$.*

Our first step is to prove the following specification of the bound in [21, Proposition 3.1]. The details are relegated to Appendix 3.C.

**Proposition 7** (Adaptation of Proposition 3.1 in [21])**.** *Let $U \subseteq \mathbb{R}^d$ be an open subset and let $f : U \to \mathbb{R}$ be three times continuously differentiable. Let $M = \{w \in \mathbb{R}^d : f(w) = \inf_{\theta\in\mathbb{R}^d} f(\theta)\}$ and suppose that $U \cap M$ is a nonempty differentiable submanifold of $\mathbb{R}^d$ of dimension $\mathfrak{d} < d$. Suppose also that for all $p \in M \cap U$, $d - \mathfrak{d} = \mathrm{rk}(\nabla^2 f(p))$ holds. Then, for any $x_0 \in M \cap U$ there exists $R_0, \delta_0, \lambda \in (0,\infty)$ such that: for all $\delta \in (0,\delta_0]$, $R \in (0, R_0]$ and $\theta : (0,\infty) \to \mathbb{R}^d$ satisfying $\mathrm{d}\theta_t/\mathrm{d}t = -\nabla f(\theta_t)$ and $\theta_0 \in V_{R/2,\delta}(x_0)$, it holds that*

$$d(\theta_t, M \cap U) \le \exp(-\lambda t) d(\theta_0, M \cap U) \quad \text{for all} \quad t \in (0,\infty), \tag{3.27}$$

*where specifically*

$$\lambda = \min_{w\in\bar{V}_{R_0,\delta_0}(W)} \min_{\substack{\|v\|=1 \\ v\in\ker\nabla^2 f(w)^\perp}} \left| v^T \nabla^2 f(w) v \right|. \tag{3.28}$$

Observe now that Theorem 10 almost follows from Proposition 7 by identifying the function $f$ with the loss function $\mathcal{I}$—that is, up to the conditions of Theorem 10 and up to (3.23). Eq. (3.23) is in fact a lower bound for (3.28), and the conditions are what allow us to lower bound (3.28) in the first place.

To see where the conditions of Theorem 10 come from and how the bound in (3.23) is obtained, consider the following approach. Suppose for a moment that we are given some open subset $U$ that meets the conditions of Proposition 7 and that $M$ is nondegenerate. If these hypotheses were true, then the convergence rate in (3.28) could be bounded by providing for each $W \in M \cap U$ a lower bound for the Hessian $\nabla^2\mathcal{I}$ restricted to $T_W^\perp M$. This is because the nondegeneracy of $M$ would imply that for any $W \in M$, $\ker \nabla^2 \mathcal{I}(w) = \mathrm{T}_W M$ and therefore $\ker \nabla^2 \mathcal{I}(w)^{\perp} = \mathrm{T}_W^{\perp} M$, and (3.20) would then imply that $\overline{V}_{R,\delta}(x_0) \subseteq M \cap U$.

The two hypotheses used in the approach above have however not been proven. Instead, we will first prove that for a generic $W \in M$ there exists a neighborhood $U$ satisfying the conditions of Proposition 7 (*Steps 2, 3*), and this turns out to be sufficient. After this, we will establish that $\nabla^2 \mathcal{I}(W)|_{T_W^{\perp} M}$ is positive definite (*Step 4*), and then we lower bound its minimum eigenvalue (*Step 5*) which allows us to approximately characterize $\omega$ in Theorem 10.

**Step 2.** We start by characterizing $M$ using $M_b$ and a Lie group action on $M$. Let $H \simeq (\mathbb{R}^*)^f$ be the Lie group of invertible diagonal matrices, where $\mathbb{R}^* = \mathbb{R}\backslash\{0\}$ is the multiplicative group of invertible elements in $\mathbb{R}$. We embed $H$ in $\mathbb{R}^{f\times f}$ via the diagonal inclusion $(a_1,\dots,a_f) \to \mathrm{Diag}(a_1,\dots,a_f) \in \mathbb{R}^{f\times f}$, and define the action $\pi$ of $C \in H$ on $M$ by

$$\pi(C)(W_2,W_1) = (W_2 C, C^{-1} W_1). \tag{3.29}$$

The action $\pi$ can be used to reduce $M$ to $M_b$, as formalized in Proposition 8. We refer to Appendix 3.D.1 for its proof.

**Proposition 8.** *For every $W \in M$ there exists a unique $C_W \in H$ such that $\pi(C_W)(W) \in M_b$.*

For a subgroup G of $\mathrm{O}(f)$, we abuse notation and let $L \in \mathrm{O}(f)/\mathrm{G}$ be a representative $L \in \mathrm{O}(f)$ of the equivalence class $[L] \in \mathrm{O}(f)/\mathrm{G}$ of cosets. Via the group action in (3.29) we can now characterize the set $M_b$: see Proposition 9, which is proven in Appendix 3.D.2.

**Proposition 9.** *If Assumption 9 holds, then*

$$M_b = \Big\{(U\Sigma_2 L, L^T \Sigma_1 V) : L \in \frac{\mathrm{O}(f)}{\mathrm{I}_\rho \oplus \mathrm{O}(\mathrm{f}-\rho)}, \mathrm{Diag}\big(L^T \left(\begin{smallmatrix}\Sigma^2 & 0\\ 0 & 0\end{smallmatrix}\right) L\big) = \frac{\|\Sigma^2\|_1}{f} \mathrm{I}_\mathrm{f}\Big\} \neq \emptyset. \tag{3.30}$$

*Here, the columns of $U$ and $V$ contain the left- and right-singular vectors of $Y = U\Sigma_Y V$, respectively,*

$$\Sigma_2 = \begin{pmatrix} \Sigma & 0_{\rho\times(f-\rho)} \\ 0_{(e-\rho)\times\rho} & 0_{(e-\rho)\times(f-\rho)} \end{pmatrix}, \quad \Sigma_1 = \begin{pmatrix} \Sigma & 0_{\rho\times(h-\rho)} \\ 0_{(f-\rho)\times\rho} & 0_{(f-\rho)\times(h-\rho)} \end{pmatrix}, \tag{3.31}$$

*where $0_{n\times m}$ denotes the all zero matrix of size $n\times m$, and*

$$\Sigma^2 = \mathrm{Diag}\Big(\sigma_1 - \rho\frac{\lambda\kappa_\rho}{f+\rho\lambda}, \dots, \sigma_\rho - \rho\frac{\lambda\kappa_\rho}{f+\rho\lambda}\Big) \in \mathbb{R}^{\rho\times\rho}. \tag{3.32}$$

**Step 3.** Next, we identify $\mathrm{T}_W M_b$, the tangent space of $M_b$, whenever it is well defined for $W \in M_b$. To do so, we find a manifold $\bar{M}_b \simeq \mathrm{O(f)}/(\mathrm{I}_\rho \oplus \mathrm{O}(\mathrm{f}-\rho))$ such that $M_b \subseteq \bar{M}_b$ and a map $T : \bar{M}_b \to \mathbb{R}^f$, whose preimage defines $M_b$ and $\mathrm{T}_W M_b$ implicitly up to a set of singular points $\mathrm{Sing}(M_b)$. In particular, since for any $W \in M_b$ we have

$$\mathrm{Diag}(W_2^T W_2) = \mathrm{Diag}(W_1 W_1^T) = \frac{\|\Sigma^2\|_1}{f} \mathrm{I}_f, \tag{3.33}$$

we will define the map $T : \bar{M}_b \to \mathbb{R}^f$ by

$$T(W_2,W_1) = \mathrm{Diag}(W_2^T W_2) = \mathrm{Diag}(W_1 W_1^T). \tag{3.34}$$

This map is well defined for each equivalence class in $\mathrm{O(f)}/(\mathrm{I}_\rho \oplus \mathrm{O(f}-\rho)) \simeq \bar{M}_b$ and has at most rank $f-1$ instead of $f$, since the trace of (3.33) is fixed in $\bar{M}_b$. We can next use the *implicit function theorem* [113, Theorem 5.5] to prove in Appendix 3.D.3 that:

**Proposition 10.** *Let $W \in M_b \backslash \mathrm{Sing}(M_b)$, where*

$$\mathrm{Sing}(M_b) = \{W \in M_b : \mathrm{rk}(\mathrm{D}_W T) < f-1\}. \tag{3.35}$$

*If Assumption 9 holds, then there exists an open neighborhood $U_W \subset \mathcal{P}$ of $W \in U_W$ such that:*

*(a) $U_W \cap M_b$ is a submanifold of $\bar{M}_b$ of codimension $f-1$, and*

*(b) $\mathrm{T}_W M_b = \ker \mathrm{D}_W T$, where the differential map $\mathrm{D}_W T : \mathrm{T}_W \bar{M}_b \to \mathrm{T}_{T(W)}\mathbb{R}^f$ at $W = (U\Sigma_2 S, S^T \Sigma_1 V)$ given by*

$$\begin{aligned}\mathrm{D}_W T(V_2, V_1) &= \mathrm{D}_W T\left(U\Sigma_2 \begin{pmatrix} X & E \\ -E^T & 0 \end{pmatrix} S, S^T \begin{pmatrix} X^T & -E \\ E^T & 0 \end{pmatrix} \Sigma_1 V\right) \\ &= 2\mathrm{Diag}\left(S^T \begin{pmatrix} \Sigma^2 X & \Sigma^2 E \\ 0 & 0 \end{pmatrix} S\right).\end{aligned} \tag{3.36}$$

The set $\mathrm{Sing}(M_b)$ contains the singular points of $M_b$, which are points where the usual tangent space cannot be defined in local coordinates. Consequently, $M_b$ cannot be a manifold at these points. Finally, we prove that most points in $M_b$ are *regular*, that is, they are nonsingular. The proof is relegated to Appendix 3.D.4:

**Proposition 11.** *If Assumption 9 holds, then:*

*(a) $\mathrm{Sing}(M_b)$ is a proper closed set in $M_b$.*

*(b) $M_b$ is a manifold up to an algebraic set of lower dimension than that of $M_b$ (i.e., any generic point in $M_b$ is regular).*

*(c) $M_b$ has codimension $f-1$ in $\bar{M}_b$.*

**Step 4.** Now that we have identified $\mathrm{T}_W M_b$ in Proposition 10, we can use the fact that $M$ can be reduced to $M_b$ via the group action $\pi$ of *Step 2*. This allows us to compute the tangent space $\mathrm{T}_W M$ at a regular point $W \in M_b \backslash \mathrm{Sing}(M_b) \subset M$, and to also compute the cotangent space $\mathrm{T}_W^\perp M$. The latter task is done in Lemma 26 in Appendix 3.D.6.

Having now characterized the cotangent space $\mathrm{T}_W^\perp M$, we continue by computing a lower bound for the Hessian. We start by calculating the Hessian in Appendix 3.D.5, and identify it as follows:

**Proposition 12.** *For $W = (W_2, W_1) \in \mathcal{P}$, $(V_1, V_2) \in \mathrm{T}_W \mathcal{P}$, the Hessian $\nabla^2 \mathcal{I}(W)$ satisfies*

$$\begin{aligned}&\big(\mathrm{vec}(V_1), \mathrm{vec}(V_2)\big)^T \nabla^2 \mathcal{I}(W) \big(\mathrm{vec}(V_1), \mathrm{vec}(V_2)\big) = 2\,\|W_2 V_1 + V_2 W_1\|_F^2 \\ &+ 2\lambda \mathrm{Tr}[V_1^T \mathrm{Diag}(W_2^T W_2) V_1] + 2\lambda \mathrm{Tr}[V_2 \mathrm{Diag}(W_1 W_1^T) V_2^T] - 4\mathrm{Tr}[V_1^T V_2^T (Y - \mathcal{S}_\alpha[Y])] \\ &+ 2\lambda\big(\|\mathrm{Diag}(V_2^T W_2) + \mathrm{Diag}(W_1^T V_1)\|_\mathrm{F}^2 - \|\mathrm{Diag}(V_2^T W_2) - \mathrm{Diag}(W_1^T V_1)\|_\mathrm{F}^2\big)\end{aligned} \tag{3.37}$$

*as a bilinear form. Here, for any $A \in \mathbb{R}^{m \times n}$ we consider vectorization notation, that is,*

$$\mathrm{vec}(A) = [a_{1,1}, \ldots, a_{m,1}, \ldots, a_{1,n}, \ldots, a_{m,n}]^T \in \mathbb{R}^{mn}. \tag{3.38}$$

Finally, we lower bound the Hessian in the directions normal to the manifold of minima. The proof of the following is relegated to Appendix 3.D.6:

**Proposition 13.** *Suppose Assumption 9 holds. For any $W \in M_b \cap M \backslash \mathrm{Sing}(M) \subseteq M$, $\nabla^2 \mathcal{I}(W)$ restricted to $\mathrm{T}^{\perp}_{W} M$ is a positive definite bilinear form. Furthermore,*

$$\nabla^2 \mathcal{I}(W)|_{\mathrm{T}^{\perp}_{W} M} \geq \omega, \tag{3.39}$$

*where*

$$\omega = \begin{cases} \min\Big\{\zeta_W, 2\frac{\lambda \kappa_\rho \rho}{f+\lambda\rho} - 2\sigma_{\rho+1}\Big\} & \text{if } \rho < f \\ \min\Big\{\zeta_W, 2(\sigma_\rho - \sigma_{\rho+1})\Big\} & \text{if } \rho = f \end{cases}. \tag{3.40}$$

*Here, $\zeta_W > 0$ is a positive constant that depends on $W$, $\lambda$ and $\Sigma$. If $\rho = r$ (recall from* (3.32) *that we have $\rho \leq r$), then we set $\sigma_{\rho+1} = \sigma_{r+1} = 0$.*

*In the case that $\rho = 1$, the result holds with* (3.40) *replaced by*

$$\omega = \begin{cases} 2\sigma_1 \frac{\lambda}{f+\lambda} & \text{if } r = 1, \\ 2\sigma_1 \frac{\lambda}{f+\lambda} - 2\sigma_2 & \text{otherwise.} \end{cases} \tag{3.41}$$

**Step 5.** Proposition 13 reveals that for $W \in M_b \cap M \backslash \mathrm{Sing}(M) \subseteq M$, $M$ is locally nondegenerate at $W$—recall Definition 11. But in order to apply Proposition 7, we need to also prove that $M$ is nondegenerate in a large enough neighborhood around such regular point. By continuity, we then obtain a lower bound of the Hessian in a neighborhood of $\omega$: that is, the bound in (3.28) will hold with $\lambda \in [\omega - \epsilon, \omega + \epsilon]$ for some $\epsilon > 0$. The following is proved in Appendix 3.D.7.

**Proposition 14.** *Suppose Assumption 9 holds. If $W \in M_b \cap M \backslash \mathrm{Sing}(M)$, then there exists a neighborhood $U_W \subseteq \mathcal{P}$ of $W$ such that:*

*(a) for any $W' \in U_W \cap M$, $\ker \nabla^2 \mathcal{I}(W') = \mathrm{T}_{W'} M$;*

*(b) $U_W \cap M$ is a locally nondegenerate manifold; and*

*(c) for any $W' \in U_W \cap M$,*

$$\min_{\substack{\|v\|=1 \\ v \in \mathrm{T}^{\perp}_{W'} M}} v^T \nabla^2 \mathcal{I}(W') v = \omega_{W'} > 0. \tag{3.42}$$

Proposition 14 covers only regular points in $M_b \cap M$. We will now extend the results to $M$ in the generic sense. By using the action $\pi$ from (3.29) we can show that if a point $W \in M_b$ is regular, so is $\pi(C)(W) \in M$ for any $C \in H$. The action on $M_b$ generates $M$ and by Proposition 11 $M_b$ is regular for generic points and moreover nondegenerate by Proposition 14. We prove the following in Appendix 3.D.8.

**Proposition 15.** *If Assumption 9 holds, then the set $M$ is a nondegenerate manifold for generic points.*

**Step 6.** We are now in a position to prove Theorem 10 and Proposition 6 by applying Proposition 7. Proposition 15 yields that $M$ is a nondegenerate manifold for generic points. Together with Proposition 14, this implies that up to an algebraic set of lower dimension than the dimension of $M$, for each $W \in M$ there exists a neighborhood $U_W \subseteq \mathcal{P}$ such that for any $W' \in U_W \cap M$ there exists a constant $\omega_{W'}$ so that (3.42) holds. Hence, setting $\lambda = \min_{W' \in U_W \cap M} \omega_{W'}$, we obtain a lower bound for (3.28) and a proof of convergence close to $M$. If moreover $W \in M_b$, then the lower bound (3.40) for (3.28) in Proposition 13 proves that if $U_W$ is small enough for $W \in M_b$, the bound (3.23) in Theorem 10 holds by continuity.

In case $\rho = 1$, Proposition 13's lower bound (3.41) to (3.28) proves (3.24) in Proposition 6.

This concludes the proof. □

## 3.5 Numerical experiments

In this section we implement the gradient descent algorithm

$$W^{\{t+1\}} = W^{\{t\}} - \eta \nabla \mathcal{J}(W^{\{t\}}), \tag{3.43}$$

numerically,[6] and apply it to *Dropout*'s objective function in (3.11). We measure the convergence rate of gradient descent for different widths $f$ and dropout probabilities $1-p$ and conduct a comparison of these measurements to our bound for the convergence rate in (3.25).

Related experimental results for the convergence of *Dropout* can be found in [31]. The number of iterations required for convergence, as well as the dependence of the performance on the initialization, have been investigated experimentally both for the linear NN case [35] as well as for *Dropout* [60].

### 3.5.1 Setup

*Datasets.* We use the following datasets obtained from the UCI Machine Learning Repository:[7]

(i) the Super Conductivity (SC) dataset [55], which describes the critical temperature of superconductors, with input dimension $h = 81$ and output dimension $e = 1$;

(ii) the Modified National Institute of Standards and Technology (MNIST) digit dataset [125], which contains images of handwritten digits, with $h = 784$ and $e = 10$; and

(iii) the Canadian Institute For Advanced Research (CIFAR)-100 image dataset [128], which is a collection of images of items associated with 100 different classes. We convert the RGB images to grayscale images yielding $h = 1024$ and $e = 100$.

After first whitening and then normalizing the data, we obtain for each dataset a matrix $Y \in \mathbb{R}^{e \times h}$ that satisfies $\|Y\|_{\mathrm{F}} = 1$. This matrix is used in the risk function in (3.11). We remark that Assumption 9 holds numerically for each dataset.

*Stopping criteria.* In all experiments, we stop the gradient descent algorithm in (3.43) either when the Frobenius norm of the gradient $\|\nabla \mathcal{J}(W^{\{t\}})\|_{\mathrm{F}}$ drops below a threshold, or when it reaches a maximum number of iterations. Specifically, we let $T = \inf_t\{t : \|\nabla \mathcal{J}(W^{\{t\}})\|_{\mathrm{F}} < 10^{-5}\} \wedge T_{\max}$ be the random termination time of any one run of the

[6] The source code of our implementation is available at https://gitlab.tue.nl/20061069/asymptotic-convergence-rate-of-dropout-on-shallow-linear-neural-networks.

[7] The UCI Machine Learning Repository repository is located at https://archive.ics.uci.edu/.

gradient descent algorithm with $T_{\max} = 10^6/2$ for SC and MNIST, and $T_{\max} = 10^5$ for CIFAR.
*Initialization.* In each experiment we set the initial weights $W^{\{0\}}$ according to one of two methods. The first method we will call *Gaussian initialization*: we set every weight $W_{ijk} \sim \text{Normal}(0,\xi^2)$ in an independent manner. The second method we will call *$\epsilon$-initialization*: when $\rho = 1$, we use Lemma 22 to sample a random balanced minimizer $W^*_{ijk}$ in $M_b$ and then set every weight $W_{ijk} \sim \text{Normal}(W^*_{ijk},\epsilon^2)$ in an independent fashion; when $\rho > 1$, we use [60, Algorithm 2] with random orthogonal initialization to sample a random balanced minimizer $W^*_{ijk}$ in $M_b$.
*Step size.* In each experiment the step size is $\eta = 10^{-2}$ and fixed.

### 3.5.2 Results

Figures 3.5.1 and 3.5.2 show convergence rate fit results for different parameters $p, f$ and the different datasets for one or two different initialization methods with different values of $\xi$ and $\epsilon$. Our fitting procedure worked as follows:
*Step 1.* For various $f \in \mathcal{F} \subset \mathbb{N}_+$, $p \in \mathcal{P} \subset [0,1]$, we ran gradient descent as explained in Section 3.5.1. If the run terminated at a time $T < T_{\max}$, then we fitted the model

$$G(t;a,\beta_{f,p}) = a\mathrm{e}^{-\beta_{f,p}t} \tag{3.44}$$

to the points $\{(t,\|\nabla\mathcal{J}(W^{\{t\}})\|_{\mathrm{F}}) : t = \lfloor\gamma T\rfloor,\dots,T\}$. Here, $\gamma \in [0,1)$. In this way, we obtain an estimate $\hat{\beta}_{f,p}$ for the parameter $\beta_{f,p}$ with which the model in (3.44) best fits the measured convergence rate. Note that the estimate $\hat{\beta}_{f,p}$ is random because of our initialization. By conducting independent runs, we obtain a set of sample averages $\{\langle\hat{\beta}_{f,p}\rangle\}_{f\in\mathcal{F},p\in\mathcal{P}}$.
*Step 2.* To obtain each Figure 3.5.1(a) we fixed $f \in \mathbb{N}_+$ and fitted

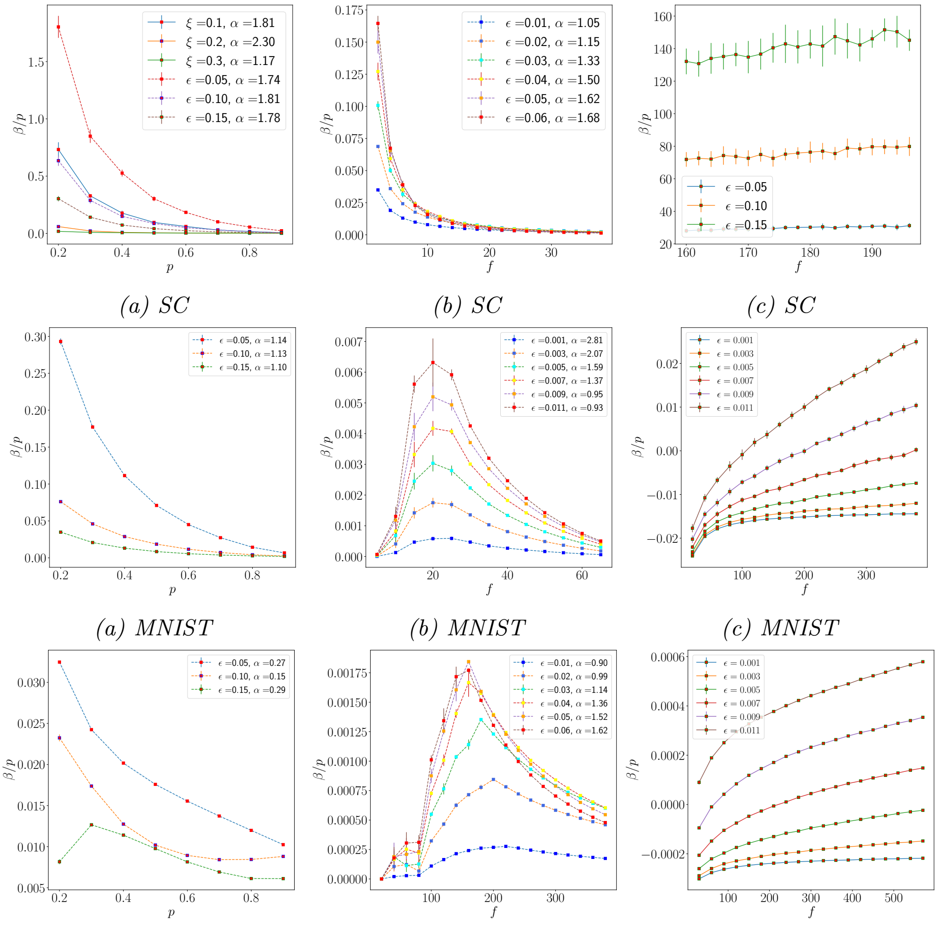
$$\beta_f(p;b,\alpha) = \frac{bp}{f(\frac{p}{1-p})^\alpha + 1} \tag{3.45}$$

to $\{(p,\langle\hat{\beta}_{f,p}\rangle)\}_{p\in\mathcal{P}}$. If a fit did not result in a positive estimate $\hat{\beta}_{f,p} > 0$, then this estimate was discarded. This eliminates runs that pass close to a saddle point. Estimates $\hat{b},\hat{\alpha}$ are obtained for the parameters $b,\alpha$ for which the model in (3.45) best describes the sample average convergence rate.
*Step 3.* To obtain each of the Figures 3.5.1(b) and 3.5.2, we fixed $p \in [0,1]$ and then fitted the model

$$\beta_p(f;b,c,\alpha) = \frac{bp(1-p)}{pf^\alpha + 1 - p} + c \tag{3.46}$$

to $\{(f,\langle\hat{\beta}_{f,p}\rangle)\}_{f\in\mathcal{F}}$. If a fit did not result in a positive estimate $\hat{\beta}_{f,p} > 0$, then this estimate was discarded. This eliminates runs that pass close to a saddle point. This similarly yields estimates $\hat{b},\hat{c},\hat{\alpha}$ for the best model parameters $b,c,\alpha$ in (3.46).
*Step 4.* To obtain each Figure 3.5.1(c), we fitted the model in (3.44) with $p = 0.7$ to the points $\{(t,\|\nabla\mathcal{J}(W^{\{t\}})\|_{\mathrm{F}}) : t = 0,\dots,10^3\}$ for different $f$. After obtaining $\hat{\beta}_{f,p}$ we then depicted $\hat{\beta}_{f,p}/p$ as a function of $f$.

Note after substituting (3.45) or (3.46) into (3.44), that both exponents have an extra factor $p$ when compared to our bound in Proposition 6. This is because we implemented the objective function $\mathcal{J}(W)$ in (3.11) as opposed to $\mathcal{I}(W)$ in (3.13). Also, if our bound in Proposition 6 turns out to be sufficiently sharp, then we can expect that $\hat{\alpha} \approx 1$ in either model.

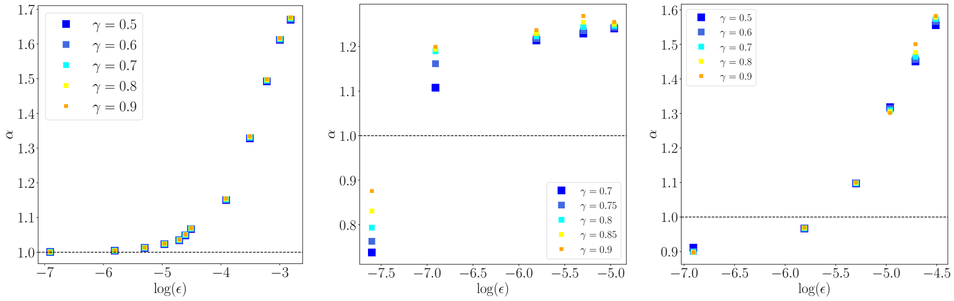


*(a) SC* *(b) SC* *(c) SC*

*(a) MNIST* *(b) MNIST* *(c) MNIST*

*(a) CIFAR* *(b) CIFAR* *(c) CIFAR*

*Figure 3.5.1:* **Column (a):** *Sample average convergence rate as a function of $p$ for fixed widths $f$ when $\gamma = 0.9$. Here, $f = 20$ for the SC dataset, $f = 40$ for the MNIST dataset and $f = 400$ for the CIFAR dataset. Our fits of* (3.45) *are also shown, and the inset shows the chosen $\xi$ or $\epsilon$ as well as the resulting fit parameters $\hat{\alpha}$. Observe that the sample averages are mostly decreasing in $p$, in agreement with our bound in* (3.25)*. It is worth noting, however, that for CIFAR, the number of iterations was $T_{\max} = 10^5$ instead of $T_{\max} = 10^6/2$. Hence, more iterations may have been required to achieve plots similar to those for SC and MNIST.* **Column (b):** *Sample average convergence rate as a function of $f$ for fixed $p = 0.7$ for the SC, MNIST, and CIFAR datasets. The fits of* (3.46) *are again shown and the inset gives the resulting fit parameters. Recall that by* (3.25)*, we may expect for sufficiently small $\epsilon$ that $\beta/p \sim 1/f$ as $f \to \infty$ and consequently $\hat{\alpha} \approx 1$. This is confirmed by the different values of $\hat{\alpha}$ shown in Figure 3.5.2. Observe for the MNIST and CIFAR datasets that when the output dimension $f \le e = 10, 100$, respectively, the convergence rate does not yet agree with a $1/f$ dependency. We expect that for small $f$ in these cases the minimum in* (3.40) *is dominated by the term $\zeta_W$.* **Column (c):** *Sample average convergence rate as a function of $f$ for fixed $p = 0.7$ and each of the datasets. Differently from columns (a,b), we fit here the* initial *iterations—iterations $0, \ldots, 10^3$. Observe that overparametrization improves the convergence rate during the initial iterations of gradient descent.*



*(a) SC* *(b) MNIST* *(c) CIFAR*

*Figure 3.5.2: All resulting fit parameters $\hat{\alpha}$ as a function of $\log \epsilon$ for fixed $p = 0.7$ and various $\gamma$ in the regime $f \geq e$ for each of the three datasets. Observe that indeed $\hat{\alpha} \simeq 1$ as $\epsilon \downarrow 0$, as predicted analytically by our bound in (3.25) for the SC dataset. This shows that the bound is sharp in this regime. The other datasets show similar behavior, though the uncertainty is relatively high for the MNIST dataset. This is due to saddle points we encountered during our simulations, whose effect becomes less pronounced as $\epsilon \downarrow 0$ and in the tail of the convergence rate ($\gamma \uparrow 1$).*

### 3.5.3 Discussion

Figures 3.5.1(a,b) and 3.5.2 show that the local bound in (3.25) characterizes the convergence rate of gradient descent close to convergence qualitatively:

– The characteristic decay of the convergence rate as $f$ and $p/(1-p)$ increase, as predicted by (3.25), is confirmed experimentally with the fits in columns (a) and (b), respectively. We see that the expected dependence of the decay of $\hat{\beta}$ on $f$ occurs whenever $f \geqslant e$. Observe that $\hat{\beta}$ and $\hat{\alpha}$ depend strongly on the initialization, which is determined here by $\xi$ and $\epsilon$. This may be explained as follows. For larger $\xi, \epsilon$, we initialize farther away from minima. Trajectories of gradient descent that follow valleys of the objective function $\mathcal{J}(W)$—regions where the loss is close to the minima and where slower convergence rates are expected—are then favored, explaining why we see a *smaller* sample average convergence rate. When $f \leqslant \min\{e, h\}$, thus in the matrix factorization regime, the convergence rate appears to behave differently as seen in column (b) with the MNIST and CIFAR datasets.

– These various decay rates that were measured were used to calculate fit parameters $\hat{\alpha}$. These tend approximately to one as $\epsilon$ becomes smaller as shown in Figure 3.5.2, in accordance with Theorem 10. Note that while the values of $\hat{\alpha}$ tend close to 1 as $\epsilon \downarrow 0$ independently of $\gamma$ with various degrees, there is a shift to larger values of $\hat{\alpha}$ as we increase $\epsilon$. This is because we are seeing an average convergence rate of the trajectories instead of the lowest convergence rate in (3.25).

In Figure 3.5.1(c) we observe that for the first iterations of gradient descent, the convergence rate improves with $f$ in all three datasets. In particular, if we initialize far away from minima, then gradient descent converges faster to what are most likely flat areas of the objective function. Even when the step size is too large and the rate is negative (we increase the loss function), the rate still increases with $f$. This is in contrast to close to convergence in column (b) where the flatness close to minima also increases with $f$ which translates to lower convergence rates.

Finally, Figure 3.5.1 indicates that independently of the regularization properties of *Dropout* and the inherent scaling of $1/p$ in the number of iterations [29], for $p \uparrow 1$ or $f \to \infty$, the landscape of *Dropout* close to the minimum becomes less rough. The objective function seems to consist of deep valleys with flat bottoms. Early on, gradient descent descends the steep valleys quickly, and once at the bottom the convergence rate becomes worse in a manner described by Theorem 1. The steepness and flatness both depend on $f$, and the numerical results in Figures (b) and (d) suggest that both become more pronounced as $f \to \infty$.

### 3.5.4 Beyond the current model

In the previous three subsections we have discussed numerical experiments that used a gradient descent algorithm to train shallow linear NNs. While not exactly the same, this is quite similar to the model behind Theorem 10 (recall that Theorem 10 is about trajectories of a gradient flow). We will therefore now briefly investigate the behavior of the convergence rate in stochastic *Dropout*, which comes closer to practice. This gives insight into the descriptive limitations of our results also.

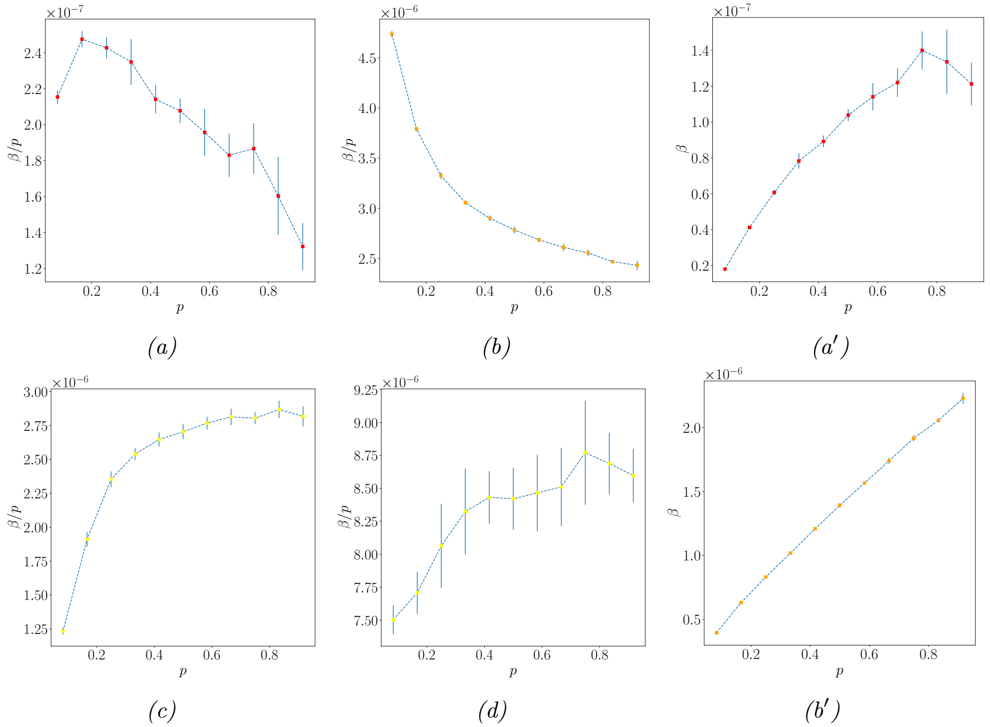
Specifically, for the SC dataset we have implemented stochastic *Dropout* for different values of $p$ on: (a) a linear, 500 unit wide, shallow NN with quadratic loss; (b) a linear, 500 unit wide, shallow NN with softmax at the output with quadratic loss; (c) a modified LeNet [145] containing two convolutional layers, two pooling layers and two dense layers, of which the first dense layer is a 500 wide linear layer and the second is an output linear layer with quadratic loss; and (d) the same modified LeNet but with softmax activation at the output and with quadratic loss. Note that setups (a)–(d) all extend beyond our modeling assumptions, and each does so to an increasingly degree.

Figure 3.5.3 shows that the scaled convergence rate $\beta/p$ decreases in setups (a) and (b), and increases in setups (c) and (d). A decrease in convergence rate is shared qualitatively by our results; an increase in convergence rate is not. By next plotting $\beta$ versus $p$ for setups (a) and (b), we can examine the convergence rate's asymptotic behavior with more precision. Observe from (a$'$) and (b$'$) that a maximizer is observed for setup (a), which is in qualitative agreement with our results; whereas for setup (b) such maximizer is not observed. The different activation function may be the cause for such different behavior.

## 3.6 Conclusion

In this chapter we have analyzed the convergence rate of the gradient flow on the objective functions induced by *Dropout* and *Dropconnect* for shallow linear NNs. Theorem 10 gives a lower bound for the convergence rate that depends implicitly on the data matrix, the probability of dropping nodes or edges, and the structure parameters of the NN. We provide in Proposition 6 a closed-form expression for our lower bound for the convergence rate in the case of a one-dimensional output. This gives insight into the dependencies of the convergence rate. The simulations show that indeed, the convergence rate of the gradient descent counterpart exhibits similar qualitative dependencies as our bound.

After the results of this chapter, one may guess if the results of this chapter could be extended to NNs with nonlinear activation functions. To do so, an analysis of the



*(a)* *(b)* *(a′)*

*(c)* *(d)* *(b′)*

*Figure 3.5.3: Exponents $\beta/p$ of the model in* (3.44) *for different NNs trained with stochastic* Dropout *depending on the remain probability $p$ (or dropout probability $1-p$) for setups (a)–(d). For setups (a) and (b), we also plot $\beta$ versus $p$ in (a′) and (b′) respectively. The error bars indicate 98% confidence intervals of the simulations' outcomes.*

minima of the objective in the nonlinear case should be first conducted. While certainly challenging, a recent paper in this direction is [15], where Rectified Linear Unit (ReLU) neural networks are considered. This extension is, however, outside the scope of this thesis.

# Appendix

# Notation for the appendix

For any vector $a=(a_1,\cdots,a_f)$, we denote by $\mathrm{Diag}(a_1,\cdots,a_f)$ the matrix in $\mathbb{R}^{f\times f}$ with the vector $a$ in the diagonal and zeroes everywhere else. For a matrix $A\in\mathbb{R}^{f\times f}$, we denote $\mathrm{Diag}(A)=\mathrm{Diag}(A_{11},\dots,A_{ff})$. For a matrix $A$ with singular values $\lambda_1,\dots,\lambda_r$ we denote the 1-norm as $\|A\|_1=\sum_{i=1}^r\lambda_i$.

## 3.A On assumption 9

We used Assumption 9 to establish Theorem 10 and Proposition 6 and for these results to be applicable in a range of scenarios.

Assumption 9 is sufficient for our proofs and reduces the complexity of the analysis without loss of the key features. The case that some of the singular values are equal can conceivably be tackled with techniques similar to those in this chapter. This would add more degeneracy to the minima and would change the subgroup representation in (3.30) by adding several additional orthogonal subgroups of lower dimensions in the stabilizer subgroup in (3.97).

Assumption 9 also allows for fairly generic data matrices $Y$. To see this, consider first that the subset of matrices of fixed rank that do not satisfy Assumption 9 has measure zero. If the data is, for example, drawn randomly from a data distribution with continuous support—even from a continuous distribution in an affine subspace—then the assumption will hold with high probability.

## 3.B Proofs of Section 3.2

### 3.B.1 Proof of (3.8) – Data whitening

For simplicity, let $\mathcal{W}=W_2W_1$. Recall also that $Y=\mathcal{Y}\mathcal{X}^{\mathrm{T}}(\mathcal{X}\mathcal{X}^{\mathrm{T}})^{-1/2}$. We now apply the identity

$$\|A-B\|_{\mathrm{F}}^2=\|A\|_{\mathrm{F}}^2-2\langle A,B\rangle_{\mathrm{F}}+\|B\|_{\mathrm{F}}^2=\mathrm{Tr}[AA^{\mathrm{T}}]-2\mathrm{Tr}[A^{\mathrm{T}}B]+\mathrm{Tr}[BB^{\mathrm{T}}] \tag{3.47}$$

twice, to obtain

$$\begin{aligned}
\hat{\mathcal{R}}_n(W)&=\|\mathcal{Y}-\mathcal{W}\mathcal{X}\|_{\mathrm{F}}^2\overset{(3.47)}{=}\mathrm{Tr}[\mathcal{Y}\mathcal{Y}^{\mathrm{T}}]-2\mathrm{Tr}[\mathcal{Y}\mathcal{X}^{\mathrm{T}}\mathcal{W}]+\mathrm{Tr}[\mathcal{W}\mathcal{X}\mathcal{X}^{\mathrm{T}}\mathcal{W}^{\mathrm{T}}]\\
&=\mathrm{Tr}[\mathcal{Y}\mathcal{Y}^{\mathrm{T}}]-2\mathrm{Tr}[\mathcal{Y}\mathcal{X}^{\mathrm{T}}(\mathcal{X}\mathcal{X}^{\mathrm{T}})^{-1/2}(\mathcal{W}(\mathcal{X}\mathcal{X}^{\mathrm{T}})^{1/2})^{\mathrm{T}}]+\mathrm{Tr}[(\mathcal{W}(\mathcal{X}\mathcal{X}^{\mathrm{T}})^{1/2})(\mathcal{W}(\mathcal{X}\mathcal{X}^{\mathrm{T}})^{1/2})^{\mathrm{T}}]\\
&=\mathrm{Tr}[YY^{\mathrm{T}}]-2\mathrm{Tr}[Y(\mathcal{W}(\mathcal{X}\mathcal{X}^{\mathrm{T}})^{1/2})^{\mathrm{T}}]\\
&\quad+\mathrm{Tr}[(\mathcal{W}(\mathcal{X}\mathcal{X}^{\mathrm{T}})^{1/2})(\mathcal{W}(\mathcal{X}\mathcal{X}^{\mathrm{T}})^{1/2})^{\mathrm{T}}]+\mathrm{Tr}[\mathcal{Y}\mathcal{Y}^{\mathrm{T}}]-\mathrm{Tr}[YY^{\mathrm{T}}]\\
&\overset{(3.47)}{=}\|Y-\mathcal{W}(\mathcal{X}\mathcal{X}^{\mathrm{T}})^{1/2}\|_{\mathrm{F}}+\mathrm{Tr}[\mathcal{Y}\mathcal{Y}^{\mathrm{T}}]-\mathrm{Tr}[YY^{\mathrm{T}}].
\end{aligned} \tag{3.48}$$

### 3.B.2 Proof of Lemma 15

*Proof of* (3.11). Note that (3.11) is a known expression for *Dropout* in the literature. In particular, see [51, Eq. (10)], [60, Eq. (10)] and [42, Lemma A.1].

*Proof of* (3.12). When using *Dropconnect*, we have for $i \in \{1,2\}$ that each matrix element $F_{ijk} \sim \mathrm{Ber}(p)$ is independent and identically distributed as indicated. We find that

$$
\begin{aligned}
\mathcal{J}(W) &= \mathbb{E}[\|Y-(W_2\odot F_2)(W_1\odot F_1)\|_{\mathrm{F}}^2] \\
&= \mathbb{E}\Big[\|Y-p^2W_2W_1+p^2W_2W_1-(W_2\odot F_2)(W_1\odot F_1)\|_{\mathrm{F}}^2\Big] \\
&= \mathbb{E}\Big[\|Y-p^2W_2W_1\|_{\mathrm{F}}^2+\|p^2W_2W_1-(W_2\odot F_2)(W_1\odot F_1)\|_{\mathrm{F}}^2 \\
&\qquad +2\sum_{ij}\big((Y-p^2W_2W_1)^{\mathrm{T}}(p^2W_2W_1-(W_2\odot F_2)(W_1\odot F_1)\big)_{ij}\Big].
\end{aligned}
\tag{3.49}
$$

Note that $\mathbb{E}[(W_2\odot F_2)(W_1\odot F_1)] = p^2W_2W_1$, so the right-most term equals zero. Furthermore, we can expand

$$
\begin{aligned}
\mathbb{E}\Big[\|p^2W_2W_1-(W_2\odot F_2)(W_1\odot F_1)\|_{\mathrm{F}}^2\Big] &= \|p^2W_2W_1\|_{\mathrm{F}}^2+\mathbb{E}[\|(W_2\odot F_2)(W_1\odot F_1)\|_{\mathrm{F}}^2] \\
&\quad -2\mathbb{E}\Big[\sum_{ij}((p^2W_2W_1)^{\mathrm{T}}(W_2\odot F_2W_1\odot F_1))_{ij}\Big].
\end{aligned}
\tag{3.50}
$$

After now (i) substituting (3.50) into (3.49) and rearranging terms, and then (ii) writing out the Frobenius norm, it follows that

$$
\begin{aligned}
&\mathcal{J}(W)-\|Y-p^2W_2W_1\|_{\mathrm{F}}^2 \overset{\text{(i)}}{=} \mathbb{E}[\|(W_2\odot F_2)(W_1\odot F_1)\|_F^2] \\
&= \mathbb{E}\Big[\mathbb{E}\Big[\|(W_2\odot F_2)(W_1\odot F_1)\|_{\mathrm{F}}^2\Big|F_1\Big]\Big] \overset{\text{(ii)}}{=} \mathbb{E}\Big[\mathbb{E}\Big[\sum_{a,b}\Big(\sum_i W_{2ai}F_{2ai}W_{1ib}F_{1ib}\Big)^2\Big|F_1\Big]\Big].
\end{aligned}
\tag{3.51}
$$

Use (iii) the fact that $(\sum a_ib_i)^2 = \sum_i a_i^2b_i^2+\sum_{i\neq j}a_ib_ia_jb_j$ now twice, to conclude that

$$
\begin{aligned}
&\mathcal{J}(W)-\|Y-p^2W_2W_1\|_{\mathrm{F}}^2 \\
&\overset{\text{(iii)}}{=} \mathbb{E}\Big[\sum_{a,b}\Big((p-p^2)\sum_i W_{2ai}^2W_{1ib}^2F_{1ib}^2+p^2\big(\sum_i W_{2ai}W_{1ib}F_{1ib}\big)^2\Big)\Big] \\
&= \sum_{a,b}\Big(p(p-p^2)\sum_i W_{2ai}^2W_{1ib}^2+p^2\Big((p-p^2)\sum_i W_{2ai}^2W_{1ib}^2+p^2\big(\sum_i W_{2ai}W_{1ib}\big)^2\Big)\Big) \\
&= (p^2-p^4)\mathrm{Tr}\big[\mathrm{Diag}(W_1W_1^{\mathrm{T}})\mathrm{Diag}(W_2W_2^{\mathrm{T}})\big]+p^4\|W_2W_1\|_{\mathrm{F}}^2.
\end{aligned}
\tag{3.52}
$$

Substituting (3.52) into (3.49) results in (3.12). This completes the proof.

### 3.B.3 Proof of Lemma 16

Let $W(t) = (W_2(t), W_1(t))$ denote a solution to (3.21). We will now prove the following facts:

(i) If $\mathrm{Diag}(W_1(0)W_1^{\mathrm{T}}(0)) = \mathrm{Diag}(W_2^{\mathrm{T}}(0)W_2(0))$, then $\mathrm{Diag}(W_1(t)W_1^{\mathrm{T}}(t)) = \mathrm{Diag}(W_2^{\mathrm{T}}(t)$ $W_2(t))$ for any $t \geq 0$.

(ii) If $W_1(0)W_1^{\mathrm{T}}(0) = W_2^{\mathrm{T}}(0)W_2(0)$, then $W_1(t)W_1^{\mathrm{T}}(t) = W_2^{\mathrm{T}}(t)W_2(t)$ for any $t \geq 0$.

(iii) If $\mathrm{Diag}(W_1(0)W_1^{\mathrm{T}}(0)) = \mathrm{Diag}(W_2^{\mathrm{T}}(0)W_2(0))$ and $W(t)$ converges as $t \to \infty$, then also $\lim_{t\to\infty} W_1(t)W_1^{\mathrm{T}}(t) = \lim_{t\to\infty} W_2^{\mathrm{T}}(t)W_2(t)$.

Note that (i), (ii) show that $M_{db}, M_b$ are invariant sets for the differential equation (3.21), respectively. In the argumentation that follows, let $\nabla_i$ denote the gradient operator in matrix form for $i \in \{1,2\}$. For example, $\nabla_2\mathcal{I}(W) \in \mathbb{R}^{e\times f}$ and $(\nabla_2\mathcal{I}(W))_{ij} = \partial\mathcal{I}(W)/\partial(W_2)_{ij}$. The negative gradients of (3.13) are computed e.g. in [60] and are given by:

$$\begin{aligned} -\nabla_1\mathcal{I}(W) &= 2W_2^{\mathrm{T}}(Y - W_2W_1) - 2\lambda\mathrm{Diag}(W_2^{\mathrm{T}}W_2)W_1, \\ -\nabla_2\mathcal{I}(W) &= 2(Y - W_2W_1)W_1^{\mathrm{T}} - 2\lambda W_2\mathrm{Diag}(W_1W_1^{\mathrm{T}}). \end{aligned} \tag{3.53}$$

*Proof of (i).* We take time derivatives of $W_1(t)W_1^{\mathrm{T}}(t)$, $W_2^{\mathrm{T}}(t)W_2(t)$ and substitute (3.21), i.e., $\mathrm{d}W_i/\mathrm{d}t = -\nabla_i\mathcal{I}(W(t))$ for $i = 1,2$. This results in

$$\frac{\mathrm{d}}{\mathrm{d}t}\big(W_1(t)W_1^{\mathrm{T}}(t)\big) = -W_1(t)\nabla_1\mathcal{I}(W(t))^{\mathrm{T}} - \nabla_1\mathcal{I}(W(t))W_1(t)^{\mathrm{T}}, \tag{3.54}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\big(W_2^{\mathrm{T}}(t)W_2(t)\big) = -\nabla_2\mathcal{I}(W(t))^{\mathrm{T}}W_2(t) - W_2(t)^{\mathrm{T}}\nabla_2\mathcal{I}(W(t)), \tag{3.55}$$

respectively. We subtract (3.55) from (3.54) and then substitute (3.53), to find that

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t}\big(W_1(t)W_1^{\mathrm{T}}(t) - W_2^{\mathrm{T}}(t)W_2(t)\big) \\ &= -2\lambda\Big(\mathrm{Diag}(W_2^{\mathrm{T}}(t)W_2(t))W_1(t)W_1^{\mathrm{T}}(t) + W_1(t)W_1^{\mathrm{T}}(t)\mathrm{Diag}(W_2^{\mathrm{T}}(t)W_2(t)) \\ &\qquad - W_2^{\mathrm{T}}(t)W_2(t)\mathrm{Diag}(W_1(t)W_1^{\mathrm{T}}(t)) - \mathrm{Diag}(W_1(t)W_1^{\mathrm{T}}(t))W_2^{\mathrm{T}}(t)W_2(t)\Big). \end{aligned} \tag{3.56}$$

Conclude in particular that

$$\frac{\mathrm{d}}{\mathrm{d}t}\big(\mathrm{Diag}(W_1(t)W_1^{\mathrm{T}}(t)) - \mathrm{Diag}(W_2^{\mathrm{T}}(t)W_2(t))\big) = 0, \tag{3.57}$$

by taking diagonals. Its solution is given by

$$\mathrm{Diag}(W_1(t)W_1^{\mathrm{T}}(t)) - \mathrm{Diag}(W_2^{\mathrm{T}}(t)W_2(t)) = \mathrm{Diag}(W_1(0)W_1^{\mathrm{T}}(0)) - \mathrm{Diag}(W_2^{\mathrm{T}}(0)W_2(0)), \tag{3.58}$$

i.e., a constant. This proves (i).

*Proof of (ii).* The implied and weaker assumption $\mathrm{Diag}(W_1(0)W_1^{\mathrm{T}}(0)) = \mathrm{Diag}(W_2^{\mathrm{T}}(0)W_2(0))$ combined with (3.58) reveals that

$$\mathrm{Diag}(W_1(t)W_1^{\mathrm{T}}(t)) = \mathrm{Diag}(W_2^{\mathrm{T}}(t)W_2(t)) = A(t) \tag{3.59}$$

say, for any $t \geq 0$. Combining $W_1(t)W_1^{\mathrm{T}}(t) - W_2^{\mathrm{T}}(t)W_2(t) = S(t)$, say, with (3.59) lets us reduce (3.56) to

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -2\lambda\big(A(t)S(t) + S(t)A(t)\big). \tag{3.60}$$

The solution of (3.60) in a neighborhood $V$ of 0 is given by

$$S(t) = \mathrm{e}^{-2\lambda\int_0^t A(s)\,\mathrm{d}s} S(0) \mathrm{e}^{-2\lambda\int_0^t A(s)\,\mathrm{d}s}. \tag{3.61}$$

Since $S(0) = 0$ by assumption, we have that $S(t) = 0$ for all $t \geq 0$. This proves (ii).
*Proof of (iii).* We distinguish several cases.
Case 1: Suppose that there exists an $l$ such that both the row $W_{1l\cdot}(t)$ and the column $W_{2\cdot l}(t)$ converge to 0. Then

$$\big(W_1(t)W_1^{\mathrm{T}}(t)\big)_{ij} = \sum_k W_{1ik}(t)W_{1jk}(t) \to 0 \quad \text{whenever} \quad i = l \quad \text{or} \quad j = l \tag{3.62}$$

and similarly

$$\big(W_2^{\mathrm{T}}(t)W_2(t)\big)_{ij} = \sum_k W_{2ki}(t)W_{2kj}(t) \to 0 \quad \text{whenever} \quad i = l \quad \text{or} \quad j = l. \tag{3.63}$$

In particular, we have that

$$\lim_{t\to\infty}\big(W_2^{\mathrm{T}}(t)W_2(t)\big)_{ij} = \lim_{t\to\infty}\big(W_1(t)W_1^{\mathrm{T}}(t)\big)_{ij} \quad \text{whenever} \quad i = l \quad \text{or} \quad j = l. \tag{3.64}$$

Case 2: Consider now any $l$ for which either the row $W_{1l\cdot}(t)$ or the column $W_{2\cdot l}(t)$ does not converge to zero. In particular, there must then exist a sufficiently large $t_l \geq 0$ and $\epsilon_l > 0$ such that

$$A_{ll}(t) = \sum_k W_{1lk}^2(t) \overset{\text{(i)}}{=} \sum_k W_{2kl}^2(t) \geq \epsilon_l \tag{3.65}$$

for all $t \geq t_l$. We therefore also have by (3.61) that

$$\begin{aligned} S_{lj}(t) &= \mathrm{e}^{-2\lambda\int_{t_l}^t A_{ll}(s)\,\mathrm{d}s} S_{l,j}(t_l) \mathrm{e}^{-2\lambda\int_{t_l}^t A_{jj}(s)\,\mathrm{d}s} \\ &\leq |S_{lj}(t_l)|\mathrm{e}^{-2\lambda\varepsilon_l(t-t_l)} \to 0 \quad \text{for} \quad j = 1,\ldots,f. \end{aligned} \tag{3.66}$$

Hence, we obtain $\lim_{t\to\infty} S_{lj}(t) = 0$ for any $j$ and so (iii) is proven.
*Proof that $M_b = M_{db}$.* Fact (i) implies that $M_{db}$ is an invariant set for (3.21). Fact (iii) tells us that if $W(0) \in M_{db}$ and $W(t)$ converges, then $\lim_{t\to\infty} W(t) \in M_b$. Combining facts (i) and (iii), it must be that $M_{db} \subseteq M_b$.

The inclusion $M_b \subseteq M_{db}$ follows immediately from Definition 8. This concludes the proof.

## 3.C Proof of Proposition 7

Proposition 7 is a specification of [21, Proposition 3.1]. To arrive at Proposition 7, all we need to do is prove that [21, Proposition 3.1] holds with the implicit convergence rate there ($\lambda$) replaced by the convergence rate

$$\min_{w\in\bar{V}_{R_0,\delta_0}(x_0)} \min_{\substack{\|v\|=1 \\ v\in\ker\nabla^2 f(w)^{\perp}}} \big|v^{\mathrm{T}}\nabla^2 f(w)v\big|, \tag{3.67}$$

where $V_{R,\delta}(x_0)$ is defined in (3.20). The convergence rate appears implicitly in the proof of [21, Proposition 3.1] after the application of [21, Lemma 2.9] at [21, (3.11)]. Hence, we need to make appropriate modifications to these steps in the proof of [21, Lemma 2.9].

*Modifications to the proof of [21, Lemma 2.9].* Let $x_0 \in M \cap U$. Since $M \cap U$ is a nonempty $\mathfrak{d}$-dimensional submanifold of $\mathbb{R}^d$, we have by [21, Proposition 2.1] that there exists a neighborhood $V_*(x_0)$ of $x_0$ such that:

(a) For every $x \in V_*(x_0)$, there exists a unique projection $x_* \in M \cap U$ say such that $\|x - x_*\| = d(x, M \cap U)$.

(b) This projection map $x \to x_*$ is locally $C^1$-smooth.

Fix $R_0, \delta_0 > 0$ such that for any $\delta \in (0, \delta_0], R \in (0, R_0]$, it holds that $\bar{V}_{R,\delta}(x_0) \subset V_*(x_0)$. There exists an $r \in (0, \infty)$ such that

$$\max_{y \in \bar{B}_{R_0}(x_0) \cap M \cap U} \left\|\nabla^2 f(y_*)\right\| \leq \frac{1}{r}. \tag{3.68}$$

Our modified proof will be complete when we find a $\lambda$ that satisfies the following conditions:

(i) $0 < \lambda \leq \max_{y \in \bar{B}_{R_0}(x_0) \cap M \cap U} \|\nabla^2 f(y_*)\|$;

(ii) for any $x \in V_{R,\delta}(x_0)$, $\|(x - x_*) - r\nabla^2 f(x_*) \cdot (x - x_*)\| \leq (1 - r\lambda)\|x - x_*\|$; and

(iii) for any $x \in V_{R,\delta}(x_0)$, $\big(\nabla^2 f(x_*) \cdot (x - x_*)\big) \cdot (x - x_*) \geq \lambda \|x - x_*\|^2$.

Condition (iii) is essentially our addendum to the proof of [21, Lemma 2.9].

Note that by assumption, $M \cap U$ is a nondegenerate submanifold of $\mathcal{P}$ (see Definition 11), so there is an embedding $M \cap U \to \mathcal{P}$ inducing an orthogonal decomposition

$$\mathrm{T}_{w_*}\mathbb{R}^d = \mathrm{T}_{w_*}(M \cap U) \oplus N_{w_*} = P_{w_*} \oplus N_{w_*} \tag{3.69}$$

for which $\nabla^2 f(w_*)|_{P_{w_*}} > 0$ and $\nabla^2 f(w_*)|_{N_{w_*}} = 0$ for any $w_*$. It holds moreover that for any $w' \in \bar{B}_{R_0}(x_0) \cap M \cap U$ that $\dim(\ker \nabla^2 f(w')) = s$.

Taking inspiration from the decomposition in (3.69), we will now prove that the candidate

$$\tilde{\lambda} = \min_{w \in \bar{V}_{R_0,\delta_0}(x_0)} \min_{\substack{\|v\|=1 \\ v \in \ker \nabla^2 f(w_*)^{\perp} = P_{w_*}}} \left|v^{\mathrm{T}} \nabla^2 f(w_*) v\right| \tag{3.70}$$

satisfies Conditions (i)–(iii).

Condition (i): The orthogonal decomposition in (3.69) together with the compactness of $\bar{V}_{R_0,\delta_0}$ guarantees the strict positivity of (3.70). That is, $\tilde{\lambda} > 0$.

For any $w \in \bar{V}_{R_0,\delta_0}(x_0)$, it holds that $w_* \in \bar{B}_R(x_0) \cap M \cap U$. This implies that

$$\tilde{\lambda} \leq \max_{w \in \bar{V}_{R_0,\delta_0}(x_0)} \|\nabla^2 f(w_*)\| \leq \max_{w \in \bar{B}_{R_0}(x_0) \cap M \cap U} \|\nabla^2 f(w_*)\|. \tag{3.71}$$

Condition (ii): Let $x \in V_{R,\delta}(x_0)$. Since $x - x_* \in P_{x_*}$, it follows that

$$\|(x - x_*) - r\nabla^2 f(x_*) \cdot (x - x_*)\|^2 = \|(1 - r\nabla^2 f(x_*)) \cdot (x - x_*)\|^2. \tag{3.72}$$

Recall now that we have the positive bilinear form $\nabla^2 f(x_*))|_{P_{x_*}}$ on $P_{x_*}$. Let the minimal eigenvalue of $\nabla^2 f(x_*))|_{P_{x_*}}$ be $\lambda_{\min}(\nabla^2 f(x_*)|_{P_{x_*}}) > 0$. By (3.68),

$$0 < (1 - r\nabla^2 f(x_*))|_{P_{x_*}} \leq 1 - r\lambda_{\min}(\nabla^2 f(x_*)|_{P_{x_*}}) \tag{3.73}$$

as a positive bilinear form, so that

$$\|(1-r\nabla^2 f(x_*))\cdot(x-x_*)\|^2 \le \big(1-r\lambda_{\min}(\nabla^2 f(x_*)|_{P_{x_*}})\big)\|x-x_*\|^2. \tag{3.74}$$

We have by nondegeneracy that $P_{x_*} = \ker\nabla^2 f(x_*)^\perp$. Therefore $\lambda_{\min}(\nabla^2 f(x_*)|_{P_{x_*}}) \ge \tilde{\lambda}$ for any $x \in V_{R,\delta}(x_0)$.

Condition (iii). Let $x \in V_{R,\delta}(x_0)$. Similar to (ii), from $\lambda_{\min}(\nabla^2 f(x_*)|_{P_{x_*}}) \ge \tilde{\lambda}$ we conclude also

$$\big(\nabla^2 f(x_*)\cdot(x-x_*)\big)\cdot(x-x_*) \ge \tilde{\lambda}\|x-x_*\|^2. \tag{3.75}$$

This completes the proof.

# 3.D Proofs of Section 3.4

## 3.D.1 Proof of Proposition 8 – Reduction from $M$ to $M_b$

Let $W=(W_2,W_1)\in M$ and let $\pi$ be the action from (3.29). Note that $\pi(C)(W_2,W_1)\in M$, since $\pi$ preserves the conditions in (3.17) for $W$ to be a minimum. Hence, $\pi$ is well defined. Note now also that the same conditions imply that for $i=1,\dots,f$,

$$\big(\mathrm{Diag}(W_2^{\mathrm{T}}W_2^{\mathrm{T}})\big)_{ii} > 0, \quad \big(\mathrm{Diag}(W_1W_1^{\mathrm{T}})\big)_{ii} > 0. \tag{3.76}$$

This enables us to define

$$C_W = \mathrm{Diag}(W_1W_1^{\mathrm{T}})^{1/4}\mathrm{Diag}(W_2W_2^{\mathrm{T}})^{-1/4} \tag{3.77}$$

and then consider the point $\pi(C_W)(W)=(\tilde{W}_2,\tilde{W}_1)$ say. For this particular point,

$$\begin{aligned}\mathrm{Diag}(\tilde{W}_2^{\mathrm{T}}\tilde{W}_2) &\overset{(3.29)}{=} C_W^{\mathrm{T}}\mathrm{Diag}(W_2^{\mathrm{T}}W_2)C_W \\ &\overset{(3.77)}{=} \mathrm{Diag}(W_2^{\mathrm{T}}W_2)^{1/2}\mathrm{Diag}(W_1W_1^{\mathrm{T}})^{1/2} \overset{(3.17)}{=} \frac{\|\mathcal{W}^*\|_1}{f}\mathrm{I}_f &(3.78)\\ &= \mathrm{Diag}(\tilde{W}_1\tilde{W}_1^{\mathrm{T}}). &(3.79)\end{aligned}$$

Here, (3.79) follows using the same (but appropriately modified) argumentation as for (3.78). Consequently, $\pi(C_W)(W)\in M_{db}$. Recalling that $M_b = M_{db}$ by Lemma 16 concludes the proof. □

## 3.D.2 Proof of Proposition 9 – Characterization of $M_b$.

Recall $M_b$, $M_{db}$'s definitions in (3.18), (3.19), respectively. We now introduce the following two extended sets:

$$\bar{M}_b = \{W=(W_2,W_1)\in\mathcal{P} : W_2^{\mathrm{T}}W_2 = W_1W_1^{\mathrm{T}}, W_2W_1 = \mathcal{S}_\alpha[Y]\}, \quad \text{and} \tag{3.80}$$

$$\bar{M}_{db} = \{W=(W_2,W_1)\in\mathcal{P} : \mathrm{Diag}(W_2^{\mathrm{T}}W_2) = \mathrm{Diag}(W_1W_1^{\mathrm{T}}), W_2W_1 = \mathcal{S}_\alpha[Y]\}. \tag{3.81}$$

The sets $\bar{M}_{db}, \bar{M}_b$ also contain diagonally balanced and balanced points respectively, but these points are not necessarily minima. They are extensions because

$$M_{db} = \bar{M}_{db}\cap M, \quad \text{and} \quad M_b = \bar{M}_b \cap M. \tag{3.82}$$

Recall the definitions of $\rho$ in (3.15), $\Sigma_2, \Sigma_1$ in (3.31), and $\Sigma$ in (3.32).

**Lemma 17.** *If Assumption 9 holds, then there exist a full SVD of $W = (W_2, W_1) \in \bar{M}_b$ of the form $(U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V)$ where $S \in \mathrm{O}(f)$.*

*Proof.* Let $W = (W_2, W_1) \in \bar{M}_b$. Consider a compact SVD of the following form $W = (U_2\tilde{\Sigma}_2 S_2, S_1^{\mathrm{T}}\tilde{\Sigma}_1 V_1)$. Note that for this compact SVD in particular

$$S_2 S_2^{\mathrm{T}} = \mathrm{Id}_{\dim\tilde{\Sigma}_2} \quad \text{and} \quad S_1 S_1^{\mathrm{T}} = \mathrm{Id}_{\dim\tilde{\Sigma}_1}. \tag{3.83}$$

We also suppose (without loss of generality) that the singular values of $\tilde{\Sigma}_2$ and $\tilde{\Sigma}_1$ are both ordered in the diagonal from largest to smallest.

Observe that

$$S_2^{\mathrm{T}}\tilde{\Sigma}_2^2 S_2 = W_2^{\mathrm{T}} W_2 \overset{(3.80)}{=} W_1 W_1^{\mathrm{T}} = S_1^{\mathrm{T}}\tilde{\Sigma}_1^2 S_1. \tag{3.84}$$

Uniqueness of the singular values combined with (3.84) implies that there exists a permutation matrix $P$ such that $\tilde{\Sigma}_2 = P\tilde{\Sigma}_1$. Moreover, because the singular values of $\tilde{\Sigma}_2$ $\tilde{\Sigma}_1$ are ordered by construction, we must have that (i) $\tilde{\Sigma}_2 = \tilde{\Sigma}_1 = \tilde{\Sigma}$ say. From (3.84), it follows in particular that

$$S_2^{\mathrm{T}}\tilde{\Sigma}^2 S_2 = S_1^{\mathrm{T}}\tilde{\Sigma}^2 S_1. \tag{3.85}$$

Suppose now that $\tilde{\Sigma} \in \mathbb{R}^{l\times l}$, that the singular values are given by $\lambda_1, \ldots, \lambda_s$ (each distinct), and that their multiplicities are given by $r_1, \ldots, r_s$. Recall that $\sum_{i=1}^{s} r_i = l$ necessarily. After left-, right-multiplying (3.85) by $S_2$, $S_1^{\mathrm{T}}$, respectively, it follows that the matrix $L = S_2 S_1^{\mathrm{T}}$ commutes with $\tilde{\Sigma}^2$: $\tilde{\Sigma}^2 L = L\tilde{\Sigma}^2$. Combining this fact with the fact that

$$\tilde{\Sigma}^2 = \begin{pmatrix} \lambda_1^2 \mathrm{I}_{r_1\times r_1} & & & \\ & \lambda_2^2 \mathrm{I}_{r_2\times r_2} & & \\ & & \ddots & \\ & & & \lambda_s^2 \mathrm{I}_{r_s\times r_s} \end{pmatrix}, \tag{3.86}$$

in which all off-diagonal elements are equal to zero, leads to the conclusion that the matrix $L$ must be a conformally partitioned block-diagonal matrix of the form

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_s \end{pmatrix}. \tag{3.87}$$

Furthermore, $L$ must have strictly positive entries and $L_1 \in O(r_1), \ldots, L_s \in O(r_s)$ because of the uniqueness of the eigenspaces for each eigenvalue and therefore $L \in O(l)$. Consequently, $L$ also commutes with $\tilde{\Sigma}$:

$$\tilde{\Sigma} L = L\tilde{\Sigma}. \tag{3.88}$$

Let $U_c\Sigma^2 V_c$ now be a compact SVD of $\mathcal{S}_\alpha[Y]$. Recall (3.15) and (3.16), and conclude that $\Sigma^2$ is given by (3.32). Observe that

$$U_c\Sigma^2 V_c = \mathcal{S}_\alpha[Y] \overset{(3.80)}{=} W_2 W_1 \overset{(\mathrm{SVD})}{=} U_2\tilde{\Sigma}_2 S_2 S_1^{\mathrm{T}}\tilde{\Sigma}_1 V_1 \overset{(\mathrm{i})}{=} U_2\tilde{\Sigma} L\tilde{\Sigma} V_1 \overset{(3.88)}{=} U_2 L\tilde{\Sigma}^2 V_1. \tag{3.89}$$

Remark now that $(U_2 L)^{\mathrm{T}}(U_2 L) = \mathrm{Id}_l$. Consequently, the left-hand side as well as the right-hand side of (3.89) are compact SVDs. By uniqueness of the singular values we must again have that there exists a permutation matrix $P'$ such that $\tilde{\Sigma}^2 = P'\Sigma^2$. By construction, the singular values of both diagonal matrices $\tilde{\Sigma}^2$ and $\Sigma^2$ were put in the same order. This implies that we must have $\tilde{\Sigma}^2 = \Sigma^2$. By positivity of the entries, we must consequently also have (ii) $\tilde{\Sigma} = \Sigma$. The singular values in $\Sigma^2$ have no multiplicity by Assumption 9, so equating multiplicities yields $r_1 = \ldots = r_s = 1$, $l = \rho$. Moreover, $L \in O(r)$ is a diagonal matrix with $\{-1,+1\}$-valued entries.

The uniqueness of the left and right eigenvectors in the left-hand side as well as the right-hand side of (3.89) together with the fact that all eigenvalues of $\Sigma$ have multiplicity one, implies that there exists a diagonal matrix $D$ with entries in $\{-1,+1\}$ such that $U_{\mathrm{c}}D = U_2L$ and (iii) $DV_{\mathrm{c}} = V_1$. In particular, (iv) $U_2 = U_{\mathrm{c}}DL^{\mathrm{T}} = U_{\mathrm{c}}DL$. Also, from the facts that $L$ is a diagonal matrix with $\{-1,+1\}$-valued entries and both $S_1$, $S_2$ have orthonormal rows, we obtain from $L = S_2S_1^{\mathrm{T}}$ that (v) $LS_1 = S_2$. Utilizing (i–v), together with (vi) the fact that $D, L, \Sigma$ are diagonal matrices which are thus symmetric and commute, we can rewrite the compact SVD of $W$ as

$$\begin{aligned}\big(U_2\tilde{\Sigma}_2S_2, S_1^{\mathrm{T}}\tilde{\Sigma}_1V_1\big) &\overset{\text{(i,ii)}}{=} \big(U_2\Sigma S_2, S_1^{\mathrm{T}}\Sigma V_1\big) \overset{\text{(iii,iv)}}{=} \big(U_{\mathrm{c}}DL\Sigma S_2, S_1^{\mathrm{T}}\Sigma DV_{\mathrm{c}}\big) \\ &\overset{\text{(v)}}{=} \big(U_{\mathrm{c}}DL\Sigma S_2, S_2^{\mathrm{T}}L\Sigma DV_{\mathrm{c}}\big) \overset{\text{(vi)}}{=} \big(U_{\mathrm{c}}\Sigma(DLS_2), (S_2LD)^{\mathrm{T}}\Sigma V_{\mathrm{c}}\big). \quad (3.90)\end{aligned}$$

We can extend the compact SVD in (3.90) to a full SVD by noting that $S$ will be the extension of $DLS_2$ to an orthogonal matrix in $\mathrm{O}(f)$, and $U$ and $V$ will be the extensions of $U_{\mathrm{c}}$ and $V_{\mathrm{c}}$ to $\mathrm{O}(e)$ and $\mathrm{O}(h)$, respectively. Similarly, $\Sigma_2$ and $\Sigma_1$ will be the extension to a full SVD. □

We next characterize $\bar{M}_b$ from (3.80) as a homogeneous manifold. Let us summarize the method first. Suppose that $G$ is a finite-dimensional Lie group, that is, a group with a smooth manifold structure (for example, $\mathrm{GL}(n)$ or $\mathrm{SL}(n)$). Suppose for a moment that $\bar{M}_b$ is a set, and that there is a *transitive* Lie group action $\pi : G \times \bar{M}_b \to \bar{M}_b$. A transitive action means that for any $a, b \in \bar{M}_b$, there exists a $g \in G$ such that $\pi(g)(a) = b$. We define the *stabilizer subgroup* (also called *isotropy subgroup*) of $\pi$ at $a \in \bar{M}_b$ as $\mathrm{Stab}_G(a) = \{g \in G : \pi(g)(a) = a\}$. We will use that if for a point $a \in \bar{M}_b$, $\mathrm{Stab}_G(a) \subseteq G$ is a closed smooth Lie subgroup (closed in the topology of $G$), then there exists a smooth manifold structure on $\bar{M}_b$ which is that of the homogeneous manifold $G/\mathrm{Stab}_G(a)$ [113, Thm. 21.20]. Once we have a *diffeomorphism* $\bar{M}_b \simeq G/\mathrm{Stab}_G(a)$ (a differentiable isomorphism with differentiable inverse) we can consider the projection map $\Pi : G \to G/\mathrm{Stab}_G(a) \simeq \bar{M}_b$ and look at the differential $\mathrm{D}\Pi : \mathfrak{g} \to \mathrm{T}_0(G/\mathrm{Stab}_G(a))$ at $\Pi(Id) = [\mathrm{Stab}_G(a)] = 0$, where $\mathfrak{g}$ is the Lie algebra of $G$. The linear map $\mathrm{D}\Pi$ is surjective and the kernel is the Lie algebra of $\mathrm{Stab}_G(a)$, denoted by $\mathrm{Lie}(\mathrm{Stab}_G(a))$. Hence as vector spaces

$$\frac{\mathfrak{g}}{\mathrm{Lie}(\mathrm{Stab}_G(a))} \simeq \mathrm{T}_a\bar{M}_b. \quad (3.91)$$

We refer the reader to [137, Ch. 4] for more details on homogeneous spaces.

**Lemma 18.** *If Assumption 9 holds, then there is a diffeomorphism of manifolds* $\bar{M}_b \simeq \mathrm{O(f)}/(\mathrm{I}_\rho \oplus \mathrm{O(f}-\rho))$, *i.e., the manifold* $\bar{M}_b$ *is a homogeneous space.*

*Proof.* Consider the smooth Lie group action $\pi : \mathrm{O}(f) \times \bar{M}_b \to \bar{M}_b$ given by

$$\pi(L)(W_2, W_1) = (W_2L, L^{\mathrm{T}}W_1). \quad (3.92)$$

For $W = (W_2, W_1) \in \bar{M}_b$, we are first going to determine the stabilizer subgroup

$$\mathrm{Stab}_{\mathrm{O}(f)}(W) = \{S' \in \mathrm{O}(f) : \pi(S')(W) = W\}. \tag{3.93}$$

Let to that end $(U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V)$ be an SVD of $W$, which exists by Lemma 17. Note then that for any orthogonal matrix $S' \in \mathrm{O}(f)$ of the form

$$S' = S^{\mathrm{T}} \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right) S \quad \text{where} \quad A \in \mathbb{R}^{\rho\times\rho}, \tag{3.94}$$

we have that

$$\begin{aligned}\pi(S')(W) &\overset{(3.92)}{=} \big(W_2 S', (S')^{\mathrm{T}} W_1\big) \overset{(\mathrm{SVD})}{=} \big(U\Sigma_2 S S', (S')^{\mathrm{T}} S^{\mathrm{T}} \Sigma_1 V\big) \\ &\overset{(3.94)}{=} \big(U\Sigma_2 \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right) S, S^{\mathrm{T}} \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)^{\mathrm{T}} \Sigma_1 V\big) \overset{(3.31)}{=} \big(U \left(\begin{smallmatrix} \Sigma A & \Sigma B \\ 0 & 0 \end{smallmatrix}\right) S, S^{\mathrm{T}} \left(\begin{smallmatrix} A^{\mathrm{T}}\Sigma & 0 \\ B^{\mathrm{T}}\Sigma & 0 \end{smallmatrix}\right) V\big) = W\end{aligned} \tag{3.95}$$

if and only if

$$\left(\begin{smallmatrix} \Sigma A & \Sigma B \\ 0 & 0 \end{smallmatrix}\right) = \Sigma_2 \quad \text{and} \quad \left(\begin{smallmatrix} A^{\mathrm{T}}\Sigma & 0 \\ B^{\mathrm{T}}\Sigma & 0 \end{smallmatrix}\right) = \Sigma_1. \tag{3.96}$$

Because of our Assumption 9 on the multiplicity of the eigenvalues, (3.96) holds if and only if $B = 0$ and $A = \mathrm{Id}_\rho$. We must then furthermore have that $C = 0$ and $D \in \mathrm{O}(f - \rho)$ because $S' \in \mathrm{O}(f)$. We have shown that

$$\mathrm{Stab}_{\mathrm{O}(f)}(W) \simeq S^{\mathrm{T}}(\mathrm{I}_\rho \oplus \mathrm{O}(\mathrm{f} - \rho))S, \tag{3.97}$$

the right-hand side of which is a closed, smooth Lie subgroup of $\mathrm{O}(f)$.

Next, we prove the diffeomorphism. Lemma 17 ensures that $\pi$ is transitive. Transitiveness ensures that the choice of $L$ in (3.92) only changes the stabilizer subgroup by conjugation, i.e.,

$$\mathrm{Stab}_{\mathrm{O}(f)}(\pi(L)W) = L^{-1}\mathrm{Stab}_{\mathrm{O}(f)}(W)L. \tag{3.98}$$

The set $\bar{M}_b$ admits therefore a smooth manifold structure, and we have the following diffeomorphism of smooth manifolds [113, Thm. 21.20]:

$$\bar{M}_b \simeq \frac{\mathrm{O}(f)}{\mathrm{I}_\rho \oplus \mathrm{O}(\mathrm{f} - \rho)}. \tag{3.99}$$

This completes the proof. □

Lemma 17 guarantees that each point $W \in \bar{M}_b$ has an SVD of the form $(U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V)$ where $[S] \in \mathrm{O}(\mathrm{f})/(\mathrm{I}_\rho \oplus \mathrm{O}(\mathrm{f} - \rho))$. Here, $S$ denotes any representative of the equivalence class $[S]$. Conclude using (3.17), (3.18) and (3.80) that if $W \in \bar{M}_b$, then $W \in M_b$ also if and only if moreover $\mathrm{Diag}(W_2^{\mathrm{T}} W_2) = \mathrm{Diag}(W_1 W_1^{\mathrm{T}}) = \|\Sigma^2\|_1 \mathrm{I}_f / f$. Combined with the isomorphism in Lemma 18, this provides us with the alternative representation in (3.30).

All that remains is to prove that $M_b \neq \emptyset$. This fact was also proven in [60], but for completeness we will prove it here using the *theory of majorization* instead. For any vector $a \in \mathbb{R}^f$, denote by $a^\downarrow \in \mathbb{R}^f$ the vector with the same components but sorted in descending order. Given two vectors $a, b \in \mathbb{R}^f$, we say that $a$ is *majorized* by $b$, written as $a \prec b$, if

$$\sum_{i=1}^{l} a_i^\downarrow \leq \sum_{i=1}^{l} b_i^\downarrow \quad \text{for} \quad l = 1, \ldots, f \quad \text{and furthermore} \quad \sum_{i=1}^{f} a_i = \sum_{i=1}^{f} b_i. \tag{3.100}$$

**Lemma 19.** *It holds that $M_b \neq \emptyset$.*

*Proof.* We temporarily abuse our notation and let Diag denote the map that: (a) maps vectors $y \in \mathbb{R}^f$ to an $\mathbb{R}$-valued diagonal $f \times f$ matrices with $y_1,\dots,y_f$ for its diagonal entries, and (b) maps matrices $A \in \mathbb{R}^{f\times f}$ to $\mathbb{R}$-valued vectors with entries $A_{11},\dots,A_{ff}$.

If $a \in [0,\infty)^f$ and $L \in \mathrm{O}(f)$, then as a linear map $\mathrm{Diag}(L\mathrm{Diag}(y)L^{\mathrm{T}}) = Py$ for some *orthostochastic* matrix $P$; specifically, the doubly stochastic matrix that is formed by taking the square of the entries of $L \in \mathrm{O}(f)$ [163, Definition B.5, p.34]. Because $P$ is doubly stochastic, we have that $Pa \prec a$. Note now that *Horn's theorem* states that the converse is also true [163, Theorem B.6, p.35]: if $a \prec b$, then there exists a orthostochastic matrix $Q$ such that $Qb = a$. In particular, there exists some $L \in \mathrm{O}(f)$ satisfying $\mathrm{Diag}(L\mathrm{Diag}(b)L^{\mathrm{T}}) = a$ whenever $a \prec b$.

Consider now the two $f$-dimensional vectors

$$a = \Big(\frac{\|\Sigma^2\|}{f},\dots,\frac{\|\Sigma^2\|}{f}\Big), \tag{3.101}$$

$$b = \Big(\sigma_1 - \frac{\rho\lambda\kappa_\rho}{f+\rho\lambda},\dots,\sigma_\rho - \frac{\rho\lambda\kappa_\rho}{f+\rho\lambda},0,\dots,0\Big) \tag{3.102}$$

specifically, and note in particular that $a \prec b$. Applying Horn's theorem proves that there exists an orthogonal matrix $L \in O(f)$ such that

$$\mathrm{Diag}\big(L\big(\begin{smallmatrix}\Sigma^2 & 0\\ 0 & 0\end{smallmatrix}\big)L^{\mathrm{T}}\big) = \frac{\|\Sigma^2\|}{f}\mathrm{I}_f. \tag{3.103}$$

In particular, we have shown that the condition in (3.30) holds. Consequently, $M_b \neq \emptyset$. □

### 3.D.3 Proof of Proposition 10 – Characterization of $\mathrm{T}_W M_b$.

We start by describing the tangent space of $\bar{M}_b$. Using the diffeomorpishm in Lemma 18 together with (3.91), we find that for any $W \in \bar{M}_b$,

$$\mathrm{T}_W\bar{M}_b \simeq \mathrm{T}_0\Big(\frac{\mathrm{O}(f)}{\mathrm{I}_\rho \oplus \mathrm{O(f}-\rho)}\Big) \simeq \frac{\mathfrak{o}(f))}{0_\rho \oplus \mathfrak{o}(f-\rho)}. \tag{3.104}$$

Here, $\mathfrak{o}(s)$ denotes the Lie algebra of the orthogonal group O(s), and

$$\frac{\mathfrak{o}(f)}{0_\rho \oplus \mathfrak{o}(f-\rho)} = \Big\{\Big(\begin{smallmatrix}X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\Big) : X \in \mathrm{Skew}(\mathbb{R}^{\rho\times\rho}), E \in \mathbb{R}^{\rho\times(f-\rho)}\Big\}. \tag{3.105}$$

Note also that the isomorphism in (3.104) is given by the differential $\mathrm{D}_{\mathrm{Id}}\pi$ of the action $\pi$ in (3.29) at the identity of $\mathrm{O}(f)$; that is, $\mathrm{T}_W(\bar{M}_b) = \mathrm{D}_{\mathrm{Id}}\pi(\mathfrak{o}(f)/0_\rho \oplus \mathfrak{o}(f-\rho))(W)$.

Recall now that $W$ has a SVD decomposition of the form $(U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V)$ by Lemma 17. We therefore have that for any $(X,E;-E^{\mathrm{T}},0) \in \mathfrak{o}(f)/0_\rho \oplus \mathfrak{o}(f-\rho)$,

$$\begin{aligned}\mathrm{D}_{\mathrm{Id}}\pi\Big(S^{\mathrm{T}}\Big(\begin{smallmatrix}X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\Big)S\Big)(W) &\overset{(3.92)}{=} \Big(W_2S^{\mathrm{T}}\Big(\begin{smallmatrix}X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\Big)S, S^{\mathrm{T}}\Big(\begin{smallmatrix}X^{\mathrm{T}} & -E\\ E^{\mathrm{T}} & 0\end{smallmatrix}\Big)SW_1\Big)\\ &= \Big(U\Sigma_2\Big(\begin{smallmatrix}X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\Big)S, S^{\mathrm{T}}\Big(\begin{smallmatrix}X^{\mathrm{T}} & -E\\ E^{\mathrm{T}} & 0\end{smallmatrix}\Big)\Sigma_1 V\Big).\end{aligned} \tag{3.106}$$

Consequently,

$$\begin{aligned}\mathrm{T}_W(\bar{M}_b) &= \mathrm{D}_{\mathrm{Id}}\pi(\mathfrak{o}(f)/0_\rho \oplus \mathfrak{o}(f-\rho))(W)\\ &= \left\{\left(U\Sigma_2\left(\begin{smallmatrix} X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\right)S, S^{\mathrm{T}}\left(\begin{smallmatrix} X^{\mathrm{T}} & -E\\ E^{\mathrm{T}} & 0\end{smallmatrix}\right)\Sigma_1 V\right) : X \in \mathrm{Skew}(\mathbb{R}^{\rho\times\rho}), E \in \mathbb{R}^{\rho\times(f-\rho)}\right\}. \end{aligned} \tag{3.107}$$

Next, recall that $\mathrm{D}_W T : \mathrm{T}_W \bar{M}_b \to \mathrm{T}_{T(W)}\mathbb{R}^f$. Concretely, for any

$$(V_2, V_1) = \left(U\Sigma_2\left(\begin{smallmatrix} X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\right)S, S^{\mathrm{T}}\left(\begin{smallmatrix} X^{\mathrm{T}} & -E\\ E^{\mathrm{T}} & 0\end{smallmatrix}\right)\Sigma_1 V\right) \in \mathrm{T}_W \bar{M}_b \tag{3.108}$$

say, we have that

$$\begin{aligned}&\mathrm{D}_W T(V_2, V_1)\\ &\overset{(3.34)}{=} \mathrm{Diag}\left(\left(\mathrm{D}_{\mathrm{Id}}\pi\left(S^{\mathrm{T}}\left(\begin{smallmatrix} X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\right)S\right)(W)\right)_1 W_1^{\mathrm{T}} + W_1\left(\mathrm{D}_{\mathrm{Id}}\pi\left(S^{\mathrm{T}}\left(\begin{smallmatrix} X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\right)S\right)(W)\right)_1^{\mathrm{T}}\right)\\ &\overset{(3.106)}{=} \mathrm{Diag}\left(S^{\mathrm{T}}\left(\begin{smallmatrix} X^{\mathrm{T}} & -E\\ E^{\mathrm{T}} & 0\end{smallmatrix}\right)\Sigma_1 V W_1^{\mathrm{T}} + W_1 V^{\mathrm{T}}\Sigma_1^{\mathrm{T}}\left(\begin{smallmatrix} X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\right)S\right)\\ &\overset{(\mathrm{SVD})}{=} \mathrm{Diag}\left(S^{\mathrm{T}}\left(\begin{smallmatrix} X^{\mathrm{T}} & -E\\ E^{\mathrm{T}} & 0\end{smallmatrix}\right)\Sigma_1 V V^{\mathrm{T}}\Sigma_1^{\mathrm{T}} S + S^{\mathrm{T}}\Sigma_1 V V^{\mathrm{T}}\Sigma_1^{\mathrm{T}}\left(\begin{smallmatrix} X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\right)S\right)\\ &\overset{(3.31,3.32)}{=} \mathrm{Diag}\left(S^{\mathrm{T}}\left(\begin{smallmatrix} X^{\mathrm{T}} & -E\\ E^{\mathrm{T}} & 0\end{smallmatrix}\right)\left(\begin{smallmatrix} \Sigma^2 & 0\\ 0 & 0\end{smallmatrix}\right)S + S^{\mathrm{T}}\left(\begin{smallmatrix} \Sigma^2 & 0\\ 0 & 0\end{smallmatrix}\right)\left(\begin{smallmatrix} X & E\\ -E^{\mathrm{T}} & 0\end{smallmatrix}\right)S\right)\\ &= \mathrm{Diag}\left(S^{\mathrm{T}}\left(\begin{smallmatrix} X^{\mathrm{T}}\Sigma^2 & 0\\ E^{\mathrm{T}}\Sigma^2 & 0\end{smallmatrix}\right)S + S^{\mathrm{T}}\left(\begin{smallmatrix} \Sigma^2 X & \Sigma^2 E\\ 0 & 0\end{smallmatrix}\right)S\right) = 2\mathrm{Diag}\left(S^{\mathrm{T}}\left(\begin{smallmatrix} \Sigma^2 X & \Sigma^2 E\\ 0 & 0\end{smallmatrix}\right)S\right).\end{aligned} \tag{3.109}$$

Let now $W \in M_b \backslash \mathrm{Sing}(M_b)$. By (3.35), $\mathrm{D}_W T$ has maximal rank $f-1$. By continuity, the full rank property holds in an open set. There thus exists an open neighborhood $\mathcal{N}_W \subseteq \bar{M}_b$ say of $W$ such that for any $W' \in \mathcal{N}_W$ the rank of $\mathrm{D}_{W'}T$ is constant and equal to $f-1$. Note now that $T : \mathcal{N}_W \to \mathbb{R}^f$ is a smooth function and we have $T(W) = \|\Sigma^2\|_1/f$ by (3.17) and (3.18). In particular for any $W \in \mathcal{N}_W \cap M_b$ we have $\mathrm{D}_{W'}T$ is maximal and $T(W') = \|\Sigma^2\|_1/f$. The *constant rank theorem* [113, Theorem 5.22] therefore applies, and there exists an open neighborhood $\mathcal{U}_W \subseteq \mathcal{N}_W$ of $W$ such that

$$T^{-1}\left(\frac{\|\Sigma^2\|_1}{f}\right) \cap \mathcal{U}_W = M_b \cap U_W \tag{3.110}$$

is a smooth embedded manifold in $\bar{M}_b$ of codimension $f-1$.

Note now furthermore that for any $W \in \bar{M}_b$, $\mathrm{Tr}[T(W)] = \|\Sigma^2\|$ by the diffeomorphism in Lemma 18. The map $\mathrm{D}_W T$ can therefore have rank $f-1$ at most in particular. That is, any $f-1$ components of $T$ are regular at $W$ and we can therefore consider $M_b$ as being an embedded manifold in $\bar{M}_b$ [113, Proposition 5.28]. Hence, by [113, Lemma 5.29] we also have that for any $Q \in T^{-1}(\|\Sigma^2\|_1/f) \cap U_W = M_b \cap U_W$ we have the representation

$$\ker \mathrm{D}_Q T = \mathrm{T}_Q M_b, \tag{3.111}$$

where we understand $\mathrm{T}_Q M_b$ as a subspace of $\mathrm{T}_Q \bar{M}_b$.

This concludes the proof. □

### 3.D.4 Proof of Proposition 11 – The set Sing($M_b$)

We start by proving that if there exists a point $W \in M_b$ such that $\mathrm{rk}(\mathrm{D}_W T) = f-1$, or in other words $M_b \backslash \mathrm{Sing}(M_b) \neq \emptyset$, then Proposition 11 holds. The proof relies on an established fact for the singular loci in affine algebraic varieties, of which $M_b$ is one.

**Lemma 20.** *If there exists a point $W \in M_b$ such that $\operatorname{rk}(\mathrm{D}_W T) = f-1$, then Proposition 11 holds.*

*Proof.* Fix any point $W = (W_2, W_1) \in \bar{M}_b$. Let $\pi$ be the action defined in (3.29), and recall the representation of $M_b$ in (3.30) as well as the definition of $\bar{M}_b$ in (3.80). Observe that the set $M_b$ can be defined as the set of solutions to the algebraic equations

$$L \in [L] \in \frac{\mathrm{O}(f)}{\mathrm{I}_\rho \oplus \mathrm{O}(f-\rho)} \simeq \bar{M}_b$$

$$\frac{\left\|\Sigma^2\right\|_1}{f} \mathrm{I}_{f\times f} = \operatorname{Diag}(L^{\mathrm{T}} W_2^{\mathrm{T}} W_2 L) = \operatorname{Diag}(L^{\mathrm{T}} W_1 W_1^{\mathrm{T}} L) \overset{(3.34)}{=} T(\pi(L)(W)). \tag{3.112}$$

We may therefore consider $M_b$ as a *real algebraic variety* of $\mathcal{P}$; that is, the zero loci (the set of real solutions) of a set of real polynomials of finite degree with variables in $\mathcal{P}$. Let $P_1, \ldots, P_s$ with $s = \dim \mathcal{P} - \dim(\bar{M}_b)$ be the polynomials defining $\bar{M}_b$ at zero, that is, $\bar{M}_b = P_1^{-1}(0) \cap \ldots \cap P_s^{-1}(0)$. If we denote the gradient with respect to the coordinates in $\mathcal{P}$ by $\nabla$, then the matrix composed by $(\nabla P_i)_{i=1}^s$ has rank $\dim \mathcal{P} - \dim(\bar{M}_b)$ at $W$ whenever $P_1(W) = \ldots = P_s(W) = 0$. Eq. (3.112) also shows that $T$ defines $M_b$ and its differential $\mathrm{D}_W T$ via $f-1$ polynomials $Q_1, \ldots, Q_{f-1}$ say (one less than $f$, since the trace of $T$ is fixed) plus the polynomials $\{P_i\}_{i=1}^s$ needed to define $\bar{M}_b$. Recall that $\bar{M}_b$ is a smooth manifold and has no singular points. In particular, we have a matrix

$$\mathcal{S} = \big(\nabla P_1, \ldots, \nabla P_s, \nabla Q_1, \ldots, \nabla Q_{f-1}\big) \tag{3.113}$$

that satisfies the following: if $W \in \mathcal{P}$ is such that $P_i(W) = Q_j(W) = 0$ for all $i, j$, then $\mathcal{S}$ has rank at most $\dim \mathcal{P} - \dim(\bar{M}_b) + f - 1$. The set of singular points $\operatorname{Sing}(M_b)$ can be then understood as the set of points $W \in \mathcal{P} \cap M_b$ that are not regular points; that is, the set of points $W \in M_b$ where $\mathcal{S}$ does not have maximal rank. This is a closed Zariski set in the algebraic variety $M_b$.

Recall now that there exists $W^* \in M_b$ such that $\operatorname{rk}(\mathrm{D}_{W^*} T) = f-1$ is maximal by assumption. This implies that for $W'$ in a neighborhood of $W^*$, $\mathrm{D}_{W'} T$ has also constant rank $f-1$. Noting that the polynomials $Q_i$ for $i = 1, \ldots, f-1$ are defined as $f-1$ components of $T(\pi(L)(W))$ up to a constant, we then have that $\mathcal{S}$ has exactly rank $\dim \mathcal{P} - \dim(\bar{M}_b) + f - 1$ at $W^*$. By [111, Prop. 3.3.10, (iii) → (ii)], there is an irreducible component of $M_b$ of codimension $f-1$ in $\bar{M}_b$ (or of dimension $\dim(\bar{M}_b) - (f-1)$ in $\mathcal{P}$), and there is a unique component containing $W^*$. By [111, Prop. 3.3.14], $\operatorname{Sing}(M_b)$ is then an algebraic set of codimension strictly larger than $f-1$ in $\bar{M}_b$ (see also [136, §6.2]). Alternatively we can say that $\operatorname{Sing}(M_b)$ is a proper closed Zariski set in $M_b$. Hence, $M_b$ is generically smooth or up to a closed algebraic set of dimension smaller than $M_b$. From the previous computation moreover, $M_b$ is then a smooth manifold of codimension $f-1$ in $\bar{M}_b$ up to the closed lower-dimensional algebraic set $\operatorname{Sing}(M_b)$. □

We will next prove that the condition of Lemma 20 holds under Assumption 9, i.e., that there exists a point $W \in M_b$ such that $\operatorname{rk}(\mathrm{D}_W T) = f-1$. The proof consists of two steps. First, we prove that a sufficient condition is the existence of a particular orthogonal matrix (Lemma 21). Second, we prove that such orthogonal matrix indeed exists by iteratively constructing said matrix (Lemma 23).

**Lemma 21.** *Suppose Assumption 9 holds. Let $W \in \bar{M}_b \cap T^{-1}(\|\Sigma^2\|_1/f)$ have an SVD of the type $(U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V)$. If one of the first $\rho$ rows of $S \in O(f)$ has no zeros, then there exists a point $W \in M_b$ such that* $\operatorname{rk}(\mathrm{D}_W T) = f-1$.

*Proof.* The SVD exists by Lemma 17. Let

$$X = \begin{pmatrix} 0 & X_{12} & \cdots & X_{1\rho} \\ -X_{12} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ -X_{1\rho} & 0 & \cdots & 0 \end{pmatrix} \in \operatorname{Skew}(\mathbb{R}^{\rho\times\rho}), \quad B = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1\rho} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{\rho\times(f-\rho)}. \tag{3.114}$$

Hence, the first row of $X$ is the vector $X_{1,\cdot} = (0, X_{1,2:\rho})$ where $X_{1,2:\rho} = (X_{12},\dots,X_{1\rho}) \in \mathbb{R}^{\rho-1}$ say, and the first row of $B$ is a vector $B_{1\cdot} \in \mathbb{R}^{(f-\rho)}$ say. Let $\Sigma^2 = \operatorname{Diag}(\theta_1,\dots,\theta_\rho) \in \mathbb{R}^{\rho\times\rho}$ be as in (3.32), and define $\theta_{2:\rho} = (\theta_2,\dots,\theta_\rho) \in \mathbb{R}^{\rho-1}$.

Consider now the map

$$\begin{aligned} &\operatorname{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix} \Sigma^2 X & \Sigma^2 B \\ 0_{(f-\rho)\times\rho} & 0_{(f-\rho)\times(f-\rho)} \end{pmatrix} S\Big) \\ &= \operatorname{Diag}\big(S^{\mathrm{T}}\begin{pmatrix} \Sigma^2 X & 0_{\rho\times(f-\rho)} \\ 0_{(f-\rho)\times\rho} & 0_{(f-\rho)\times(f-\rho)} \end{pmatrix} S\big) + \operatorname{Diag}\big(S^{\mathrm{T}}\begin{pmatrix} 0_{\rho\times\rho} & \Sigma^2 B \\ 0_{(f-\rho)\times\rho} & 0_{(f-\rho)\times(f-\rho)} \end{pmatrix} S\big) \\ &= \operatorname{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix} 0_{1\times1} & \theta_1 X_{1,2:\rho} & 0_{1\times(f-\rho)} \\ -(\theta_{2:\rho}\odot X_{1,2:\rho})^{\mathrm{T}} & 0_{(\rho-1)\times(\rho-1)} & 0_{(\rho-1)\times(f-\rho)} \\ 0_{(f-\rho)\times1} & 0_{(f-\rho)\times(\rho-1)} & 0_{(f-\rho)\times(f-\rho)} \end{pmatrix} S\Big) \\ &\quad + \operatorname{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix} 0_{1\times1} & 0_{1\times(\rho-1)} & \theta_1 B_{1,:} \\ 0_{(\rho-1)\times1} & 0_{(\rho-1)\times(\rho-1)} & 0_{(\rho-1)\times(f-\rho)} \\ 0_{(f-\rho)\times1} & 0_{(f-\rho)\times(\rho-1)} & 0_{(f-\rho)\times(f-\rho)} \end{pmatrix} S\Big). \end{aligned} \tag{3.115}$$

Observe now that because $\operatorname{Diag}(A) = \operatorname{Diag}(A^{\mathrm{T}})$ for any square matrix $A \in \mathbb{R}^{f\times f}$, we have

$$\begin{aligned} &\operatorname{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix} 0_{1\times1} & \theta_1 X_{1,2:\rho} & 0_{1\times(f-\rho)} \\ -(\theta_{2:\rho}\odot X_{1,2:\rho})^{\mathrm{T}} & 0_{(\rho-1)\times(\rho-1)} & 0_{(\rho-1)\times(f-\rho)} \\ 0_{(f-\rho)\times1} & 0_{(f-\rho)\times(\rho-1)} & 0_{(f-\rho)\times(f-\rho)} \end{pmatrix} S\Big) \\ &= \operatorname{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix} 0_{1\times1} & \theta_1 X_{1,2:\rho} - \theta_{2:\rho}\odot X_{1,2:\rho} & 0_{1\times(f-\rho)} \\ 0_{(f-1)\times1} & 0_{(f-1)\times(\rho-1)} & 0_{(f-1)\times(f-\rho)} \end{pmatrix} S\Big). \end{aligned} \tag{3.116}$$

Define now $x = ((\theta_1-\theta_2)X_{12},\dots,(\theta_1-\theta_\rho)X_{1\rho})$, $b = \theta_1 B_{1,:}$. We have shown that

$$\operatorname{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix} \Sigma^2 X & \Sigma^2 B \\ 0_{(f-\rho)\times\rho} & 0_{(f-\rho)\times(f-\rho)} \end{pmatrix} S\Big) = \operatorname{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix} 0_{1\times1} & x & b \\ 0_{(f-1)\times1} & 0_{(f-1)\times(\rho-1)} & 0_{(f-1)\times(f-\rho)} \end{pmatrix} S\Big). \tag{3.117}$$

Let $S_{\cdot,1},\dots,S_{\cdot,f}$ be the columns of $S$, and denote the $j$-th component of the column $S_{\cdot,i}$ by $S_{ij}$. We have now

$$\begin{aligned} &\operatorname{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix} \Sigma^2 X & \Sigma^2 B \\ 0_{(f-\rho)\times\rho} & 0_{(f-\rho)\times(f-\rho)} \end{pmatrix} S\Big) = \operatorname{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix} (0,x,b) \\ 0_{(f-1)\times f} \end{pmatrix} S\Big) \\ &= \operatorname{Diag}\Big(\begin{pmatrix} S_{11}(0,x,b) \\ \cdots \\ S_{1f}(0,x,b) \end{pmatrix} S\Big) = \operatorname{Diag}\Big(S_{11}\langle S_{\cdot,1},(0,x,b)\rangle,\dots,S_{1f}\langle S_{\cdot,f},(0,x,b)\rangle\Big). \end{aligned} \tag{3.118}$$

Recall that $\theta_i \neq \theta_j$ for $i \neq j$, and note that $x \in \mathbb{R}^{\rho-1}$, $b \in \mathbb{R}^{f-\rho}$ are free variables. To prove Lemma 21, we need to find $x$, $b$ such that the map in (3.118) has rank $f-1$. Note that the vector

$$\big(S_{11}\langle S_{\cdot,1},(0,x,b)\rangle,\dots,S_{1f}\langle S_{\cdot,f},(0,x,b)\rangle\big)^{\mathrm{T}} = (0,x,b)S\operatorname{Diag}(S_{11},\dots,S_{1f}). \tag{3.119}$$

The subspace spanned by vectors of the form $(0,x,b) \in \mathbb{R}^f$ has dimension $f-1$. Therefore, since $S$ is an orthogonal matrix (note that taking any representative of $S$ from the homogeneous space $\bar{M}_b$ in (3.80) also works), we have that the linear map in (3.119) has rank $f-1$ if $\mathrm{Diag}(S_{11},\dots,S_{1f})$ has maximal rank. This happens whenever $S_{1i} \neq 0$ for all $i=1,\dots,f$.

The argument above also works if $s \leq \rho$: consider then instead

$$X = \begin{pmatrix} 0 & \cdots & 0 & -X_{s1} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & -X_{s(s-1))} & 0 & \cdots & 0 \\ X_{s1} & \cdots & X_{s(s-1)} & 0 & X_{s(s+1))} & \cdots & X_{s\rho} \\ 0 & \cdots & 0 & -X_{s(s+1))} & 0 & \cdots & 0 \\ \vdots & & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & -X_{s\rho} & 0 & \cdots & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \\ B_{s1} & B_{s2} & \cdots & B_{s\rho} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{\rho\times(f-\rho)}. \tag{3.120}$$

The diagonal matrix in (3.119) will then turn out to be $\mathrm{Diag}(S_{s1},\dots,S_{sf})$ instead. Hence, in case one of the first $\rho$ rows of $S$ has no zeros then the map has rank $f-1$. This concludes the proof. □

In the case that $\rho = 1$, we can verify the requirement of Lemma 21 in (3.33) on the rows of $S$ by computing $M_b$ exactly. This is implied by the following lemma, which will be useful later also in the computations for Proposition 6.

**Lemma 22.** *Suppose Assumption 9 holds and $\rho = 1$. For any $W = (U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V) \in M_b$ that satisfies* (3.33), *it holds that $|S_{1j}|^2 = 1/f$ for $j \in \{1,\dots,f\}$. Furthermore, $M_b$ is a union of finitely many points.*

*Proof.* We first calculate $S$'s entries. Recall $\Sigma_2$'s and $\Sigma_1$'s expressions in (3.31). If $\rho = 1$, then $\Sigma^2 = \Sigma_{\mathcal{W}^*} = \eta \in \mathbb{R}$. Consequently,

$$\mathrm{Diag}(W_2^{\mathrm{T}} W_2) \overset{\text{(SVD)}}{=} \mathrm{Diag}\big(S^{\mathrm{T}} \left(\begin{smallmatrix} \eta & 0 \\ 0 & 0 \end{smallmatrix}\right) S\big) = \mathrm{Diag}\big(|S_{11}|^2\eta, \cdots, |S_{1f}|^2\eta\big). \tag{3.121}$$

By Assumption 9 and (3.33), (3.121) must equal $(\eta/f)\mathrm{I}_f$. Consequently,

$$|S_{11}|^2 = \ldots = |S_{1f}|^2 = \frac{1}{f}. \tag{3.122}$$

We next show that $M_b$ is a union of finitely many points. Whatever choice for $S_{1,\cdot}$ is made (as long as it satisfies (3.122)), we can then complete the system $\{S_{1,\cdot}\}$ to an orthonormal basis $\{S_{1,\cdot},\dots,S_{f,\cdot}\}$ say. In other words, this procedure constructs a matrix $S$ that is moreover in $\mathrm{O}(f)$. The resulting $S$ also gives an element of $\bar{M}_b$ and under the quotient $\mathrm{O}(f) \to \mathrm{O}(f)/(1 \oplus \mathrm{O}(f-1))$ all the basis completions of $S$ are equivalent modulo $1 \oplus \mathrm{O}(f-1)$. Note that we can *a priori* choose the signs of the one dimensional 'seed' subspace corresponding to the singular value $\Sigma$ in $U$ and $V$. Moreover, for each choice of signs in the vector $S_{1,\cdot}$, we can select a unique disjoint element in $M_b$. Hence, there are only finitely many points in $M_b$ when $\rho = 1$. □

In the general case when $\rho > 0$, we can verify the requirement of Lemma 21 on the rows of $S$ by showing that $M_b$ will always have a point $W = (U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V) \in M_b$ such that the first row of $S$ has no zeros. This is implied in by following lemma:

**Lemma 23.** *Let $\rho > 0$. There exists $W = (U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V) \in M_b$ that satisfies (3.33). Moreover, $S \in O(f)$ can be chosen so that the first $\rho$ rows of $S$ have no zeros.*

*Proof.* If $(U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V) \in M_b$, then necessarily

$$\operatorname{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix}\Sigma^2 & 0\\ 0 & 0\end{pmatrix}S\Big) = \frac{\left\|\Sigma^2\right\|_1}{f}\mathrm{I_f} \tag{3.123}$$

by the definition of $M_b$ in (3.30). Recall furthermore that here, $\Sigma^2$ is a diagonal matrix.

We will prove the following result that is stronger than required: Let $b = (b_1,\ldots,b_f) \in \mathbb{R}^f$ be such that $b_1 \geq \ldots \geq b_f \geq 0$, and consider $\bar{b} = ((\sum_i b_i)/f,\ldots,(\sum_i b_i)/f) \in \mathbb{R}^f$. Then there exists $S \in O(f)$ satisfying

$$\operatorname{Diag}\big(S^{\mathrm{T}}\operatorname{Diag}(b)S\big) = \bar{b} \tag{3.124}$$

such that $S_{1i} \neq 0$ for $i = 1,\ldots,f$. We can construct such orthogonal matrix iteratively by extending the proof idea from [155, Theorem 1.4].

Let us begin the iterative construction. For vectors $a, b \in \mathbb{R}^f$ recall the definition of $a \prec b$ in (3.100). Let $\bar{b}' = ((\sum_{i=2}^f \beta_i)/(f-1),\ldots,(\sum_{i=2}^f \beta_i)/(f-1)) \in \mathbb{R}^{f-1}$ and consider $b' = (b_2,\ldots,b_f) \in \mathbb{R}^{f-1}$. Observe that $\bar{b}' \prec b'$, and so by the proof of Lemma 19(see (3.100)–(3.103) specifically), there exists a $W \in O(f-1)$ such that $\bar{b}' = \operatorname{Diag}(W^{\mathrm{T}}\operatorname{Diag}(b')W)$. Consider now the matrix $A_1$ defined by

$$A_1 = \begin{pmatrix}1 & 0\\ 0 & W\end{pmatrix}^{\mathrm{T}}\operatorname{Diag}(b)\begin{pmatrix}1 & 0\\ 0 & W\end{pmatrix} = \begin{pmatrix}b_1 & 0\\ 0 & W^{\mathrm{T}}\operatorname{Diag}(b')W\end{pmatrix}. \tag{3.125}$$

Observe that $\operatorname{Diag}(A_1) = (b_1, \bar{b}')$ is not yet the precise vector we need. But, by using rotations one coordinate at a time, we can move 'mass' of $b_1$ towards the other coordinates to ultimately arrive at the vector $\bar{b}$.

Specifically, the iterative construction proceeds as follows. Consider first the matrix $T_2(t) \in O(f)$ for $t \in [0,1]$ defined by

$$T_2(t) = \begin{pmatrix}\sqrt{t} & \sqrt{1-t} & 0_{1\times f-2}\\ -\sqrt{1-t} & \sqrt{t} & 0_{1\times f-2}\\ 0^{\mathrm{T}}_{1\times f-2} & 0^{\mathrm{T}}_{1\times f-2} & I_{f-2}\end{pmatrix}. \tag{3.126}$$

Since $b_1 \geq (\sum_{i=1}^f b_i)/f \geq (\sum_{i=2}^f b_i)/(f-1)$, there exists $t_2 \in (0,1)$ such that $(\sum_{i=1}^f b_i)/f = (1-t_2)b_1 + t_2(\sum_{i=2}^f b_i)/(f-1)$. We can then compute

$$\begin{pmatrix}1 & 0\\ 0 & W\end{pmatrix}T_2(t_2) = \begin{pmatrix}\sqrt{t_2} & \sqrt{1-t_2} & 0_{1\times(f-2)}\\ * & * & 0_{1\times(f-2)}\\ *^{\mathrm{T}}_{1\times(f-2)} & *^{\mathrm{T}}_{1\times(f-2)} & (W_{i,j})_{i,j>2}\end{pmatrix}, \tag{3.127}$$

where $*$ denote entries which may be different from zero (their precise values are unimportant). In particular, it holds that

$$\begin{aligned}
&A_2 = T_2(t_2)^{\mathrm{T}} A_1 T_2(t_2)\\
&= T_2(t_2)^{\mathrm{T}} \begin{pmatrix} \sqrt{t_2} b_1 & \sqrt{1-t_2} b_1 & 0_{1\times(f-2)} \\ -\sqrt{1-t_2}(\sum_{i=2}^f b_i)/(f-1) & \sqrt{t_2}(\sum_{i=2}^f b_i)/(f-1) & *_{1\times(f-2)} \\ *^{\mathrm{T}}_{1\times(f-2)} & *^{\mathrm{T}}_{1\times(f-2)} & (W^{\mathrm{T}}\mathrm{Diag}(b')W)_{i,j>2} \end{pmatrix}\\
&= \begin{pmatrix} t_2 b_1 + (1-t_2)(\sum_{i=2}^f b_i)/(f-1) & * & 0_{1\times(f-2)} \\ * & (\sum_{i=1}^f b_i)/f & *_{1\times f-2} \\ 0^{\mathrm{T}}_{1\times(f-2)} & *^{\mathrm{T}}_{1\times(f-2)} & (W^{\mathrm{T}}\mathrm{Diag}(b')W)_{i,j>2} \end{pmatrix}. \qquad (3.128)
\end{aligned}$$

Observe furthermore from the definition of $t_2$ and the fact that $\mathrm{Tr}(A_2) = \sum_{i=1}^f b_i$, that

$$\begin{aligned}
t_2 b_1 + (1-t_2)\frac{\sum_{i=2}^f b_i}{f-1} &= \sum_{i=1}^f b_i - (f-2)\frac{\sum_{i=2}^f b_i}{f-1} - \frac{\sum_{i=1}^f b_i}{f}\\
&= b_1 + \frac{\sum_{i=2}^f b_i}{f-1} - \frac{\sum_{i=1}^f b_i}{f} = \frac{f-2}{f-1} b_1 + \frac{1}{(f-1)f}(\sum_{i=1}^n b_i) \geq \frac{1}{f}(\sum_{i=1}^n b_i). \qquad (3.129)
\end{aligned}$$

To arrive at the last inequality, we used here that $b_1 \geq (\sum_{i=1}^n b_i)/f$.

Now consider $T_3(t), \ldots, T_f(t) \in O(f)$ where

$$T_i(t)_{11} = T_i(t)_{ii} = \sqrt{t}, \quad T_i(t)_{1i} = -T_i(t)_{i1} = \sqrt{1-t}, \qquad (3.130)$$

and $T_{jk}(t) = \delta_{jk}$ for $j,k \notin \{1,i\}$ elsewhere. We define recursively

$$A_i = T_i(t_i)^{\mathrm{T}} A_{i-1} T_i(t_i). \qquad (3.131)$$

An argument analogous to (3.129) shows that $(A_{i-1})_{11} \geq (\sum_{i=1}^f b_i)/f \geq (\sum_{i=2}^f b_i)/(f-1)$ for $i = 3, \ldots, f$. Hence, we can find $t_i \in (0,1)$ such that $(1-t_i)(A_{i-1})_{11} + t_i(\sum_{i=2}^f b_i)/(f-1) = (\sum_{i=1}^f b_i)/f$. In particular, the first row of

$$S = \begin{pmatrix} 1 & 0 \\ 0 & W \end{pmatrix} \prod_{i=2}^f T_i(t_i) \in O(f) \qquad (3.132)$$

will satisfy $S_{1j} > 0$ for $j = 1, \ldots, f$. Moreover, we will have for $A_f$ that

$$\mathrm{Diag}(A_f) = \mathrm{Diag}(S^{\mathrm{T}}\mathrm{Diag}(b)S) = \bar{b}. \qquad (3.133)$$

To complete the proof, identify $\mathrm{Diag}(\Sigma^2, 0) = \mathrm{Diag}(b)$. In summary, we have thus found a matrix $S \in O(f)$ satisfying (3.33). □

### 3.D.5 Proof of Proposition 12 – Computing $\nabla^2 \mathcal{I}(W)$

We first recall several properties of vectorization in (3.38):

- If $A, B \in \mathbb{R}^{a\times b}$, then $\mathrm{Tr}[A^{\mathrm{T}}B] = \mathrm{vec}(A)^{\mathrm{T}}\mathrm{vec}(B)$.
- If $A \in \mathbb{R}^{e\times f}, B \in \mathbb{R}^{f\times h}$, then $\mathrm{vec}(AB) = B^{\mathrm{T}} \otimes \mathrm{I}_{e\times e}\mathrm{vec}(A) = \mathrm{I}_{h\times h} \otimes A\mathrm{vec}(B)$. Here, $\otimes$ is understood as the Kronecker tensor product compatible with the vectorization vec.

- If $A \in \mathbb{R}^{e\times f}, B \in \mathbb{R}^{f\times h}, C \in \mathbb{R}^{h\times g}$, then $\mathrm{vec}(ABC) = (C^{\mathrm{T}} \otimes A)\mathrm{vec}(B)$.
- Let $K : \mathbb{R}^{ab} \to \mathbb{R}^{ba}$ be the linear map such that $\mathrm{vec}(A^{\mathrm{T}}) = K\mathrm{vec}(A)$ holds for any $A \in \mathbb{R}^{a\times b}$.

We also rewrite (3.53) using the vectorization notation. Specifically, we have that

$$\begin{aligned}\mathrm{vec}(\nabla_1\mathcal{I}(W)) &= \mathrm{vec}(-2W_2^{\mathrm{T}}Y + 2W_2^{\mathrm{T}}W_2W_1) + \mathrm{vec}(2\lambda\mathrm{Diag}(W_2^{\mathrm{T}}W_2)W_1)\\ &= -2\mathrm{vec}(W_2^{\mathrm{T}}Y) + 2\mathrm{I}_{h\times h} \otimes W_2^{\mathrm{T}}W_2\mathrm{vec}(W_1) + 2\lambda\mathrm{I}_{h\times h} \otimes \mathrm{Diag}(W_2^{\mathrm{T}}W_2)\mathrm{vec}(W_1), \quad (3.134)\end{aligned}$$

and

$$\begin{aligned}\mathrm{vec}(\nabla_2\mathcal{I}(W)) &= \mathrm{vec}(-2YW_1^{\mathrm{T}} + 2W_2W_1W_1^{\mathrm{T}}) + \mathrm{vec}(2\lambda W_2\mathrm{Diag}(W_1W_1^{\mathrm{T}}))\\ &= -2\mathrm{vec}(YW_1^{\mathrm{T}}) + 2W_1W_1^{\mathrm{T}} \otimes \mathrm{I}_{e\times e}\mathrm{vec}(W_2) + 2\lambda\mathrm{Diag}(W_1W_1^{\mathrm{T}}) \otimes \mathrm{I}_{e\times e}\mathrm{vec}(W_2). \quad (3.135)\end{aligned}$$

As a final preparatory step, let us denote for $i, j \in \{1, 2\}$ the partial derivatives with respect to matrices $i$ and $j$ by $\partial_{i,j}$. For example,

$$\partial_{1,1}\mathcal{I} \in \mathbb{R}^{(f\times h)\times(f\times h)} \quad \text{and} \quad (\partial_{1,1}\mathcal{I})_{kl,mn} = \frac{\partial^2\mathcal{I}(W)}{\partial W_{1kl}\partial W_{1mn}}. \quad (3.136)$$

*Step 1: Calculating the partial derivatives $\partial_{ij}\mathcal{I}(W)$.* We start by computing the partial derivatives $\partial_{11}\mathcal{I}, \partial_{22}\mathcal{I}$ directly from (3.134), (3.135). For any vector $v \in \mathbb{R}^n$ and matrix $A \in \mathbb{R}^{n\times n}$, it holds that $\partial(Av)/\partial v = A$. Therefore

$$\begin{aligned}\partial_{1,1}\mathcal{I}(W) &= 2\mathrm{I}_{h\times h} \otimes W_2^{\mathrm{T}}W_2 + 2\lambda\mathrm{I}_{h\times h} \otimes \mathrm{Diag}(W_2^{\mathrm{T}}W_2),\\ \partial_{2,2}\mathcal{I}(W) &= 2W_1W_1^{\mathrm{T}} \otimes \mathrm{I}_{e\times e} + 2\lambda\mathrm{Diag}(W_1W_1^{\mathrm{T}}) \otimes \mathrm{I}_{e\times e}. \quad (3.137)\end{aligned}$$

Next we are going to calculate $\partial_{1,2}\mathcal{I}$. We first rewrite terms of $\nabla_1\mathcal{I}(W)$ and $\nabla_2\mathcal{I}(W)$ in (3.53). Specifically, note that

$$\begin{aligned}\mathrm{vec}\big(-2W_2^{\mathrm{T}}(Y - W_2W_1)\big) &\overset{(i)}{=} -2\mathrm{vec}(W_2^{\mathrm{T}}Y) + 2W_1^{\mathrm{T}} \otimes W_2^{\mathrm{T}}\mathrm{vec}(W_2),\\ &\overset{(ii)}{=} -2((Y - W_2W_1)^{\mathrm{T}} \otimes \mathrm{I}_{f\times f})K\mathrm{vec}(W_2). \quad (3.138)\end{aligned}$$

Here, we have isolated (i) $W_2$ by using the identity $\mathrm{vec}(ABC) = C^{\mathrm{T}} \otimes A\mathrm{vec}(B)$; and (ii) $W_2^{\mathrm{T}}$ using the tensor $K$ that satisfies $\mathrm{vec}(W_2^{\mathrm{T}}) = K\mathrm{vec}(W_2)$. Similarly, note that

$$\begin{aligned}\mathrm{vec}(2\lambda\mathrm{Diag}(W_2^{\mathrm{T}}W_2)W_1) &\overset{(iii)}{=} \mathrm{vec}(2\lambda\sum_i P_iW_2^{\mathrm{T}}W_2P_iW_1) \quad (3.139)\\ &\overset{(iv)}{=} 2\lambda\sum_i(P_iW_1)^{\mathrm{T}} \otimes P_iW_2^{\mathrm{T}}\mathrm{vec}(W_2)\\ &\overset{(v)}{=} 2\lambda\sum_i((W_2P_iW_1)^{\mathrm{T}} \otimes P_i)K\mathrm{vec}(W_2). \quad (3.140)\end{aligned}$$

Here, we (iii) utilized the fact that that $\mathrm{Diag}(A) = \sum_i^d P_iAP_i$ for some set of symmetric matrices $\{P_i\}_i$, and then isolated (iv) $W_2$ as a vector from (3.139) as well as (v) $W_2^{\mathrm{T}}$ and using the tensor $K$.

Recall (3.134). We take the derivative of $\mathrm{vec}(\nabla_1\mathcal{I}(W))$ with respect to $W_2$ in vectorization notation. While some terms are linear in $W_2$, we use Leibniz's rule on terms including $W_2^{\mathrm{T}}W_2$. Leibniz's rule yields the expressions in (3.138), (3.140) resulting in the tensor

$$\begin{aligned}\partial_{2,1}\mathcal{I}(W) &= -2\big((Y-W_2W_1)^{\mathrm{T}}\otimes \mathrm{I}_{f\times f}\big)K + 2W_1^{\mathrm{T}}\otimes W_2^{\mathrm{T}} \\ &\quad + 2\lambda\sum_i \Big(\big((W_2P_iW_1)^{\mathrm{T}}\otimes P_i\big)K + (P_iW_1)^{\mathrm{T}}\otimes P_iW_2^{\mathrm{T}}\Big). \end{aligned} \tag{3.141}$$

*Step 2: Evaluation at a vector.* Now that we have the partial derivatives of the Hessian, we want to apply it to vectors of the form $(V_2,V_1)\in \mathrm{T}_W\mathbb{R}^{e\times f}\times\mathbb{R}^{f\times h}$. Concretely, we will consider the vectorization of $(V_2,V_1)$ and then compute the elements of the left-hand side of (3.37) one by one.

First,

$$\begin{aligned}&\mathrm{vec}(V_1)^{\mathrm{T}}\partial_{1,1}\mathcal{I}(W)\mathrm{vec}(V_1)\\ &\overset{(3.137)}{=} \mathrm{vec}(V_1)^{\mathrm{T}}\big(2\mathrm{I}_{h\times h}\otimes W_2^{\mathrm{T}}W_2 + 2\lambda\mathrm{I}_{h\times h}\otimes \mathrm{Diag}(W_2^{\mathrm{T}}W_2)\big)\mathrm{vec}(V_1)\\ &= \mathrm{vec}(V_1)^{\mathrm{T}}\big(2\mathrm{I}_{h\times h}\otimes W_2^{\mathrm{T}}W_2\mathrm{vec}(V_1) + 2\lambda\mathrm{I}_{h\times h}\otimes \mathrm{Diag}(W_2^{\mathrm{T}}W_2)\mathrm{vec}(V_1)\big)\\ &= \mathrm{vec}(V_1)^{\mathrm{T}}\big(2\mathrm{vec}(W_2^{\mathrm{T}}W_2V_1) + 2\lambda\mathrm{vec}(\mathrm{Diag}(W_2^{\mathrm{T}}W_2)V_1)\big)\\ &= 2\mathrm{Tr}[V_1^{\mathrm{T}}W_2^{\mathrm{T}}W_2V_1] + 2\lambda\mathrm{Tr}[V_1^{\mathrm{T}}\mathrm{Diag}(W_2^{\mathrm{T}}W_2)V_1],\end{aligned} \tag{3.142}$$

where we have used that $\mathrm{vec}(B)^{\mathrm{T}}\mathrm{vec}(A) = \mathrm{Tr}[B^{\mathrm{T}}A]$. Similarly

$$\mathrm{vec}(V_2)^{\mathrm{T}}\partial_{2,2}\mathcal{I}(W)\mathrm{vec}(V_2) \overset{(3.137)}{=} 2\mathrm{Tr}[V_2W_1W_1^{\mathrm{T}}V_2^{\mathrm{T}}] + 2\lambda\mathrm{Tr}[V_2\mathrm{Diag}(W_1W_1^{\mathrm{T}})V_2^{\mathrm{T}}], \tag{3.143}$$

and

$$\begin{aligned}\mathrm{vec}(V_1)^{\mathrm{T}}\partial_{1,2}\mathcal{I}(W)\mathrm{vec}(V_2) &\overset{(3.141)}{=} -2\mathrm{Tr}[V_1^{\mathrm{T}}V_2^{\mathrm{T}}(Y-W_2W_1)] + 2\mathrm{Tr}[V_1^{\mathrm{T}}W_2^{\mathrm{T}}V_2W_1] \qquad (3.144)\\ &\quad + 2\lambda\big(\mathrm{Tr}[V_1^{\mathrm{T}}\mathrm{Diag}(V_2^{\mathrm{T}}W_2)W_1] + \mathrm{Tr}[V_1^{\mathrm{T}}\mathrm{Diag}(W_2^{\mathrm{T}}V_2)W_1]\big).\end{aligned}$$

We also have that $\partial_{2,1}\mathcal{I}(W) = \partial_{1,2}\mathcal{I}(W)^{\mathrm{T}}$ because $\mathcal{I}$ is a smooth function. Therefore

$$\mathrm{vec}(V_2)^{\mathrm{T}}\partial_{2,1}\mathcal{I}(W)\mathrm{vec}(V_1) = \mathrm{vec}(V_1)^{\mathrm{T}}\partial_{1,2}\mathcal{I}(W)\mathrm{vec}(V_2). \tag{3.145}$$

Adding (3.141)–(3.145) yields:

$$\begin{aligned}&\big(\mathrm{vec}(V_1),\mathrm{vec}(V_2)\big)^{\mathrm{T}}\nabla^2\mathcal{I}(W)\big(\mathrm{vec}(V_1),\mathrm{vec}(V_2)\big)\\ &= 2\mathrm{Tr}[V_1^{\mathrm{T}}W_2^{\mathrm{T}}W_2V_1] + 2\lambda\mathrm{Tr}[V_1^{\mathrm{T}}\mathrm{Diag}(W_2^{\mathrm{T}}W_2)V_1]\\ &\quad + 2\mathrm{Tr}[V_2W_1W_1^{\mathrm{T}}V_2^{\mathrm{T}}] + 2\lambda\mathrm{Tr}[V_2\mathrm{Diag}(W_1W_1^{\mathrm{T}})V_2^{\mathrm{T}}]\\ &\quad - 4\mathrm{Tr}[V_1^{\mathrm{T}}V_2^{\mathrm{T}}(Y-W_2W_1)] + 4\mathrm{Tr}[V_1^{\mathrm{T}}W_2^{\mathrm{T}}V_2W_1]\\ &\quad + 4\lambda\big(\mathrm{Tr}[V_1^{\mathrm{T}}\mathrm{Diag}(V_2^{\mathrm{T}}W_2)W_1] + \mathrm{Tr}[V_1^{\mathrm{T}}\mathrm{Diag}(W_2^{\mathrm{T}}V_2)W_1]\big).\end{aligned} \tag{3.146}$$

Finally, note that

$$\begin{aligned}2\|W_2V_1+V_2W_1\|_{\mathrm{F}}^2 &= 2\mathrm{Tr}[(W_2V_1+V_2W_1)^{\mathrm{T}}(W_2V_1+V_2W_1)]\\ &= 2\mathrm{Tr}[V_1^{\mathrm{T}}W_2^{\mathrm{T}}W_2V_1] + 2\mathrm{Tr}[W_1^{\mathrm{T}}V_2^{\mathrm{T}}V_2W_1]\\ &\qquad + 2\mathrm{Tr}[V_1^{\mathrm{T}}W_2^{\mathrm{T}}V_2W_1] + 2\mathrm{Tr}[W_1^{\mathrm{T}}V_2^{\mathrm{T}}W_2V_1]\\ &= 2\mathrm{Tr}[V_1^{\mathrm{T}}W_2^{\mathrm{T}}W_2V_1] + 2\mathrm{Tr}[V_2W_1W_1^{\mathrm{T}}V_2^{\mathrm{T}}] + 4\mathrm{Tr}[V_1^{\mathrm{T}}W_2^{\mathrm{T}}V_2W_1],\end{aligned} \tag{3.147}$$

where in the last equality we have used the cyclic property of the trace. Now, for any $A, B \in \mathbb{R}^n$, $\|A+B\|_{\mathrm{F}}^2 - \|A-B\|_{\mathrm{F}}^2 = 4\langle A, B\rangle$, so that

$$\begin{aligned}
&2\big(\|\mathrm{Diag}(V_2^{\mathrm{T}}W_2)+\mathrm{Diag}(W_1V_1^{\mathrm{T}})\|_{\mathrm{F}}^2 - \|\mathrm{Diag}(V_2^{\mathrm{T}}W_2)-\mathrm{Diag}(W_1V_1^{\mathrm{T}})\|_{\mathrm{F}}^2\big) \\
&= 8\big\langle \mathrm{Diag}(W_1V_1^{\mathrm{T}}), \mathrm{Diag}(V_2^{\mathrm{T}}W_2)\big\rangle = 8\mathrm{Tr}[\mathrm{Diag}(W_1V_1^{\mathrm{T}})\mathrm{Diag}(V_2^{\mathrm{T}}W_2)] \\
&\overset{\text{(vi)}}{=} 8\mathrm{Tr}[W_1V_1^{\mathrm{T}}\mathrm{Diag}(V_2^{\mathrm{T}}W_2)] \overset{\text{(vii)}}{=} 8\mathrm{Tr}[V_1^{\mathrm{T}}\mathrm{Diag}(V_2^{\mathrm{T}}W_2)W_1]. \qquad (3.148)
\end{aligned}$$

Here, we have used (vi) that $\mathrm{Tr}[A\mathrm{Diag}(B)] = \mathrm{Tr}[\mathrm{Diag}(A)\mathrm{Diag}(B)]$ for any $A, B$ square matrices of the same dimension, and (vii) the cyclic property of the trace. Substituting (3.147) and (3.148) into (3.146) completes the proof. □

### 3.D.6 Proof of Proposition 13

**Obtaining** $\mathrm{T}_W M$

We compute first $\mathrm{T}_W M_b$ in Lemma 24, which we will use to compute $\mathrm{T}_W M$ later:

**Lemma 24.** *If Assumption 9 holds, then for any* $W \in M_b \backslash \mathrm{Sing}(M_b)$

$$\begin{aligned}
\mathrm{T}_W M_b = \Bigg\{ &\left(U\begin{pmatrix} \Sigma X & \Sigma E \\ 0_{(e-\rho)\times\rho} & 0_{(e-\rho)\times(f-\rho)} \end{pmatrix} S, S^{\mathrm{T}}\begin{pmatrix} X^{\mathrm{T}}\Sigma & 0_{\rho\times(h-\rho)} \\ E^{\mathrm{T}}\Sigma & 0_{(f-\rho)\times(h-\rho)} \end{pmatrix} V\right) \\
&: X \in \mathrm{Skew}(\mathbb{R}^{\rho\times\rho}), E \in \mathbb{R}^{\rho\times(f-\rho)}, (X, E) \in \ker \mathrm{D}_W T \Bigg\}. \qquad (3.149)
\end{aligned}$$

*Proof.* Let $W = (U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V) \in M_b \backslash \mathrm{Sing}(M_b)$; such SVD exists by Lemma 17. By Proposition 10, $\mathrm{T}_W M_b = \ker \mathrm{D}_W T$ where $\mathrm{D}_W T : \mathrm{T}_W \bar{M}_b \to \mathrm{T}_{T(W)}\mathbb{R}^f$. Next, write

$$\begin{aligned}
\ker \mathrm{D}_W T &= \{(V_2, V_1) \in \mathrm{T}_W \bar{M}_b : \mathrm{D}_W T(V_2, V_1) = 0\} \\
&\overset{(3.36,3.107)}{=} \Bigg\{ \left(U\Sigma_2\begin{pmatrix} X & E \\ -E^{\mathrm{T}} & 0 \end{pmatrix} S, S^{\mathrm{T}}\begin{pmatrix} X^{\mathrm{T}} & -E \\ E^{\mathrm{T}} & 0 \end{pmatrix}\Sigma_1 V\right) : \\
&X \in \mathrm{Skew}(\mathbb{R}^{\rho\times\rho}), E \in \mathbb{R}^{\rho\times(f-\rho)}, 2\mathrm{Diag}\Big(S\begin{pmatrix} \Sigma^2 X & \Sigma^2 E \\ 0 & 0 \end{pmatrix} S\Big) = 0 \Bigg\}. \qquad (3.150)
\end{aligned}$$

Hence, from the bilinear form $\mathrm{D}_W T$ defined in (3.36) in Proposition 10(b), we take the pairs $(X, E) \in \mathrm{Skew}(\mathbb{R}^{\rho\times\rho}) \times \mathbb{R}^{\rho\times(f-\rho)}$ that also belong to $\ker \mathrm{D}_W T$. The last step required to arrive at (3.149) is to substitute the definitions of $\Sigma_2$ and $\Sigma_1$, recall (3.31), into (3.150). □

Observe from (3.18) and (3.29) that, under the action of $H = (\mathbb{R}^*)^f$, we always have $\pi(H)(M_b) \subseteq M$. Proposition 8 implies that $M \subseteq \pi(H)(M_b)$, and hence $\pi(H)(M_b) = M$. Proposition 8 also yields that the group action is free and so the map $\pi : H \times M_b \to M$ is bijective. We have moreover that on the open set $\pi(H \times M_b)$, $\pi$ has a continuous inverse given by

$$\pi^{-1}(W) = (C_W, \pi(C_W)(W)). \qquad (3.151)$$

Here, $C_W = \mathrm{Diag}(W_1W_1^{\mathrm{T}})^{1/4}\mathrm{Diag}(W_2W_2^{\mathrm{T}})^{-1/4}$, which is discussed in the proof of Proposition 8. If $\pi$ is smooth, this allows us to obtain the tangent space of $\mathrm{T}_W M$ at every point $W = \pi(C)(W') \in M$ such that $W' \in M_b \backslash \mathrm{Sing}(M_b)$.

For every point $W \in M_b \backslash \mathrm{Sing}(M_b)$, the action $\pi$ restricted to a smooth neighborhood $R_{\mathrm{Id}} \times U_W \subset H \times M_b$ is a map $\mathrm{D}_{(\mathrm{Id},W)}\pi : \mathcal{H} \times \mathrm{T}_W\mathcal{P} \to \mathrm{T}_W\mathcal{P}$ with $\mathcal{H} = \mathrm{T}_{\mathrm{Id}}H = \mathrm{Lie}((\mathbb{R}^*)^f)$ the Lie algebra of $H$. Furthermore, for every point $W \in M_b$, the differential $\mathrm{D}_{(\mathrm{Id},W)}\pi(D,V)$ at $D \in \mathcal{H}, V \in \mathrm{T}_W M_b$ is given by

$$\begin{aligned} &\mathrm{D}_{(\mathrm{Id},W)}\pi(0,V) \overset{(3.29)}{=} V, \\ &\mathrm{D}_{(\mathrm{Id},W)}\pi(D,0) \overset{(3.29)}{=} (W_2 D, -D W_1) \overset{(3.30)}{=} \left(U\left(\begin{smallmatrix} \Sigma & 0 \\ 0 & 0 \end{smallmatrix}\right) S D, -D S^{\mathrm{T}} \left(\begin{smallmatrix} \Sigma & 0 \\ 0 & 0 \end{smallmatrix}\right) \Sigma_1 V\right) \end{aligned} \tag{3.152}$$

say. For every point $W \in M_b$, we will define the vector space

$$\mathrm{D}_W\pi(\mathcal{H}) = \{\mathrm{D}_{(\mathrm{Id},W)}\pi(D,0) : D \in \mathcal{H}\}. \tag{3.153}$$

Recall now finally that for $V = (V_2, V_1)$ and $R = (R_2, R_1) \in \mathrm{T}_W\mathcal{P}$ we have the Euclidean inner product $\langle \cdot, \cdot \rangle : \mathrm{T}_W\mathcal{P} \times \mathrm{T}_W\mathcal{P} \to \mathbb{R}$ defined as $\langle V, R \rangle = \langle V_2, R_2 \rangle_{\mathrm{F}} + \langle V_1, R_1 \rangle_{\mathrm{F}}$.

We are now in a position to prove the following:

**Lemma 25.** *Suppose Assumption 9 holds. Let $\pi$ be the Lie group action of $H$ on $M$ defined in* (3.29) *and $\mathcal{H} = \mathrm{Lie}((\mathbb{R}^*)^f)$. If $W = (U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V) \in M_b \backslash \mathrm{Sing}(M_b)$, then*

$$\begin{aligned} &\mathrm{T}_W M \\ &= \mathrm{T}_W M_b \oplus \mathrm{D}_W\pi(\mathcal{H}) \\ &= \Bigg\{ \left( U \begin{pmatrix} \Sigma X & \Sigma B \\ 0_{(e-\rho)\times\rho} & 0_{(e-\rho)\times(f-\rho)} \end{pmatrix} S, S^{\mathrm{T}} \begin{pmatrix} X^{\mathrm{T}}\Sigma & 0_{\rho\times(h-\rho)} \\ B^{\mathrm{T}}\Sigma & 0_{(f-\rho)\times(h-\rho)} \end{pmatrix} \Sigma_1 V \right) \\ &\qquad : X \in \mathrm{Skew}(\mathbb{R}^{\rho\times\rho}), B \in \mathbb{R}^{\rho\times(f-\rho)}, (X,B) \in \ker \mathrm{D}_W T \Bigg\} \\ &\quad \oplus \Bigg\{ \left( U \begin{pmatrix} \Sigma & 0_{\rho\times(f-\rho)} \\ 0_{(e-\rho)\times\rho} & 0_{(e-\rho)\times(f-\rho)} \end{pmatrix} S D, -D S^{\mathrm{T}} \begin{pmatrix} \Sigma & 0_{\rho\times(h-\rho)} \\ 0_{(f-\rho)\times\rho} & 0_{(f-\rho)\times(h-\rho)} \end{pmatrix} V \right) : D \in \mathrm{Diag}(\mathbb{R}^{f\times f}) \Bigg\}. \end{aligned} \tag{3.154}$$

*Proof.* Let $W = (U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V) \in M_b \backslash \mathrm{Sing}(M_b)$. Start by noting that Proposition 10 implies that $\mathrm{T}_W M_b = \ker \mathrm{D}_W T$. Here, we understand that $\ker \mathrm{D}_W T \subseteq \mathrm{T}_W \bar{M}_b$. In order to expand this result to $\mathrm{T}_W M$, we will use the smooth action of $H = (\mathbb{R}^*)^f$ on $M_b$.

Let $R_{\mathrm{Id}} \times U_W \subset H \times M_b$ be a neighborhood such that we can compute the differential of $\pi$ in (3.152). Note by combining (3.149), (3.152) and (3.153) that for any $K = (K_2, K_1) \in \mathrm{T}_W M_b$ and $Q = (Q_2, Q_1) \in \mathrm{D}_W\pi(\mathcal{H})$ we have that $\langle K, Q \rangle = 0$. In other words, $\mathrm{T}_W M_b$ is orthogonal to $\mathrm{D}_W\pi(\mathcal{H})$. Hence, the sum of the subspaces $\mathrm{T}_W M_b$ and $\mathrm{D}\pi(\mathcal{H})$ is orthogonal. We also know that $R_{\mathrm{Id}} \times U_W$ is a smooth submanifold of $H \times M_b$ and $\pi$ is smooth, bijective and with continuous inverse. Therefore, by dimension counting, we must have that $\pi$ is a local diffeomorphism and so $\mathrm{T}_W M = \mathrm{T}_W M_b \oplus \mathrm{D}_W\pi(\mathcal{H})$.

To arrive at the expression in (3.154), we simply use the expressions for $\mathrm{T}_W M_b$ from Lemma 24 together with (3.152). □

### Obtaining $\mathrm{T}_W^{\perp} M$

We now compute the cotangent space $\mathrm{T}_W^{\perp} M$ by embedding $\mathrm{T}_W^{\perp} M \subset \mathrm{T}_W\mathcal{P}$ and obtaining the orthogonal complement of $\mathrm{T}_W M$.

**Lemma 26.** *Suppose Assumption 9 holds. For* $W = (U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V) \in M \cap M_b \backslash \mathrm{Sing}(M)$,

$$\begin{aligned}\mathrm{T}_W^{\perp} M = \Big\{ &(K_2, K_1) = \Big(U \begin{pmatrix} A_2 & B_2 \\ C_2 & D_2 \end{pmatrix} S, S^{\mathrm{T}} \begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix} V\Big) \in \mathrm{T}_W \mathcal{P} : \\ &X \in \mathrm{Skew}(\mathbb{R}^{\rho\times\rho}), B \in \mathbb{R}^{\rho\times(f-\rho)}, \\ &\langle \Sigma(A_2 + A_1^{\mathrm{T}}), X\rangle + \langle \Sigma(B_2 + C_1^{\mathrm{T}}), B\rangle = 0, \\ &\mathrm{Diag}(K_2^{\mathrm{T}} W_2) = \mathrm{Diag}(K_1 W_1^{\mathrm{T}}), 2\mathrm{Diag}\Big(S \begin{pmatrix} \Sigma^2 X & \Sigma^2 B \\ 0_{(f-\rho)\times\rho} & 0_{(f-\rho)\times(f-\rho)} \end{pmatrix} S\Big) = 0 \Big\}.\end{aligned} \tag{3.155}$$

*Proof.* Let $W = (U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V) \in M \cap M_b \backslash \mathrm{Sing}(M)$. Taking the orthogonal complement in (3.154), we obtain that

$$\mathrm{T}_W^{\perp} M = \mathrm{T}_W^{\perp} M_b \cap (\mathrm{D}\pi(\mathcal{H}))^{\perp}. \tag{3.156}$$

We will now determine both subspaces in the right-hand side of (3.156). Taking the intersection of these two sets will then immediately result in (3.155).
*Determining* $\mathrm{T}_W^{\perp} M_b$. Recall first the definition of a cotangent space, that is

$$\mathrm{T}_W^{\perp} M_b = \big\{ K \in \mathrm{T}_W \mathcal{P} = \mathrm{T}_{W_2}\mathbb{R}^{e\times f} \times \mathrm{T}_{W_1}\mathbb{R}^{f\times h} : \forall R \in \mathrm{T}_W M_b, \langle K, R\rangle = 0 \big\}. \tag{3.157}$$

Furthermore, note that for any $K = (K_2, K_1) \in T_W \mathcal{P}$, there exist matrices $A_1, A_2 \in \mathbb{R}^{\rho\times\rho}$ and matrices $B_2, B_1, C_2, C_1, D_2, D_1$ of appropriate dimensions such that

$$K = \Big(U \begin{pmatrix} A_2 & B_2 \\ C_2 & D_2 \end{pmatrix} S, S^{\mathrm{T}} \begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix} V\Big). \tag{3.158}$$

This is because $U$, $S$, and $V$ are orthogonal matrices and thus

$$(a, b) \in \mathrm{T}_W \mathcal{P} \Rightarrow (UaS, S^{\mathrm{T}} bV) \in \mathrm{T}_W \mathcal{P}. \tag{3.159}$$

We now investigate the inner product condition in (3.157). Lemma 24 implies that if $R \in \mathrm{T}_W M_b$, then there exist $(X, E) \in \mathrm{Skew}(\mathbb{R}^{\rho\times\rho}) \times \mathbb{R}^{\rho\times(f-\rho)}$ such that

$$R = (R_2, R_1) = \Big(U \begin{pmatrix} \Sigma X & \Sigma E \\ 0 & 0 \end{pmatrix} S, S^{\mathrm{T}} \begin{pmatrix} X^{\mathrm{T}}\Sigma & 0 \\ E^{\mathrm{T}}\Sigma & 0 \end{pmatrix} V\Big) \quad \text{and} \quad 2\mathrm{Diag}\Big(S \begin{pmatrix} \Sigma^2 X & \Sigma^2 E \\ 0 & 0 \end{pmatrix} S\Big) = 0. \tag{3.160}$$

For any $K \in \mathrm{T}_W \mathcal{P}$, the inner product condition in (3.157) reduces to

$$\begin{aligned} 0 = \langle K, R\rangle &= \langle K_2, R_2\rangle + \langle K_1, R_1\rangle \\ &\overset{\text{(i)}}{=} \langle A_2, \Sigma X\rangle + \langle B_2, \Sigma E\rangle + \langle A_1, X^{\mathrm{T}}\Sigma\rangle + \langle C_1, E^{\mathrm{T}}\Sigma\rangle \\ &\overset{\text{(ii)}}{=} \langle \Sigma A_2, X\rangle + \langle A_1\Sigma, X^{\mathrm{T}}\rangle + \langle \Sigma B_2, E\rangle + \langle C_1\Sigma, E^{\mathrm{T}}\rangle \\ &\overset{\text{(iii)}}{=} \langle \Sigma(A_2 + A_1^{\mathrm{T}}), X\rangle + \langle \Sigma(B_2 + C_1^{\mathrm{T}}), E\rangle. \end{aligned} \tag{3.161}$$

Here, we used (i) the representations in (3.158) and (3.160), (ii) that $\langle M, \Sigma N\rangle = \langle \Sigma M, N\rangle$ for any matrices $M, N$ of appropriate size because $\Sigma$ is diagonal, and (iii) that $\langle M, N\rangle = \langle M^{\mathrm{T}}, N^{\mathrm{T}}\rangle$ for any matrices $M, N$ of the same size.

Summarizing, we have that

$$\begin{aligned} T_W^{\perp} M_b = \Big\{ & (K_2, K_1) = \Big(U \big(\begin{smallmatrix} A_2 & B_2 \\ C_2 & D_2 \end{smallmatrix}\big) S, S^{\mathrm{T}} \big(\begin{smallmatrix} A_1 & B_1 \\ C_1 & D_1 \end{smallmatrix}\big) V\Big) \in \mathrm{T}_W \mathcal{P} : \\ & \langle \Sigma(A_2 + A_1^{\mathrm{T}}), X \rangle + \langle \Sigma(B_2 + C_1^{\mathrm{T}}), E \rangle = 0, \\ & X \in \mathrm{Skew}(\mathbb{R}^{\rho \times \rho}), E \in \mathbb{R}^{\rho \times (f-\rho)}, 2\mathrm{Diag}\Big(S \big(\begin{smallmatrix} \Sigma^2 X & \Sigma^2 E \\ 0 & 0 \end{smallmatrix}\big) S\Big) = 0 \Big\}. \end{aligned} \tag{3.162}$$

*Determining* $(\mathrm{D}\pi(\mathcal{H}))^{\perp}$. Recall the definition of an orthogonal complement, that is

$$(\mathrm{D}_W \pi(\mathcal{H}))^{\perp} = \big\{ K \in \mathrm{T}_W \mathcal{P} : \langle K, \mathrm{D}_W \pi(D)(W) \rangle = 0 \forall D \in \mathrm{Diag}(\mathbb{R}^f) \big\}. \tag{3.163}$$

We now investigate the inner product condition in (3.163). For any $K \in \mathrm{T}_W \mathcal{P}, D \in \mathcal{H} = \mathrm{Diag}(\mathbb{R}^f)$, recalling (3.152), this condition reduces to

$$\begin{aligned} 0 = \langle K, \mathrm{D}_W \pi(D)(W) \rangle &= \langle K_2, W_2 D \rangle + \langle K_1, -D W_1 \rangle \\ &\overset{\text{(i)}}{=} \langle W_2^{\mathrm{T}} K_2, D \rangle - \langle K_1 W_1^{\mathrm{T}}, D \rangle = \langle W_2^{\mathrm{T}} K_2 - K_1 W_1^{\mathrm{T}}, D \rangle. \end{aligned} \tag{3.164}$$

Here, we used (i) that $\langle M, NO \rangle = \langle N^{\mathrm{T}} M, O \rangle = \langle M O^{\mathrm{T}}, N \rangle$ for any matrices $M, N, O$ with compatible dimensions. Now, because (3.164) holds for any $D \in \mathrm{Diag}(\mathbb{R}^f)$, we must have that

$$\mathrm{Diag}(W_2^{\mathrm{T}} K_2 - K_1 W_1^{\mathrm{T}}) = 0. \tag{3.165}$$

Summarizing, we have that

$$(\mathrm{D}\pi(\mathcal{H}))^{\perp} = \{ (K_2, K_1) \in T_W \mathcal{P} : \mathrm{Diag}(K_1 W_1^{\mathrm{T}}) = \mathrm{Diag}(W_2^{\mathrm{T}} K_2) \}. \tag{3.166}$$

*Concluding.* As mentioned before, taking the intersection of (3.162) and (3.166) results in (3.155). This completes the proof. □

### Lower bound of $\nabla^2 \mathcal{I}(W)$ restricted to $\mathrm{T}_W^{\perp} M$

We require the following lemma. This will be used in an optimization problem we encounter when looking for a lower bound for $\nabla^2 \mathcal{I}(W)|_{T_W^{\perp} M}$.

**Lemma 27.** *Suppose Assumption 9 holds. Let*

$$\begin{aligned} \mathcal{K} = \Big\{ & \Big(U \Big(\begin{smallmatrix} A_2 & B_2 \\ 0_{(e-\rho) \times \rho} & 0_{(e-\rho) \times (f-\rho)} \end{smallmatrix}\Big) S, S^{\mathrm{T}} \Big(\begin{smallmatrix} A_1 & 0_{\rho \times (h-\rho)} \\ C_1 & 0_{(f-\rho) \times (h-\rho)} \end{smallmatrix}\Big) V\Big) \in \mathrm{T}_W^{\perp} M \\ & : A_1, A_2 \in \mathbb{R}^{\rho \times \rho}, B_2, C_1^{\mathrm{T}} \in \mathbb{R}^{\rho \times (f-\rho)} \Big\}. \end{aligned} \tag{3.167}$$

*For any* $W = (U \Sigma_2 S, S^{\mathrm{T}} \Sigma_1 V) \in M_b \backslash \mathrm{Sing}(M_b)$*, the following holds: if*

$$K = \Big(U \Big(\begin{smallmatrix} A_2 & B_2 \\ 0_{(e-\rho) \times \rho} & 0_{(e-\rho) \times (f-\rho)} \end{smallmatrix}\Big) S, S^{\mathrm{T}} \Big(\begin{smallmatrix} A_1 & 0_{\rho \times (h-\rho)} \\ C_1 & 0_{(f-\rho) \times (h-\rho)} \end{smallmatrix}\Big) V\Big) \in \mathcal{K} \tag{3.168}$$

*say and*

$$\|A_2 \Sigma + \Sigma A_1\|_{\mathrm{F}} = 0, \quad A_1 = A_2^{\mathrm{T}} \quad \text{and} \quad B_2 = C_1^{\mathrm{T}}, \tag{3.169}$$

*then*

$$A_2 = A_1^{\mathrm{T}} = \Sigma X' \quad \text{and} \quad B_2 = C_1^{\mathrm{T}} = \Sigma E' \tag{3.170}$$

*for some* $X' \in \mathrm{Skew}(\mathbb{R}^{\rho \times \rho})$ *and* $E' \in \mathbb{R}^{\rho \times (f-\rho)}$*. If additionally*

$$\mathrm{Diag}\Big(S^{\mathrm{T}} \Big(\begin{smallmatrix} \Sigma(A_2 + A_1^{\mathrm{T}}) & \Sigma(B_2 + C_1^{\mathrm{T}}) \\ 0_{(f-\rho) \times \rho} & 0_{(f-\rho) \times (f-\rho)} \end{smallmatrix}\Big) S\Big) = 0, \tag{3.171}$$

*then* $K = 0$.

*Proof.* We first prove (3.170). It follows from (3.169) that if $\|A_2\Sigma + \Sigma A_1\|_{\mathrm{F}} = 0$, then $A_2\Sigma + \Sigma A_1 = A_2\Sigma + \Sigma A_2^{\mathrm{T}} = 0$ by property of the Frobenius norm. If now $A_2 = \Sigma X'$ say, then $\Sigma X'\Sigma + \Sigma(X')^{\mathrm{T}}\Sigma = 0$. Since $\Sigma$ is invertible, left and right multiplication with its inverse shows that $X' \in \mathrm{Skew}(\mathbb{R}^{\rho\times\rho})$. The identity $B_2 = \Sigma B' = C_1^{\mathrm{T}}$ follows similarly.

We next prove that if (3.171) holds besides (3.169), then in fact $K = 0$. We will do so by showing that $K \in \mathrm{T}_W M$, because then

$$K \in \mathcal{K} \cap \mathrm{T}_W M \overset{(3.167)}{\subseteq} \mathrm{T}_W^{\perp} M \cap \mathrm{T}_W M = \{0\}. \tag{3.172}$$

*Verification that $K \in \mathrm{T}_W M$.* Recall that

$$\begin{aligned} \ker \mathrm{D}_W T &= \big\{(V_2, V_1) \in \mathrm{T}_W \bar{M}_b : \mathrm{D}_W T(V_2, V_1) = 0\big\} \qquad (3.173) \\ &\overset{(3.36)}{=} \Big\{\Big(U\Sigma_2\Big(\begin{smallmatrix} X & E \\ -E^{\mathrm{T}} & 0 \end{smallmatrix}\Big)S, S^{\mathrm{T}}\Big(\begin{smallmatrix} X^{\mathrm{T}} & -E \\ E^{\mathrm{T}} & 0 \end{smallmatrix}\Big)\Sigma_1 V\Big) \in T_W \bar{M}_b : 2\mathrm{Diag}\big(S^{\mathrm{T}}\big(\begin{smallmatrix} \Sigma^2 X & \Sigma^2 E \\ 0 & 0 \end{smallmatrix}\big)S\big) = 0\Big\}. \end{aligned}$$

Thus since

$$2\mathrm{Diag}\big(S^{\mathrm{T}}\big(\begin{smallmatrix} \Sigma^2 X' & \Sigma^2 E' \\ 0 & 0 \end{smallmatrix}\big)S\big) \overset{(3.170)}{=} \mathrm{Diag}\Big(S^{\mathrm{T}}\Big(\begin{smallmatrix} \Sigma(A_2 + A_1^{\mathrm{T}}) & \Sigma(B_2 + C_1^{\mathrm{T}}) \\ 0 & 0 \end{smallmatrix}\Big)S\Big) = 0 \tag{3.174}$$

by assumption (3.171), clearly also

$$\Big(U\Sigma_2\Big(\begin{smallmatrix} X' & E' \\ -(E')^{\mathrm{T}} & 0 \end{smallmatrix}\Big)S, S^{\mathrm{T}}\Big(\begin{smallmatrix} X^{\mathrm{T}} & -E' \\ (E')^{\mathrm{T}} & 0 \end{smallmatrix}\Big)\Sigma_1 V\Big) \overset{(3.173)}{\in} \ker \mathrm{D}_W T. \tag{3.175}$$

Note now lastly that

$$K \overset{(3.170)}{=} \big(U\big(\begin{smallmatrix} \Sigma X' & \Sigma E' \\ 0 & 0 \end{smallmatrix}\big)S, S^{\mathrm{T}}\Big(\begin{smallmatrix} (\Sigma X')^{\mathrm{T}} & 0 \\ (\Sigma E')^{\mathrm{T}} & 0 \end{smallmatrix}\Big)V\big). \tag{3.176}$$

Utilizing (3.175) and (3.176) together with (3.149) of Lemma 24, we conclude that $K \in \mathrm{T}_W M_b \subseteq \mathrm{T}_W M$. This finishes the proof. □

We now define a bilinear form that will appear in the computation of the lower bound of the Hessian $\nabla^2 \mathcal{I}(W)$.

**Definition 12.** *Let $W = (U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V) \in \bar{M}_b$ where $S \in \mathrm{O}(f)$ and let $\Sigma \in \mathrm{Diag}(\mathbb{R}^{\rho\times\rho})$ be defined as in* (3.32)*. Define the map $\bar{T}_W : \mathbb{R}^{\rho\times\rho} \times \mathbb{R}^{\rho\times(f-\rho)} \to \mathbb{R}^f$ by*

$$\bar{T}_W(A, B) = \mathrm{Diag}\big(S^{\mathrm{T}}\big(\begin{smallmatrix} \Sigma A & \Sigma B \\ 0 & 0 \end{smallmatrix}\big)S\big), \tag{3.177}$$

*and the bilinear form $\mathcal{T} : (\mathbb{R}^{\rho\times\rho} \times \mathbb{R}^{\rho\times(f-\rho)}) \times (\mathbb{R}^{\rho\times\rho} \times \mathbb{R}^{\rho\times(f-\rho)}) \to \mathbb{R}$ by*

$$\begin{aligned} \mathcal{T}_W\big((A, B), (A', B')\big) &= \big\langle \bar{T}_W(A, B), \bar{T}_W(A', B') \big\rangle \\ &= \mathrm{Tr}\Big[\mathrm{Diag}\big(S^{\mathrm{T}}\big(\begin{smallmatrix} \Sigma A & \Sigma B \\ 0 & 0 \end{smallmatrix}\big)S\big)\mathrm{Diag}\big(S^{\mathrm{T}}\big(\begin{smallmatrix} \Sigma A' & \Sigma B' \\ 0 & 0 \end{smallmatrix}\big)S\big)\Big]. \end{aligned} \tag{3.178}$$

Observe that, when using notation as in (3.36), we have $\mathrm{D}_W T(V_2, V_1) = \bar{T}_W(\Sigma X, \Sigma B)$.

We also introduce some extra notation. For a positive definite symmetric bilinear form $A : E \times E \to \mathbb{R}$ on a real vector space $E$ with norm $\|\cdot\|$, we denote $A > l$ for $l \in \mathbb{R}_+$ to indicate that $v^{\mathrm{T}}Av > l\|v\|^2$ for all $v \in E$. We are now in position to prove a lower bound for the Hessian using $\mathrm{T}_W^{\perp} M$.

**Lemma 28.** *Suppose Assumption 9 holds. Let $W \in M_b \cap M \backslash \mathrm{Sing}(M) \subseteq M$. We have that $\nabla^2 \mathcal{I}(W)$ restricted to $\mathrm{T}_W^\perp M$ is a positive definite bilinear form. Furthermore,*

$$\nabla^2 \mathcal{I}(W)|_{\mathrm{T}_W^\perp M} \geq \omega \tag{3.179}$$

*where*

$$\omega = \begin{cases} \min\{\zeta_W, 2\frac{\lambda\kappa_\rho\rho}{f+\lambda\rho} - 2\sigma_{\rho+1}\} & \text{if} \quad \rho < f, \\ \min\{\zeta_W, 2(\sigma_\rho - \sigma_{\rho+1})\} & \text{if} \quad \rho = f. \end{cases} \tag{3.180}$$

*Here, $\zeta_W > 0$ is strictly positive and depends on the point $W$, $\lambda$ and $\Sigma$. If $\rho = r$ (recall from* (3.32) *that we have $\rho \leq r$), then we set $\sigma_{\rho+1} = \sigma_{r+1} = 0$.*

*Proof.* To arrive at the result, we will give a lower bound for the solution

$$\mathcal{H}_W^{\mathrm{opt}} = \begin{cases} \text{minimum of} & \big(\mathrm{vec}(V_1), \mathrm{vec}(V_2)\big)^{\mathrm{T}} \nabla^2 \mathcal{I}(W) \big(\mathrm{vec}(V_1), \mathrm{vec}(V_2)\big) \\ \text{obtained over} & (V_2, V_1) \in \mathrm{T}_W \mathcal{P} \\ \text{subject to} & \|(V_2, V_1)\|_{\mathrm{F}} = 1, (V_2, V_1) \in \mathrm{T}_W^\perp M \end{cases} \tag{3.181}$$

say, that holds for any $W \in M_b \cap M \backslash \mathrm{Sing}(M)$. We consider first the case that $\rho < f$.
*Step 1: Simplifying the objective function.* Let $(V_2, V_1) \in T_W \mathcal{P}$, $W \in M_b \cap M \backslash \mathrm{Sing}(M)$. Since $W \in M_b$, we have by (3.33) that

$$\begin{aligned} \mathrm{Tr}\big[V_1^{\mathrm{T}} \mathrm{Diag}(W_2^{\mathrm{T}} W_2) V_1\big] &= \frac{\|\Sigma^2\|_1}{f} \|V_1\|_{\mathrm{F}}^2 \quad \text{and similarly} \\ \mathrm{Tr}\big[V_2 \mathrm{Diag}(W_1 W_1^{\mathrm{T}}) V_2^{\mathrm{T}}\big] &= \frac{\|\Sigma^2\|_1}{f} \|V_2\|_{\mathrm{F}}^2. \end{aligned} \tag{3.182}$$

Substituting (3.182) into (3.37), we find that

$$\begin{aligned} &\big(\mathrm{vec}(V_1), \mathrm{vec}(V_2)\big)^{\mathrm{T}} \nabla^2 \mathcal{I}(W) \big(\mathrm{vec}(V_1), \mathrm{vec}(V_2)\big) \\ &= 2\|W_2 V_1 + V_2 W_1\|_{\mathrm{F}}^2 + 2\lambda \frac{\|\Sigma^2\|_1}{f} \big(\|V_1\|_{\mathrm{F}}^2 + \|V_2\|_{\mathrm{F}}^2\big) - 4\mathrm{Tr}\big[V_1^{\mathrm{T}} V_2^{\mathrm{T}} (Y - \mathcal{S}_\alpha[Y])\big] \\ &\quad + 2\lambda\big(\|\mathrm{Diag}(V_2^{\mathrm{T}} W_2) + \mathrm{Diag}(W_1^{\mathrm{T}} V_1)\|_{\mathrm{F}}^2 - \|\mathrm{Diag}(V_2^{\mathrm{T}} W_2) - \mathrm{Diag}(W_1^{\mathrm{T}} V_1)\|_{\mathrm{F}}^2\big). \end{aligned} \tag{3.183}$$

Substituting (3.183) into (3.181) and using the facts that:

- if $\|(V_2, V_1)\|_{\mathrm{F}} = 1$, then $\|(V_2, V_1)\|_{\mathrm{F}}^2 = \|V_1\|_{\mathrm{F}}^2 + \|V_2\|_{\mathrm{F}}^2 = 1$;
- if $(V_2, V_1) \in \mathrm{T}_W^\perp M$, then $\mathrm{Diag}(V_2^{\mathrm{T}} W_2) - \mathrm{Diag}(W_1 V_1^{\mathrm{T}}) = 0$ by Lemma 26;
- and $\|\Sigma^2\|_1 = \sum_{i=1}^{\rho} (\sigma_i - \lambda\rho\kappa_\rho/(f + \lambda\rho)) = \rho\kappa_\rho - \rho\frac{\lambda\rho\kappa_\rho}{f+\lambda\rho} = \rho\kappa_\rho f/(f + \lambda\rho)$, which can be seen from $\Sigma^2$'s singular values shown in (3.32) and then recalling (3.15);

we find that

$$\mathcal{H}_W^{\mathrm{opt}} = \begin{cases} \text{minimum of} & 2\|W_2 V_1 + V_2 W_1\|_{\mathrm{F}}^2 + 2\lambda \frac{\rho\kappa_\rho}{f+\lambda\rho} \\ & -4\mathrm{Tr}[V_1^{\mathrm{T}} V_2^{\mathrm{T}} (Y - \mathcal{S}_\alpha[Y])] + 8\lambda\|\mathrm{Diag}(V_2^{\mathrm{T}} W_2)\|_{\mathrm{F}}^2 \\ \text{obtained over} & V_2, V_1 \in \mathrm{T}_W \mathcal{P} \\ \text{subject to} & \|V_2\|_{\mathrm{F}}^2 + \|V_1\|_{\mathrm{F}}^2 = 1, (V_2, V_1) \in T_W^\perp M. \end{cases} \tag{3.184}$$

*Step 2: Change of variables.* We now apply a change of variables to the minimization problem in (3.184). Specifically, we utilize the orthogonal matrices $U, S, V$ of the SVD $W = (U\Sigma_2 S, S^{\mathrm{T}}\Sigma_1 V)$ by letting

$$U\tilde{V}_2 S = V_2 \quad \text{and} \quad S^{\mathrm{T}}\tilde{V}_1 V = V_1 \tag{3.185}$$

say. We examine next the consequences of this change of variables to the three relevant terms in (3.184).

Under the change of variables in (3.185), the first term in (3.184) satisfies

$$\begin{aligned} 2\|W_2V_1 + V_2W_1\|_{\mathrm{F}}^2 &\overset{\text{(SVD)}}{=} 2\|U\Sigma_2 SV_1 + V_2 S^{\mathrm{T}}\Sigma_1 V\|_{\mathrm{F}}^2 \\ &\overset{(3.185)}{=} 2\|U\Sigma_2 SS^{\mathrm{T}}\tilde{V}_1 V + U\tilde{V}_2 SS^{\mathrm{T}}\Sigma_1 V\|_{\mathrm{F}}^2 \overset{\text{(i,ii)}}{=} 2\|\Sigma_2\tilde{V}_1 + \tilde{V}_2\Sigma_1\|_{\mathrm{F}}^2, \end{aligned} \tag{3.186}$$

since (i) $SS^{\mathrm{T}} = \mathrm{Id}$ and (ii) the Frobenius norm is unitarily invariant, i.e., $\|U(\cdot)V\|_{\mathrm{F}} = \|\cdot\|_{\mathrm{F}}$.

Recall the definition of $\Sigma_Y$ in Section 3.2.4. Introducing

$$\Lambda = \begin{pmatrix} \Sigma_Y & 0_{r\times(h-r)} \\ 0_{(e-r)\times r} & 0_{(e-r)\times(h-r)} \end{pmatrix}, \tag{3.187}$$

note that

$$\begin{aligned} U^{\mathrm{T}}\big(Y - \mathcal{S}_\alpha[Y]\big)V^{\mathrm{T}} &\overset{\text{(iii)}}{=} U^{\mathrm{T}}\big(U\big(\begin{smallmatrix}\Sigma_Y & 0\\ 0 & 0\end{smallmatrix}\big)V - \mathcal{S}_\alpha[Y]\big)V^{\mathrm{T}} \\ &\overset{\text{(iv)}}{=} U^{\mathrm{T}}(U\Lambda V - U\Sigma_2\Sigma_1 V)V^{\mathrm{T}} \overset{\text{(v)}}{=} \Lambda - \Sigma_2\Sigma_1 \end{aligned} \tag{3.188}$$

by (iii) lifting $Y$'s compact SVD defined in Section 3.2.4 to a full SVD, and since (iv) $W \in M$ and therefore $\mathcal{S}_\alpha[Y] = W_2W_1 = U\Sigma_2 SS_T\Sigma_1 V = U\Sigma_2\Sigma_1 V$ by (3.17), and (v) $U^{\mathrm{T}}U = \mathrm{Id}_{e\times e}$ and $VV^{\mathrm{T}} = \mathrm{Id}_{h\times h}$. Conclude then that under the change of variables in (3.185) the third term in (3.184) satisfies

$$\begin{aligned} &-4\mathrm{Tr}[V_1^{\mathrm{T}}V_2^{\mathrm{T}}(Y - \mathcal{S}_\alpha[Y])] \overset{(3.185)}{=} -4\mathrm{Tr}[(S^{\mathrm{T}}\tilde{V}_1 V)^{\mathrm{T}}(U\tilde{V}_2 S)^{\mathrm{T}}(Y - \mathcal{S}_\alpha[Y])] \\ &\overset{\text{(vi)}}{=} -4\mathrm{Tr}[\tilde{V}_1^{\mathrm{T}}\tilde{V}_2^{\mathrm{T}}U^{\mathrm{T}}(Y - \mathcal{S}_\alpha[Y])V] \overset{(3.188)}{=} -4\mathrm{Tr}[\tilde{V}_1^{\mathrm{T}}\tilde{V}_2^{\mathrm{T}}(\Lambda - \Sigma_2\Sigma_1)], \end{aligned} \tag{3.189}$$

because (vi) of $SS^{\mathrm{T}} = \mathrm{Id}$ and the cyclic property of the trace.

Under the change of variables in (3.185), the fourth term in (3.184) satisfies

$$\begin{aligned} 8\lambda\|\mathrm{Diag}(V_2^{\mathrm{T}}W_2)\|_{\mathrm{F}}^2 &\overset{(3.185)}{=} 8\lambda\|\mathrm{Diag}((U\tilde{V}_2 S)^{\mathrm{T}}W_2)\|_{\mathrm{F}}^2 \overset{\text{(W's SVD)}}{=} 8\lambda\|\mathrm{Diag}((U\tilde{V}_2 S)^{\mathrm{T}}U\Sigma_2 S)\|_{\mathrm{F}}^2 \\ &\overset{\text{(vi)}}{=} 8\lambda\|\mathrm{Diag}(S^{\mathrm{T}}\tilde{V}_2^{\mathrm{T}}\Sigma_2 S)\|_{\mathrm{F}}^2, \end{aligned} \tag{3.190}$$

since (vi) $U^{\mathrm{T}}U = \mathrm{Id}_{e\times e}$.

Applying the change of coordinates in (3.185) to (3.184)—by substituting (3.186), (3.189), and (3.190) into (3.184)—thus yields

$$\mathcal{H}_W^{\mathrm{opt}} = \begin{cases} \text{minimum of} & 2\|\Sigma_2\tilde{V}_1 + \tilde{V}_2\Sigma_1\|_{\mathrm{F}}^2 + 2\frac{\rho\kappa_\rho\lambda}{f+\lambda\rho} \\ & -4\mathrm{Tr}[\tilde{V}_1^{\mathrm{T}}\tilde{V}_2^{\mathrm{T}}(\Lambda - \Sigma_2\Sigma_1)] + 8\lambda\|\mathrm{Diag}(S^{\mathrm{T}}\tilde{V}_2^{\mathrm{T}}\Sigma_2 S)\|_{\mathrm{F}}^2 \\ \text{obtained over} & \tilde{V}_2, \tilde{V}_1 \\ \text{subject to} & \|\tilde{V}_2\|_{\mathrm{F}}^2 + \|\tilde{V}_1\|_{\mathrm{F}}^2 = 1, (U\tilde{V}_2 S, S^{\mathrm{T}}\tilde{V}_1 V) \in T_W^{\perp}M. \end{cases} \tag{3.191}$$

*Step 3: Block matrix parametrization.* We will now write $\tilde{V}_2$ and $\tilde{V}_1$ as block matrices in a manner similar to the parametrization in Lemma 26. In particular, we let

$$\tilde{V}_2 = \left(\begin{smallmatrix} A_2 & B_2 \\ C_2 & D_2 \end{smallmatrix}\right) \text{ where } A_2 \in \mathbb{R}^{\rho\times\rho}, B_2 \in \mathbb{R}^{\rho\times(f-\rho)}, C_2 \in \mathbb{R}^{(e-\rho)\times\rho}, D_2 \in \mathbb{R}^{(e-\rho)\times(f-\rho)}, \tag{3.192}$$

and

$$\tilde{V}_1 = \left(\begin{smallmatrix} A_1 & B_1 \\ C_1 & D_1 \end{smallmatrix}\right) \text{ where } A_1 \in \mathbb{R}^{\rho\times\rho}, B_1 \in \mathbb{R}^{\rho\times(h-\rho)}, C_1 \in \mathbb{R}^{(f-\rho)\times\rho}, D_1 \in \mathbb{R}^{(f-\rho)\times(h-\rho)}. \tag{3.193}$$

We expand the first term of (3.191). Utilizing $\Sigma_2, \Sigma_1$'s definitions in (3.32), we find that

$$\begin{aligned}\|\Sigma_2\tilde{V}_1 + \tilde{V}_2\Sigma_1\|_{\mathrm{F}}^2 &= \|\left(\begin{smallmatrix} \Sigma & 0 \\ 0 & 0 \end{smallmatrix}\right)\left(\begin{smallmatrix} A_1 & B_1 \\ C_1 & D_1 \end{smallmatrix}\right) + \left(\begin{smallmatrix} A_2 & B_2 \\ C_2 & D_2 \end{smallmatrix}\right)\left(\begin{smallmatrix} \Sigma & 0 \\ 0 & 0 \end{smallmatrix}\right)\|_{\mathrm{F}}^2 = \|\left(\begin{smallmatrix} \Sigma A_1 + A_2\Sigma & \Sigma B_1 \\ C_2\Sigma & 0 \end{smallmatrix}\right)\|_{\mathrm{F}}^2 \\ &= \|\Sigma A_1 + A_2\Sigma\|_{\mathrm{F}}^2 + \|\Sigma B_1\|_{\mathrm{F}}^2 + \|C_2\Sigma\|_{\mathrm{F}}^2. \end{aligned} \tag{3.194}$$

We now tackle the third term of (3.191). Recall the definitions of $\Sigma_2$, $\Sigma_1$, $\Lambda$ in (3.32), (3.187) respectively, and let $\Sigma_{\min} \in \mathbb{R}^{(e-\rho)\times(h-\rho)}$ be defined such that

$$\Lambda - \Sigma_2\Sigma_1 = \begin{pmatrix} \lambda\kappa_\rho\rho\mathrm{Id}_{\rho\times\rho}/(f+\lambda\rho) & 0 \\ 0 & \Sigma_{\min} \end{pmatrix}. \tag{3.195}$$

Note that $\Sigma_{\min}$ consists of values $\sigma_{\rho+1}, \ldots, \sigma_r$ in its upper left diagonal. Substituting (3.195) into the third term of (3.191), we find that

$$\begin{aligned}\mathrm{Tr}[\tilde{V}_1^{\mathrm{T}}\tilde{V}_2^{\mathrm{T}}(\Lambda - \Sigma_2\Sigma_1)] &= \mathrm{Tr}\left[\begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix}^{\mathrm{T}}\begin{pmatrix} A_2 & B_2 \\ C_2 & D_2 \end{pmatrix}^{\mathrm{T}}\begin{pmatrix} \lambda\kappa_\rho\rho\mathrm{Id}_{\rho\times\rho}/(f+\lambda\rho) & 0 \\ 0 & \Sigma_{\min} \end{pmatrix}\right] \\ &= \mathrm{Tr}\left[\begin{pmatrix} A_2A_1 + B_2C_1 & A_2B_1 + B_2D_1 \\ C_2A_1 + D_2C_1 & C_2B_1 + D_2D_1 \end{pmatrix}^{\mathrm{T}}\begin{pmatrix} \lambda\kappa_\rho\rho\mathrm{Id}_{\rho\times\rho}/(f+\lambda\rho) & 0 \\ 0 & \Sigma_{\min} \end{pmatrix}\right] \\ &\overset{\text{(i)}}{=} \frac{\lambda\kappa_\rho\rho}{f+\lambda\rho}\mathrm{Tr}[A_2A_1 + B_2C_1] + \mathrm{Tr}[\Sigma_{\min}^{\mathrm{T}}(C_2B_1 + D_2D_1)], \end{aligned} \tag{3.196}$$

where (i) we used that $\mathrm{Tr}[A^{\mathrm{T}}B] = \mathrm{Tr}[B^{\mathrm{T}}A]$ for any pair of matrices $A, B$ of compatible dimensions.

We finally simplify the fourth term of (3.191). Recall again $W$'s SVD $(U\Sigma_2S, S^{\mathrm{T}}\Sigma_1V)$; and that if $(V_2, V_1) \in \mathrm{T}_W^{\perp}M$, then $\mathrm{Diag}(V_2^{\mathrm{T}}W_2) = \mathrm{Diag}(W_1V_1^{\mathrm{T}})$ by Lemma 26. If we use the parametrization from (3.192) and (3.193), this latter relation on diagonals is equivalent to equating

$$\begin{aligned}\mathrm{Diag}(W_2^{\mathrm{T}}V_2) &\overset{\text{(W's SVD, 3.185)}}{=} \mathrm{Diag}(S^{\mathrm{T}}\Sigma_2^{\mathrm{T}}U^{\mathrm{T}}U\tilde{V}_2S) \overset{(3.32, 3.192)}{=} \mathrm{Diag}\big(S^{\mathrm{T}}\left(\begin{smallmatrix} \Sigma^{\mathrm{T}} & 0 \\ 0 & 0 \end{smallmatrix}\right)\left(\begin{smallmatrix} A_2 & B_2 \\ C_2 & D_2 \end{smallmatrix}\right)S\big) \\ &= \mathrm{Diag}\big(S^{\mathrm{T}}\left(\begin{smallmatrix} \Sigma^{\mathrm{T}}A_2 & \Sigma^{\mathrm{T}}B_2 \\ 0 & 0 \end{smallmatrix}\right)S\big) \end{aligned} \tag{3.197}$$

to the expression

$$\mathrm{Diag}(W_1V_1^{\mathrm{T}}) = \mathrm{Diag}\big(S^{\mathrm{T}}\left(\begin{smallmatrix} \Sigma A_1^{\mathrm{T}} & \Sigma C_1^{\mathrm{T}} \\ 0 & 0 \end{smallmatrix}\right)S\big), \tag{3.198}$$

the latter of which can be shown in a similar fashion to (3.197). Recall (i) that for any pair of matrices $A, B$, $\mathrm{Diag}(A^{\mathrm{T}}B) = \mathrm{Diag}(B^{\mathrm{T}}A)$. Therefore,

$$\begin{aligned} 2\mathrm{Diag}(V_2^{\mathrm{T}}W_2) &\overset{\text{(i)}}{=} \mathrm{Diag}(W_1V_1^{\mathrm{T}}) + \mathrm{Diag}(W_2^{\mathrm{T}}V_2) \\ &\overset{(3.197, 3.198)}{=} \mathrm{Diag}\big(S^{\mathrm{T}}\left(\begin{smallmatrix} \Sigma^{\mathrm{T}}A_2 + \Sigma A_1^{\mathrm{T}} & \Sigma^{\mathrm{T}}B_2 + \Sigma C_1^{\mathrm{T}} \\ 0 & 0 \end{smallmatrix}\right)S\big) \\ &\overset{\text{(ii)}}{=} \mathrm{Diag}\big(S^{\mathrm{T}}\left(\begin{smallmatrix} \Sigma(A_2 + A_1^{\mathrm{T}}) & \Sigma(B_2 + C_1^{\mathrm{T}}) \\ 0 & 0 \end{smallmatrix}\right)S\big), \end{aligned} \tag{3.199}$$

where (ii) we have used that $\Sigma$ is a diagonal matrix.

Applying the matrix parametrization in (3.192), (3.193) to (3.191)—by substituting (3.194), (3.196) and (3.199) into (3.191)—yields

$$\mathcal{H}_W^{\mathrm{opt}} = \begin{cases} \text{minimum of} & 2\big(\|\Sigma A_1 + A_2\Sigma\|_{\mathrm F}^2 + \|\Sigma B_1\|_{\mathrm F}^2 + \|C_2\Sigma\|_{\mathrm F}^2\big) \\ & +2\frac{\lambda\rho\kappa_\rho}{f+\lambda\rho} \\ & -4\frac{\lambda\kappa_\rho\rho}{f+\lambda\rho}\mathrm{Tr}[A_2A_1 + B_2C_1] \\ & -4\mathrm{Tr}[\Sigma_{\min}^{\mathrm T}(C_2B_1 + D_2D_1)] \\ & +2\lambda\mathrm{Tr}\Big[\mathrm{Diag}\big(S^{\mathrm T}\left(\begin{smallmatrix}\Sigma(A_2+A_1^{\mathrm T}) & \Sigma(B_2+C_1^{\mathrm T}) \\ 0 & 0\end{smallmatrix}\right)S\big)^2\Big] \\ \text{obtained over} & A_1, B_1, C_1, D_1; A_2, B_2, C_2, D_2 \\ \text{subject to} & \|A_1\|_{\mathrm F}^2 + \|B_1\|_{\mathrm F}^2 + \cdots + \|D_2\|_{\mathrm F}^2 = 1, \\ & \big(U\left(\begin{smallmatrix}A_2 & B_2 \\ C_2 & D_2\end{smallmatrix}\right)S, S^{\mathrm T}\left(\begin{smallmatrix}A_1 & B_1 \\ C_1 & D_1\end{smallmatrix}\right)V\big) \in T_W^\perp M. \end{cases} \tag{3.200}$$

*Step 5: The first bounds.* We now start with bounding the objective function in (3.200). Here, we utilize an auxiliary lemma—Lemma 30—twice. Lemma 30 and its proof can be found in Appendix 3.E .

First, we lower bound the second part of the first term in (3.200). Denote the singular values of $\Sigma$ by $\chi_1, \ldots, \chi_\rho$; these satisfy $\chi_i > \chi_{i+1}$ and $\chi_i^2 = \sigma_i - (\lambda\kappa_\rho\rho)/(f+\lambda\rho)$ for $i = 1, \ldots, \rho - 1$. From the fact that $\Sigma$ is an invertible, positive and diagonal matrix with minimal eigenvalue $\chi_\rho$, we conclude using (i) Lemma 30(c) that

$$\|\Sigma B_1\|_{\mathrm F}^2 + \|C_2\Sigma\|_{\mathrm F}^2 \overset{\text{(i)}}{\geq} \chi_\rho^2\big(\|B_1\|_{\mathrm F}^2 + \|C_2\|_{\mathrm F}^2\big) = \Big(\sigma_\rho - \frac{\lambda\kappa_\rho\rho}{f+\lambda\rho}\Big)\big(\|B_1\|_{\mathrm F}^2 + \|C_2\|_{\mathrm F}^2\big). \tag{3.201}$$

Next, we upper bound the next-to-last term in (3.200). Recall that the largest singular value of $\Sigma_{\min}$ is $\sigma_{\rho+1}$; all of its singular values are in the set $\{\sigma_{\rho+1}, \sigma_{\rho+2}, \ldots, \sigma_r, 0\}$. Using (ii) the cyclic property of the trace, and (iii) Lemma 30(b), we therefore have

$$\begin{aligned} \mathrm{Tr}[\Sigma_{\min}^{\mathrm T}(C_2B_1 + D_2D_1)] &= \mathrm{Tr}[\Sigma_{\min}^{\mathrm T}C_2B_1] + \mathrm{Tr}[\Sigma_{\min}^{\mathrm T}D_2D_1] \\ &\overset{\text{(ii)}}{=} \mathrm{Tr}[C_2B_1\Sigma_{\min}^{\mathrm T}] + \mathrm{Tr}[D_2D_1\Sigma_{\min}^{\mathrm T}] \\ &\overset{\text{(iii)}}{\leq} \frac{\sigma_{\rho+1}}{2}\big(\mathrm{Tr}[C_2C_2^{\mathrm T}] + \mathrm{Tr}[B_1B_1^{\mathrm T}] + \mathrm{Tr}[D_2D_2^{\mathrm T}] + \mathrm{Tr}[D_1D_1^{\mathrm T}]\big) \\ &= \frac{\sigma_{\rho+1}}{2}\big(\|B_1\|_{\mathrm F}^2 + \|C_2\|_{\mathrm F}^2 + \|D_1\|_{\mathrm F}^2 + \|D_2\|_{\mathrm F}^2\big). \end{aligned} \tag{3.202}$$

Using (3.201) and (3.202) to bound their respective terms in (3.200), together with the constraint $\|A_1\|_{\mathrm F}^2 + \|B_1\|_{\mathrm F}^2 + \cdots + \|D_2\|_{\mathrm F}^2 = 1$, we obtain the following lower bound for (3.200):

$$\mathcal{H}_W^{\mathrm{opt}} \geq \begin{cases} \text{minimum of} & 2\|\Sigma A_1 + A_2\Sigma\|_{\mathrm F}^2 + 2\big(\sigma_\rho - \frac{\lambda\kappa_\rho\rho}{f+\lambda\rho}\big)\big(\|B_1\|_{\mathrm F}^2 + \|C_2\|_{\mathrm F}^2\big) \\ & +2\frac{\lambda\rho\kappa_\rho}{f+\lambda\rho}\big(\|A_1\|_{\mathrm F}^2 + \|B_1\|_{\mathrm F}^2 + \cdots + \|D_2\|_{\mathrm F}^2\big) \\ & -4\frac{\lambda\kappa_\rho\rho}{f+\lambda\rho}\mathrm{Tr}[A_2A_1 + B_2C_1] \\ & -2\sigma_{\rho+1}\big(\|B_1\|_{\mathrm F}^2 + \|C_2\|_{\mathrm F}^2 + \|D_1\|_{\mathrm F}^2 + \|D_2\|_{\mathrm F}^2\big) \\ & +2\lambda\mathrm{Tr}\big[\mathrm{Diag}\big(S^{\mathrm T}\left(\begin{smallmatrix}\Sigma(A_2+A_1^{\mathrm T}) & \Sigma(B_2+C_1^{\mathrm T}) \\ 0 & 0\end{smallmatrix}\right)S\big)^2\big] \\ \text{obtained over} & A_1, B_1, C_1, D_1; A_2, B_2, C_2, D_2 \\ \text{subject to} & \|A_1\|_{\mathrm F}^2 + \|B_1\|_{\mathrm F}^2 + \cdots + \|D_2\|_{\mathrm F}^2 = 1, \\ & \big(U\left(\begin{smallmatrix}A_2 & B_2 \\ C_2 & D_2\end{smallmatrix}\right)S, S^{\mathrm T}\left(\begin{smallmatrix}A_1 & B_1 \\ C_1 & D_1\end{smallmatrix}\right)V\big) \in T_W^\perp M. \end{cases} \tag{3.203}$$

*Step 6: Splitting the minimization over two subspaces, with two quadratic forms.* We examine now (3.203) closely. We split the objective function into the sum of

$$\begin{aligned}\mathcal{B}_1(B_1,C_2,D_1,D_2) &= 2(\sigma_\rho-\sigma_{\rho+1})\big(\|B_1\|_{\mathrm{F}}^2+\|C_2\|_{\mathrm{F}}^2\big)\\ &\quad+2\Big(\frac{\lambda\rho\kappa_\rho}{f+\lambda\rho}-\sigma_{\rho+1}\Big)\big(\|D_1\|_{\mathrm{F}}^2+\|D_2\|_{\mathrm{F}}^2\big)\end{aligned}\tag{3.204}$$

and

$$\begin{aligned}&\mathcal{B}_2(A_1,A_2,B_2,C_1)=2\|\Sigma A_1+A_2\Sigma\|_{\mathrm{F}}^2+2\frac{\lambda\rho\kappa_\rho}{f+\lambda\rho}\Big(\|A_1\|_{\mathrm{F}}^2+\|A_2\|_{\mathrm{F}}^2-2\mathrm{Tr}[A_2A_1]\\ &+\|B_2\|_{\mathrm{F}}^2+\|C_1\|_{\mathrm{F}}^2-2\mathrm{Tr}[B_2C_1]\Big)+2\lambda\mathrm{Tr}\Big[\mathrm{Diag}\Big(S^{\mathrm{T}}\begin{pmatrix}\Sigma(A_2+A_1^{\mathrm{T}}) & \Sigma(B_2+C_1^{\mathrm{T}})\\ 0 & 0\end{pmatrix}S\Big)^2\Big].\end{aligned}\tag{3.205}$$

Also observe in (3.204) that the coefficients in front of $\|B_1\|_{\mathrm{F}}^2+\|C_2\|_{\mathrm{F}}^2$ and $\|D_1\|_{\mathrm{F}}^2+\|D_2\|_{\mathrm{F}}^2$ are both strictly positive; visit Section 3.2.4 and recall (3.15) specifically.

In the dimensions provided in (3.192) and (3.193), let

$$\begin{aligned}\mathcal{V}_1=\mathrm{span}\Big\{&\Big(U\begin{pmatrix}0&0\\C_2&0\end{pmatrix}S,S^{\mathrm{T}}\begin{pmatrix}0&0\\0&0\end{pmatrix}V\Big),\Big(U\begin{pmatrix}0&0\\0&D_2\end{pmatrix}S,S^{\mathrm{T}}\begin{pmatrix}0&0\\0&0\end{pmatrix}V\Big),\\ &\Big(U\begin{pmatrix}0&0\\0&0\end{pmatrix}S,S^{\mathrm{T}}\begin{pmatrix}0&B_1\\0&0\end{pmatrix}V\Big),\Big(U\begin{pmatrix}0&0\\0&0\end{pmatrix}S,S^{\mathrm{T}}\begin{pmatrix}0&0\\0&D_1\end{pmatrix}V\Big)\\ &:B_1\in\mathbb{R}^{\rho\times(h-\rho)},C_2\in\mathbb{R}^{(e-\rho)\times\rho},D_1\in\mathbb{R}^{(f-\rho)\times(h-\rho)},D_2\in\mathbb{R}^{(e-\rho)\times(f-\rho)}\Big\}.\end{aligned}\tag{3.206}$$

Note now that $\mathcal{V}_1\subseteq\mathrm{T}_W^\perp M$. Indeed, when we examine the definition of $\mathrm{T}_W^\perp M$ in Lemma 26, we can see that nearly every condition pertains to $A_1,A_2,B_2,C_1$ only and not to $B_1,C_2,D_1$, $D_2$—the only exception is possibly the condition $\mathrm{Diag}(V_2^{\mathrm{T}}W_2)=\mathrm{Diag}(W_1V_1^{\mathrm{T}})$. But in fact in (3.197) and (3.198), we can see that the matrices $B_1,C_2,D_1,D_2$ do not appear in this constraint. Hence, any $v\in\mathcal{V}_1$ will satisfy the conditions in the definition of $\mathrm{T}_W^\perp M$ in Lemma 26. This observation yields therefore that $\mathcal{V}_1\subseteq\mathrm{T}_W^\perp M$.

Consider now the orthogonal complement of $\mathcal{V}_1$ in $\mathrm{T}_W^\perp M$ given by

$$\mathcal{V}_2=\Big\{\Big(U\begin{pmatrix}A_2&B_2\\C_2&D_2\end{pmatrix}S,S^{\mathrm{T}}\begin{pmatrix}A_1&B_1\\C_1&D_1\end{pmatrix}V\Big)\in\mathrm{T}_W^\perp M:B_1,C_2,D_1,D_2=0\Big\}\cap\mathrm{T}_W^\perp M.\tag{3.207}$$

From the definitions of $\mathcal{V}_1$ and $\mathcal{V}_2$ we have:

- $\mathcal{V}_1\subseteq\mathrm{T}_W^\perp M$,
- $\mathcal{V}_1\oplus\mathcal{V}_2=\mathrm{T}_W^\perp M$, and
- $\mathcal{V}_1\perp\mathcal{V}_2$.

Using (3.204)–(3.207) we can then lower bound

$$\mathcal{H}_W^{\mathrm{opt}}\geq\begin{cases}\text{minimum of} & \xi\big(\|B_1\|_{\mathrm{F}}^2+\|C_2\|_{\mathrm{F}}^2+\|D_1\|_{\mathrm{F}}^2+\|D_2\|_{\mathrm{F}}^2\big)+\mathcal{B}_2(A_1,A_2,B_2,C_1)\\ \text{obtained over} & (0,0,B_1,0,C_2,0,D_1,D_2)\in\mathcal{V}_1,\\ & (A_1,A_2,0,B_2,0,C_1,0,0)\in\mathcal{V}_2,\\ \text{subject to} & \|A_1\|_{\mathrm{F}}^2+\|B_1\|_{\mathrm{F}}^2+\cdots+\|D_2\|_{\mathrm{F}}^2=1.\end{cases}$$

Here,

$$\xi=2\min\Big\{\frac{\lambda\rho\kappa_\rho}{f+\lambda\rho}-\sigma_{\rho+1},\sigma_\rho-\sigma_{\rho+1}\Big\}>0.\tag{3.208}$$

Now critically, note that $\mathcal{B}_1$ and $\mathcal{B}_2$ are quadratic forms, i.e., for any $\eta \in \mathbb{R}$, $\mathcal{B}_1(\eta\cdot) = \eta^2\mathcal{B}_1(\cdot)$ and $\mathcal{B}_2(\eta\cdot) = \eta^2\mathcal{B}_2(\cdot)$. We can therefore apply Lemma 31, to find that

$$\mathcal{H}_W^{\text{opt}} \geq \min\Big\{\xi, \min_{\substack{\|v_2\|_{\text{F}}^2=1\\ v_2\in\mathcal{V}_2}} \mathcal{B}_2(v))\Big\}. \tag{3.209}$$

*Step 7: Lower bounding the minimum of* $\mathcal{B}_2$*.* We will now prove that

$$\begin{aligned}\zeta_W = \min_{\substack{\|v_2\|_{\text{F}}^2=1\\ v_2\in\mathcal{V}_2}} \mathcal{B}_2(v) = \min_{\substack{\|v_2\|_{\text{F}}^2=1\\ v_2\in\mathcal{V}_2}} 2\|\Sigma A_1 + A_2\Sigma\|_{\text{F}}^2 + 2\frac{\lambda\rho\kappa_\rho}{f+\lambda\rho}\Big(\|A_1\|_{\text{F}}^2 + \|A_2\|_{\text{F}}^2 - 2\text{Tr}[A_2A_1]\\ + \|B_2\|_{\text{F}}^2 + \|C_1\|_{\text{F}}^2 - 2\text{Tr}[B_2C_1]\Big) + 2\lambda\text{Tr}\Big[\text{Diag}\Big(S^{\text{T}}\Big(\begin{smallmatrix}\Sigma(A_2+A_1^{\text{T}}) & \Sigma(B_2+C_1^{\text{T}})\\ 0 & 0\end{smallmatrix}\Big)S\Big)^2\Big]\end{aligned} \tag{3.210}$$

has a strictly positive lower bound. Note that (3.210) can only equal zero if and only if at every solution $(A_1^*, A_2^*, B_2^*, C_1^*)$, the following conditions hold

C1. $2\|\Sigma A_1^* + A_2^*\Sigma\|_{\text{F}}^2 = 0$, and

C2. $2\frac{\lambda\rho\kappa_\rho}{f+\lambda\rho}\Big(\|A_1^*\|_{\text{F}}^2 + \|A_2^*\|_{\text{F}}^2 - 2\text{Tr}[A_2^*A_1^*] + \|B_2^*\|_{\text{F}}^2 + \|C_1^*\|_{\text{F}}^2 - 2\text{Tr}[B_2^*C_1^*]\Big) = 0$, and

C3. $2\lambda\text{Tr}\Big[\text{Diag}\Big(S^{\text{T}}\Big(\begin{smallmatrix}\Sigma(A_2^*+(A_1^*)^{\text{T}}) & \Sigma(B_2^*+(C_1^*)^{\text{T}})\\ 0 & 0\end{smallmatrix}\Big)S\Big)^2\Big] = 0$.

This is because both

$$\|A_1\|_{\text{F}}^2 + \|A_2\|_{\text{F}}^2 - 2\text{Tr}(A_2A_1) \geq 0 \quad \text{and} \quad \|B_2\|_{\text{F}}^2 + \|C_1\|_{\text{F}}^2 - 2\text{Tr}(B_2C_1) \geq 0 \tag{3.211}$$

are nonnegative: see Lemma 30(a) in Appendix 3.E. We next prove that if conditions C1–C3 hold, then necessarily $(A_1^*, A_2^*, B_1^*, C_2^*) = 0$. Consequently, we must have a positive lower bound as there are no such solutions in the optimization domain of (3.210).

Condition C1 equals zero if and only if $A_2^*\Sigma + \Sigma A_1^* = 0$. This is a standard property of a norm. Condition C2 equals zero if and only if $(A_1^*)^{\text{T}} = A_2^*$, $B_2^* = (C_1^*)^{\text{T}}$. This is an additional consequence of Lemma 30(a). Equivalent to conditions C1, C2 are therefore the statements that

$$A_2^*\Sigma + \Sigma A_1^* = 0, \quad (A_1^*)^{\text{T}} = A_2^* \quad \text{and} \quad B_2^* = (C_1^*)^{\text{T}}. \tag{3.212}$$

Condition C3 is equivalent to

$$\begin{aligned}&2\lambda\text{Tr}\Big[\text{Diag}\Big(S^{\text{T}}\Big(\begin{smallmatrix}\Sigma(A_2^*+(A_1^*)^{\text{T}}) & \Sigma(B_2^*+(C_1^*)^{\text{T}})\\ 0 & 0\end{smallmatrix}\Big)S\Big)^2\Big]\\ &= 2\lambda\mathcal{T}_W\big(A_2^*+(A_1^*)^{\text{T}}, B_2^*+(C_1^*)^{\text{T}}, A_2^*+(A_1^*)^{\text{T}}, B_2^*+(C_1^*)^{\text{T}}\big) = 0\end{aligned} \tag{3.213}$$

by Definition 12. By $\bar{T}_W$'s definition in (3.177) and $\mathcal{T}_W$'s definition in (3.178), we must then have that

$$\bar{T}_W\big(A_2^*+(A_1^*)^{\text{T}}, B_2^*+(C_1^*)^{\text{T}}\big) = \text{Diag}\Big(S^{\text{T}}\Big(\begin{smallmatrix}\Sigma(A_2^*+(A_1^*)^{\text{T}}) & \Sigma(B_2^*+(C_1^*)^{\text{T}})\\ 0 & 0\end{smallmatrix}\Big)S\Big) = 0 \tag{3.214}$$

also. We have proven that if conditions C1–C3 are all met, then all prerequisites of Lemma 27 are met; compare (3.212) to (3.169) and (3.214) to (3.171). Lemma 27 implies that $(A_1^*, A_2^*, B_2^*, C_1^*) = 0$.

We finally form the lower bound. We have proven that there is no solution in the optimization domain that satisfies conditions C1–C3 simultaneously. Consequently,

$$\zeta_W = \min_{\substack{\|v_2\|_{\mathrm{F}}^2=1 \\ v_2\in\mathcal{V}_2}} \mathcal{B}_2(v) > 0. \tag{3.215}$$

Substituting (3.215) into (3.209), we obtain that

$$\mathcal{H}_W^{\mathrm{opt}} \geq \min\Big\{\zeta_W, 2\frac{\lambda\kappa_\rho\rho}{f+\lambda\rho} - 2\sigma_{\rho+1}, 2(\sigma_\rho - \sigma_{\rho+1})\Big\}. \tag{3.216}$$

Because $\sigma_\rho \geq \lambda\kappa_\rho\rho/(f+\lambda\rho)$, we also have that

$$\mathcal{H}_W^{\mathrm{opt}} \geq \min\Big\{\zeta_W, 2\frac{\lambda\kappa_\rho\rho}{f+\lambda\rho} - 2\sigma_{\rho+1}\Big\}. \tag{3.217}$$

This concludes the case that $\rho < f$.

Now consider the case $\rho = f$. The proof is mostly the same except for the fact that in Lemmas 24–27, all coordinates indicated to 'have dimension $f-\rho = 0$' need to be removed from the subsequent calculations. Furthermore, we then also use that $\mathcal{W}^*$ equals the rank-$f$ approximation of $S_\alpha[Y]$ as $f = \rho \leq r$—recall the discussion below (3.17). Concretely, the matrices $B_2, C_1, D_1, D_2$ do not appear in the calculations and ultimately this will yield functions $\mathcal{B}_1$, $\mathcal{B}_2$ different from (3.204), (3.205), respectively. Specifically, we find the function

$$\mathcal{B}_1(B_1, C_2) = 2(\sigma_\rho - \sigma_{\rho+1})\big(\|B_1\|_{\mathrm{F}}^2 + \|C_2\|_{\mathrm{F}}^2\big) \tag{3.218}$$

and similarly the function

$$\begin{aligned}\mathcal{B}_2(A_1, A_2) = {}& 2\|\Sigma A_1 + A_2\Sigma\|_{\mathrm{F}}^2 + 2\frac{\lambda\rho\kappa_\rho}{f+\lambda\rho}\Big(\|A_1\|_{\mathrm{F}}^2 + \|A_2\|_{\mathrm{F}}^2 - 2\mathrm{Tr}[A_2A_1]\Big) \\ & + 2\lambda\mathrm{Tr}\big[\mathrm{Diag}\big(S^{\mathrm{T}}\Sigma(A_2 + A_1^{\mathrm{T}})S\big)^2\big].\end{aligned} \tag{3.219}$$

Observe now that in (3.218) there is only one quadratic term involving $B_1$ and $C_2$, and its coefficient results in a replacement for (3.208):

$$\xi = 2(\sigma_\rho - \sigma_{\rho+1}). \tag{3.220}$$

These changes carry over to (3.217), and the minimum becomes

$$\mathcal{H}_W^{\mathrm{opt}} \geq \min\Big\{\zeta_W, 2(\sigma_\rho - \sigma_{\rho+1})\Big\}. \tag{3.221}$$

This concludes the case that $\rho = f$.

Note finally that if $\rho = r$, then $\sigma_{\rho+1} = 0$ by (3.32). This concludes the proof. □

We can improve the result in Lemma 28 in case $\rho = 1$. Note that whenever $e = 1$, then necessarily $\rho = 1$ also by (3.17). In this case we can explicitly calculate the minima. Note that this case occurs when $p$ is either sufficiently small or when we have rank one data (that is, when $r = 1$).

**Lemma 29.** *Suppose that Assumption 9 holds and that $\rho=1$. If $W\in M_b\backslash\mathrm{Sing}(M_b)$, then Lemma 28 holds with*

$$\omega=\begin{cases}\frac{2\lambda\sigma}{f+\lambda} & \text{if } r=1,\\ \frac{2\lambda\sigma_1}{f+\lambda}-2\sigma_2 & \text{otherwise}\end{cases}\tag{3.222}$$

*instead.*

*Proof.* Recall from (3.104) and (3.105) that we are able to characterize elements of $\mathrm{T}_W\bar{M}_b$ using pairs $(X,E)$ where $X\in\mathrm{Skew}(\mathbb{R}^{\rho\times\rho})$ and $E\in\mathbb{R}^{\rho\times(f-\rho)}$. For the particular case $\rho=1$, we have that $\mathfrak{o}(1)=\{0\}$ and $E\in\mathbb{R}^{f-1}$. Conclude therefore from (3.36) that if

$$\mathrm{Diag}\big(S^{\mathrm{T}}\begin{pmatrix}0 & \eta^2E\\ 0 & 0\end{pmatrix}S\big)=0,\tag{3.223}$$

then $(0,E)\in\ker\mathrm{D}_WT$. Here, $\eta^2=\Sigma^2=f\sigma/(f+\lambda)\in\mathbb{R}$. Note now that in fact $\eta\neq0$ under Assumption 9: this allows us next to argue that in the present case $\rho=1$ and $\eta\neq0$, (3.223) holds if and only if $E=0$. This critical observation for the case $\rho=1$ allows us to extend Lemma 28, since we will see that the term in (3.223) is proportional to $\|E\|_{\mathrm{F}}^2$. This allows us to explicitly compute $\zeta_W$.

*Proof that* (3.223) *holds if and only if* $E=0$. Let $W=(U\Sigma_2S,S^{\mathrm{T}}\Sigma_1V)\in M_b\backslash\mathrm{Sing}(M_b)$—by Lemma 21 $M_b\backslash\mathrm{Sing}(M_b)\neq\emptyset$—and refer to the rows of $S$ as $S_{1\cdot},\dots,S_{f\cdot}$. By Lemma 22, these satisfy

$$|S_{1j}|=\frac{1}{\sqrt{f}}\quad\text{for}\quad j=1,\dots,f.\tag{3.224}$$

We therefore have for any $s\neq0$,

$$\begin{aligned}\mathrm{Tr}\Big[\mathrm{Diag}\big(S^{\mathrm{T}}\left(\begin{smallmatrix}0 & sE\\ 0 & 0\end{smallmatrix}\right)S\big)^2\Big]&\overset{(3.224)}{=}\mathrm{Tr}\Big[\mathrm{Diag}\Big(\begin{pmatrix}0 & sE/f\\ \cdots & \cdots\\ 0 & sE/f\end{pmatrix}(S_{\cdot1},\dots,S_{\cdot f})\Big)^2\Big]\\ &=\sum_{i=1}^{f}\frac{s^2}{f}\big\langle(0,E),S_{\cdot i}^{\mathrm{T}}\big\rangle^2\overset{\text{(i)}}{=}\frac{s^2}{f}\|E\|_2^2\end{aligned}\tag{3.225}$$

because (i) the columns of $S$ form an orthonormal basis and we could therefore use Parseval's identity. Consequently,

$$\mathrm{Diag}\big(S^{\mathrm{T}}\left(\begin{smallmatrix}0 & sE\\ 0 & 0\end{smallmatrix}\right)S\big)=0\tag{3.226}$$

if and only if $E=0$.

*Modification of step 4:* The equality of (3.197) and (3.198) implies that

$$\mathrm{Diag}\big(S^{\mathrm{T}}\left(\begin{smallmatrix}\eta A_2 & \eta B_2\\ 0 & 0\end{smallmatrix}\right)S\big)=\mathrm{Diag}\big(S^{\mathrm{T}}\begin{pmatrix}\eta A_1^{\mathrm{T}} & \eta C_1^{\mathrm{T}}\\ 0 & 0\end{pmatrix}S\big).\tag{3.227}$$

By taking traces in (3.227), we find that $\eta A_1=\eta A_2$. Because in the present case we have $A_1,A_2\in\mathbb{R}$, we can restrict to the solutions of the form $A_1=A_2=a$ say as $\eta\neq0$. Thus, we can optimize over vectors of the type $v=(a,a,B_2,C_1)$ in a similar way as explained in Step 6 of the proof ofLemma 28. Next, we conduct the minimization (mimicking Step 7 of Lemma 28's proof).

*Modification of step 7: Lower bounding the minimum of $\mathcal{B}_2$.* In the present setting,

$$\|A_1\|_{\mathrm{F}}^2+\|A_2\|_{\mathrm{F}}^2-2\mathrm{Tr}[A_2A_1]=|a|^2+|a|^2-2a^2=0.\tag{3.228}$$

Furthermore, $\kappa_\rho = \sigma$ since $\rho = 1$. The optimization problem in (3.210) therefore reduces to

$$\begin{aligned}\min_{\substack{\|v\|=1\\ v\in\mathcal{V}}} \mathcal{B}_2(v) = \min_{\substack{\|v\|=1\\ v\in\mathcal{V}}} 8\eta^2 a^2 + 2\frac{\lambda\sigma}{f+\lambda}\big(\|B_1\|_{\mathrm{F}}^2 + \|C_1\|_{\mathrm{F}}^2 - 2\mathrm{Tr}(B_2 C_1)\big)\\ + 2\lambda\mathrm{Tr}\Big[\mathrm{Diag}\Big(S^{\mathrm{T}}\Big(\begin{smallmatrix} 2\eta a & \eta(B_2+C_1^{\mathrm{T}}) \\ 0 & 0\end{smallmatrix}\Big)S\Big)^2\Big]\end{aligned} \tag{3.229}$$

in the present setting. We next simplify (3.229) term by term.

Observe first that

$$\|B_1\|_{\mathrm{F}}^2 + \|C_1\|_{\mathrm{F}}^2 - 2\mathrm{Tr}[B_2 C_1] = \|B_2 - C_1^{\mathrm{T}}\|_{\mathrm{F}}^2 \tag{3.230}$$

by property of the Frobenius norm.

Next, let us inspect the trace in (3.229). Its argument satisfies

$$\begin{aligned}\mathrm{Diag}\Big(S\Big(\begin{smallmatrix} 2\eta a & \eta(B_2+C_1^{\mathrm{T}}) \\ 0 & 0\end{smallmatrix}\Big)S^{\mathrm{T}}\Big)^2 &= \Big(\mathrm{Diag}\Big(S^{\mathrm{T}}\big(\begin{smallmatrix} 2\eta a & 0 \\ 0 & 0\end{smallmatrix}\big)S\Big) + \mathrm{Diag}\Big(S^{\mathrm{T}}\Big(\begin{smallmatrix} 0 & \eta(B_2+C_1^{\mathrm{T}}) \\ 0 & 0\end{smallmatrix}\Big)S\Big)\Big)^2\\ &= \mathrm{Diag}\Big(S^{\mathrm{T}}\big(\begin{smallmatrix} 2\eta a & 0 \\ 0 & 0\end{smallmatrix}\big)S\Big)^2 + \mathrm{Diag}\Big(S^{\mathrm{T}}\Big(\begin{smallmatrix} 0 & \eta(B_2+C_1^{\mathrm{T}}) \\ 0 & 0\end{smallmatrix}\Big)S\Big)^2\\ &\quad + 2\mathrm{Diag}\Big(S^{\mathrm{T}}\big(\begin{smallmatrix} 2\eta a & 0 \\ 0 & 0\end{smallmatrix}\big)S\Big)\mathrm{Diag}\Big(S^{\mathrm{T}}\Big(\begin{smallmatrix} 0 & \eta(B_2+C_1^{\mathrm{T}}) \\ 0 & 0\end{smallmatrix}\Big)S\Big).\end{aligned} \tag{3.231}$$

We may thus split the analysis of the trace in (3.229) by giving attention to the three terms in the right-hand side of (3.231). Since $W \in M_b \backslash \mathrm{Sing}(M_b)$, the trace of the first term in the right-hand side of (3.231) satisfies

$$\mathrm{Tr}\Big[\mathrm{Diag}\big(S^{\mathrm{T}}\big(\begin{smallmatrix} 2\eta a & 0 \\ 0 & 0\end{smallmatrix}\big)S\big)^2\Big] = \frac{4\eta^2 a^2}{f} \quad \text{because} \quad \mathrm{Diag}\big(S^{\mathrm{T}}\big(\begin{smallmatrix} 2\eta a & 0 \\ 0 & 0\end{smallmatrix}\big)S\big)^2 \overset{(3.30)}{=} \frac{(2\eta a)^2}{f^2}\mathrm{I}_{f\times f}. \tag{3.232}$$

The trace of the second term in the right-hand side of (3.231) satisfies

$$\mathrm{Tr}\Big[\mathrm{Diag}\Big(S^{\mathrm{T}}\Big(\begin{smallmatrix} 0 & \eta(B_2+C_1^{\mathrm{T}}) \\ 0 & 0\end{smallmatrix}\Big)S\Big)^2\Big] \overset{(3.225)}{=} \frac{\eta^2}{f}\|B_2 + C_1^{\mathrm{T}}\|_{\mathrm{F}}. \tag{3.233}$$

The trace of the third term in the right-hand side of (3.231) satisfies

$$\mathrm{Tr}\Big[2\mathrm{Diag}\Big(S^{\mathrm{T}}\big(\begin{smallmatrix} 2\eta a & 0 \\ 0 & 0\end{smallmatrix}\big)S\Big)\mathrm{Diag}\Big(S^{\mathrm{T}}\Big(\begin{smallmatrix} 0 & \eta(B_2+C_1^{\mathrm{T}}) \\ 0 & 0\end{smallmatrix}\Big)S\Big)\Big] = 0. \tag{3.234}$$

Substituting (3.230)–(3.234) into (3.229) yields

$$\min_{\substack{\|v\|=1\\ v\in\mathcal{V}}} \mathcal{B}_2(v) = \min_{\substack{\|v\|=1\\ v\in\mathcal{V}}} 8\eta^2 a^2 + \frac{2\lambda\sigma}{f+\lambda}\|B_2 - C_1^{\mathrm{T}}\|_{\mathrm{F}}^2 + \frac{8\lambda\eta^2 a^2}{f} + \frac{2\lambda\eta^2}{f}\big\|B_2 + C_1^{\mathrm{T}}\big\|_2^2. \tag{3.235}$$

Recall now that $\eta^2 = f/(f+\lambda)\sigma$ and observe that

$$8\eta^2 + \frac{8\lambda\eta^2}{f} = \frac{8f\sigma}{\lambda+f} + \frac{8\lambda\sigma}{f+\lambda} = 8\sigma. \tag{3.236}$$

Substituting (3.236) into (3.235), we find therefore that

$$\min_{\substack{\|v\|=1\\ v\in\mathcal{V}}} \mathcal{B}_2(v) = \min_{\substack{\|v\|=1\\ v\in\mathcal{V}}} 8\sigma a^2 + \frac{2\lambda\sigma}{f+\lambda}\|B_2 - C_1^{\mathrm{T}}\|_{\mathrm{F}}^2 + \frac{2\lambda\sigma}{f+\lambda}\|B_2 + C_1^{\mathrm{T}}\|_2^2. \tag{3.237}$$

Finally, note that solutions of the optimization problem in (3.237) are subject to the constraint $2a^2 + \|B_2\|_2^2 + \|C_1\|_2^2 = 1$. By identifying $s_1 = 8\sigma$, $s_2 = s_3 = 2\lambda\sigma/(f+\lambda)$ and applying Lemma 32, see Appendix 3.E, we find that

$$\zeta_W = \min_{\substack{\|v\|=1 \\ v\in\mathcal{V}}} \mathcal{B}_2(v) = \min\left\{4\sigma, \frac{4\lambda\sigma}{f+\lambda}\right\}. \tag{3.238}$$

Replacing $\zeta_W$ in (3.217) by (3.238), we find that

$$\mathcal{H}_W^{\text{opt}} \geq \min\left\{\frac{2\sigma_1\lambda}{f+\lambda} - 2\sigma_2, 4\sigma_1, \frac{4\sigma_1\lambda}{f+\lambda}\right\} = \frac{2\sigma_1\lambda}{f+\lambda} - 2\sigma_2. \tag{3.239}$$

If $r = 1$, then $\sigma_2 = 0$. This completes the proof. □

*Proof of Proposition 13:* Finally, Proposition 13 follows directly by combining Lemmas 28, 29. □

## 3.D.7 Proof of Proposition 14 – Looking at a neighborhood of $W \in M \cap U$

A consequence of Proposition 13 is that the manifold $M$ is nondegenerate at $W \in M_b \cap M \backslash \mathrm{Sing}(M)$:

**Corollary 2.** *Suppose Assumption 9 holds. If $W \in M_b \cap M \backslash \mathrm{Sing}(M)$, then* $\ker \nabla^2 \mathcal{I}(W) = \mathrm{T}_W M$. *Furthermore, the manifold $M$ is locally nondegenerate at $W$.*

*Proof.* Recall that Proposition 13 implies that $\nabla^2\mathcal{I}(W)$ is a positive definite bilinear form when restricted to $\mathrm{T}_W^\perp M$, and that Proposition 13 provides a lower bound $\omega$ for it. This implies in particular that $\mathrm{T}_W^\perp M \subseteq (\ker \nabla^2\mathcal{I}(W))^\perp$. Because moreover $\mathrm{T}_W M \subseteq \ker \nabla^2 \mathcal{I}(W)$, we find that

$$\mathrm{T}_W M = \ker \nabla^2 \mathcal{I}(W). \tag{3.240}$$

Now, because $W$ is nonsingular by assumption and $\mathrm{Sing}(M)$ is a closed set (recall Proposition 11(a)), there exists a neighborhood $U_W \subset \mathcal{P}$ of $W$ such that for any $W' \in U_W \cap M$, $W'$ is also nonsingular in $M$. In particular $U_W \cap M$ is a submanifold of $\mathcal{P}$. Hence,

$$\dim \mathrm{T}_{W'} M = \dim \mathrm{T}_W M \tag{3.241}$$

is constant for all $W' \in U_W \cap M$.

By continuity of $\nabla^2\mathcal{I}$, we have furthermore that for any $W' \in U_W \cap M$,

$$\mathrm{rk}(\nabla^2\mathcal{I}(W')) \geq \mathrm{rk}(\nabla^2\mathcal{I}(W)) = \dim \mathrm{T}_W M. \tag{3.242}$$

Now, (i) the *rank–nullity theorem* together with (3.242) implies that

$$\dim\ker\nabla^2\mathcal{I}(W') \overset{\text{(i)}}{\leq} \dim\ker\nabla^2\mathcal{I}(W) \overset{(3.240)}{=} \dim \mathrm{T}_W M \overset{(3.241)}{=} \dim \mathrm{T}_{W'} M. \tag{3.243}$$

Since $\mathrm{T}_{W'} M \subseteq \ker \nabla^2\mathcal{I}(W')$ also, (3.243) implies that $\ker \nabla^2\mathcal{I}(W') = \mathrm{T}_{W'} M$ for any $W' \in U_W \cap M$. Hence, $M$ is locally nondegenerate at $W$ according to Definition 11. □

*Proof of Proposition 14.* Let $W \in M_b \cap M \backslash \mathrm{Sing}(M)$ and let $U_W$ be the open neighborhood from Corollary 2, where $M$ is nondegenerate at $W$.

Proposition 14(a) follows from the definition of nondegeneracy of Definition 11 and the existence of $U_W$ in Corollary 2. Proposition 14(b) follows because an immediate consequence of Proposition 14(a) is that for any $W' \in U_W \cap M$, $U_W \cap M$ is also locally nondegenerate at $W'$.

Corollary 2 yields that for any $W' \in U_W \cap M$, $\ker \nabla^2 \mathcal{I}(W') = \mathrm{T}_{W'} M$. Hence, there exists an $\omega_{W'}$ such that

$$\min_{\substack{\|v\|=1 \\ v \in T^{\perp}_{W'} M}} v^{\mathrm{T}} \nabla^2 \mathcal{I}(W') v = \omega_{W'} > 0. \tag{3.244}$$

This is in fact Proposition 14(c), and this completes the proof of Proposition 14. □

### 3.D.8 Proof of Proposition 15 – Extension in generic sense

Proposition 11 implies that $M_b$ is nonsingular for generic points. Together with Proposition 14, this implies that up to a closed algebraic set with lower dimension than that of $M_b$, for every $W \in M_b \cap M$, there exists a neighborhood of $U_W \in \mathcal{P}$ of $W$ such that $U_W \cap M$ is a manifold that is locally nondegenerate at $W$. We will now extend these results to $M$ by using the group action in (3.92).

The result of Proposition 8 implies that the action of $\pi$ extends $M_b$ to $M$ as defined in (3.92). We look at the action on the Hessian. We need to prove that:

(a) a given point $W' \in M$ is regular if the point in $M_b$ corresponding to $W' \in M$ under the group action $\pi$ in (3.29) is also regular; and

(b) $W' \in M$ is nondegenerate.

*Proof of (a).* Recall that $H = \mathrm{Diag}((\mathbb{R}^*)^f)$ as a Lie group. Proposition 8 provides the bijective map $\pi^{-1} : M \to M_b \times H$ given by

$$\pi^{-1}(W) = (\pi(C_W)(W), C_W), \tag{3.245}$$

where $C_W = \mathrm{Diag}(W_2^T W_2)^{-1/4} \mathrm{Diag}(W_1 W_1^T)^{1/4}$. From (3.151), $\pi^{-1}$ has a continuous inverse in the open set $\pi(M_b \times H)$. Recall that by Proposition 14, for each $W_b \in M_b \backslash \mathrm{Sing}(M_b)$ there is a neighborhood $Q_{W_b}$ of $W_b$ such that every $W_b' \in Q_{W_b} \cap M_b$ is nonsingular in $M_b$. In particular, for any $C \in H$, there is a neighborhood $Q_C \subset H$ of $C$ such that $\pi^{-1} : Q_{W_b} \times Q_C \to M$ is smooth. Hence, $\pi^{-1}$ is a local diffeomorphism at $(W_b, C)$ and so $M$ is smooth at $W = \pi(C)(W_b)$. Hence, $W$ is regular in $M$ whenever $W_b \in M_b$ is regular.

This implies that if $W \in M_b \cap M$ is generic, then so is $\pi(C)(W) \in M$ for any $C \in H$.

*Proof of (b).* We start by computing the effect of the action $\pi : M \times H \to M$ on the Hessian. For any fixed $C \in H$ and $W \in M$ regular, there is an induced smooth map $\mathrm{D}_{(C,W)}\pi : \mathrm{T}_W M \to \mathrm{T}_{\pi(C)(W)} M$ for $V = (V_2, V_1) \in \mathrm{T}_W M$ given by

$$\mathrm{D}_{(C,W)}\pi(V) = (V_2 C, C^{-1} V_1), \tag{3.246}$$

In vectorization notation for $V$ and denoting $\mathrm{vec}(A,B) = (\mathrm{vec}(A), \mathrm{vec}(B))$ for any $A, B$, the map $\mathrm{D}_W \pi$ is given by

$$\begin{aligned} \mathrm{vec}(\mathrm{D}_W \pi(V)) &= \mathrm{vec}((V_2 C, C^{-1} V_1)) = (\mathrm{vec}(V_2 C), \mathrm{vec}(C^{-1} V_1))) \\ &= \begin{pmatrix} C \otimes \mathrm{I}_{e\times e} & 0 \\ 0 & C^{-1} \otimes \mathrm{I}_{h\times h} \end{pmatrix} \mathrm{vec}(V_2, V_1) = \mathcal{C} \mathrm{vec}(V_2, V_1) \end{aligned} \tag{3.247}$$

say.

We next consider the Hessian of the map $\mathcal{I}(\pi(C)(\cdot)) : \mathcal{P} \to \mathbb{R}$ and compare it to $\nabla^2\mathcal{I}$. We let $\nabla(g(W))(V)$ be the differential of a function $g(W)$ depending on $W$ in the direction $V$; note that we use only Euclidean coordinates in $\mathcal{P}$ and so we can understand the differential as a gradient. First, use the chain rule to conclude that for $V \in \mathrm{T}_W\mathcal{P}$ we have

$$\nabla\big(\mathcal{I}(\pi(C)(W))\big)(V) = \nabla\mathcal{I}(\pi(C)(W))\big(\mathrm{D}_W\pi(V)\big). \tag{3.248}$$

For the Hessian $\nabla^2\big(\mathcal{I}(\pi(C)(\cdot))\big) : \mathrm{T}_W\mathcal{P} \times \mathrm{T}_W\mathcal{P} \to \mathbb{R}$, we have that similarly that for $V, R \in \mathrm{T}_W\mathcal{P}$ and $W \in M$,

$$\begin{aligned}\nabla^2\Big(\mathcal{I}(\pi(C)(W))\Big)(V,R) &= \nabla\Big(\nabla\big(\mathcal{I}(\pi(C)(W))\big)(V)\Big)(R)\\ &\overset{(3.248)}{=} \nabla\Big(\nabla\mathcal{I}(\pi(C)(W))\big(\mathrm{D}_W\pi(V)\big)\Big)(R)\\ &\overset{\text{(i)}}{=} \nabla\Big(\nabla\mathcal{I}(\pi(C)(W))\Big)(R)\big(\mathrm{D}_W\pi(V)\big) + \nabla\mathcal{I}(\pi(C)(W))\Big(\nabla(\mathrm{D}_W\pi(V))(R)\Big)\\ &\overset{\text{(ii)}}{=} \nabla^2\mathcal{I}(\pi(C)(W))\big(\mathrm{D}_W\pi(V), \mathrm{D}_W\pi(R)\big) + \nabla\mathcal{I}(\pi(C)(W))\big(\nabla(\mathrm{D}_W\pi(V))(R)\big),\end{aligned} \tag{3.249}$$

where we have (i) used Leibniz's rule in the one-to-last step and (ii) the chain rule. Since $\nabla\mathcal{I}(\pi(C)(W)) = 0$ at any minimizer $\pi(C)(W)$, we have that (3.249) reduces to

$$\nabla^2\big(\mathcal{I}(\pi(C)(W))\big)(V,R) = \nabla^2\mathcal{I}(\pi(C)(W))(\mathrm{D}_W\pi(V), \mathrm{D}_W\pi(R)). \tag{3.250}$$

We abuse now the vectorization notation from (3.247) and consider the Hessian as a bilinear form in terms of $\operatorname{vec}(V)$ and $\operatorname{vec}(R)$ in (3.250). This means specifically that (3.250) can be written as

$$\nabla^2\big(\mathcal{I}(\pi(C)(W))\big) = \mathcal{C}^{\mathrm{T}}\big(\nabla^2\mathcal{I}(\pi(C)(W))\big)\mathcal{C}. \tag{3.251}$$

Recall now that for any $C \in H$ and $W \in \mathcal{P}$, $\mathcal{I}(\pi(C)(W)) = \mathcal{I}(W)$. Consequently, as bilinear forms

$$\nabla^2\mathcal{I}(W) = \mathcal{C}^{\mathrm{T}}\nabla^2\mathcal{I}(\pi(C)(W))\mathcal{C}. \tag{3.252}$$

Finally, note that $\mathcal{C}$ in (3.252) is invertible for any $C \in H$. Therefore, we the ranks are equal:

$$\operatorname{rk}(\nabla^2\mathcal{I}(W)) = \operatorname{rk}\big(\nabla^2\mathcal{I}(\pi(C)(W))\big) \tag{3.253}$$

for any $C \in H$. Now, if $W \in M_b \cap M$ is nondegenerate (so the rank is maximal), we can repeat the arguments of Proposition 14 and in particular conclude that $\pi(C)(W)$ is a nondegenerate point in $M$.

Combining (a) and (b) implies that $M$ is nondegenerate at generic points. □

# 3.E Auxiliary statements

## 3.E.1 Inequalities pertaining to the Frobenius norm

**Lemma 30.** *The following inequalities hold:*

(a) *For any* $C \in \mathbb{R}^{a\times b}$, $D \in \mathbb{R}^{b\times a}$, *it holds that*

$$2\mathrm{Tr}[CD] \leq \|C\|_{\mathrm{F}}^2 + \|D\|_{\mathrm{F}}^2. \tag{3.254}$$

*with equality if and only if* $C = D^{\mathrm{T}}$.

(b) *For any* $A \in \mathbb{R}^{h\times f}$, $B \in \mathbb{R}^{e\times f}$, $\Lambda \in \mathbb{R}^{e\times h}$, *it holds that*

$$\mathrm{Tr}[A^{\mathrm{T}}B^{\mathrm{T}}\Lambda] \leq \frac{\sigma_1(\Lambda)}{2}\big(\mathrm{Tr}[B^{\mathrm{T}}B] + \mathrm{Tr}[AA^{\mathrm{T}}]\big). \tag{3.255}$$

(c) *For any* $B \in \mathbb{R}^{e\times f}$, *and diagonal matrix* $\Lambda \in \mathbb{R}^{e\times e}$ *with positive entries and minimal eigenvalue* $s = \min_{i=1,\dots,e}\Lambda_{ii}$, *it holds that*

$$\|B^{\mathrm{T}}\Lambda\|_{\mathrm{F}}^2 \geq s^2\|B\|_{\mathrm{F}}^2. \tag{3.256}$$

We prove the inequalities in Lemma 30 one by one.

*Proof of (a).* Recall that the Frobenius norm satisfies $\|A+B\|_{\mathrm{F}}^2 = \|A\|_{\mathrm{F}}^2 + \|B\|_{\mathrm{F}}^2 - 2\langle A,B\rangle_{\mathrm{F}}$, where $\langle A,B\rangle_{\mathrm{F}} = \mathrm{Tr}[A^{\mathrm{T}}B]$ (for real matrices) denotes the *Frobenius inner product*. We have in particular that

$$0 \leq \|C - D^{\mathrm{T}}\|_{\mathrm{F}}^2 = \|C\|_{\mathrm{F}}^2 + \|D\|_{\mathrm{F}}^2 - 2\mathrm{Tr}[CD], \tag{3.257}$$

with equality if and only if $C = D^{\mathrm{T}}$ (by property of a norm).

*Proof of (b).* Consider any square matrix $R \in \mathbb{R}^{f\times f}$ and let $\sigma_{\max}(R)$ be its spectral norm, i.e., its largest singular value. Recall that

$$\sigma_{\max}(R) = \sup\big\{\|Rx\|_2 : x \in \mathbb{R}^f, \|x\|_2 = 1\big\} = \sup\big\{\frac{x^{\mathrm{T}}R^{\mathrm{T}}Rx}{x^{\mathrm{T}}x} : x \in \mathbb{R}^f, x \neq 0\big\}^{1/2}. \tag{3.258}$$

Note that

$$\begin{aligned}\sigma_{\max}\big(\begin{smallmatrix}0 & R\\ R^{\mathrm{T}} & 0\end{smallmatrix}\big)^2 &= \sup\Big\{\|\big(\begin{smallmatrix}0 & R\\ R^{\mathrm{T}} & 0\end{smallmatrix}\big)\big(\begin{smallmatrix}a\\ b\end{smallmatrix}\big)\|_2^2 : a,b \in \mathbb{R}^f, \|a\|_2^2 + \|b\|_2^2 = 1\Big\}\\ &= \sup\Big\{\|\big(\begin{smallmatrix}Rb\\ R^{\mathrm{T}}a\end{smallmatrix}\big)\|_2^2 : a,b \in \mathbb{R}^f, \|a\|_2^2 + \|b\|_2^2 = 1\Big\}\\ &= \sup\Big\{\|R^{\mathrm{T}}a\|_2^2 + \|Rb\|_2^2 : a,b \in \mathbb{R}^f, \|a\|_2^2 + \|b\|_2^2 = 1\Big\}\\ &\leq \sigma_{\max}(R)^2, \end{aligned}\tag{3.259}$$

where the inequality follows because $\sigma_{\max}(R) = \sigma_{\max}(R^{\mathrm{T}})$ and by definition of $\sigma_{\max}(R)$, $\|Rx\|_2^2 \leq \sigma_{\max}(R)^2\|x\|_2^2$ for any $x \in \mathbb{R}^f$.

Recall the properties of the vectorization notation in Appendix 3.D.5. We have then in vectorization notation

$$\begin{aligned}\mathrm{Tr}[A^{\mathrm{T}}B^{\mathrm{T}}\Lambda] &= \frac{1}{2}\big(\mathrm{Tr}[A^{\mathrm{T}}(B^{\mathrm{T}}\Lambda)] + \mathrm{Tr}[B(A\Lambda^T)]\big)\\ &= \frac{1}{2}\big(\mathrm{vec}(A)^T\Lambda^T \otimes \mathrm{I}_{f\times f}\mathrm{vec}(B^T) + \mathrm{vec}(B^T)\Lambda \otimes \mathrm{I}_{f\times f}\mathrm{vec}(A)\big)\\ &= \big(\mathrm{vec}(A), \mathrm{vec}(B^{\mathrm{T}})\big)^{\mathrm{T}}\tfrac{1}{2}\Big(\begin{smallmatrix}0 & \Lambda^{\mathrm{T}}\otimes \mathrm{I}_{f\times f}\\ \Lambda\otimes \mathrm{I}_{f\times f} & 0\end{smallmatrix}\Big)\big(\mathrm{vec}(A), \mathrm{vec}(B^{\mathrm{T}})\big)\\ &\leq \frac{\sigma_{\max}(\Lambda)}{2}\big(\mathrm{Tr}[AA^{\mathrm{T}}] + \mathrm{Tr}[B^{\mathrm{T}}B]\big), \end{aligned}\tag{3.260}$$

where we have used that $\sigma_{\max}(Y \otimes I) = \sigma_{\max}(Y)$.

*Proof of (c).* Suppose without loss of generality that the diagonal elements of $\Lambda$ are ordered, i.e., $\Lambda_{11} \geq \ldots \geq \Lambda_{ee} > 0$. Denote the columns of $B$ by $B_{\cdot 1}, \ldots, B_{\cdot e}$. Calculating the Frobenius norm directly, we find that

$$\|B^{\mathrm{T}}\Lambda\|_{\mathrm{F}}^2 = \sum_{j=1}^{e} \Lambda_j^2 \|B_{\cdot j}\|_2^2 \geq \Lambda_e^2 \sum_{j=1}^{e} \|B_{\cdot j}\|_2^2 = \Lambda_e^2 \|B\|_{\mathrm{F}}^2. \tag{3.261}$$

This completes the proof of Lemma 30. □

## 3.E.2 Subspace minimization

**Lemma 31.** *Let $\mathcal{V}_1$, $\mathcal{V}_2$ be two orthogonal subspaces, and let $\{v_1, \ldots, v_d\}$ be an orthonormal basis of $\mathcal{V}_1 \oplus \mathcal{V}_2$ such that $\mathcal{V}_1 = \operatorname{span}\{v_1, \ldots, v_s\}$ and $\mathcal{V}_2 = \operatorname{span}\{v_{s+1}, \ldots, v_d\}$. Assume that $l_1, \ldots, l_s \in (0, \infty)$, and let $\mathcal{B}_2 : \mathcal{V}_2 \to [0, \infty)$ be a function that satisfies $\mathcal{B}_2(\zeta u_2) = \zeta^2 \mathcal{B}_2(u_2)$ for $\zeta \in \mathbb{R}$. Then,*

$$\min_{\substack{\|u_1\|_{\mathrm{F}}^2 + \|u_2\|_{\mathrm{F}}^2 = 1 \\ u_1 \in \mathcal{V}_1, u_2 \in \mathcal{V}_2}} \sum_{i=1}^{s} l_i |\langle v_i, u_1 \rangle|^2 + \mathcal{B}_2(u_2) \geq \min\Big\{l_1, \ldots, l_s, \min_{\substack{\|u_2\|_{\mathrm{F}}^2 = 1 \\ u_2 \in \mathcal{V}_1}} \mathcal{B}_2(u_2)\Big\}. \tag{3.262}$$

*Proof.* Note that

$$\begin{aligned} &\{(u_1, u_2) \in \mathcal{V}_1 \times \mathcal{V}_2 : \|u_1\|_{\mathrm{F}}^2 + \|u_2\|_{\mathrm{F}}^2 = 1\} \\ &= \cup_{\zeta \in [0,1]} \{(u_1, u_2) \in \mathcal{V}_1 \times \mathcal{V}_2 : \|u_1\|_{\mathrm{F}}^2 = \zeta^2, \|u_2\|_{\mathrm{F}}^2 = 1 - \zeta^2\} \\ &= \cup_{\zeta \in [0,1]} \{(\zeta w_1, \sqrt{1-\zeta^2} w_2) : (w_1, w_2) \in \mathcal{V}_1 \times \mathcal{V}_2, \|w_1\|_{\mathrm{F}}^2 = 1, \|w_2\|_{\mathrm{F}}^2 = 1\}. \end{aligned} \tag{3.263}$$

The left-hand side of (3.262) therefore equals

$$\min_{\substack{\|w_1\|_{\mathrm{F}}^2 = 1, \|w_2\|_{\mathrm{F}}^2 = 1, \\ w_1 \in \mathcal{V}_1, w_2 \in \mathcal{V}_2, \zeta \in [0,1]}} \zeta^2 \sum_{i=1}^{s} l_i |\langle v_i, w_1 \rangle|^2 + (1 - \zeta^2) \mathcal{B}_2(w_2). \tag{3.264}$$

Observe in (3.264) a convex combination in terms of $\zeta^2$. The minimum of (3.264) therefore occurs at either $\zeta = 0$ or $\zeta = 1$. Note additionally that if $\|w_1\|_{\mathrm{F}} = 1$, then

$$\sum_{i=1}^{s} l_i |\langle v_i, w_1 \rangle|^2 \overset{\text{(i)}}{\geq} \min\{l_1, \ldots, l_s\} \sum_{i=1}^{s} |\langle v_i, w_1 \rangle|^2 \overset{\text{(ii)}}{=} \min\{l_1, \ldots, l_s\} \tag{3.265}$$

by (i) strict positivity of the summands and (ii) an application of Parseval's identity—which is warranted since $\{v_1, \ldots, v_s\}$ is an orthonormal basis of $\mathcal{V}_1$. Together, this proves the lower bound for the right-hand side in (3.262). □

## 3.E.3 Minimization

**Lemma 32.** *For $a \in \mathbb{R}$, $B, C \in \mathbb{R}^{f-1}$ and $s_1, s_2, s_3 > 0$,*

$$\min_{\substack{a, B, C \\ 2a^2 + \|B\|_2^2 + \|C\|_{\mathrm{F}}^2 = 1}} \Big\{ s_1 a^2 + s_2 \|B - C\|_{\mathrm{F}}^2 + s_3 \, \|B + C\|_2^2 \Big\} = \min\Big\{\frac{s_1}{2}, 2s_2, 2s_3\Big\}. \tag{3.266}$$

*Proof.* We can decouple the minimization over $a$ and over $(B,C)$ in (3.266), respectively. To see this, suppose that $(a_0,B_0,C_0)$ is a minimizer of (3.266). If so, then $(B_0,C_0)$ must also be a minimizer of

$$\min_{\substack{B,C\\ \|B\|_2^2+\|C\|_2^2=1-2a_0^2}} \quad s_2\|B-C\|_{\mathrm{F}}^2+s_3\|B+C\|_{\mathrm{F}}^2, \tag{3.267}$$

for otherwise $(a_0,B_0,C_0)$ would not be a minimizer of (3.266) by linearity.

For fixed $a_0$, the following holds:

- if $s_2>s_3$, then the minimizer $(B_0,C_0)$ of (3.267) satisfies $B_0=C_0$ and the minimum is $4s_3\|B_0\|_{\mathrm{F}}^2=2s_3(1-2a_0^2)$;
- if $s_2<s_3$, then the minimizer $(B_0,C_0)$ of (3.267) satisfies $B_0=-C_0$ and the minimum is $4s_2\|B_0\|_{\mathrm{F}}^2=2s_2(1-2a_0^2)$;
- if $s_2=s_3$, then any point $(B_0,C_0)$ that satisfies $\|B_0\|_2^2+\|C_0\|_2^2=1-2a_0^2$ is a minimizer of (3.267) by the parallelogram law, and the minimum is in fact $2s_2(1-2a_0^2)$.

Thus, we have that the left-hand side of (3.266) reduces to:

- if $s_2>s_3$, then $\min_{a\in[-1/\sqrt{2},1/\sqrt{2}]}\{s_1a^2+2s_3(1-2a^2)\}=\min\{s_1/2,2s_3\}$;
- if $s_2<s_3$, then $\min_{a\in[-1/\sqrt{2},1/\sqrt{2}]}\{s_1a^2+2s_2(1-2a^2)\}=\min\{s_1/2,2s_2\}$;
- if $s_2=s_3$, then $\min_{a\in[-1/\sqrt{2},1/\sqrt{2}]}\{s_1a^2+2s_2(1-2a^2)\}=\min\{s_1/2,2s_2\}$.

Combining cases, we observe that the left-hand side of (3.266) equals $\min\{s_1/2,2s_2,2s_3\}$. □

# Chapter 4

# Universal approximation of dropout neural networks

Based on [5]:
*"Universal approximation of dropout neural networks"*
by O.A. Manita, M.A. Peletier, J.W. Portegies, J. Sanders, and A. Senen–Cerda

In Chapters 2 and 3 we have analyzed dropout from the stochastic optimization perspective and examined convergence properties of the algorithm. In this chapter we consider dropout and in particular the randomness created by dropout in a more abstract manner. We will investigate if Neural Networks (NNs) with the additional randomness from dropout can still approximate functions and if so, in which sense.

## 4.1 Introduction

The class of NNs satisfies a well-known *universal approximation property*: any given function can be approximated to arbitrary precision by a NN [156, 153]. This property partially explains why NNs are effective as approximators of implicitly given functions.

We have seen in Chapters 3 and 4 that dropout converts a deterministic NN into a random one while training by randomly 'dropping' nodes. In this chapter we address the following question: Does this randomness interfere with the universal approximation property? Or, to formulate it in the affirmative: does the class of dropout NNs still satisfy a universal approximation property?

To provide a first quantification of this question, let us explain the expectation–variance split, which in the context of dropout goes back to the theoretical analysis by [114]. We will think of a dropout NN as a function $\Psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ together with a $\{0,1\}^n$-valued random variable $f$.[1] Here, we will think of $\mathbb{R}^d$ as the data space, and $\mathbb{R}^n$ as the parameter space (the space of weights and biases of the NN). The parameters of

[1] Differently from Chapter 3 where $f$ denotes the width of the NN, in this chapter we let $f$ denote a vector of *filters*.

the NN are multiplied componentwise with the vector of filter variables $f$. That means that when $\zeta : \mathbb{R}^d \to \mathbb{R}$ is a function we want to approximate, we try to approximate it with the stochastic function that maps $x$ to $\Psi(x, w \odot f)$. For fixed $x$ and $w$, the expectation–variance split reads

$$\mathbb{E}\left[\big(\Psi(x,w\odot f)-\zeta(x)\big)^2\right] = \big(\mathbb{E}[\Psi(x,w\odot f)]-\zeta(x)\big)^2 + \mathrm{Var}[\Psi(x,w\odot f)]. \tag{4.1}$$

As both terms on the right-hand side of (4.1) are nonnegative, both terms have to be small in order to have a good approximation.

In [22], Foong et al. showed that deep Rectified Linear Unit (ReLU) NNs with node-dropout can still approximate functions arbitrarily well, by showing that both the expectation term and the variance term in the expectation–variance split can be made arbitrarily small (see [22, Theorem 3]). In fact, the two terms are arbitrarily small uniformly over $x$ in the unit cube in $\mathbb{R}^d$. With this statement, Foong et al. effectively showed a universal-approximation result.

In this chapter we show two universal-approximation results for wider classes of dropout NNs. Where Foong et al. made specific use of the ReLU activation, assume Bernoulli filter variables (thus equidistributed, independent, and with finite variance), and restrict to one hidden layer, we show that the property of universal approximation holds under more general assumptions. Our distinguishing insight is that certain classes of random NNs satisfy an algebraic property, which enables us to deal with arbitrary depth, generic activation functions, and dependent filter variables not necessarily equidistributed and possibly with infinite variance. Notably, our techniques allow for dropout of edges from the input layer.

For the theorems we prove below the structural assumptions on the NN reduce to the assumption that the underlying deterministic NN has the universal-approximation property; necessary and sufficient conditions for the latter to hold are well-known, for example, see [153]. In addition, our main theorems allow for general classes of filters, including the original node-based dropout [117], the edge-based dropconnect [115], and many others, including sets of filters with strong dependence. We show that the class of dropout NNs can *exactly* match a given deterministic NN, at least in expectation as shown in Corollary 3 below. We also show that we can construct NNs that approximate a given function arbitrarily well, both as a random NN and as a deterministic NN (see Corollaries 4 and 5). Finally, we provide control over the error both in probability and in $L^q$.

### 4.1.1 Approximation by random neural networks

In a deterministic context, a universal-approximation theorem for some class $\mathsf{C}$ is a density statement, stating that any function $\zeta$ can be approximated to arbitrary precision by NNs in $\mathsf{C}$, where the approximation is measured in some seminormed function space $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$. Such approximation statements can be generalized in a stochastic context in multiple ways. We will namely focus on two of these: approximation in probability, and in $L^q$ for $q \in [1,\infty)$.

Universal approximation in probability in a function space $\mathcal{F}$ is the property that for every function $\zeta \in \mathcal{F}$, and every $\epsilon > 0$, there exists a NN $\Psi$, a weight vector $w$ and a random vector $f$ (all with certain extra properties to make the statement nontrivial), such that[2]

$$\mathbb{P}[\|\zeta(\cdot) - \Psi(\cdot, w \odot f)\|_{\mathcal{F}} > \epsilon] < \epsilon.$$

A stronger approximation statement involves approximation in $L^q$ for $q \in [1, \infty)$: for any $\epsilon > 0$, there exist $\Psi$, $w$ and $f$ such that

$$\mathbb{E}\left[\|\zeta(\cdot) - \Psi(\cdot, w \odot f)\|_{\mathcal{F}}^q\right]^{\frac{1}{q}} < \epsilon.$$

We prove two main classes of approximation results, corresponding to the two main ways that dropout NNs can be used in practice. In the first class of results, the NN is a random object as described above, and is used in a random fashion during training as well as for prediction; we call this *random-approximation* dropout.[3] In the second class of results, the training is conducted with random NNs of the form $\Psi(\cdot, w \odot f)$, but the deterministic network $\Psi(\cdot, w \odot \mathbb{E}[f])$ is used for prediction instead. In this latter case, the filters are thus replaced by their expectations. We call this type of dropout *expectation-replacement.*

## 4.1.2 Random-approximation

We start with uniform *random-approximation* dropout, that is the property that any function $\zeta$ in an appropriate set $\mathcal{F}$ can be approximated by random NNs of the form $\Psi(\cdot, w \odot f)$. The first result in this direction states that there exist real constants $\{a_U\}_U$ such that

$$\mathbb{E}\left[\sum_{U \in 2^{[n]}} a_U \Psi\big(\cdot, (w \odot \mathbf{1}_U) \odot f^U\big)\right] = \Psi(\cdot, w). \tag{4.2}$$

See Theorem 13 in Chapter 4 for the full statement. (4.2) establishes the following fact: for *any* collection of random filter variables $f^U$, for *any* function $\Psi$, for *any* parameter point $w$, the function $\Psi(\cdot, w)$ can be matched exactly by the expectation of a sum of filtered versions of the same function. The important caveat is that one needs to take into account all reduced versions of the functions $\Psi$, i.e., the whole hierarchy of deterministically modified versions indexed by subsets $U$.

The equality in (4.2) suggests a special role for ‘classes of networks’, with the property that given a ‘network’ $\Psi(\cdot, w)$ we can in some sense define a new (random) network $\tilde{\Psi}(\cdot, \tilde{w} \odot \tilde{f})$ by

$$\tilde{\Psi}(\cdot, \tilde{w} \odot \tilde{f}) := \sum_{U \in 2^{[n]}} a_U \Psi(\cdot, w \odot \mathbf{1}_U \odot f^U). \tag{4.3}$$

To formalize this, we assume that we have chosen a set $\mathsf{DDNN}$ (a ‘set of random NNs’), which can be any collection of tuples $(n, \Psi, f)$ that satisfy the following properties:

(i) $n \in \mathbb{N}$ is a natural number;

(ii) $\Psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ is a function such that for every $w \in \mathbb{R}^n$, $\Psi(\cdot, w) \in \mathcal{F}$;

[2] For convenience, in this chapter we use $\epsilon$ for both the event $\{\|\zeta(\cdot) - \Psi(\cdot, w \odot f)\|_{\mathcal{F}} > \epsilon\}$ and a bound of its probability. An equivalent definition with, e.g. $\epsilon, \eta > 0$ for each term may also be used.

[3] This has also been called *Monte Carlo dropout* because of the close connection with Monte Carlo estimation of integrals [23].

(iii) $f$ is a $\{0,1\}^n$-valued random variable such that

$$\mathbb{P}[f = (1,\ldots,1)] > 0. \tag{4.4}$$

Moreover, we assume that DDNN is closed under linear, independent combinations. By this we mean that whenever $a,b \in \mathbb{R}$ and $(m,\Phi,f)$ and $(n,\Psi,g)$ are in DDNN, then also $(n',a\Phi+b\Psi,h) \in \mathsf{DDNN}$ with $n' \geq n+m$, where $h$ is an $\{0,1\}^{n'}$-valued random variable that is the independent concatenation of $f$ and $g$. In this case, $a\Phi+b\Psi : \mathbb{R}^d \times \mathbb{R}^{n'} \to \mathbb{R}$ is given by

$$(x,(w_1,w_2)) \mapsto a\Phi(x,w_1)+b\Psi(x,w_2).$$

This closure assumption implies that a definition of the form (4.3) is meaningful.

The range of possible subclasses in DDNN satisfying these requirements is vast. Typical examples are NNs with dropout and *dropconnect*, but many other choices also are possible. Note that the function $\Psi$ may be extremely general, implying that there are no restrictions on e.g. the form of the activation function or the structure of the NN. In fact, nothing in the requirements on DDNN restricts to functions $\Psi$ generated by NNs; other approximation methodologies may also be used, for instance based on Fourier or wavelet expansions. See Section 4.2 for a detailed description and examples in the class DDNN.

By combining a result of the type in (4.2) with the law of large numbers we find Corollary 3 below, which expresses the following insight: if the class DDNN is rich enough to approximate any function in $\mathcal{F}$ when all filter variables are set to 1, then any function in $\mathcal{F}$ can also be approximated by a (random) dropout NN in DDNN.

**Corollary 3.** *Let $\zeta \in \mathcal{F}$ and $\epsilon > 0$. Assume there exists a $(m,\Phi,g) \in \mathsf{DDNN}$ and a $v \in \mathbb{R}^m$ such that $\|\Phi(\cdot,v)-\zeta\|_{\mathcal{F}} < \epsilon/2$. Then there exist a $(n,\Psi,f) \in \mathsf{DDNN}$ and a $w \in \mathbb{R}^n$ such that*

$$\mathbb{P}\left[\|\Psi(\cdot,w\odot f)-\zeta\|_{\mathcal{F}} > \epsilon\right] < \epsilon \tag{4.5}$$

*and*

$$\mathbb{E}\left[\|\Psi(\cdot,w\odot f)-\zeta\|_{\mathcal{F}}^q\right]^{\frac{1}{q}} < \epsilon.$$

Section 4.4 is devoted to these results, but develops them in more generality. There we also give some examples and calculate the coefficients $a_U$ explicitly for the case of independent Bernoulli filters.

### 4.1.3 Expectation-replacement

In the previous subsection we considered a dropout NN to be a random object, which is also used as such during prediction. By contrast, it is common practice to choose the filter variables to be *random* during training and to be *deterministic* during prediction and equal to their expectations; see e.g., Goodfellow, Bengio, and Courville [74, Sec. 7.12]. We call this *expectation-replacement* dropout, and Corollary 3 above does not say anything about this situation.

In fact, we show with an example that the construction at the heart of Corollary 3 may lead to NNs that are 'bad approximators' in this specific sense: given a function $\zeta$, the constructed NNs approximate $\zeta$ with high probability with random filters, but do not approximate $\zeta$ at all when replacing the filters by their expectations (see Example 19 further in this chapter).

At the same time, expectation-replacement dropout is both very widespread and very successful; see [41]. How can these two observations be reconciled?

To address this, we describe in Section 4.4 the construction of dropout NNs that approximate not only in probability and in $L^q$, but also in this expectation-replacement sense. As in the case of Corollary 3, the construction builds on existing density results for deterministic NNs: we start with a given deterministic NN $\Psi(\cdot, w)$ that is close to a given function $\zeta$. Differently from Corollary 3, however, the nonlinearity of $\Psi$ forces us to apply the law of large numbers to each edge (or weight in this context) separately, instead of simultaneously for the whole NN.

The main result pertaining to expectation-replacement dropout allows for a wide range of choices of activation functions and filter-variable distributions. For example, the following are simple, specific corollaries for a ReLU activation function with dropconnect and node-dropout respectively.

**Corollary 4.** *Let $\mathcal{F}$ be the space of continuous functions $\mathbb{R}^d \to \mathbb{R}$, and endow it with a seminorm $\|\cdot\|_{\mathcal{F}}$ equal to supremum of the function on the unit cube. Then for every $\zeta \in \mathcal{F}$ and every $\epsilon > 0$ there exists a dropconnect ReLU NN $(\Psi, f)$ and a parameter vector $w$ such that*

$$\mathbb{P}\left[\left\|\Psi(\cdot, w \odot f) - \zeta\right\|_{\mathcal{F}} > \epsilon\right] < \epsilon \tag{4.6}$$

*and*

$$\mathbb{E}\left[\|\Psi(\cdot, w \odot f) - \zeta\|_{\mathcal{F}}^q\right]^{\frac{1}{q}} < \epsilon,$$

*while*

$$\|\Psi(\cdot, w \odot \mathbb{E}[f]) - \zeta\|_{\mathcal{F}} < \epsilon.$$

**Corollary 5.** *Corollary 4 also holds when using dropout instead of dropconnect.*

Note that where the construction of the section on random-approximation dropout is applied to a very wide class of functions $\Psi$—not only those generated by NNs, the construction underlying Corollaries 4 and 5 depends in a detailed manner on the fact that $\Psi$ has the structure of a NN.

In this chapter we show that dropout NNs have sufficient representational capacity to approximate well simultaneously in probability, in $L^q$, *and* in the expectation-replacement sense. While this does not explain why any given training algorithm finds parameter points that approximate well in the expectation-replacement sense, at least it shows that the contrast between random training and deterministic prediction is not an obstacle to good performance.

## Structure of this chapter

The following section is devoted to the definition of of a dropout NN. In Section 4.3 we show universal approximation results for random-approximation dropout, whereas Section 4.4 is devoted to universal approximation results for expectation-replacement dropout. We discuss our results and their limitations in Section 4.5 and conclude in Section 4.6.

## 4.2 Specification of dropout neural networks

We consider first general functions $\Psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ together with a $\{0,1\}^n$-valued random variable $f$. Some of the results will hold in such general setting. Later on, we will specify functions $\Psi$ that arise from a NN. We specify now the NN structure that we will use and introduce the corresponding notation.

### 4.2.1 Neural networks

We briefly recall that a (feedforward) NN is a function $\Psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$, where

$$\Psi(\cdot, w) = \Psi_L\left(\cdot, w^{(L)}\right) \circ \Psi_{L-1}\left(\cdot, w^{(L-1)}\right) \circ \cdots \circ \Psi_1\left(\cdot, w^{(1)}\right). \tag{4.7}$$

Here, $L$ is the depth, the parameter $w$ is the concatenation of the individual parameter vectors $w^{(j)} = (W^{(j)}, b^{(j)})$ for $j = [L]$, which in turn consist of a $d_j \times d_{j-1}$ *weight matrix* $W^{(j)}$ and a bias vector $b^{(j)} \in \mathbb{R}^{d_j}$. We set $d_0 = d$ and $d_L = 1$.

In (4.7) every $\Psi_j$ is a function from $\mathbb{R}^{d_{j-1}}$ to $\mathbb{R}^{d_j}$ given by

$$\Psi_j(x, w^{(j)}) := \sigma_j\left(W^{(j)}x + b^{(j)}\right), \tag{4.8}$$

where the function $\sigma_j : \mathbb{R} \to \mathbb{R}$ is the *activation function*, which is applied componentwise. This definition differs compared to Chapter 2 since we include biases and possibly different activation functions. These, however, will become important in Section 4.4.

### 4.2.2 Dropout neural networks

A dropout NN consists of a NN $\Psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ as above together with a random vector $f \in \{0,1\}^n$. The components of $f$ are called filter variables. The NN $\Psi$, the filter variables $f$, and a parameter vector $w \in \mathbb{R}^n$ together form a stochastic function from $\mathbb{R}^d$ to $\mathbb{R}$ given by

$$x \mapsto \Psi(x, w \odot f).$$

The most well-known examples of dropout NNs are NNs with dropout and dropconnect, which we have also defined in Chapters 2 and 3, albeit in different contexts. We briefly recall their formal definitions.

The original version of dropout [117], which in this chapter we will call from this point on *node-dropout* —to differentiate from more general versions of dropout—can be understood in the notation of (4.8) as follows. For any $j = 1, \dots, L$, choose dropout probabilities $1 - p^j$, and let $f_1^j, \dots, f_{d_j}^j$ be independent Bernoulli filters with remain probability $p^j$. Let $D_j \in \mathbb{R}^{d_j \times d_j}$ be the diagonal matrix with entries $f_1^j, \dots, f_{d_j}^j$ in the diagonal. If we arrange all nodes per block, then node-dropout implements for $j = 1, \dots, L$,

$$\Psi_j(\cdot, w^{(j)} \odot f^{(j)}) = \sigma_j\left(W^{(j)} D^j(\cdot) + b^{(j)}\right). \tag{4.9}$$

Note in this definition that if $p^1 < 1$ then with positive probability an input is masked. For this reason we call the case $p^1 < 1$ node-dropout *with* dropout on the inputs. We call the case $p^1 = 1$ node-dropout *without* dropout on the inputs.

Similarly, in the notation of (4.8) we can understand *dropconnect* as follows. For $j = 1,\dots,L$, let $F^{(j)} \in \mathbb{R}^{d_{j+1}\times d_j}$ be random matrices composed of entries $(F^j)_{ik}$, all of which are mutually independent Bernoulli random variables with the same success probability $1-p$. Dropconnect then implements for $j = 1,\dots,L$,

$$\Psi_j(\cdot, w^{(j)} \odot f^{(j)}) = \sigma_j\left((W^{(j)} \odot F^{(j)})(\cdot) + b^{(j)}\right). \tag{4.10}$$

## 4.3 Universal approximation for random-approximation dropout

The aim of this section is to derive the abstract universal approximation statement for random-approximation dropout already mentioned in the introduction (Corollary 3).

At the highest level the proof strategy is the same as in Foong et al. [22], and consists of the following three steps. Given a function $\zeta \in \mathcal{F}$ to be approximated:

1. Approximate $\zeta$ by a NN $\Psi(\cdot,w)$ using classical deterministic universal approximation results (e.g., [153]);
2. Use $\Psi(\cdot,w)$ to construct a larger, random dropout NN $\tilde{\Psi}(\cdot,\tilde{w}\odot\tilde{f})$ that matches $\Psi(\cdot,w)$ in expectation;
3. Construct an even larger random NN $\widehat{\Psi}(\cdot,\widehat{w}\odot\widehat{f})$ consisting of many independent copies of the network $\tilde{\Psi}(\cdot,\tilde{w}\odot\tilde{f})$ to obtain an approximation of $\zeta$ that is close in expectation and also has small variance.

We consider Step 1 as given by existing results, and Step 3 is a standard procedure. The novelty of this chapter for *random-approximation* lies in Step 2, which we describe in the rest of this section.

Step 2 is based on an algebraic property, which is illustrated by the following simpler version of the central theorem. We write $2^{[n]}$ for the collection of subsets of $[n] = \{1,\dots,n\}$, and for any such a subset $U$, we write $\mathbf{1}_U \in \{0,1\}^n$ for the vector with entries $(1_{j\in U})_{j\in[n]}$.

**Theorem 13.** *Let $\Psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ be any given function. Let $(f^U)_{U\in 2^{[n]}}$ be a collection of $\{0,1\}^n$-valued random variables indexed by subsets $U \in 2^{[n]}$ such that for every $U$*

$$\mathbb{P}[f^U = (1,\dots,1)] > 0.$$

*Then there exist constants $(a_U)_{U\in 2^{[n]}}$, independent of $w$, such that for all $w$,*

$$\mathbb{E}\left[\sum_{U\in 2^{[n]}} a_U \Psi\left(\cdot,(w\odot \mathbf{1}_U)\odot f^U\right)\right] = \Psi(\cdot,w). \tag{4.11}$$

This theorem should be read as follows. The right-hand side in (4.11) plays the role of a deterministic function that we want to approximate. The left-hand side is the expectation of a linear combination of many copies of $\Psi(\cdot,w)$. Each copy has two ‘dropout’ modifications: the vector $\mathbf{1}_U$ implements a deterministic dropout, and the random filter variables $f^U$ a stochastic one. With a view to generality, the random filter vector $f^U$ is allowed to be a different random vector for each subset $U$ of edges, but note that the distribution of $f^U$ on $\{0,1\}^n$ can be completely unrelated to the subset $U \subset [n]$; the subset $U$ only serves as label.

### 4.3.1 Key approximation result

Theorem 13 can be extended together with a convergence statement to yield the following theorem.

**Theorem 14.** *Let $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$ be a seminormed vector space of functions from $\mathbb{R}^d$ to $\mathbb{R}$. Let $\Psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ be a given function such that $\Psi(\cdot, w) \in \mathcal{F}$ for every $w \in \mathbb{R}^n$. Let $(f^U)_{U \in 2^{[n]}}$ be a collection of $\{0,1\}^n$-valued random variables indexed by subsets $U \in 2^{[n]}$, such that for every $U$*

$$\mathbb{P}[f^U = (1, \dots, 1)] > 0. \tag{4.12}$$

*Then there exist constants $(a_U)_{U \in 2^{[n]}}$ independent of $w$ such that*

$$\mathbb{E}\left[\sum_{U \in 2^{[n]}} a_U \Psi(\cdot, (w \odot \mathbf{1}_U) \odot f^U)\right] = \Psi(\cdot, w). \tag{4.13}$$

*In particular, by the weak law of large numbers, if $f^{i,U}$ are independent copies of $f^U$, then as $M \to \infty$,*

$$\frac{1}{M} \sum_{i=1}^{M} \sum_{U \in 2^{[n]}} a_U \Psi(\cdot, (w \odot \mathbf{1}_U) \odot f^{i,U}) \to \Psi(\cdot, w) \tag{4.14}$$

*in probability in $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$ and in $L^q$ for every $q \in [1, \infty)$.*

A proof of Theorem 14 can be found in Appendix 4.A.1. The main observation in Theorem 14 is the existence of the constants $(a_U)_{U \in 2^{[n]}}$. This purely algebraic statement follows by induction, as explained by Lemma 37 in Appendix 4.A.1. From Theorem 14, it follows that one can see a dropout NN as a linear combination of dropout NNs with weights $(w \odot \mathbf{1}_U)_{U \in 2^{[n]}}$, such that the linear combination equals the original NN in expectation as shown in (4.13).

To get a dropout NN that is close to the original network in probability, in (4.14) one makes a large average of independent copies of the dropout network that approximates the original network in expectation. The convergence in probability of (4.14) follows then from the weak law of large numbers. The convergence in $L^q$ finally follows because the sum in (4.14) is also uniformly bounded in $\mathcal{F}$ for any realization of the filter variables $f^{i,U}$, so that the convergence in probability immediately implies the convergence in $L^q$ by dominated convergence.

Note that the number of parameters in this construction increases exponentially, which limits the potential applicability of this theorem. On the other hand, for particular cases the coefficients can be calculated explicitly, as will be shown in Section  4.3.4.

### 4.3.2 Examples

We further illustrate the construction of Theorem 14 with the following examples:

**Example 15** (One-hidden-layer dropconnect NNs)**.** *Consider the function $\Psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ given by*

$$\Psi(x, w) := \sum_{j=1}^{N} c^j \sigma(w^j x + b^j), \tag{4.15}$$

*where the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is continuous with $\sigma(x) \to 0$ as $x \to -\infty$ and $\sigma(x) \to 1$ as $x \to \infty$. In* (4.15) *we have biases $b_j \in \mathbb{R}$ and weights made up from the constants $c^j \in \mathbb{R}$ and the $1 \times d$-matrices $w^j$.*

*The well-known result by [156] implies that the class of all such functions is dense in $C([0,1]^d)$ endowed with the supremum norm. An example of an approximation by functions in* (4.15) *is depicted in Figure 4.3.1.*

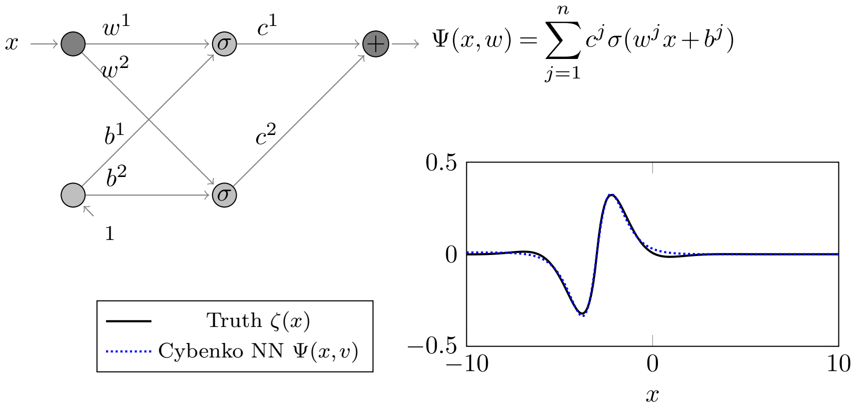


*Figure 4.3.1: A Cybenko NN as in* (4.15)*, trained to approximate a function $\zeta$; here, $n = 2, d = 1,$ and $\zeta(x) = \sin(x+3)\exp|x-3|$.*

*We suppose that the distribution of the filters follows the case of dropconnect, as described in Section 4.2.2. Theorem 14 directly yields that by choosing appropriate weights $c^{j,U}$ and weight matrices $w^{j,U}$, the one-hidden-layer dropconnect NN given by*

$$\frac{1}{M} \sum_{i=1}^{M} \sum_{U \in 2^{[n]}} \sum_{j=1}^{N} a_U c^{j,U} g^{j,U} \sigma\left((w^{j,U} \odot f^{j,U})x + b^j\right) \tag{4.16}$$

*can be chosen to be close to $\Psi$ in $L^q$ for large $M$. Here $g^{j,U}$ are independent Bernouilli random variables, and $f^{j,U}$ are random vectors with independent Bernoulli-distributed components, all with success probability $1-p$. This result is illustrated by Figures 4.3.2 and 4.3.3, where for simplicity we have used filters only on the weights $w^{j,U}$, while leaving the biases $b^j$ and $c^j$ with constant filters* 1. *Figure 4.3.2 shows a single realization of the NN in* (4.13) *with dropconnect while in Figure 4.3.3 a 'blow up'—the average of $M$ independent copies of the network in* (4.16)*—of the previous construction is depicted.*

In a similar way, we can also consider more general dropconnect networks.

**Example 16** (Dropconnect networks)**.** *Consider a deep NN $\Psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ as introduced in* (4.7) *with dropconnect filters as described in Section 4.2.2. Here, the filter variables, i.e., the components of $f^{i,U}$ in* (4.7)*, are independent Bernoulli distributed with success probability $1-p$ if they multiply elements of the weight matrices $W^{(j)}$, and are equal to* 1 *if they multiply biases $b^{(j)}$.*

*Let $w \in \mathbb{R}^n$. We choose for $\mathcal{F}$ the vector space of continuous functions on $\mathbb{R}^d$, endowed with the supremum seminorm over the closed unit cube. Then the dropconnect random network in* (4.14) *is close to the network $\Psi(\cdot, w)$ in $L^q$ for large $M$.*

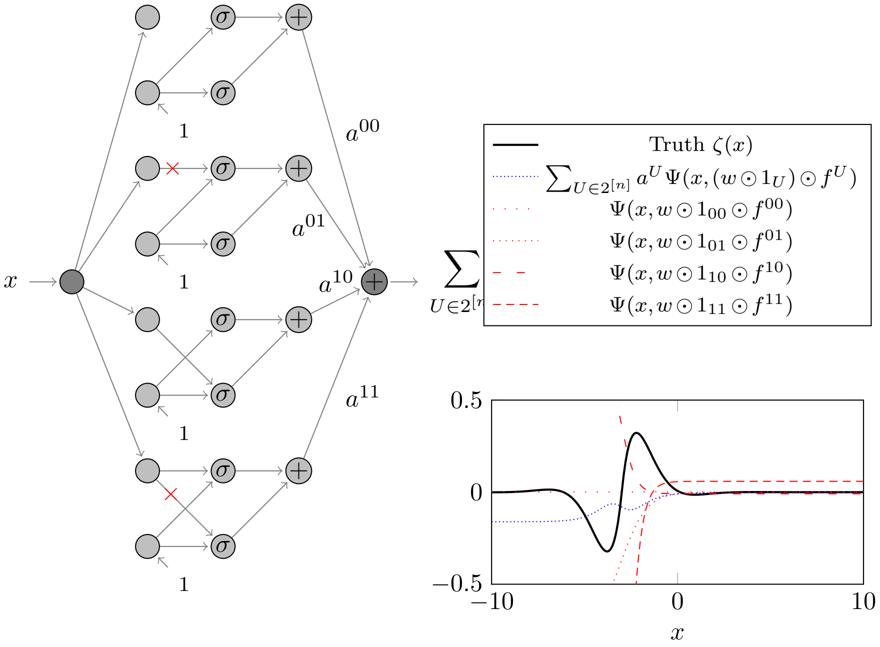

$x$
$\sigma$
$+$
1
1
1
1
$a^{00}$
$a^{01}$
$a^{10}$
$a^{11}$
$\sum_{U\in 2^{[n]}}$
Truth $\zeta(x)$
$\sum_{U\in 2^{[n]}} a^U \Psi(x,(w\odot 1_U)\odot f^U)$
$\Psi(x, w\odot 1_{00}\odot f^{00})$
$\Psi(x, w\odot 1_{01}\odot f^{01})$
$\Psi(x, w\odot 1_{10}\odot f^{10})$
$\Psi(x, w\odot 1_{11}\odot f^{11})$
0.5
0
$-0.5$
$-10$
0
10
$x$

*Figure 4.3.2: A single realization of the random NN in* (4.13) *using Dropconnect. Based on the trained Cybenko NN in Figure 4.3.1, for simplicity, we only apply dropout to the weights* $w^j$ *of* (4.15)*, which we denote by* $w$ *and which correspond to the edges joining the nodes connected to the input* $x$*. With* $n = 2$ *and* $d = 1$*, there are four different random NNs with their respective independent filters. All of them use as base* $\Psi(\cdot, w)$ *in Figure 4.3.1. In this realization, some of the edges are filtered, which are depicted with red crosses. The explicit coefficients* $a_U$ *used for dropconnect are computed in Section 4.3.4.*

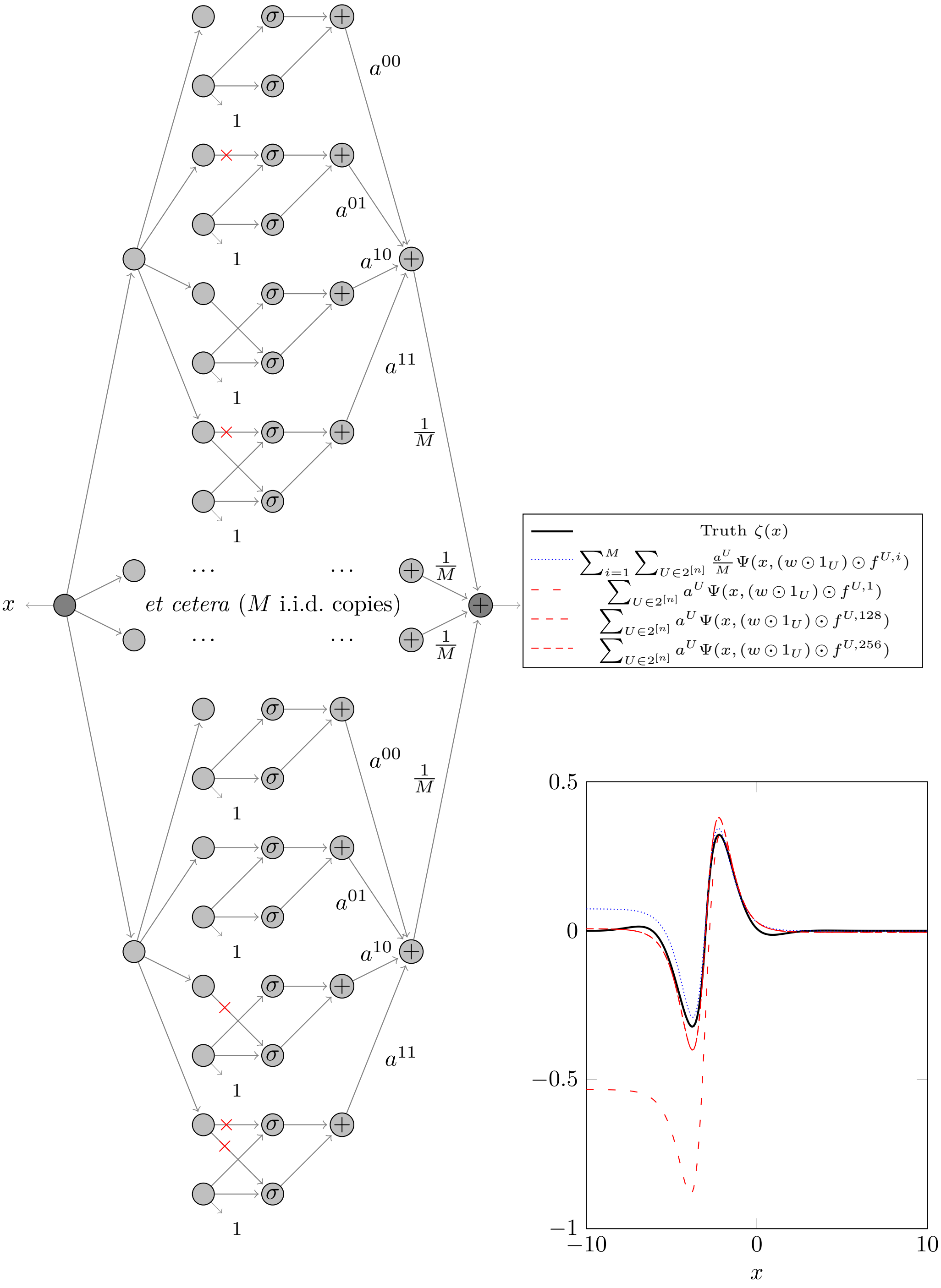

$a^{00}$
$a^{01}$
$a^{10}$
$a^{11}$
1
$\frac{1}{M}$
$x$
$\sigma$
$\cdots$
*et cetera* ($M$ i.i.d. copies)
Truth $\zeta(x)$
$\sum_{i=1}^{M}\sum_{U\in 2^{[n]}} \frac{a^U}{M}\Psi(x,(w\odot 1_U)\odot f^{U,i})$
$\sum_{U\in 2^{[n]}} a^U\Psi(x,(w\odot 1_U)\odot f^{U,1})$
$\sum_{U\in 2^{[n]}} a^U\Psi(x,(w\odot 1_U)\odot f^{U,128})$
$\sum_{U\in 2^{[n]}} a^U\Psi(x,(w\odot 1_U)\odot f^{U,256})$
0.5
0
−0.5
−1
−10
0
10
$x$

*Figure 4.3.3: An approximation of $\zeta$ with a large NN using dropconnect, based on the base Cybenko NN in* (4.15) *depicted in Figure 4.3.1. Adding many independent copies of the network from Figure 4.3.2, we are leveraging the law of large numbers as in* (4.16)*. Different independent copies of the network may have a different realization of the filters, which is here depicted by the red crosses on the edges.*

**Example 17** (Node-dropout networks)**.** *Consider again the deep NN in* (4.7) *with node-dropout as described in Section 4.2.2. The random NN in* (4.14) *is then again a node-dropout NN. In this way, we recover [22, Theorem 3] (with* $h \equiv 0$*), which for ReLU activation functions and a target function* $\zeta$ *bounds*

$$\sup_{x\in[0,1]^d} \operatorname{Var}(\zeta(x) - \Psi(x, w \odot f)). \tag{4.17}$$

*When* $\mathcal{F}$ *is the space of continuous functions with supremum norm,* (4.17) *can be bounded by a constant times the square of the* $L^2$*-norm. Hence, Theorem 14 approximates in a stronger sense, namely, in* $L^q$ *for any* $q \in [1,\infty)$*. Moreover, Theorem 14 also allows for activation functions other than ReLU.*

**Example 18** (Dropout networks with dropout on *input*)**.** *In contrast, if there is also dropout on the input, then the NN in* (4.14) *is* not *again a dropout NN with dropout on the inputs. Results by [22] imply that in general NNs with dropout on the* input *cannot satisfy a universal approximation property.*

*We remark that this kind of stochastic network is* not *a dropout NN defined in Section 4.2.2 as the following example shows: Suppose that* $\Psi_1, \Psi_2 : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ *are two different dropout NNs with weights* $w_1, w_2$ *and with respective filter random variables* $f, g$ *with values in* $\{0,1\}^n$*. Then we can define the dropout NN* $\Psi$ *with value*

$$\Psi(x, (w_1 \odot f, w_2 \odot g)) = \Psi_1(x, w_1 \odot f) + \Psi_2(x, w_1 \odot g). \tag{4.18}$$

*Suppose that, additionally, we add independent filters* $h_1$ *and* $h_2$ *with values in* $\{0,1\}^d$ *to* $\Psi_1, \Psi_2$ *for their respective inputs. Then,* $\Psi_1(x \odot h_1, w_1) + \Psi_2(x \odot h_2, w_2)$ *is not necessarily of the type* $\Psi(x \odot h, (w_1, w_2))$ *for some random variable* $h$ *with values in* $\{0,1\}^d$*.*

As the above examples illustrate, a crucial aspect of whether a certain class of dropout NNs (such as dropconnect or node-dropout) satisfy a universal approximation property, is whether linear, independent combinations of such networks are again networks in the same class. On the other hand, many details of the NNs, such as them being a composition of simpler functions, are irrelevant for the proof of Theorem 14.

### 4.3.3 The classes $\mathsf{DDNN}$

The classes $\mathsf{DDNN}$ of tuples $(n, \Psi, f)$ are closed under linear, independent combinations and they are the basic objects with which we want to approximate a given function $\zeta \in \mathcal{F}$.

The convergence statement (4.14) of Theorem 14 then immediately implies Corollary 3 in Section 4.1. It expresses that if the class $\mathsf{DDNN}$ is rich enough to approximate any function in $\mathcal{F}$ when all filter variables are set to 1 (the event in (4.4)), then for every function in $\mathcal{F}$ there exists a dropout NN such that with high probability with regard to the filter variables, the dropout NN also approximates the function.

The proof of Corollary 3 can be found in Appendix 4.A.2. This corollary can be combined with deterministic universal approximation properties of certain classes of NNs to obtain concrete universal approximation properties of dropout NNs. For instance, because both the class of node-dropout networks and the class of dropconnect networks form examples of a set $\mathsf{DDNN}$, we obtain the following universal approximation property by combining Corollary 3 in the introduction with a known universal approximation result in [153, Proposition 1].

**Corollary 6.** *Assume $\mu$ is a nonnegative probability measure on $\mathbb{R}^n$ with compact support, absolutely continuous with respect to the Lebesgue measure. Take $\mathcal{F} = L^r(\mu)$ for some $r \in [1,\infty)$. Assume that the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is not equal to a polynomial almost everywhere. Then for every $\epsilon > 0$ there exists a one-hidden-layer dropconnect NN $(n,\Psi,g)$ such that*

$$\mathbb{P}[\|\Psi(\cdot,w \odot g) - \zeta\|_{L^r(\mu)} > \epsilon] < \epsilon, \tag{4.19}$$

*and*

$$\mathbb{E}\left[\|\zeta(\cdot) - \Psi(\cdot,w \odot h)\|^q_{L^r(\mu)}\right]^{\frac{1}{q}} < \epsilon.$$

*There also exists a one-hidden-layer node-dropout NN with the same properties.*

To further illustrate Corollaries 3 and 6, in Figure 4.3.4 we look at the approximation in probability of our construction from Theorem 14.

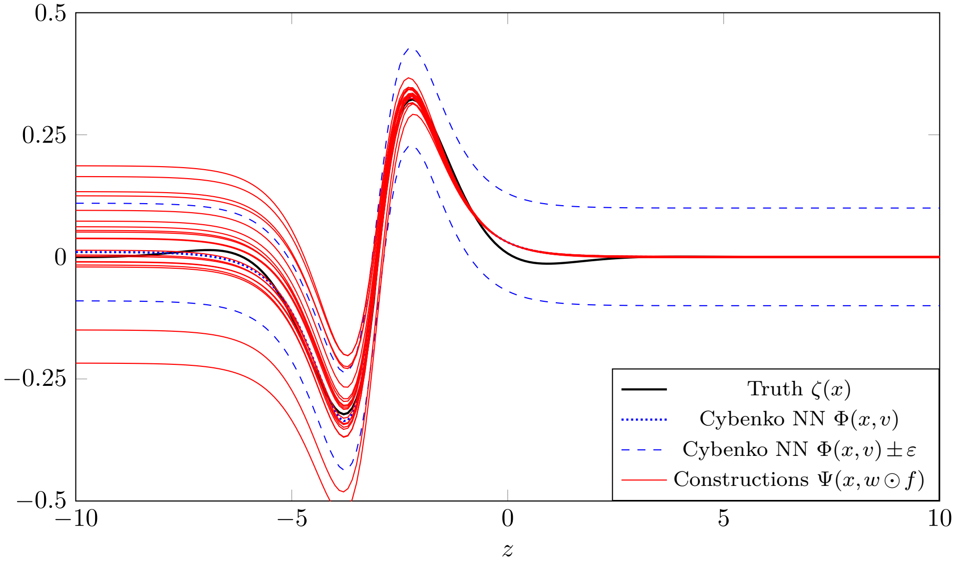


*Figure 4.3.4: An illustration of function approximation in probability with our construction. Here, $M = 256$, and 20 independent runs of the construction are shown in red. Here, $\epsilon = 0.1$ was chosen for illustrative purposes. Most of the runs lie within $\epsilon$ distance around the Cybenko NN from* (4.15)*, which we approximate with our construction in* (4.16)*. Altogether, we are approximating the target function $\zeta$.*

### 4.3.4 Explicit computation of coefficients

To further illustrate Theorem 14, we will compute the coefficients $a_U$ in (4.13) explicitly for a special case of dropout NNs for which the filter variables are partitioned into independent blocks. All variables in one block $i$ are all simultaneously off with probability $1 - p_i$ and simultaneously on with probability $p_i$. Both node-dropout and dropconnect are special cases.

**Proposition 16.** *Let $f$ be a $\{0,1\}^n$-valued random variable with a distribution specified as follows. Let $[n] = I_1 \cup \ldots \cup I_r$ be a disjoint partition and suppose that $f_i = f_j$ whenever $i,j \in I_s$ for any $i,j \in [n]$ and $s \in [r]$. Let $f = (f_{I_1}, \ldots, f_{I_r})$ denote the random variables ordered as blocks and suppose that $\mathbb{P}(f_{I_s} = 1) = p_s > 0$ for all $s \in [r]$ and that $\{f_{I_i}\}_{i \in [r]}$ are mutually independent. Then*

$$\Psi(\cdot, w) = \sum_{V \in 2^{[r]}} \prod_{i \in V} \left(\frac{1}{p_i}\right) \prod_{i \in [r] \setminus V} \left(-\frac{1 - p_i}{p_i}\right) \mathbb{E}\left[\Psi(\cdot, (w \odot \mathbf{1}_{\iota(V)}) \odot f^V)\right]$$

*where $\iota : 2^{[r]} \to 2^{[n]}$ is the embedding characterized by $j \in \iota(V)$ if $j \in I_i$ for some $i \in V$.*

We prove Proposition 16 in Appendix 4.A.3. Note that as the remain probability $p_i \to 0$ or equivalently the dropout probability $1 - p_i \to 1$, the coefficients $a_U$ become large. From this fact, together with the observation that the sum is taken over the large set $2^{[r]}$, it is clear that the construction is computationally intensive. Still, small examples in the case of dropconnect are shown in Figures 4.3.2, 4.3.3 and 4.3.4.

### 4.3.5 Limitation of the results to random-approximation dropout

In this section, we have shown a random-approximation universal approximation result, i.e., a universal approximation result that is relevant when the dropout NN is also used at prediction time by sampling from the random NN. In practice, as explained in Section 4.1.3, the replacement of the filter random variables $f$ by their expected values $\mathbb{E}[f]$ is common practice after having trained a NN with dropout for prediction. The following example shows that the previous construction in this section can lead to a bad approximation when doing expectation-replacement.

**Example 19.** *Let $\sigma$ be the standard ReLU activation function. The approximation procedure in Corollary 3 would yield that the function $\zeta : \mathbb{R} \to \mathbb{R}$ given by $\zeta(x) := \sigma(x-1)$ can be approximated well by an average of many independent copies of the dropout NN*

$$x \mapsto 4 f_1 \sigma(f_2 x - 1),$$

*where $f_1$ and $f_2$ are independent Bernoulli random variables with success probability $1/2$. However, replacing $f_1$ and $f_2$ by $1/2$, we just obtain the function*

$$x \mapsto 2\sigma(x/2 - 1),$$

*which is not a good approximation to the function $\zeta$ at all.*

This example motivates us to look for a different approximation scheme that also has the property that we can replace the random variables by their expectation. In the next section we explain how this can be accomplished.

## 4.4 Universal approximation for expectation-replacement dropout

We now get into the problem of approximating a NN by a dropout NN that is also close to the original NN if the filter variables are replaced by their expected values. The main result in this section is Theorem 28 below. Informally, it states that for any base NN $\Psi(\cdot, w)$, there exists a larger NN, denoted by $\mathsf{NN}_{\Gamma,\Xi}(\cdot, v)$, and filter variables $f$ such that

$$\mathsf{NN}_{\Gamma,\Xi}(x, v \odot f) \approx \Psi(x, w) \approx \mathsf{NN}_{\Gamma,\Xi}(x, v \odot \mathbb{E}[f]). \tag{4.20}$$

The basis for this approximation technique is by adding repeated weights. Namely, we iteratively replace each edge in the deterministic NN $\Psi$ by a set of parallel edges, with edge-weights $w$ taken from the original edge, and with independent filter variables on each of them. In this way we can use the law of large numbers to obtain convergence estimates for each edge separately, and then combine these estimates into a single convergence estimate for the whole network.

The convergence estimate for a single edge arises from the following statement (which is a simplified version of Lemma 35 below). It describes how the error encountered by averaging $N$ independently filtered edges can be controlled in probability. At the same time it also allows for small perturbations of the inputs to this edge. This latter perturbation freedom is needed in order to apply this lemma progressively, moving from edge to edge through the network.

**Lemma 33.** *Consider any continuous function $\sigma : \mathbb{R}^m \to \mathbb{R}^m$ and let $W \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Let $\{F^i\}_{i \in [N]}$ be a collection of independent copies of a random matrix $F \in \{0,1\}^{m \times n}$. Then for every $K > 0$ there exists a $\delta > 0$ such that*

$$\sup_{x \in \overline{B(0,K)}} \sup_{(\tilde{x}^i) \in B(x,\delta)^N} \left| \sigma\left( \frac{1}{N} \sum_{i=1}^{N} (W \odot F^i)\tilde{x}^i + b \right) - \sigma\Big( (W \odot \mathbb{E}[F^i])x + b \Big) \right| \tag{4.21}$$

*converges to zero in probability as $N \to \infty$.*

This construction is described in detail in the following sections. A separate part of this description is how to connect the resulting dropout NN to the inputs of the now random original layer; for this we introduce a single additional layer that implements this connection.

## Global variables

In order to improve the readability, we fix for the entire section a few (otherwise arbitrary) variables. Throughout this section:

(i) The base NN $\Psi$ is assumed to be a fixed $(L-1)$-hidden layer NN as described in Section 4.2.1. We assume that its activation functions $\sigma_j$ are continuous. We also keep the weights $W^{(j)}$ and biases $b^{(j)}$ fixed.

(ii) We fix a number $R > 0$, which will play the role of the radius of a ball in the input space.

(iii) We fix a number $\beta \in (0,1)$ and assume that for every random filter matrix $F$ in this section, each one entry is on with a probability that is larger than or equal to $\beta$, i.e., for all $r, c$,

$$\mathbb{P}[F_{rc} = 1] \geq \beta > 0.$$

(iv) We assume there is a constant $Q > 1$, which will encode the differentiability of the activation function of the input layer at the origin. We postpone its formal definition for now.

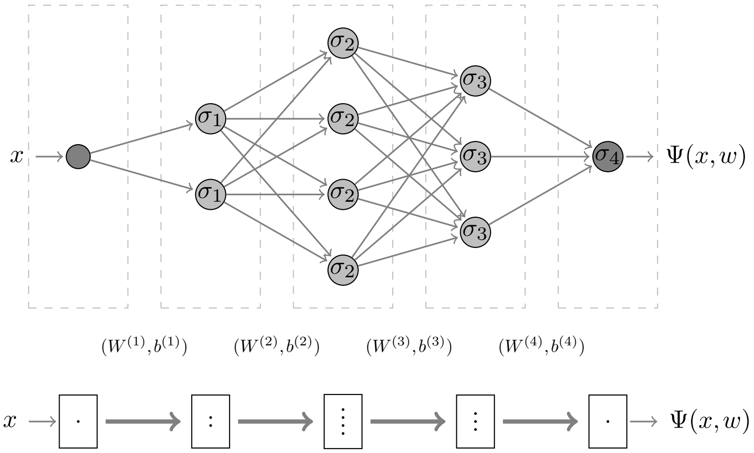


*Figure 4.4.1: An example base NN* $\Psi$ *where* $L=4$. *The top diagram indicates the individual activation functions and the dimensions of each layer of nodes:* $d=d_0=1$, $d_1=2$, $d_2=4$, $d_3=3$, *and* $d_4=1$. *The set of all arrows connecting layer* $k-1$ *of nodes to layer* $k$ *correspond to the parameters* $(W^{(k)},b^{(k)})$. *The bottom diagram shows the same network, with the edges between layers compressed to single arrows; this compressed notation is the basis for the diagram in Figure 4.4.2 below.*

### 4.4.1 Heuristic description of the construction

In this section we describe the construction of the larger dropout NN $\mathsf{NN}_{\Gamma,\Xi}$ in heuristic terms; the full details are given in the subsequent sections. The construction starts at the last layer of the base NN $\Psi$, which is a function $\Psi_L : \mathbb{R}^{d_{L-1}} \to \mathbb{R}^{d_L}$ given by

$$x \mapsto \sigma_L\big(W^{(L)}x + b^{(L)}\big). \tag{4.22}$$

We construct the last layer of the larger dropout NN such that it remains close to (4.22) as follows. Let $N \in \mathbb{N}$ and consider any collection $\{F^{(L),i}\}_{i\in[N]}$ of i.i.d. random filter matrices such that for each $i$, $F^{(L),i}$ has the same dimension as $W^{(L)}$. By the law of large numbers, we can expect that the function

$$x \mapsto \sigma_L\left(\frac{1}{N}\sum_{i=1}^{N}\Big(\big(W^{(L)} \div \mathbb{E}[F^{(L),i}]\big)\odot F^{(L),i}\Big)x + b^{(L)}\right) \tag{4.23}$$

will be close to the function $\Psi_L$ for sufficiently large $N$. Here, we write $\div$ for element-wise division.

Viewed as a one-layer NN, the function (4.23) is a one-layer dropout NN with $N$ times as many edges as $\Psi_L$, and it can replace (4.22), i.e., $\Psi_L$, while staying close to $\Psi_L$.

A further adaptation is necessary, however, because in (4.23) each copy $W^{(L)} \div \mathbb{E}[F^{(L),i}]$ takes the *same* input $x \in \mathbb{R}^{d_{L-1}}$. To make (4.23) a *bona fides* dropout network, different edges should take different inputs, and therefore we generalize (4.23) to

$$(\mathbb{R}^{d_{L-1}})^N \ni (x^i)_{i\in[N]} \mapsto \sigma_L\left(\frac{1}{N}\sum_{i=1}^{N}\Big(\big(W^{(L)} \div \mathbb{E}[F^{(L),i}]\big)\odot F^{(L),i}\Big)x^i + b^{(L)}\right). \tag{4.24}$$

By precomposing each of the inputs $x^i$ with $\Psi^{(L-1)}$, and performing the same construction as above (copying the input to these copies of $\Psi^{(L-1)}$), we can inductively create our larger dropout NN $\mathsf{NN}_{\Gamma,\Xi}$ that will be close to $\Psi$.

There are now three points of attention:

- The intuitive statement 'repeating this construction' needs a formalization by an inductive construction. This requires a mathematical object that can record the intermediate stages of the construction.
- We need to show inductively that the resulting intermediate NNs are close to (a network closely related to) the original network. In particular, we need to introduce a mathematical specification of 'close' that is compatible with an inductive argument.
- The input space to the NN in this construction grows with each step, whereas we still aim to have a final NN with data space $\mathbb{R}^d$. This requires us to deal with the first layer of the network differently.

These points are the topics of the subsequent sections.

### 4.4.2 Dropout–trees

We will encode the intermediate stages of our inductive construction by a mathematical object that we will refer to as a *dropout–tree*. The idea is that we start with a root, then attach incoming edges labeled with random filter matrices to it (creating leaves), and then recursively attach even more edges to the leaves. To be consistent with the numbering of layers in Section 4.2.1, here, we will prefer to speak about the *level $j$* of a vertex or an edge in a tree rather than its depth $L-j$ (the latter is also an established term in graph theory, and this aligns our notation with that of the NN). In this numbering, the root is therefore at level $L$.

**Definition 20.** *A vertex $v$ of a rooted tree is at* level $j \in \{0,1,\dots,L\}$ *if the path from $v$ to the root $v_0$ has length $L-j$. An edge $e=(u,v)$ of a rooted tree is at* level $j \in \{0,1,\dots,L\}$ *if its target vertex $v$ is at level $j$.*

From now on we will write $\sigma_v$ for $\sigma_{\mathsf{level}(v)}$, $W^v$ for $W^{(\mathsf{level}(v))}$, $b^v$ for $b^{(\mathsf{level}(v))}$, *et cetera.* This simplifies the notation at only a minor cost of abuse of notation.

**Definition 21.** *A* dropout–tree $\Gamma$ *of an $(L-1)$-hidden layer NN $\Psi$ is a directed graph $\mathcal{G}$ together with a labeling of the edges such that:*

- *the graph $\mathcal{G}$ is connected and acyclic;*
- *one of the vertices, say $v_0$, is designated as the* root*;*
- *the depth of the tree is at most $L-1$;*
- *all directed edges point towards the root;*
- *every edge $e$ is labeled with a random matrix $F^e$; for each $e$*
  - *(a) $F^e$ has the same dimension as $W^{\mathsf{target}(e)}$*
  - *(b) $F^e$'s entries are $\{0,1\}$-valued*

*(c) for all entries $(r,c)$ of $F^e$, $\mathbb{P}[F^e_{rc}=1]\geq\beta>0$;*

- *for every vertex $v$ that is not a leaf, $\{F^e\}_{e\in\mathsf{into}(v)}$ is a collection of mutually independent, identically distributed random matrices.*

For convenience we recall some terminology. A directed edge points from a *source* to a *target*, and for an edge $e$ we identify them by $\mathsf{source}(e)$ and $\mathsf{target}(e)$; we write $\mathsf{into}(v)$ for the set of all edges with target vertex $v$. In the trees throughout this article, all edges point towards the root of the tree. A vertex $v$ is a *child* of a vertex $w$ if there is an edge pointing from $v$ to $w$; $w$ then is the *parent* of $v$. A *leaf* is a vertex without children.

Dropout–trees can be constructed iteratively by starting from the trivial dropout–tree consisting only of a root and then performing a so-called *$\mu$-input-copy* construction. This allows us to inductively create a larger dropout–tree from a smaller dropout–tree.

**Definition 22.** *Let $\Gamma$ be a dropout–tree and let $\ell$ be a leaf of $\Gamma$ at level $k$. Let $\mu$ be a distribution of a random matrix $F\in\{0,1\}^{d_k\times d_{k-1}}$ that satisfies for all $r,c$, $\mathbb{P}[F_{rc}=1]\geq\beta>0$. A dropout–tree $\Gamma'$ is a* $\mu$-input-copy to the leaf $\ell$ of $\Gamma$ *if one can obtain $\Gamma'$ from $\Gamma$ by: (a) attaching child vertices to $\ell$, and (b) labeling each edge going into $\ell$ by an independent copy of $F$. The* size of a $\mu$-input-copy to $\ell$ at $\Gamma$ *refers to the number of children of $\ell$ in $\Gamma'$.*

Let us describe the precise meaning of procedure (b) in Definition 22. For that, it may be useful to recall that random matrices are nothing but measurable functions defined on the probability space $(\Omega,\mathcal{F}_\Omega,\mathbb{P})$. The procedure (b) precisely means that the sigma-algebras generated by the filter variables $F^e$ with $e\in\mathsf{into}(\ell)$ are independent, and that for every $e\in\mathsf{into}(\ell)$ the law of $F^e$ equals $\mu$. In particular, this condition allows for some correlation between filter variables labeling edges in the dropout–tree that do not go into $\ell$. Moreover, in general there can be many different dropout trees $\Gamma'$ that are $\mu$-input-copies of $\Gamma$.

We will now describe how a dropout–tree encodes a dropout NN.

### Dropout NNs encoded by dropout–trees

Each dropout–tree $\Gamma$ will induce a stochastic function $\Phi^{v_0}_\Gamma$—a dropout NN—as follows. For any edge $e=(u,v)$ of $\Gamma$, let

$$V^e_\Gamma := W^e \div \mathbb{E}[F^e] \tag{4.25}$$

be rescaled weights for the dropout NN. We define

$$\Phi^v_\Gamma := \begin{cases}\sigma_v\left(\frac{1}{\#\mathsf{into}(v)}\sum_{e\in\mathsf{into}(v)}(V^e\odot F^e)\Phi^{\mathsf{source}(e)}_\Gamma + b^v\right) & \text{if } v \text{ is not a leaf,}\\ \mathrm{Identity}_{\mathbb{R}^{d_v}} & \text{if } v \text{ is a leaf.}\end{cases}$$

Figure 4.4.2 illustrates this construction, based on the network $\Psi$ depicted in Figure 4.4.1.

## 4.4.3 Dropout NNs from dropout–trees are close to deterministic NNs

We will give an inductive argument that $\Phi^{v_0}_\Gamma$ is close to $\Phi^{v_0}_{\Gamma_{\mathsf{det}}}$. Here, $\Gamma_{\mathsf{det}}$ denotes the same dropout–tree as $\Gamma$ except for the fact that we have replaced each and every filter variable deterministically by its expectation. Loosely speaking, the inductive argument implies that dropout NNs induced by dropout–trees are close to their deterministic counterparts.

As a technical preparation, we define a sequence of radii $R_0, R_1, \dots, R_L$. The idea is that these provide bounds on the output after applying several layers, no matter the choice of filter variables or weights in the upcoming construction. Denote the Hilbert–Schmidt norm of a matrix $A$ by $\|A\|_{\mathrm{HS}} = \sqrt{\mathrm{Tr}(A^T A)}$ and let $I$ be an identity matrix of suitable size. Given the radius $R > 0$, defined as a global variable at the start of this section, we set

$$R_0 := \frac{Q}{\beta} R + 1,$$

and then choose $R_j$ inductively such that for all $j \in [L]$, $x \in B(0, \beta^{-1}\|W^{(j)}\|_{\mathrm{HS}} R_{j-1} + 1)$ it holds that

$$\left|\Psi_j\Big(x, \big(I, b^{(j)}\big)\Big)\right| < R_j - 1, s \tag{4.26}$$

for all $j = [L]$, where $\beta \in (0,1)$ and $Q > 1$ were two of the global variables that we defined at the beginning of the section.

We denote the input space of a network induced by a dropout–tree $\Gamma$ by $\mathsf{Inp}_\Gamma$. That is, $\mathsf{Inp}_\Gamma$ is the vector space

$$(x^\ell \in \mathbb{R}^{d_{\mathsf{level}(\ell)}} \mid \ell \in \mathsf{leaves}(\Gamma)).$$

We endow $\mathsf{Inp}_\Gamma$ with the norm

$$\|(x^\ell)\|_{\mathsf{Inp}_\Gamma} := \max_{l \in \mathsf{leaves}(\Gamma)} \|x^\ell\|_{\mathbb{R}^{d_{\mathsf{level}(\ell)}}}.$$

We define $\mathsf{In}_\Gamma : \mathbb{R}^d \to \mathsf{Inp}_\Gamma$ to be the collection of functions, indexed by leaves $\ell$ of $\Gamma$, that are generated by those layers in the base network $\Psi$ that are *not* represented in $\Gamma$ at leaf $\ell$:

$$\mathsf{In}_\Gamma^\ell(x) := \Big(\Psi_{\mathsf{level}(\ell)}(\cdot, (W^{(\mathsf{level}(\ell))}, b^{(\mathsf{level}(\ell))})) \circ \cdots \circ \Psi_1(\cdot, (W^{(1)}, b^{(1)}))\Big)(x). \tag{4.27}$$

Note that by the definitions (4.26) of the radii $R_j$ we have

$$\mathsf{In}_\Gamma^\ell\big(\overline{B(0,R)}\big) \subset B(0, R_{\mathsf{level}(\ell)} - 1). \tag{4.28}$$

We say that a dropout–tree $\Gamma$ satisfies property $\mathsf{ApProp}_\Gamma(\delta, \epsilon)$ if

$$\mathbb{P}\Big[\sup_{x \in \overline{B(0,R)}} \sup_{\tilde{x} \in \overline{B(\mathsf{In}_\Gamma(x), \delta)}} \left|\Phi_\Gamma^{v_0}(\tilde{x}) - \Phi_{\Gamma_{\det}}^{v_0}(\mathsf{In}_\Gamma(x))\right| > \frac{\epsilon}{2}\Big] < \Big(\frac{\epsilon}{4R_L}\Big)^q. \tag{4.29}$$

Lemma 34 below shows that one can always construct a full dropout–tree that satisfies $\mathsf{ApProp}_\Gamma(\delta, \epsilon)$ for some $\delta > 0$, by copying inputs at vertices.

**Lemma 34.** *Let $\Gamma$ be a dropout–tree and let $\ell$ be a leaf of $\Gamma$ at level $k > 1$. Let $\mu$ be the distribution of a random matrix $F \in \{0,1\}^{d_k \times d_{k-1}}$ that satisfies for all $r, c$, $\mathbb{P}[F_{rc} = 1] \geq \beta > 0$. If $\Gamma$ satisfies $\mathsf{ApProp}_\Gamma(\delta, \epsilon)$ in (4.29) for some $\delta, \epsilon > 0$, then for every sufficiently large $\mu$-input-copy $\Gamma'$ of $\Gamma$ at $\ell$ there exists a $\delta' > 0$ such that $\Gamma'$ satisfies $\mathsf{ApProp}(\Gamma')(\delta', \epsilon)$.*

The proof of Lemma 34 is relegated to Appendix 4.B.1. There, we show that Lemma 34 follows from Lemma 35, which is presented next and proved in Appendix 4.B.2.

**Lemma 35.** *Consider any continuous function* $\sigma : \mathbb{R}^m \to \mathbb{R}^m$ *and let* $W \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. *Let* $\{F^i\}_{i\geq 1}$ *be a sequence of mutually independent copies of a random matrix* $F \in \mathbb{R}^{m\times n}$ *that satisfies: for* $r \in [m]$ *and* $c \in [n]$, $0 < \mathbb{E}[F_{rc}] < \infty$ *and* $0 \leq F_{rc} \leq M < \infty$ *w.p. one. Let* $V := W \div \mathbb{E}[F]$. *Then, for every* $0 \leq K < \infty$ *and* $\rho > 0$ *there exists a* $\delta > 0$ *such that*

$$\mathbb{P}\Big[\sup_{x\in\overline{B(0,K)}}\ \sup_{(\tilde{x}^i)\in\overline{B(x,\delta)}^N}\Big|\sigma\Big(\frac{1}{N}\sum_{i=1}^{N}(V\odot F^i)\tilde{x}^i+b\Big)-\sigma(Wx+b)\Big|>\rho\Big]\to 0 \tag{4.30}$$

*as* $N \to \infty$.

### 4.4.4 Replacing the first layer

We examine now the first layer of the NN and consider first the case that we do not use dropout. Later, we will consider the case when we also have dropout in the first layer.

**Copying the first layer many times, without dropout of input edges**

We will now start from a full dropout–tree and connect copies of the first layer many times to obtain a NN that approximates the original NN well, both in terms of random approximation and in terms of expectation replacement. In this section, it is crucial that there is no dropout of edges in this very first layer.

Assume therefore that we have constructed a full dropout tree, i.e., a dropout tree of which all leaves are at level 1 (at depth $L-1$). We denote by $\mathsf{NN}_{\Gamma,\Xi}$ the NN that is formed by precomposing every leaf $\ell$ of the NN $\Phi_\Gamma^{v_0}$ induced by the dropout tree by the first layer of the original network. To align with the next section, we denote this function by $\Xi^\ell := \Phi^{(1)}(\cdot,(W^{(1)},b^{(1)}))$. We also let $\mathsf{NN}_{\Gamma,\Xi}^{\mathsf{avg-filt}}$ denote almost the same network, but with the modification that every random filter variable is replaced by its expectation.

The next theorem expresses that this construction yields a dropout NN that approximates the original NN well in terms of random approximation and expectation replacement.

**Theorem 23.** *Let* $1 > \epsilon > 0$. *Let* $\Gamma$ *be a full dropout–tree satisfying* $\mathsf{ApProp}_\Gamma(\delta,\epsilon)$ *for some* $\delta > 0$. *Let* $\mathsf{NN}_{\Gamma,\Xi}$ *and* $\mathsf{NN}_{\Gamma,\Xi}^{\mathsf{avg-filt}}$ *be the networks described above, or more precisely defined in Example 25. Then*

$$\mathbb{P}\Big[\sup_{x\in\overline{B(0,R)}}\big|\mathsf{NN}_{\Gamma,\Xi}(x)-\Psi(x,w)\big|>\epsilon\Big]<\epsilon \tag{4.31}$$

*and*

$$\mathbb{E}\Big[\sup_{x\in\overline{B(0,R)}}\big|\mathsf{NN}_{\Gamma,\Xi}(x)-\Psi(x,w)\big|^q\Big]^{1/q}<\epsilon, \tag{4.32}$$

*while*

$$\sup_{x\in\overline{B(0,R)}}\Big|\mathsf{NN}_{\Gamma,\Xi}^{\mathsf{avg-filt}}(x)-\Psi(x,w)\Big|<\epsilon. \tag{4.33}$$

Theorem 23 follows from Theorem 28 below, which pertains to the more general case in which edges in the input layers can be dropped. Indeed, if in Theorem 28 one takes $\sigma_0 : \mathbb{R} \to \mathbb{R}$ to be the identity function and one sets all filter variables deterministically equal to 1, then for every $\alpha$ and $N$, the networks constructed in Theorem 28 coincide exactly with $\mathsf{NN}_{\Gamma,\Xi}$ and $\mathsf{NN}_{\Gamma,\Xi}^{\mathsf{avg-filt}}$ introduced in Theorem 23.

### First layer, with dropout of input edges

The situation becomes more complicated if we allow for dropout of edges in every layer, particularly the first. In this section we will show that we can also in that case find a dropout NN that approximates the original NN well both in terms of random approximation and in terms of expectation replacement.

Assume again that we have constructed a full dropout–tree, that is, a dropout–tree of which all leaves are at level 1 (i.e., at depth $L-1$). This means that we have constructed suitable replacements for almost every layer of the NN, except for the first layer. This layer contains the edges that have the global input as source. Replacing the first layer requires a different construction: if we would outright drop edges in the first layer, then we can not control the error with the current technique. We now describe how we replace the first layer.

For every leaf $\ell$ in the full dropout–tree, we precompose every input at $\ell$ with a stochastic function $\Xi^\ell : \mathbb{R}^{d_0} \to \mathbb{R}^{d_1}$. We record this information in what we call a *precomposition* for a dropout–tree. Figure 4.4.2 illustrates this precomposition.

**Definition 24.** *A* precomposition $\Xi$ *for a full dropout–tree* $\Gamma$ *is a map* $\Xi : \mathsf{leaves}(\Gamma) \to (\mathbb{R}^{d_0} \to \mathbb{R}^{d_1})$ *that sends every leaf* $\ell \in \mathsf{leaves}(\Gamma)$ *to a stochastic function* $\Xi^\ell : \mathbb{R}^{d_0} \to \mathbb{R}^{d_1}$.

Let $\Delta : \mathbb{R}^{d_0} \to (\mathbb{R}^{d_0})^{\mathsf{leaves}_\Gamma}$ be the diagonal map sending $x$ to copies of $x$. The NN induced by the full dropout–tree $\Gamma$ and a precomposition $\Xi$ that we consider is given by

$$\mathsf{NN}_{\Gamma,\Xi} := \Phi^{v_0}_{\Gamma,\Xi} \circ \Delta \tag{4.34}$$

where

$$\Phi^v_{\Gamma,\Xi} = \begin{cases} \sigma_v\left(\frac{1}{\#\mathsf{into}(v)} \sum_{e\in\mathsf{into}(v)} (V^e \odot F^e)\Phi^{\mathsf{source}(e)}_{\Gamma,\Xi} + b^e\right) & \text{if } v \text{ is not a leaf,} \\ \Xi^v & \text{if } v \text{ is a leaf.} \end{cases} \tag{4.35}$$

We also define $\mathsf{NN}^{\mathsf{avg-filt}}_{\Gamma,\Xi}$ as being almost the same NN as $\mathsf{NN}_{\Gamma,\Xi}$, with the only difference being that we replace each random filter variable $F^e$ in (4.35) with its expectation $\mathbb{E}[F^e]$. Recall for (4.34) that $v_0$ designates the root of the dropout–tree, and note that (4.35) constructs the NN recursively (layer by layer).

**Example 25.** *In fact, we already examined one specific precomposition* $\Xi$ *in Section 4.4.4: the one that assigns the function* $\Psi^1(\cdot,(W^{(1)},b^{(1)}))$ *to every leaf. This precomposition yields a NN* $\mathsf{NN}_{\Gamma,\Xi}$ *in which edges in the first layer, i.e., the input edges of the NN, are never dropped. In this case,* $\mathsf{NN}^{\mathsf{avg-filt}}_{\Gamma,\Xi}(w)$ *actually coincides with the original NN* $\Psi(\cdot,w)$.

We will now construct precompositions that allow for the possibility of dropping edges in the first layer and applying e.g., the ReLU function to them immediately. Concretely, we will add a zeroth layer with an activation function $\sigma_0 : \mathbb{R} \to \mathbb{R}$. We assume that $\sigma_0(0) = 0$ and that $\sigma_0$ has one-sided derivatives $\sigma_-$ and $\sigma_+$ in the point $0 \in \mathbb{R}$:

$$\sigma_- := \lim_{\alpha\downarrow 0} \frac{\sigma_0(-\alpha)-\sigma_0(0)}{\alpha}, \quad \sigma_+ := \lim_{\alpha\downarrow 0} \frac{\sigma_0(+\alpha)-\sigma_0(0)}{\alpha}. \tag{4.36}$$

Define the sign function

$$S(x) = \begin{cases} - & \text{if } x < 0, \\ + & \text{if } x \geq 0 \end{cases} \tag{4.37}$$

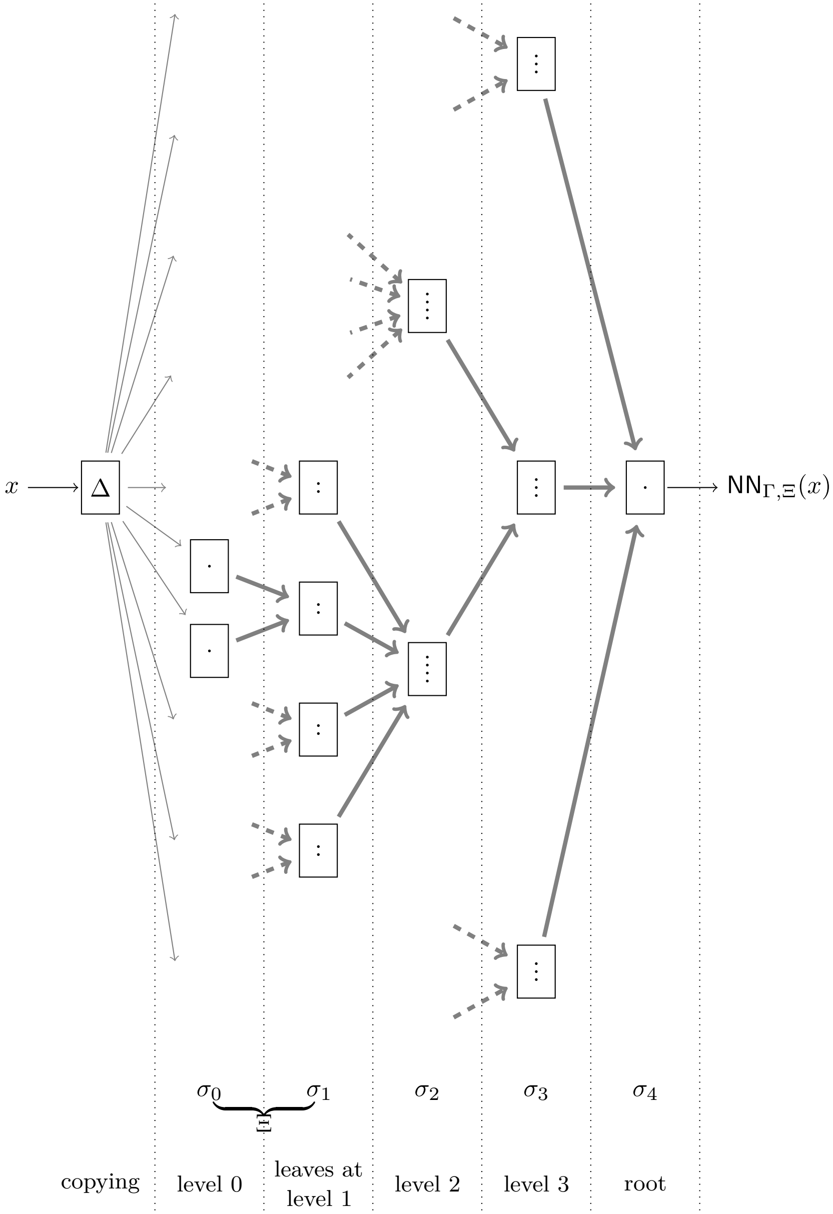


*Figure 4.4.2: An illustration of how we use the base NN $\Psi$ from Figure 4.4.1 and a dropout–tree (indicated with thicker arrows) to ultimately construct the larger NN $\mathsf{NN}_{\Gamma,\Xi}$ in which edges are being dropped stochastically. Here, $L = 4$.*

componentwise. If $x \neq 0$, then $\sigma_{S(-x)} + \sigma_{S(x)} = \sigma_- + \sigma_+$ does not depend on $x$—a critical fact that we will leverage in our construction.

**Example 26.** *Consider a zeroth layer that is the identity function, i.e., $\sigma_0(y) = y$. Then $\sigma_\pm = 1$. Choosing $\sigma_0$ as the identity function is allowed here, and means that the layer is not adapted.*

**Example 27.** *Consider a zeroth layer with ReLU activation function, $\sigma_0 := \mathrm{ReLU}(z) := z\mathbb{1}[z \geq 0]$. Then $\sigma_- = 0$ and $\sigma_+ = 1$.*

The precompositions that we employ are as follows. We call $\Xi$ an $(\alpha, N)$*-precomposition associated with a set of distributions* $\{\mu^\ell, \nu^\ell\}_{l \in \mathrm{leaves}(\Gamma)}$ if for each leaf $\ell$,

$$\Xi^\ell(x) := \sigma_1\Big(\frac{1}{N}\sum_{i=1}^{2N}(-1)^i (V^\ell \odot F^{\ell,i})\sigma_0\big((-1)^i \alpha(I \odot G^{\ell,i})x\big) + b^\ell\Big) \tag{4.38}$$

where element-wise

$$V^\ell_{rc} = \frac{W^{(1)}_{rc}}{\alpha\big(\sigma_- + \sigma_+\big)\,\mathbb{E}[F^\ell_{rc}]\,\mathbb{E}[G^\ell_{cc}]}, \tag{4.39}$$

and $\{F^{\ell,i}\}_{i \geq 1}$, $\{G^{\ell,i}\}_{i \geq 1}$ are sequences of mutually independent copies of random matrices $F^\ell, G^\ell$ that have distributions $\mu^\ell$, $\nu^\ell$, respectively. Furthermore, the $F^\ell$ are presumed to have the same size as $W^{(1)}$, and the $G^\ell$ to have size $d_0 \times d_0$. Note that these assumptions allow us to place unit mass on any particular outcome and thus to replace $F^\ell, G^\ell$ by deterministic counterparts.

The idea of (4.38) is that it represents two layers of a dropout NN that satisfies the approximation properties we are after. The functions $\sigma_1$, $\sigma_0$ can be understood as their activation functions, the matrices $V^\ell$, $\alpha I$ as their weights, $b^\ell$ as a bias, and the matrices $F^\ell$, $G^\ell$ as describing which edges and inputs are randomly removed. By scaling the weights $V^\ell$ by $1/(2N)$ and generating $2N$ independent copies of the first layer, we are preparing for an application of the law of large numbers. Furthermore, by allowing for arbitrarily small $\alpha$, we are preparing for a linearization of $\sigma_0$ around 0. Finally, the alternatingly positive and negative multiplicative factors $(-1)^i$ allow us to cover directional derivatives such as that of the ReLU activation function. All together, the construction allows us to prove the following theorem.

**Theorem 28.** *Fix $0 < \epsilon < 1$. Let $\Gamma$ be a full dropout–tree satisfying $\mathsf{ApProp}_\Gamma(\delta, \epsilon)$ for some $\delta > 0$. Let $\Xi$ be an $(\alpha, N)$-precomposition associated with a set of distributions $\{\mu^\ell, \nu^\ell\}_{\ell \in \mathrm{leaves}(\Gamma)}$. For every $\ell$, if $F$ is a matrix of filter variables distributed according to $\mu^\ell$ or $\nu^\ell$, then for every $r, c$, we assume that*

$$\mathbb{P}[F_{rc} = 1] \geq \beta > 0.$$

*Let $\sigma_0 : \mathbb{R} \to \mathbb{R}$ be a continuous function with one-sided derivatives $\sigma_-$ and $\sigma_+$ in 0, such that $\sigma_- + \sigma_+ \neq 0$ and such that $\sigma_0(0) = 0$. Assume moreover that $\sigma_-$ and $\sigma_+$ satisfy the following inequality with respect to the global variable $Q$:*

$$4\frac{|\sigma_-| + |\sigma_+|}{|\sigma_- + \sigma_+|} < Q. \tag{4.40}$$

*The following inequalities now hold for $\alpha > 0$ small enough and $N \in \mathbb{N}$ large enough:*

$$\mathbb{P}\Big[\sup_{x \in B(0,R)} \big|\mathsf{NN}_{\Gamma,\Xi}(x) - \Psi(x,w)\big| > \epsilon\Big] < \epsilon \tag{4.41}$$

*and*

$$\mathbb{E}\Big[\sup_{x \in B(0,R)} \big|\mathsf{NN}_{\Gamma,\Xi}(x) - \Psi(x,w)\big|^q\Big]^{1/q} < \epsilon, \tag{4.42}$$

*while*

$$\sup_{x \in B(0,R)} \Big|\mathsf{NN}^{\mathsf{avg-filt}}_{\Gamma,\Xi}(x) - \Psi(x,w)\Big| < \epsilon. \tag{4.43}$$

An important consequence of Theorem 28 is that we obtain for instance a universal approximation result for node-dropout and dropconnect NNs with ReLU activation functions that also guarantees a good approximation when filter variables are replaced by their averages, as formalized by Corollaries 4 and 5 in the introduction. Theorem 28 is proven in Appendix 4.B.3. There, we show that Theorem 28 follows from the following Lemma 36, which in turn is proven in Appendix 4.B.4 using compactness arguments and the law of large numbers.

**Lemma 36.** *Let $\sigma_0 : \mathbb{R} \to \mathbb{R}$ be a continuous function with $\sigma(0) = 0$ and with two one-sided derivatives $\sigma_-$ and $\sigma_+$ in $0$ satisfying $|\sigma_- + \sigma_+| > 0$. Let $\Xi$ be an $(\alpha, N)$-precomposition associated with a set of distributions $\{\mu^\ell, \nu^\ell\}_{l \in \mathrm{leaves}(\Gamma)}$ such that for all $\ell, r, c$, $\mathbb{E}[F^\ell_{rc}] > 0, \mathbb{E}[G^\ell_{rc}] > 0$, $0 \le F^\ell_{rc} \le M$ w.p. one, and $0 \le G^\ell_{rc} \le M < \infty$ w.p. one. The following now holds: for every leaf $\ell$, $0 \le K < \infty$, and $\rho > 0$, for $\alpha$ small enough and $N$ large enough,*

$$\mathbb{P}\Big[\sup_{x \in B(0,K)} \big|\Xi^\ell(x) - \Psi_1(x; (W^{(1)}, b^{(1)}))\big| > \rho\Big] < \rho \tag{4.44}$$

*and*

$$\sup_{x \in B(0,K)} \big|\Xi^{\ell,\mathsf{avg-filt}}(x) - \Psi_1(x; (W^{(1)}, b^{(1)}))\big| < \rho \tag{4.45}$$

*where $\Xi^{\ell,\mathsf{avg-filt}}$ denotes the function $\Xi^\ell$ in (4.38) but with each filter variable $F^{\ell,i}$ replaced by its expectation $\mathbb{E}[F^\ell]$ .*

## 4.5 Discussion

In this chapter, we showed that dropout NNs are rich enough for a universal approximation property to hold, both for random-approximation and expectation-replacement dropout. It is further evidence that the representational capacity of NNs is so large that approximations are possible despite significant additional constraints. In the case of dropout, these additional constraints are the implicit symmetry constraints enforced by the turning on and off of the filter variables. For instance, in dropconnect for most realizations of the filter variables the output of the dropconnect NN still approximates the original NN well after the filter variables are randomly permuted. Despite the enforced invariance with respect to this operation, there is enough room in the parameter space for the weights of the network to have a good approximation for the overwhelming majority of realizations of the filter variables.

Our proof of the universal approximation property for random-approximation dropout explicitly works with this symmetry. By this, we mean the following. The universal approximation property that we show even works when edges from the input nodes are dropped at random. The output in the first hidden layer is then inherently *random*, and in no way close to deterministic. This is in contrast with for instance the universal approximation property in [22] in which the layers are all very close to deterministic. In our case, the values in the nodes are random but we do have a good understanding of the *distribution* of the values in the nodes, and two stochastic realizations are most likely almost permutations of each other. By blowing up the first layer, i.e., repeating it many times in parallel, we then know the output very well up to this permutation symmetry and this turns out to be enough for us to show a universal approximation property.

Our results and methods have several limitations. Specifically, we only show the *existence* of dropout NNs close to a given function. It is a completely separate question whether an algorithm such as dropout stochastic gradient descent would actually be able to find such an approximation. The main message of our result is that at least there is no theoretical obstruction to approximating functions with dropout NNs.

In the proofs, we used very explicitly that filter variables only take on the values zero or one, while other forms of dropout also exist (for instance with Gaussian filter variables). Our algebraic proof does not readily generalize to this more general case, but it is possible that parts of the proof could be reused.

In this chapter we made no efforts to reduce the number of parameters of the approximating networks, and indeed the number of parameters can rapidly grow with increasing dimensions of the parameter $w$. This can for instance be recognized in the exponential number of additional parameters $a_U$ in Theorem 14, the large (but algebraic) number $M$ of copies that is required to reduce variance in (4.14), and the exponential increase of leaves in dropout–trees with increasing depth. Naturally, understanding optimal approximation rates in terms of parameter dimensions is a key question, but we leave this to future work.

## 4.6 Conclusion

We showed two types of universal approximation results for dropout NNs, one for random-approximation dropout, in which case the random filter variables are also used at prediction time, and one for expectation-replacement dropout, in which case the filter variables are replaced by their averages for prediction. Our results allow for dropout of edges from the input layer, allow for a wide class of distributions on filter variables, including dropout of edges from the input layer, and for a wide class of activation functions.

By making the distinction between random-approximation and expectation-replacement dropout explicit, our results contrast with the success of using expectation-replacement after training NNs with dropout. This fact suggests that the training procedure actually constrains the class of dropout NNs that are found in practice by more than just ensuring good random approximation properties.

# Appendix

## 4.A Proofs of Section 4.3

In this appendix we prove the results of Section 4.3. In particular, we prove Theorem 14, Corollary 3 and Proposition 16.

### 4.A.1 Proof of Theorem 14

We require the following algebraic lemma, which lies at the heart of Theorem 14, as it implies the existence of the constants $(a_U)$.

**Lemma 37.** *Let $\Psi:\mathbb{R}^d\times\mathbb{R}^n\to\mathbb{R}$ be a function. Let $(f^U)$ be a collection of $\{0,1\}^n$-valued random variables indexed by subsets $U\subset[n]$, such that for every $U$*

$$\mathbb{P}[f^U=(1,\dots,1)]>0.$$

*Then for every subset $V\subset[n]$, it holds that*

$$\Psi(\cdot,\cdot\odot\mathbf{1}_V)\in\operatorname{span}\left\{\mathbb{E}\left[\Psi(\cdot,(\cdot\odot\mathbf{1}_U)\odot f^U)\right]:U\in 2^{[n]}\right\},$$

*where for any subset $S\in 2^d$, i.e., $S\subset\{1,\dots,d\}$, we denote by $\mathbf{1}_S$ the characteristic function of $S$.*

*In other words, there exist constants $(a_{U,V})_{U\in 2^{[n]}}$ independent of $w$ and $x$ such that for all $(x,w)\in\mathbb{R}^d\times\mathbb{R}^n$*

$$\Psi(x,w\odot\mathbf{1}_V)=\mathbb{E}\left[\sum_{U\in 2^{[n]}}a_{U,V}\Psi(x,(w\odot\mathbf{1}_U)\odot f^U)\right],$$

*and in particular there exist constants $\alpha_U$ such that*

$$\Psi(x,w)=\mathbb{E}\left[\sum_{U\in 2^{[n]}}a_U\Psi(x,(w\odot\mathbf{1}_U)\odot f^U)\right].$$

*Proof.* The proof is by induction on the cardinality of $V$ and follows from the equality

$$\Big(\sum_{S:V\subset S}\mathbb{P}[f^V=\mathbf{1}_S]\Big)\Psi(\cdot,w\odot\mathbf{1}_V)=\mathbb{E}\Big[\Psi(\cdot,(w\odot\mathbf{1}_V)\odot f^V)\Big]\\ -\sum_{S:V\setminus S\neq\emptyset}\mathbb{P}[f^V=\mathbf{1}_S]\Psi(\cdot,w\odot\mathbf{1}_{S\cap V}).\qquad(4.46)$$

In particular, for the base case in which $V$ is empty, the last term vanishes. In the induction step, the functions $\Psi(\cdot,w\odot\mathbf{1}_{S\cap V})$ are by the induction hypothesis all in the required span. □

*Proof.* (of Theorem 14) By Lemma 37, we can find constants $a_U$ for $U\in 2^{[n]}$ such that (4.13) holds. We look now at (4.14). By the law of large numbers, convergence in probability in the normed vector space $(\mathcal{F},\|\cdot\|)$ follows. Moreover, for any $V\in 2^{[n]}$ we have

$$\|\Psi(\cdot,(w\odot\mathbf{1}_V)\odot f^V)\|_{\mathcal{F}}\le\max_{U\in 2^{[n]}}\|\Psi(\cdot,(w\odot\mathbf{1}_U)\odot f^U)\|_{\mathcal{F}}=:C_w,\qquad(4.47)$$

so that for any $q \in [1,\infty)$ and $M$,

$$\mathbb{E}\Big[\Big\|\frac{1}{M}\sum_{i=1}^{M}\sum_{U\in 2^{[n]}} a_U\Psi(\cdot,(w\odot \mathbf{1}_U)\odot f^{i,U})\Big\|_{\mathcal{F}}^{q}\Big]^{\frac{1}{q}} \leq \max_{U\in 2^{[n]}} |a_U| C_w. \tag{4.48}$$

With uniform boundedness for all $M$, we can use the dominated convergence theorem which implies then convergence in $L^q$ of the $\mathcal{F}$-valued random variables as $M \to \infty$. □

## 4.A.2 Proof of Corollary 3

*Proof.* Let $\zeta \in \mathcal{F}$ and let $\epsilon > 0$. Assume there exist a $(m,\Phi,f) \in \mathsf{DDNN}$ and a $v \in \mathbb{R}^m$ such that $\|\Phi(\cdot,v)-\zeta\|_{\mathcal{F}} < \epsilon$. Define $\eta := \epsilon - \|\Phi(\cdot,v)-\zeta\|_{\mathcal{F}} > 0$. Define the collection $(f^U)$ of $\{0,1\}^m$-valued filter variables, each being specifically an independent copy of $f$. By Theorem 14, there exist constants $(a_U)$, a number $M \in \mathbb{N}$ and $2^m M$ independent copies $(f^{i,U})$ of $f$ such that

$$\mathbb{P}\Big[\Big\|\frac{1}{M}\sum_{i=1}^{M}\sum_{U\in 2^{m}} a_U\Phi(\cdot,(v\odot \mathbf{1}_U)\odot f^{i,U}) - \Phi(\cdot,v)\Big\|_{\mathcal{F}} > \eta\Big] < \eta$$

and

$$\mathbb{E}\Big[\Big\|\frac{1}{M}\sum_{i=1}^{M}\sum_{U\in 2^{m}} a_U\Phi(\cdot,(v\odot \mathbf{1}_U)\odot f^{i,U}) - \Phi(\cdot,v)\Big\|_{\mathcal{F}}^{q}\Big]^{1/q} < \eta.$$

Hence by the triangle inequality, in fact

$$\mathbb{P}\Big[\Big\|\frac{1}{M}\sum_{i=1}^{M}\sum_{U\in 2^{m}} a_U\Phi(\cdot,(v\odot \mathbf{1}_U)\odot f^{i,U}) - \zeta\Big\|_{\mathcal{F}} > \epsilon\Big] < \epsilon$$

and

$$\mathbb{E}\Big[\Big\|\frac{1}{M}\sum_{i=1}^{M}\sum_{U\in 2^{m}} a_U\Phi(\cdot,(v\odot \mathbf{1}_U)\odot f^{i,U}) - \zeta\Big\|_{\mathcal{F}}^{q}\Big]^{1/q} < \epsilon.$$

We then define the tuple $(n,\Psi,g) \in \mathsf{DDNN}$ as an independent finite linear combination of $2^m M$ copies of $(m,\Phi,f)$, with coefficients $a_U/M$. Setting $\tilde{v} \in \mathbb{R}^{2^m m}$ to be the concatenation of the $2^m$ modified vectors $(v\odot \mathbf{1}_U)_{U\subset[m]}$, we then set $w \in \mathbb{R}^{2^m Mm}$ to be the subsequent concatenation of $M$ copies of $\tilde{v}$. Combining $(n,\Psi,g)$ and $w$ proves the corollary. □

## 4.A.3 Proof of Proposition 16

As we have seen in Lemma 37, we can find the map $\Psi(\cdot,w)$ in the span of $\mathbb{E}[\Psi(\cdot,(w\odot \mathbf{1}_V)\odot f^V]$ for $V \in 2^{[n]}$. We examine now specific cases where we can explicitly compute the linear combination. In particular, we consider the cases of *dropout* [117], that is, we drop nodes independently with the same probability, and *dropconnect* [115], where we drop individual weights independently with the same probability.

In both cases the filter variables $f_i$ take the same values for some disjoint subsets of $[n]$, where $n$ is the number of weights where we apply filters. That is, if we have a disjoint set decomposition $[n] = I_1 \cup \ldots \cup I_r$ with $I_k \cap I_s = \emptyset$ whenever $k \neq s$, then $f_i = f_j$ for all $i,j \in I_k$ and $f_i, f_l$ are independent if they belong to disjoint sets, $f_i \in I_k$ and $f_l \in I_s$ with $s \neq k$. In this section we drop the index $U \in 2^{[n]}$ of the random variable $f^U$ for notational convenience as they are identically distributed. We will use this property to obtain an explicit decomposition in (4.13) and in a more general setting where the probability of the filters may differ depending on which disjoint set they belong to. For $K, S \in 2^{[n]}$, we denote $K \subseteq S$ to be the usual set inclusion, that is, $i \notin K$ whenever $i \notin S$ for all $i \in [n]$ holds.

We need the following lemmas:

**Lemma 38.** *Let $1 \geq p_1, \ldots, p_r > 0$ and $r \in \mathbb{N}$. For $K \in 2^{[r]}$,*

$$\mu_K := \sum_{S:K\subseteq S} \prod_{i\in S} p_i \prod_{i\in[r]\setminus S} (1-p_i) \prod_{i\in K}\Big(\frac{1}{p_i}\Big) \prod_{i\in S\setminus K}\Big(1-\frac{1}{p_i}\Big) \tag{4.49}$$

*satisfies*

$$\mu_K = \begin{cases} 1 & \textit{if } K = [r] \\ 0 & \textit{otherwise.} \end{cases} \tag{4.50}$$

*Proof.* Let $x_1, \ldots, x_r$ be free variables. For $S \in 2^{[r]}$ we denote the monomial $x^S = \prod_{i\in S} x_i$. We will prove the identity by comparing coefficients of two equal polynomials. We have

$$q(x_1, \ldots, x_r) = \prod_{i=1}^{r} (p_i x_i + (1-p_i)) = \sum_{S\in 2^{[r]}} x^S \prod_{i\in S} p_i \prod_{i\in[r]\setminus S} (1-p_i), \tag{4.51}$$

where we have expanded all $|2^{[r]}|$ monomials appearing in the decomposition of $q$. Now we set $x_i = (1/p_i)y_i - (1-p_i)/p_i$ in (4.51). We have

$$\begin{aligned} q\Big(\frac{1}{p_1}y_1 - \frac{1-p_1}{p_1}, \ldots, \frac{1}{p_r}y_r - \frac{1-p_r}{p_r}\Big) &= \prod_{i=1}^{r}\Big(p_i\Big(\frac{1}{p_i}y_i - \frac{1-p_i}{p_i}\Big) + (1-p_i)\Big) \\ &= \prod_{i=1}^{r} y_i = y^{[r]}. \end{aligned} \tag{4.52}$$

On the other hand, if we substitute $x_i = (1/p_i)y_i - (1-p_i)/p_i$ in the monomials $x^S$ in (4.51) we have

$$\begin{aligned} \sum_{S\in 2^{[r]}} x^S \prod_{i\in S} p_i \prod_{i\in[r]\setminus S} (1-p_i) &= \sum_{S\in 2^{[r]}} \prod_{i\in S}\Big(\frac{1}{p_i}y_i - \frac{1-p_i}{p_i}\Big)p_i \prod_{i\in[r]\setminus S} (1-p_i) \\ &= \sum_{S\in 2^{[r]}} \sum_{K\in 2^{[r]}:K\subseteq S} \prod_{i\in S} p_i \prod_{i\in[r]\setminus S} (1-p_i) \prod_{i\in K} y_i \Big(\frac{1}{p_i}\Big) \prod_{i\in S\setminus K}\Big(-\frac{1-p_i}{p_i}\Big) \\ &= \sum_{K\in 2^{[r]}} y^K \sum_{S:K\subseteq S} \prod_{i\in S} p_i \prod_{i\in[r]\setminus S} (1-p_i) \prod_{i\in K}\Big(\frac{1}{p_i}\Big) \prod_{i\in S\setminus K}\Big(1-\frac{1}{p_i}\Big) \\ &= \sum_{K\in 2^{[r]}} \mu_K y^K, \end{aligned} \tag{4.53}$$

so that we must have $\mu_K = 1$ if $K = [r]$ and zero otherwise. □

Let $[n] = I_1 \cup \ldots, \cup I_r$ be a partition of $[n]$, i.e., $I_j \cap I_i = \emptyset$ if $i \neq j$. We consider $S \in 2^{[r]}$ also as an element of $2^{[n]}$ via the inclusion $\iota : 2^{[r]} \to 2^{[n]}$ given by $j \in \iota(S)$ if $j \in I_i$ and $i \in S$, i.e., we consider the index $i$ as the set of all indices $j \in I_i$. Note then that $\iota([r]) = [n]$. Recall now that the filter random variables with values in $\{0,1\}^n$ are denoted by $f = (f_1, \cdots, f_n)$ . We suppose now that the filter random variables satisfy $f_i = f_j$ whenever $i, j \in I_s$ for some $s \in [r]$. We denote by $B_s$ the $\{0,1\}$-valued random variable corresponding to the $I_s$ part of $[n]$. We suppose that $\mathbb{P}(B_s = 1) = p_s$ for all $s \in [r]$, where $p_s$ is the probability of success and $1 - p_s$ the dropout probability. Moreover, we suppose that the $(B_s)_{s \in [r]}$ are mutually independent. With this notation we have:

$$\mathbb{E}\big[\Psi(\cdot, w \odot f)\big] = \sum_{L \in 2^{[r]}} \prod_{i \in L} p_i \prod_{i \in [r] \setminus L} (1 - p_i) \Psi(\cdot, w \odot \mathbf{1}_{\iota(L)}). \tag{4.54}$$

In the following lemma, we embed $2^{[r]}$ into $2^{[n]}$ as blocks according to a partition of $[n]$ using $\iota$:

**Lemma 39.** *For $S \in 2^{[r]}$,*

$$\mathbb{E}\big[\Psi(\cdot, (w \odot \mathbf{1}_{\iota(S)}) \odot f)\big] = \sum_{K \in 2^{[r]} : K \subseteq S} \prod_{i \in K} p_i \prod_{i \in S \setminus K} (1 - p_i) \Psi\left(\cdot, w \odot \mathbf{1}_{\iota(K)}\right). \tag{4.55}$$

*Proof.* Let $S \in 2^{[r]}$. Observe that $f = \sum_{s=1}^r \mathbb{1}[B_s = 1] \mathbf{1}_{I_s}$ and note in particular that

$$g := \mathbf{1}_{\iota(S)} \odot f = \Big(\sum_{s \in S} + \sum_{s \in S^c}\Big) \mathbb{1}[B_s = 1] \mathbf{1}_{\iota(S) \cap I_s} = \sum_{s \in S} \mathbb{1}[B_s = 1] \mathbf{1}_{I_s}. \tag{4.56}$$

Hence, $g$ depends only on $(B_s)_{s \in S}$ and is thus moreover independent of $(B_t)_{t \in S^c}$ by assumption. Consequently $\Psi(\cdot, w \odot g)$ also depends only on $(B_s)_{s \in S}$ and is also independent of $(B_t)_{t \in S^c}$. The result then follows.

To see this in detail, suppose that $S = \{s_1, \ldots, s_m\}$ and $S^c = \{t_1, \ldots, t_{r-m}\}$ say. By expanding the expectation together with (i) independence to conclude that

$$\begin{aligned}
\mathbb{E}[\Psi(\cdot, w \odot g)] &\overset{(4.56)}{=} \sum_{b_1=0}^{1} \cdots \sum_{b_r=0}^{1} \Psi(\cdot, w \odot \sum_{s \in S} \mathbb{1}[b_s = 1] \mathbf{1}_{I_s}) \mathbb{P}[B_1 = b_1, \ldots, B_r = b_r] \\
&\overset{\text{(i)}}{=} \sum_{b_{s_1}=0}^{1} \cdots \sum_{b_{s_m}=0}^{1} \Psi(\cdot, w \odot \sum_{i=1}^{m} \mathbb{1}[b_{s_i} = 1] \mathbf{1}_{I_{s_i}}) \mathbb{P}[\cap_{i=1}^m \{B_{s_i} = b_{s_i}\}] \\
&\quad \times \underbrace{\sum_{b_{t_1}=0}^{1} \cdots \sum_{b_{t_{r-m}}=0}^{1} \mathbb{P}[\cap_{j=1}^{r-m} \{B_{t_j} = b_{t_j}\}]}_{=1 \text{ as an axiom of the pdf}}.
\end{aligned} \tag{4.57}$$

Substitute

$$\mathbb{P}[\cap_{i=1}^m \{B_{s_i} = b_{s_i}\}] \overset{\text{(i)}}{=} \prod_{i=1}^{m} \mathbb{P}[B_{s_i} = b_{s_i}] = \prod_{i=1}^{m} p_{s_i}^{b_{s_i}} (1 - p_{s_i})^{1 - b_{s_i}} \tag{4.58}$$

and then apply the change of variables $K(b_{s_1}, \ldots, b_{s_m}) = \cup_{i=1}^m \{s_i : b_{s_i} = 1\}$ to identify the right-hand side of (4.55). □

We can now prove Proposition 16:

*Proof.* (of Proposition 16) In the same notation as in Lemmas 38 and 39,, we use $p_s$ as the success probability. Then, we can write

$$\begin{aligned}
&\sum_{V\in 2^{[r]}}\prod_{i\in V}\Big(\frac{1}{p_i}\Big)\prod_{i\in[r]\setminus V}\Big(1-\frac{1}{p_i}\Big)\mathbb{E}(\Psi(\cdot,(w\odot\mathbf{1}_{\iota(V)})\odot f)\\
&\overset{\text{(Lemma 39)}}{=}\sum_{V\in 2^{[r]}}\prod_{i\in V}\Big(\frac{1}{p_i}\Big)\prod_{i\in[r]\setminus V}\Big(1-\frac{1}{p_i}\Big)\sum_{K:K\subseteq V\in 2^{[r]}}\prod_{i\in K}p_i\prod_{i\in V\setminus K}(1-p_i)\Psi(\cdot,w\odot\mathbf{1}_{\iota(K)})\\
&=\sum_{V\in 2^{[r]}}\sum_{K:K\subseteq V\in 2^{[r]}}\prod_{i\in V}\Big(\frac{1}{p_i}\Big)\prod_{i\in[r]\setminus V}\Big(1-\frac{1}{p_i}\Big)\prod_{i\in K}p_i\prod_{i\in V\setminus K}(1-p_i)\Psi(\cdot,w\odot\mathbf{1}_{\iota(K)})\\
&=\sum_{K\in 2^{[r]}}\Psi(\cdot,w\odot\mathbf{1}_{\iota(K)})\sum_{V:K\subseteq V\in 2^{[r]}}\Big(\frac{1}{p_i}\Big)\prod_{i\in[r]\setminus V}\Big(1-\frac{1}{p_i}\Big)\prod_{i\in K}p_i\prod_{i\in V\setminus K}(1-p_i)\\
&=\sum_{K\in 2^{[r]}}\Psi(\cdot,w\odot\mathbf{1}_{\iota(K)})\mu_K\\
&\overset{\text{(Lemma 38)}}{=}\Psi(\cdot,w\odot\mathbf{1}_{\iota([r])})\\
&=\Psi(\cdot,w).
\end{aligned}\tag{4.59}$$

With this last step we finally obtain Proposition 16. □

## 4.B Proofs of Section 4.4

In this appendix we prove the results of Section 4.4. In particular we prove Lemmas 34, 35 and 36 as well as Theorem 28.

### 4.B.1 Proof of Lemma 34

Let $\Gamma$ be a dropout tree. Let $\ell$ be a leaf of $\Gamma$ at level $k>1$. Let $\mu$ be the distribution of a random matrix $F\in\{0,1\}^{d_k\times d_{k-1}}$ that satisfies for all $r,c$, $\mathbb{P}[F_{rc}=1]\geq\beta>0$. Assume that $\Gamma$ satisfies $\mathsf{ApProp}_\Gamma(\delta,\epsilon)$, i.e.,

$$\mathbb{P}\Big[\underline{\sup}_{x\in B(0,R)}\underline{\sup}_{\tilde{x}\in B(\mathsf{In}_\Gamma(x),\delta)}\big|\Phi_\Gamma^{v_0}((\tilde{x}^\ell)_\ell)-\Phi_{\Gamma_{\text{det}}}^{v_0}(\mathsf{In}_\Gamma(x))\big|>\frac{\epsilon}{2}\Big]<\Big(\frac{\epsilon}{4R_L}\Big)^q.$$

Define $\kappa>0$ by

$$\kappa:=\Big(\frac{\epsilon}{4R_L}\Big)^q-\mathbb{P}\Big[\underline{\sup}_{x\in B(0,R)}\underline{\sup}_{\tilde{x}\in B(\mathsf{In}_\Gamma(x),\delta)}\big|\Phi_\Gamma^{v_0}(x)-\Phi_{\Gamma_{\text{det}}}^{v_0}(\tilde{x})\big|>\frac{\epsilon}{2}\Big].$$

By Lemma 35, there exists an $\eta>0$ and an $N_0\in\mathbb{N}$ such that for all $N\geq N_0$, if $F^i$ are independent, identically distributed filter matrices distributed according to $\mu$, and if $V$ is given by (4.25), then

$$\mathbb{P}\Big[\underline{\sup}_{z\in B(0,R_k)}\underline{\sup}_{(\tilde{z})^i\in B(x,\eta)^N}\Big|\sigma_k\Big(\frac{1}{N}\sum_{i=1}^N(V\odot F^i)\tilde{z}^i+b^{(k)}\Big)-\sigma_k\big(W^{(k)}z+b^{(k)}\big)\Big|>\delta\Big]<\kappa.$$

Now let $\Gamma'$ be a $\mu$-input-copy of $\Gamma$ at $\ell$ of size $N \geq N_0$. Choose $\delta' := \min(\delta, \eta)$.

Consider the event $\mathcal{A}$ that

$$\sup_{x \in \overline{B(0,R)}} \sup_{\tilde{x} \in \overline{B(\mathsf{In}_\Gamma(x),\delta)}} \left| \Phi_\Gamma^{v_0}((\tilde{x}^\ell)_\ell) - \Phi_{\Gamma_{\mathrm{det}}}^{v_0}(\mathsf{In}_\Gamma(x)) \right| > \frac{\epsilon}{2},$$

which (informally) means that the tree $\Gamma$ provides a bad approximation. Consider also the event $\mathcal{B}$ that

$$\sup_{z \in \overline{B(0,R_{k-1})}} \sup_{(\tilde{z}^m) \in \overline{B(z,\eta)^N}} \left| \sigma_k\Big( \sum_{e \in \mathsf{into}(\ell)} (V^e \odot F^e)\tilde{z}^m + b^e \Big) - \sigma_k(W^e z + b^e) \right| > \delta.$$

Here, $\mathsf{into}(\ell)$ refers to the dropout–tree $\Gamma'$. This event (informally) means that the added part provides a bad approximation. Note that

$$\mathbb{P}[\mathcal{A} \cup \mathcal{B}] \leq \mathbb{P}[\mathcal{A}] + \mathbb{P}[\mathcal{B}] < \mathbb{P}[\mathcal{A}] + \kappa = \Big( \frac{\epsilon}{4R_L} \Big)^q.$$

Next, let us show that on $(\mathcal{A} \cup \mathcal{B})^c$ one has

$$\sup_{x \in \overline{B(0,R)}} \sup_{\tilde{x} \in \overline{B(\mathsf{In}_{\Gamma'}(x),\delta')}} \left| \Phi_{\Gamma'}^{v_0}((\tilde{x}^\ell)_\ell) - \Phi_{\Gamma_{\mathrm{det}}}^{v_0}(\mathsf{In}_{\Gamma'}(x)) \right| \leq \frac{\epsilon}{2}. \tag{4.60}$$

To do this, suppose that $(\mathcal{A} \cup \mathcal{B})^c$ holds and $x \in \overline{B(0,R)}$. For every leaf $m \in \mathsf{children}(\ell)$ in $\Gamma'$ define $z := \mathsf{In}_{\Gamma'}^m(x)$ (recall the definition from (4.27)) and note that $z \in \overline{B(0,R_{k-1})}$. Let $\tilde{x} \in B(\mathsf{In}_{\Gamma'}(x), \delta')$. For every leaf $m \in \mathsf{children}(\ell)$ define $\tilde{z}^m := \tilde{x}^m \in B(z,\eta)$ by the choice of $\delta'$. Since $\mathcal{B}^c$ holds,

$$\left| \sigma_k\Big( \sum_{e \in \mathsf{into}(\ell)} (V^e \odot F^e)\tilde{z}^i + b^e \Big) - \sigma_k(W^e z + b^e) \right| \leq \delta,$$

or, in other words,

$$\sigma_k\Big( \sum_{e \in \mathsf{into}(\ell)} (V^e \odot F^e)\tilde{z}^i + b^e \Big) \in \overline{B(\mathsf{In}_\Gamma^\ell(x), \delta)}.$$

Together with $\mathcal{A}^c$ this implies (4.60).

Finally, by the law of total probability, we estimate

$$\begin{aligned}
&\mathbb{P}\Big[ \sup_{x \in \overline{B(0,R)}} \sup_{\tilde{x} \in \overline{B(\mathsf{In}_{\Gamma'}(x),\delta')}} \left| \Phi_{\Gamma'}^{v_0}((\tilde{x}^\ell)_\ell) - \Phi_{\Gamma_{\mathrm{det}}}^{v_0}(\mathsf{In}_{\Gamma'}(x)) \right| > \frac{\epsilon}{2} \Big] \\
&= \mathbb{P}\Big[ \sup_{x \in \overline{B(0,R)}} \sup_{\tilde{x} \in \overline{B(\mathsf{In}_{\Gamma'}(x),\delta')}} \left| \Phi_{\Gamma'}^{v_0}((\tilde{x}^\ell)_\ell) - \Phi_{\Gamma_{\mathrm{det}}}^{v_0}(\mathsf{In}_{\Gamma'}(x)) \right| > \frac{\epsilon}{2} \Big| \mathcal{A} \cup \mathcal{B} \Big] \mathbb{P}[\mathcal{A} \cup \mathcal{B}] \\
&\quad + \mathbb{P}\Big[ \sup_{x \in \overline{B(0,R)}} \sup_{\tilde{x} \in \overline{B(\mathsf{In}_{\Gamma'}(x),\delta')}} \left| \Phi_{\Gamma'}^{v_0}((\tilde{x}^\ell)_\ell) - \Phi_{\Gamma_{\mathrm{det}}}^{v_0}(\mathsf{In}_{\Gamma'}(x)) \right| > \frac{\epsilon}{2} \Big| (\mathcal{A} \cup \mathcal{B})^c \Big] \mathbb{P}[(\mathcal{A} \cup \mathcal{B})^c] \\
&< \Big( \frac{\epsilon}{4R_L} \Big)^q + 0 = \Big( \frac{\epsilon}{4R_L} \Big)^q.
\end{aligned} \tag{4.61}$$

This completes the proof of Lemma 34.

□

### 4.B.2 Proof of Lemma 35

Let $0 \le K < \infty$ and $\rho > 0$ be fixed. For every $N < \infty$, the suprema in (4.30) over $x, (\tilde{x}^i)$ are in fact attained—say at $X_*, (\tilde{X}_*^i)$—because $\sigma$ is continuous and the optimization domain is closed and bounded. We need to now be careful because $X_*, (\tilde{X}_*^i)$ depend on the collection $\{F^i\}_{i\in[N]}$.

Recall that the continuity of $\sigma$ implies that $\sigma$ is also uniformly continuous on each compact set, i.e., for every $\zeta > 0$ there exists an $\eta_\zeta > 0$ such that for all $x,y$ from this compact set

$$|x-y| < \eta_\zeta \Rightarrow |\sigma(y)-\sigma(x)| < \zeta. \tag{4.62}$$

Define $\bar{X}_* := (1/N)\sum_{i=1}^N \tilde{X}_*^i$. Then uniform continuity of $\sigma$ implies that

$$\big|\sigma\big(W\bar{X}_*+b\big)-\sigma(Wx+b)\big| \le \sup_{x\in\overline{B(0,K)}} \sup_{y\in\overline{B(x,\delta)}} \big|\sigma\big(Wy+b\big)-\sigma(Wx+b)\big| =: \gamma_\delta < \infty. \tag{4.63}$$

Moreover, $\gamma_\delta$ is independent of $N$. Remark also that

$$\sup_{f\in\{0,1\}^{m\times n}} \sup_{y\in\overline{B(0,\delta)}} \frac{1}{\mathbb{E}[F]}\big|\big(W\odot f - W\odot\mathbb{E}[F]\big)y\big| =: c_\delta < \infty, \tag{4.64}$$

where $f$ stands for all possible deterministic realizations of the filters $F$. Again, $c_\delta$ is independent of $N$. Finally, by construction there exists a compact set $\mathcal{C} \subset \mathbb{R}^m$ such that the points

$$\frac{1}{N}\sum_{i=1}^N (V\odot F^i)\tilde{X}_*^i + b, \quad W\bar{X}_* + b \tag{4.65}$$

lie in $\mathcal{C}$ with probability one.

First, fix $\zeta = \rho/2$. From uniform continuity of $\sigma$ on the compact set $\mathcal{C}$ there exists $\eta_\zeta > 0$ such that (4.62) holds for all $x,y\in\mathcal{C}$. Second, observe that $\gamma_\delta \to 0$ and $c_\delta \to 0$ as $\delta \to 0$. Hence we can choose $\delta$ and fix it such that

$$0 < \zeta < \rho - \gamma_\delta \quad \text{and} \quad c_\delta < \eta_\zeta. \tag{4.66}$$

Combining (4.63) with the triangle inequality and using (4.66), we arrive at

$$\text{LHS (4.30)} \le \mathbb{P}\Big[\underbrace{\Big|\sigma\Big(\frac{1}{N}\sum_{i=1}^N (V\odot F^i)\tilde{X}_*^i + b\Big) - \sigma\big(W\bar{X}_*+b\big)\Big|}_{=:Z} > \rho - \gamma_\delta\Big]. \tag{4.67}$$

Consider now the event

$$\mathcal{E} = \Big\{\Big|\frac{1}{N}\sum_{i=1}^N (V\odot F^i)\tilde{X}_*^i - W\bar{X}_*\Big| < \eta_\zeta\Big\}. \tag{4.68}$$

Then by the law of total probability and uniform continuity,

$$\begin{aligned}\mathbb{P}[Z > \rho-\gamma_\delta] &= \mathbb{P}[Z>\rho-\gamma_\delta|\mathcal{E}]\mathbb{P}[\mathcal{E}] + \mathbb{P}[Z>\rho-\gamma_\delta|\mathcal{E}^{\mathrm{c}}]\mathbb{P}[\mathcal{E}^{\mathrm{c}}] \\ &\le \mathbb{1}[\zeta > \rho-\gamma_\delta]\mathbb{P}[\mathcal{E}] + \mathbb{P}[\mathcal{E}^{\mathrm{c}}] \overset{(4.66)}{=} \mathbb{P}[\mathcal{E}^{\mathrm{c}}].\end{aligned} \tag{4.69}$$

We proceed by bounding $\mathbb{P}[\mathcal{E}^{\mathrm{c}}]$. Use the triangle inequality twice to establish that for any $x \in \overline{B(0,K)}$,

$$\begin{aligned}
&\Big|\frac{1}{N}\sum_{i=1}^{N}(V \odot F^i)\tilde{X}_*^i - W\bar{X}_*\Big| \\
&= \Big|\frac{1}{N\mathbb{E}[F]}\sum_{i=1}^{N}\big((W \odot F^i) - W \odot \mathbb{E}[F]\big)\big(x + (\tilde{X}_*^i - x)\big)\Big| \\
&\le \Big|\frac{1}{N\mathbb{E}[F]}\sum_{i=1}^{N}\big((W \odot F^i) - W \odot \mathbb{E}[F]\big)x\Big| + \Big|\frac{1}{N\mathbb{E}[F]}\sum_{i=1}^{N}\big((W \odot F^i) - W \odot \mathbb{E}[F]\big)(\tilde{X}_*^i - x)\Big| \\
&\overset{(4.64)}{\le} \Big|\frac{1}{N\mathbb{E}[F]}\sum_{i=1}^{N}\big((W \odot F^i) - W \odot \mathbb{E}[F]\big)x\Big| + c_\delta. \qquad (4.70)
\end{aligned}$$

Note now additionally that by the matrix version of the weak law of large numbers,

$$\frac{1}{N}\sum_{i=1}^{N} W \odot F^i \xrightarrow{\mathbb{P}} W \odot \mathbb{E}[F] \quad \text{as} \quad N \to \infty. \tag{4.71}$$

Therefore, using (4.66), we get as $N \to \infty$

$$\mathbb{P}[\mathcal{E}^{\mathrm{c}}] \overset{(4.70)}{\le} \mathbb{P}\Big[\Big|\frac{1}{N\mathbb{E}[F]}\sum_{i=1}^{N}\big((W \odot F^i) - W \odot \mathbb{E}[F]\big)x\Big| \ge \eta_\zeta - c_\delta\Big] \overset{(4.71)}{\to} 0. \tag{4.72}$$

Bounding (4.69) by (4.72) completes the proof. □

### 4.B.3 Proof of Theorem 28

We start by showing that for $\alpha$ small enough and for $N$ large enough,

$$\mathbb{P}\Big[\sup_{x \in \overline{B(0,R)}} \big|\mathsf{NN}_{\Gamma,\Xi}(x) - \Psi(x,w)\big| > \frac{\epsilon}{2}\Big] < \Big(\frac{\epsilon}{4R_L}\Big)^q. \tag{4.73}$$

Afterwards, we deduce the three assertions (4.41)–(4.43) from (4.73).

*Proof of* (4.73). Recall that the assumption $\mathsf{ApProp}_\Gamma(\delta,\epsilon)$ means that

$$\mathbb{P}\Big[\sup_{x \in \overline{B(0,R)}} \sup_{\tilde{x} \in \overline{B(\mathsf{In}_\Gamma(x),\delta)}} \big|\Phi^{v_0}_{\Gamma,\Xi}(\tilde{x}) - \Phi^{v_0}_{\Gamma_{\mathrm{det}},\Xi}(\mathsf{In}_\Gamma(x))\big| > \frac{\epsilon}{2}\Big] < \Big(\frac{\epsilon}{4R_L}\Big)^q.$$

Define therefore $\kappa > 0$ by

$$\kappa := \frac{1}{\#\mathsf{leaves}(\Gamma)}\bigg(\Big(\frac{\epsilon}{4R_L}\Big)^q - \mathbb{P}\Big[\sup_{x \in \overline{B(0,R)}} \sup_{\tilde{x} \in \overline{B(\mathsf{In}_\Gamma(x),\delta)}} \big|\Phi^{v_0}_{\Gamma,\Xi}(\tilde{x}) - \Phi^{v_0}_{\Gamma_{\mathrm{det}},\Xi}(\mathsf{In}_\Gamma(x))\big| > \frac{\epsilon}{2}\Big]\bigg). \tag{4.74}$$

Observe now that the function $\Psi_1$ is continuous, and the function $\Phi^{v_0}_{\Gamma_{\mathrm{det}}}$ is continuous on $(\mathbb{R}^{d_1})^{\mathsf{leaves}(\Gamma)}$. Since this implies uniform continuity on compact sets (see (4.62)), there exists a $\zeta > 0$ such that whenever a function $g : \overline{B(0,R)} \to \mathsf{Inp}_\Gamma$ satisfies

$$\sup_{x \in \overline{B(0,R)}} \sup_{\ell \in \mathsf{leaves}(\Gamma)} \big|g^\ell(x) - \Psi_1(x;(W^{(1)},b^{(1)}))\big| < \zeta,$$

then we also have

$$\sup_{x\in B(0,R)}\Big|\Phi^{v_0}_{\Gamma_{\mathrm{det}}}(g(x)) - \Psi(x,w)\Big| < \epsilon. \tag{4.75}$$

Now choose

$$\rho := \min(\delta/2, \kappa, \zeta) \qquad \text{and} \qquad K := R,$$

which we use as parameters for Lemma 36. This choice ensures that for $\alpha$ small enough and $N$ large enough, for all leaves $\ell$ of $\Gamma$, by inequality (4.44)

$$\mathbb{P}\Big[\sup_{x\in B(0,R)}\big|\Xi^{\ell}(x) - \Psi_1(x;(W^{(1)},b^{(1)}))\big| > \delta/2\Big] < \kappa, \tag{4.76}$$

and by inequality (4.45)

$$\sup_{x\in B(0,R)}\big|\Xi^{\ell,\mathsf{avg-filt}}(x) - \Psi_1(x;(W^{(1)},b^{(1)}))\big| < \zeta. \tag{4.77}$$

Consider now the event $\mathcal{A}$ that there exists a leaf $\ell$ of $\Gamma$ such that

$$\sup_{x\in B(0,R)}|\Xi^{\ell}(x) - \Psi_1(x;(W^{(1)},b^{(1)}))| > \delta/2.$$

From the law of total probability, it follows that

$$\begin{aligned}
&\mathbb{P}\Big[\sup_{x\in B(0,R)}\big|\mathsf{NN}_{\Gamma,\Xi}(x) - \Psi(x,w)\big| > \frac{\epsilon}{2}\Big]\\
&= \mathbb{P}\Big[\sup_{x\in B(0,R)}\big|\mathsf{NN}_{\Gamma,\Xi}(x) - \Psi(x,w)\big| > \frac{\epsilon}{2}\Big|\mathcal{A}\Big]\mathbb{P}[\mathcal{A}]\\
&\quad + \mathbb{P}\Big[\sup_{x\in B(0,R)}\big|\mathsf{NN}_{\Gamma,\Xi}(x) - \Psi(x,w)\big| > \frac{\epsilon}{2}\Big|\mathcal{A}^c\Big]\mathbb{P}[\mathcal{A}^c].
\end{aligned} \tag{4.78}$$

Observe that

$$\mathbb{P}\Big[\sup_{x\in B(0,R)}\big|\mathsf{NN}_{\Gamma,\Xi}(x) - \Psi(x,w)\big| > \frac{\epsilon}{2}\Big|\mathcal{A}\Big] \le 1, \tag{4.79}$$

and by (i) Boole's inequality

$$\begin{aligned}
\mathbb{P}[\mathcal{A}] &= \mathbb{P}\Big[\cup_{\ell\in\mathsf{leaves}(\Gamma)}\Big\{\sup_{x\in B(0,R)}|\Xi^{\ell}(x) - \Psi_1(x;(W^{(1)},b^{(1)}))| > \delta/2\Big\}\Big]\\
&\overset{(i)}{\le} \sum_{\ell\in\mathsf{leaves}(\Gamma)}\mathbb{P}\Big[\sup_{x\in B(0,R)}|\Xi^{\ell}(x) - \Psi_1(x;(W^{(1)},b^{(1)}))| > \delta/2\Big] \overset{(4.76)}{\le} \kappa\cdot\#\mathsf{leaves}(\Gamma).
\end{aligned} \tag{4.80}$$

Furthermore,

$$\begin{aligned}
&\mathbb{P}\Big[\sup_{x\in B(0,R)}\big|\mathsf{NN}_{\Gamma,\Xi}(x) - \Psi(x,w)\big| > \frac{\epsilon}{2}\Big|\mathcal{A}^c\Big]\\
&\overset{(4.75)}{\le} \mathbb{P}\Big[\sup_{x\in B(0,R)}\sup_{\tilde{x}\in B(\mathsf{In}_{\Gamma}(x),\delta)}\big|\Phi^{v_0}_{\Gamma,\Xi}(\tilde{x}) - \Phi^{v_0}_{\Gamma_{\mathrm{det}},\Xi}(\mathsf{In}_{\Gamma}(x))\big| > \frac{\epsilon}{2}\Big].
\end{aligned} \tag{4.81}$$

By bounding (4.78) using (4.79)–(4.81) and $\mathbb{P}[\mathcal{A}^c] \leq 1$, we find that

$$\begin{aligned}
&\mathbb{P}\Big[\sup_{x\in\overline{B(0,R)}}\big|\mathsf{NN}_{\Gamma,\Xi}(x)-\Psi(x,w)\big| > \frac{\epsilon}{2}\Big]\\
&< \kappa\cdot\#\mathsf{leaves}(\Gamma)+\mathbb{P}\Big[\sup_{x\in\overline{B(0,R)}}\sup_{\tilde{x}\in B(\mathsf{ln}_\Gamma(x),\delta)}\big|\Phi^{v_0}_{\Gamma,\Xi}(\tilde{x})-\Phi^{v_0}_{\Gamma_{\det},\Xi}(\mathsf{ln}_\Gamma(x))\big| > \frac{\epsilon}{2}\Big]\cdot 1\\
&\overset{(4.74)}{=} \Big(\frac{\epsilon}{4R_L}\Big)^q. \qquad (4.82)
\end{aligned}$$

This shows (4.73).

Next, we prove that (4.41)–(4.43) follow from (4.73).

*Proof of* (4.41). This inequality follows from (4.73) since $R_L \geq 1$ by construction and $q \geq 1$ by assumption.

*Proof of* (4.43). This inequality is a direct consequence of inequality (4.75) by choosing $g^\ell := \Xi^{\ell,\mathsf{avg-filt}}$ and using (4.77).

*Proof of* (4.42). We will prove that by the definition of $R_j$ in (4.26), for all $x \in \overline{B(0,R)}$

$$\big|\mathsf{NN}_{\Gamma,\Xi}(x)\big|^q < R_L^q \quad \text{w.p. one,} \quad \text{and} \quad \big|\Psi(x,w)\big|^q < R_L^q. \qquad (4.83)$$

Namely, if (4.83) holds true, then (4.42) follows.

To see the implication, consider the event $\mathcal{D}$ for which

$$\sup_{x\in\overline{B(0,R)}}\big|\mathsf{NN}_{\Gamma,\Xi}(x)-\Psi(x,w)\big| > \frac{\epsilon}{2}, \qquad (4.84)$$

and apply the law of total expectation:

$$\begin{aligned}
&\mathbb{E}\Big[\sup_{x\in\overline{B(0,R)}}\big|\mathsf{NN}_{\Gamma,\Xi}(x)-\Psi(x,w)\big|^q\Big] \qquad (4.85)\\
&=\mathbb{E}\Big[\sup_{x\in\overline{B(0,R)}}\big|\mathsf{NN}_{\Gamma,\Xi}(x)-\Psi(x,w)\big|^q\Big|\mathcal{D}\Big]\mathbb{P}[\mathcal{D}]+\mathbb{E}\Big[\sup_{x\in\overline{B(0,R)}}\big|\mathsf{NN}_{\Gamma,\Xi}(x)-\Psi(x,w)\big|^q\Big|\mathcal{D}^c\Big]\mathbb{P}[\mathcal{D}^c].
\end{aligned}$$

By the triangle inequality,

$$\mathbb{E}\Big[\sup_{x\in\overline{B(0,R)}}\big|\mathsf{NN}_{\Gamma,\Xi}(x)-\Psi(x,w)\big|^q\Big|\mathcal{D}\Big] \overset{(4.83)}{\leq} (2R_L)^q. \qquad (4.86)$$

On the other hand,

$$\mathbb{E}\Big[\sup_{x\in\overline{B(0,R)}}\big|\mathsf{NN}_{\Gamma,\Xi}(x)-\Psi(x,w)\big|^q\Big|\mathcal{D}^c\Big] \overset{(4.84)}{\leq} \Big(\frac{\epsilon}{2}\Big)^q. \qquad (4.87)$$

Bound now (4.85) using (4.73), (4.86), (4.87), and the elementary bound $\mathbb{P}[\mathcal{D}^c] \leq 1$ to obtain

$$\mathbb{E}\Big[\sup_{x\in\overline{B(0,R)}}\big|\mathsf{NN}_{\Gamma,\Xi}(x)-\Psi(x,w)\big|^q\Big] < (2R_L)^q\Big(\frac{\epsilon}{4R_L}\Big)^q+\Big(\frac{\epsilon}{2}\Big)^q \leq \epsilon^q.$$

That shows (4.42). What remains is to prove (4.83).

*Proof of* (4.83). Observe immediately that the right inequality in (4.83) follows immediately as $0 < \beta < 1$ and $Q > 1$ (recall the definition of $\Psi$ in (4.7)). Next, we will prove the left inequality in (4.83) by mathematical induction (recall the recursion in (4.34) and (4.35) that defines $\mathsf{NN}_{\Gamma,\Xi}$).

*Base case.* Recall from (4.34) and (4.35) that the induction starts with the functions

$$\Xi^\ell(x) := \sigma_1\Big(\frac{1}{N}\sum_{i=1}^{2N}(-1)^i(V^\ell \odot F^{\ell,i})\sigma_0\big((-1)^i\alpha(I\odot G^{\ell,i})x\big)+b^\ell\Big), \tag{4.88}$$

where element-wise

$$V^\ell_{rc} = \frac{W^{(1)}_{rc}}{\alpha\big(\sigma_- + \sigma_+\big)\mathbb{E}[F^\ell_{rc}]\mathbb{E}[G^\ell_{cc}]}. \tag{4.89}$$

We prove that for every $x \in \overline{B(0,R)}$, the point

$$\frac{1}{N}\sum_{i=1}^{2N}(-1)^i(V^\ell \odot F^{\ell,i})\sigma_0\big((-1)^i\alpha(I\odot G^{\ell,i})x\big) \in \overline{B\big(0,\beta^{-1}\|W^{(1)}\|_{\mathrm{HS}}R_0\big)} \quad \text{w.p. one.} \tag{4.90}$$

In particular, by the definition of $R_1$ in (4.26) (which implicitly deals with the bias $b^\ell$), this implies that for all $x \in \overline{B(0,R)}$,

$$\big|\Xi^\ell(x)\big| \le R_1 - 1.$$

Start by noting that there exists an $\alpha_0 > 0$ such that for all $0 < \alpha \le \alpha_0$ and all $\xi \in [-R,R] \subset \mathbb{R}$ we have

$$|\sigma_0(\alpha\xi)| \le 2(|\sigma_-| + |\sigma_+|)\alpha|\xi|. \tag{4.91}$$

It follows from the bound (4.91) that for all $x \in \overline{B(0,R)}$ and all $i \in [2N]$,

$$\Big|\frac{1}{\alpha(\sigma_- + \sigma_+)\mathbb{E}[G^\ell_{cc}]}\sigma_0\big(\alpha(I\odot G^{\ell,i})x\big)_c\Big| < 2\frac{|\sigma_-| + |\sigma_+|}{|\sigma_- + \sigma_+|}\frac{1}{\beta}|x_c| \quad \text{w.p. one.}$$

Since we assumed in (4.40) that

$$4\frac{|\sigma_-| + |\sigma_+|}{|\sigma_- + \sigma_+|} < Q,$$

it follows that for all $x \in \overline{B(0,R)}$ and all $i \in [2N]$,

$$\Big|2\frac{\mathbb{E}[I\odot G^\ell]^{-1}}{\alpha(\sigma_- + \sigma_+)}\sigma_0\big(\alpha(I\odot G^{\ell,i})x\big)\Big| < \frac{Q}{\beta}R < R_0 \quad \text{w.p. one.}$$

Therefore, for every $x \in \overline{B(0,R)}$,

$$\begin{aligned}
&\Big|\frac{1}{N}\sum_{i=1}^{2N}(-1)^i(V^\ell \odot F^{\ell,i})\sigma_0\big((-1)^i\alpha(I\odot G^{\ell,i})x\big)\Big| \\
&= \Big|\frac{1}{2N}\sum_{i=1}^{2N}(-1)^i((W^{(1)}\odot F^{\ell,i}) \div \mathbb{E}[F^{\ell,i}])\frac{2\mathbb{E}[I\odot G^{\ell,i}]^{-1}}{\alpha(\sigma_- + \sigma_+)}\sigma_0\big((-1)^i\alpha(I\odot G^{\ell,i})x\big)\Big| \\
&\le \frac{1}{2N}\sum_{i=1}^{2N}\|(W^{(1)}\odot F^{\ell,i}) \div \mathbb{E}[F^{\ell,i}]\|_{\mathrm{HS}}R_0 \le \frac{1}{\beta}\|W^{(1)}\|_{\mathrm{HS}}R_0 \quad \text{w.p. one.}
\end{aligned}$$

This proves (4.90).

*Inductive step.* In the definition of $\mathsf{NN}_{\Gamma,\Xi}$ we defined for $v$ not a leaf in $\Gamma$,

$$\Phi^v_{\Gamma,\Xi} = \sigma_v\Big(\frac{1}{\#\mathsf{into}(v)} \sum_{e\in\mathsf{into}(v)} (V^e \odot F^e)\Phi^{\mathsf{source}(e)}_{\Gamma,\Xi} + b^e\Big).$$

By an inductive argument we find that for all $x \in \overline{B(0,R)}$, it holds that

$$\Big|\frac{1}{\#\mathsf{into}(v)} \sum_{e\in\mathsf{into}(v)} (V^e \odot F^e)\Phi^{\mathsf{source}(e)}_{\Gamma,\Xi}(x)\Big| \leq \beta^{-1}\|W^e\|_{\mathrm{HS}} R_{\mathsf{level}(v)-1} \quad \text{w.p. one.}$$

so that by definition of $R_{\mathsf{level}(v)}$ it holds that

$$|\Phi^v_{\Gamma,\Xi}(x)| < R_{\mathsf{level}(v)} \quad \text{w.p. one.}$$

In particular,

$$\big|\mathsf{NN}_{\Gamma,\Xi}(x)\big|^q = \big|\Phi^{v_0}_{\Gamma,\Xi}(x)\big|^q < R_L^q \quad \text{w.p. one.}$$

This proves (4.83). With that, Theorem 28 is proven. □

### 4.B.4 Proof of Lemma 36

*Proof of* (4.44). Let $0 \leq K < \infty$, $\ell$ be a leaf of $\Gamma$, and $\rho > 0$. Recall that

$$\Psi_1(x;(W^{(1)},b^{(1)})) = \sigma_1\big(W^{(1)}x + b^{(1)}\big), \tag{4.92}$$

and for $x \in \overline{B(0,K)}$, define

$$Z^\ell_N(x) = \frac{1}{N}\sum_{i=1}^{2N}(-1)^i(V^\ell \odot F^{\ell,i})\sigma_0\big((-1)^i\alpha(I \odot G^{\ell,i})x + b\big) + b^\ell \tag{4.93}$$

so that $\Xi^\ell(x) = \sigma_1(Z^\ell_N(x))$. Note that the weights $W$ are fixed and therefore uniformly bounded.

Continuity of $\sigma_0$ and $\sigma_1$, boundedness of $F$ and $G$, positivity of $\mathbb{E}[F^\ell_{rc}]$ and $\mathbb{E}[G^\ell_{rc}]$, and compactness of the optimization domain imply that the supremum of the optimization problem is attained—say at $X_* \in \overline{B(0,K)}$. Just like in Appendix 4.B.2, note that $X_*$ is random and depends on the collections $\{F^{\ell,i}\}_{i\in[2N]}$, $\{G^{\ell,i}\}_{i\in[2N]}$. In summary, we have

$$\begin{aligned}&\mathbb{P}\Big[\sup_{x\in\overline{B(0,K)}}\big|\sigma_1(Z^\ell_N(x)) - \sigma_1(W^{(1)}x + b^{(1)})\big| > \rho\Big]\\ &= \mathbb{P}\Big[\big|\sigma_1(Z^\ell_N(X_*)) - \sigma_1(W^{(1)}X_* + b^{(1)})\big| > \rho\Big].\end{aligned} \tag{4.94}$$

Note that here we slightly abuse the notation by using $|\cdot|$ sign not only for absolute value of numbers, but also, as in the last formula, for the Euclidean norm of the vector.

By construction, there exists a compact set $\mathcal{C}$ so that the points

$$Z^\ell_N(X_*), \quad W^{(1)}X_* + b^{(1)} \tag{4.95}$$

lie in $\mathcal{C}$ with probability one. The uniform continuity of $\sigma_1$ on $\mathcal{C}$ implies that for each $\zeta > 0$ there exists $\eta_\zeta > 0$ such that (4.62) holds for $\sigma_1$ and all $x, y \in \mathcal{C}$. Fix $\zeta = \rho$ and introduce the event

$$\mathcal{D}(X_*) = \big\{ \| Z_N^\ell(X_*) - (W^{(1)} X_* + b^{(1)}) \|_2 < \eta_\rho \big\}. \tag{4.96}$$

By the law of total probability

$$\begin{aligned} &\mathbb{P}\Big[ |\sigma_1(Z_N^\ell(X_*)) - \sigma_1(W^{(1)} X_* + b^{(1)})| > \rho \Big] \\ &= \mathbb{P}\Big[ |\sigma_1(Z_N^\ell(X_*)) - \sigma_1(W^{(1)} X_* + b^{(1)})| > \rho \,|\, \mathcal{D}(X_*) \Big] \mathbb{P}\Big[\mathcal{D}(X_*)\Big] \\ &\quad + \mathbb{P}\Big[ |\sigma_1(Z_N^\ell(X_*)) - \sigma_1(W^{(1)} X_* + b^{(1)})| > \rho \,|\, \mathcal{D}^{\mathrm{c}}(X_*) \Big] \mathbb{P}\Big[\mathcal{D}^{\mathrm{c}}(X_*)\Big] \leq \mathbb{P}\Big[\mathcal{D}^{\mathrm{c}}(X_*)\Big]. \end{aligned} \tag{4.97}$$

We will next prove that for all $x \in \overline{B(0,K)}$,

$$\mathbb{P}\Big[\mathcal{D}^{\mathrm{c}}(x)\Big] \to 0 \tag{4.98}$$

as $\alpha \downarrow 0$ and $N \to \infty$. Together with (4.97), this implies the result.

Let $x \in \overline{B(0,K)}$. Componentwise,

$$\begin{aligned} &\big( Z_N^\ell(x) - (W^{(1)} x + b^{(1)}) \big)_r \\ &= \Big( \sum_{i=1}^{2N} \frac{(-1)^i}{N} (V^\ell \odot F^{\ell,i}) \sigma_0\big( (-1)^i \alpha (I \odot G^{\ell,i}) x \big) + b^\ell - (W^{(1)} x + b^{(1)}) \Big)_r \\ &= \sum_{i=1}^{2N} \sum_{c=1}^{d_0} \frac{(-1)^i}{N} V_{rc}^\ell F_{rc}^{\ell,i} \sigma_0\big( (-1)^i \alpha (I \odot G^{\ell,i}) x \big)_c + b_r^{(1)} - \sum_c W_{rc}^{(1)} x_c + b_r^{(1)}. \end{aligned} \tag{4.99}$$

Substituting (4.39) into (4.99), using the triangle inequality, and rearranging terms, we find that

$$\begin{aligned} &\big| \big( Z_N^\ell(x) - (W^{(1)} x + b^{(1)}) \big)_r \big| \\ &\leq \sum_{c=1}^{d_0} \Big| \frac{W_{rc}^{(1)}}{\frac{1}{2}(\sigma_- + \sigma_+) \mathbb{E}[G_{cc}^\ell]} \Big( \tfrac{1}{2N} \sum_{i=1}^{2N} \frac{F_{rc}^{\ell i}}{\mathbb{E}[F_{rc}^\ell]} (-1)^i \sigma_0\big( (-1)^i \alpha (I \odot G^{\ell,i}) x \big)_c \Big) - W_{rc}^{(1)} x_c \Big|. \end{aligned} \tag{4.100}$$

Note that the assumptions of the lemma imply that

$$\Big| \frac{W_{rc}^{(1)}}{\frac{1}{2}(\sigma_- + \sigma_+) \mathbb{E}[G_{cc}^\ell]} \Big| \leq C_{w,G,\sigma} < +\infty.$$

We focus now on the term within brackets in (4.100). Let $\delta_1 > 0$ and consider the event

$$\mathcal{E}_{F,N}(\delta_1) = \Big\{ \Big| \tfrac{1}{2N} \sum_{i=1}^{2N} \frac{F^{\ell i}}{\mathbb{E}[F_{rc}^\ell]} - 1 \Big| < \delta_1 \Big\}. \tag{4.101}$$

There exists $C_1 > 0$ such that, conditional on $\mathcal{E}_{F,N}(\delta_1)$,

$$\begin{aligned} &\Big| \tfrac{1}{2N} \sum_{i=1}^{2N} \frac{F^{\ell i}}{\mathbb{E}[F_{rc}^\ell]} (-1)^i \sigma_0\big( (-1)^i \alpha (I \odot G^{\ell,i}) x \big)_c - \tfrac{1}{2N} \sum_{i=1}^{2N} (-1)^i \sigma_0\big( (-1)^i \alpha (I \odot G^{\ell,i}) x \big)_c \Big| \\ &\leq \Big| \tfrac{1}{2N} \sum_{i=1}^{2N} \Big( \frac{F^{\ell i}}{\mathbb{E}[F_{rc}^\ell]} - 1 \Big) (-1)^i \sigma_0\big( (-1)^i \alpha (I \odot G^{\ell,i}) x \big)_c \Big| \leq C_1 \delta_1, \end{aligned} \tag{4.102}$$

since the argument of $\sigma_0$ is uniformly bounded, and $\sigma_0$ is continuous. Moreover, there exists $C_2 > 0$ such that conditional on $\mathcal{E}_{F,N}(\delta_1)$,

$$\begin{aligned}&\left|\left(Z_N^\ell(x) - (W^{(1)}x + b^{(1)})\right)_r\right| \\ &\le \sum_{c=1}^{d_0}\left| \frac{W^{(1)}_{rc}}{\frac{1}{2}(\sigma_- + \sigma_+)\mathbb{E}[G^\ell_{cc}]}\left(\tfrac{1}{2N}\sum_{i=1}^{2N}\tfrac{1}{\alpha}\Big((-1)^i\sigma_0\big((-1)^i\alpha(I\odot G^{\ell,i})x\big)_c\Big)\right) - W^{(1)}_{rc}x_c\right| + C_2\delta_1.\end{aligned} \tag{4.103}$$

Note that $C_1, C_2$ are independent of $N, \delta_1$.

Recall now that by (4.36) and (4.37) we can find for each $\gamma > 0$ an $\alpha > 0$ such that for all $y \in \mathbb{R}^{d_0}$, $c$,

$$\left|\frac{1}{\alpha}\Big(\sigma_0\big(\alpha y\big)\Big)_c - \sigma_{S(y_r)}y_c\right| < \gamma. \tag{4.104}$$

Recall furthermore that $\max_{rc} G^\ell_{rc} \le M < \infty$ with probability one by assumption. Together, this implies that we can find for each $\gamma > 0$ an $\alpha > 0$ such that for all $x \in \overline{B(0,K)}$, $c$,

$$\mathbb{P}\left[\left|\frac{1}{\alpha}\Big(\pm\sigma_0\big(\pm\alpha I\odot G^{\ell,i}x\big)\Big)_c - \sigma_{S(\pm G^{\ell i}_{cc}x_c)}G^{\ell i}_{cc}x_c\right| < \gamma\right] = 1. \tag{4.105}$$

Fix $\gamma \in (0,\delta_1)$ and corresponding $\alpha > 0$. Then there exists a constant $C_3 > 0$, independent of $\delta_1, \gamma, N$, such that conditional on $\mathcal{E}_{F,N}(\delta_1)$,

$$\begin{aligned}&\left|\left(Z_N^\ell(x) - (W^{(1)}x + b^{(1)})\right)_r\right| \\ &\le \sum_{c=1}^{d_0}\left| \frac{W^{(1)}_{rc}}{\frac{1}{2}(\sigma_- + \sigma_+)\mathbb{E}[G^\ell_{cc}]}\left(\tfrac{1}{2N}\sum_{i=1}^{2N}\sigma_{S((-1)^iG^{\ell i}_{cc}x_c)}G^{\ell i}_{cc}x_c\right) - W^{(1)}_{rc}x_c\right| + C_3\delta_1 \\ &\overset{\text{(i)}}{=} \sum_{c\in[d_0]:x_c>0}\left| \frac{W^{(1)}_{rc}}{\frac{1}{2}(\sigma_- + \sigma_+)\mathbb{E}[G^\ell_{cc}]}\left(\tfrac{1}{2N}\sum_{i=1}^{2N}\sigma_{S((-1)^iG^{\ell i}_{cc}x_c)}G^{\ell i}_{cc}x_c\right) - W^{(1)}_{rc}x_c\right| + C_3\delta_1.\end{aligned} \tag{4.106}$$

To conclude (i), we used the fact that $\sigma_\pm \cdot 0 = 0$. By assumption $G^{\ell i}_{cc} \ge 0$ with probability one, so if moreover $|x_c| > 0$, then $S((-1)^iG^{\ell i}_{cc}x_c) = S((-1)^ix_c)$ with probability one—recall its definition in (4.37). Thus there exists $C_4$ independent of $\delta_1, \gamma, N$ such that conditional on the event $\mathcal{E}_{F,N}(\delta_1) \cap \mathcal{E}_{G,N}(\delta_1)$,

$$\begin{aligned}&\left|Z_N^\ell(x) - (W^{(1)}x + b^{(1)})\right|_r \\ &\le \sum_{c\in[d_0]:|x_c|>0}\left| \frac{W^{(1)}_{rc}}{\frac{1}{2}(\sigma_- + \sigma_+)}\frac{1}{2N}\sum_{i=1}^{2N}\sigma_{S((-1)^ix_c)}x_c - W^{(1)}_{rc}x_c\right| + C_4\delta_1 = C_4\delta_1.\end{aligned} \tag{4.107}$$

The last equality holds because the sum is only of over $c$ such that $|x_c| > 0$.

All that remains is to prove that

$$\mathbb{P}[\mathcal{E}_{F,N}(\delta_1) \cap \mathcal{E}_{G,N}(\delta_1)] \to 1 \quad \text{as} \quad N \to \infty. \tag{4.108}$$

This fact follows immediately from the independence of $F, G$, and a subsequent application of the matrix version of the weak law of large numbers (which may be applied since $F, G$'s expectations are bounded). Note that $\delta_1$ is an arbitrary parameter: choosing it such that $C_4\delta_1 < \eta_\zeta$, and then choosing $N$ sufficiently large completes the proof of (4.44).

*Proof of* (4.45). The assertion (4.42) is proven for any $(\alpha, N)$-precomposition associated with some distributions $\mu^\ell, \nu^\ell$ with finite nonzero mean. In particular, the same argument shows that (4.42) holds when $F^\ell$ and $G^\ell$ are taken deterministic and equal to the expectations of the corresponding random variables (see also the discussion following (4.39)). This proves (4.45). □

# Part II

# Block Markov Chains

# Chapter 5

# The spectral norm of block Markov chains

Based on [1]:
*"Spectral norm bounds for block Markov chain random matrices"*
by J. Sanders, and A. Senen–Cerda

In this chapter, we show the bounds for the spectral error of Block Markov Chains (BMCs) in full generality. To do so we closely examine the technical issues of proving spectral norm bounds when we have sparse matrices in which entries can weakly depend on each other. For the motivation and an overview of the results we refer to Sections 1.5 to 1.7.

## 5.1 Introduction

In random graph theory, a graph can usually be encoded by an adjacency matrix $\hat{A}_n$, in which $(\hat{A}_n)_{ij} = 1$ if there is an edge between vertices $i$ and $j$ and $(\hat{A}_n)_{ij} = 0$ if there is not. The study of the spectral properties of random graphs is thus intimately related to random matrix theory.

We can already see this in the Erdös–Rényi random graph (ERRG) model [167], where an edge is present independently of others with probability $p_n$. If we denote this model as $\mathcal{G}_{n,p_n}$, then a phase transition in the asymptotic properties of $\hat{A}_n$ for this model can be readily observed depending on how fast the edge probabilities $p_n$ decrease as $n \to \infty$. Specifically, the connectivity of an ERRG $\mathcal{G}_{n,p_n}$ depends on the asymptotics of the average degree of the graph, which in an ERRG is $np_n$. If $p_n = \omega(\log n/n)$, then the random graph will be almost surely connected; and if $p_n = o(\log n/n)$, then the random graph will be almost surely disconnected [167]. In fact, there is a sharp threshold for the connectedness of the random graph exactly at $p_n \asymp \log n/n$. We refer to these scenarios as the dense, sparse, and critical regimes, respectively. In the ERRG, these regimes can be also characterized by the order of the average degree as we described in Section 1.6 in Chapter 1. The existence of these regimes with fundamental different global characteristics means that in order to study properties of the spectrum of $\hat{A}_n$, different approaches are needed. This will also be the case in BMCs.

For any matrix $A \in \mathbb{R}^{n\times n}$, we denote its singular values by $\sigma_1(A) \geq \ldots \geq \sigma_n(A)$, where $\|A\| = \sigma_1(A)$. Feige and Ofek established in [133] that there is a gap between the largest eigenvalue and second-largest absolute eigenvalue of the adjacency matrix $\hat{A}_n$ of ERRGs in the dense regime. In ERRGs, we have that $\|\hat{A}_n\| = (\hat{A}_n) \geq np_n$ with high probability and their results imply that if $\omega(\log n) = np_n = O(n^{1/3}/(\log n)^{5/3})$, then

$$\|\hat{A}_n - \mathbb{E}[\hat{A}_n]\| = O_{\mathbb{P}}(\sqrt{np_n}). \tag{5.1}$$

Consequently, in this dense regime, $\sigma_1(\hat{A}_n) = \omega_{\mathbb{P}}(\sigma_2(\hat{A}_n))$, where for now one can understand the notation $O_{\mathbb{P}}$ and $\omega_{\mathbb{P}}$ as referring to their usual counterparts asymptotically with high probability. We refer to the end of this subsection for precise definitions.

The investigation in [133] also goes into the sparse regime. There, some of the degrees in the graph are much larger than the average $nq_n$ and it can be proved, for example, that if $p_n = d/n$ for $d > 0$ a constant independent of $n$, then $\sigma_1(\hat{A}_n) \geq (1 + o(1))\sqrt{\log(n)/\log\log(n)}$ [139]. Therefore, a bound of the type in (5.1) can not obviously be expected in this sparse regime. However, if $\hat{A}_n$ is regularized by e.g. using only states with degrees lower than a certain threshold, then (5.1) can still be obtained in the sparse regime. Let $\Gamma^{\mathrm{c}} \subseteq [n]$ be the set of vertices in $\mathcal{G}_{n,d/n}$ of degree greater than $(1+\varepsilon)d$ for some appropriately small $\epsilon$. If $A_n^\Gamma$ denotes the adjacency matrix of the subgraph $\mathcal{G}_{n,d/n}^\Gamma$ induced by removing the vertices in $\Gamma^{\mathrm{c}}$ from $\mathcal{G}_{n,d/n}$, then $\|\hat{A}_n^\Gamma - \mathbb{E}[\hat{A}_n]\|{=}O_{\mathbb{P}}(\sqrt{d})$. We refer to [133] for the exact statement. The order of the largest singular values of $\hat{A}_n$ and $\hat{A}_n - \mathbb{E}[A_n]$, established in (5.1), thus persists in the sparse regime when high-degree vertices are removed.

Feige and Ofek's techniques hint at how to tackle the sparse regime also in BMCs. However, conducting such analysis will also become more complicated. Contrary to the ERRG or the Stochastic Block Model (SBM)—the generalization of an ERRG with several clusters— in BMCs there is additionally a time-dimension, and there are correlations between edges.

In this chapter, we aim to establish a bound similar to (5.1) for the largest singular value of a centered random matrix built from a sample path of a BMC in both dense and sparse regimes. Taking inspiration from [27], we establish bounds for the order of the largest singular value by firstly, combining the spectral techniques in [133] with concentration results for Markov chains [98], and secondly, obtaining a lower bound that matches the order of the upper bound. Together, these bounds establish the right asymptotic order of the singular values of BMCs which was, in fact, conjectured in [27]. The bounds depend on the ratio $T_n/n$, which compares the length of a sample path $T_n$ from the BMC to the number of states $n$, and can be understood as an 'average degree'. Unsurprisingly, the spectrum also suffers from similar sparsity-related issues as those that occur within the context of ERRGs, discussed above.

The result in this chapter characterizes the expected bound on the spectral error and hence, determines the limits for the error of the spectral clustering in BMCs. Our result also helps bring us a step closer to proving convergence of the entire spectrum to a limiting distribution as $n \to \infty$ [165] whenever $\Omega(n) = T_n = o(n^2)$ *or even* $\omega(n) = T_n = o(n\log n)$. The latter corresponds to scenarios with particularly few transitions and is thus especially relevant when analyzing sparse real-world data. One such example in Chapter 6 is the sequence of best performing stocks in a stock index. Finally, we will mention that the convergence of the singular values to a limiting distribution whenever there is an abundance of transitions, i.e., $T_n = \Omega(n^2)$, was recently established as $n \to \infty$ [2].

While for a BMC we can expect the order of the spectral error to be as predicted by Theorem 30, in the next chapter, we will examine in real-world sequential data if we can still see a spectral gap when we use the clustering algorithm for BMCs.

**Notation (Asymptotics).** For any two sequences of random variables $X_1, X_2, \ldots$ and $Y_1, Y_2, \ldots$, we denote $X_n = O_{\mathbb{P}}(Y_n)$ if and only if for any $\epsilon > 0$ there exist $C_\epsilon, n_\epsilon > 0$ such that $\mathbb{P}[|X_n/Y_n| > C_\epsilon] \leq \epsilon$ for any $n > n_\epsilon$. We write $X_n = \omega_{\mathbb{P}}(Y_n)$ if and only if for any $\epsilon > 0$ and any $C > 0$, there exist $n_{\epsilon,C} > 0$ such that $\mathbb{P}[|X_n/Y_n| < C] \leq \epsilon$ for any $n > n_{\epsilon,C}$. For any two deterministic sequences $a_1, a_2, \ldots$ and $b_1, b_2, \ldots$, we denote $a_n = \Theta(b_n)$ if $a_n = O(b_n)$ and $b_n = O(a_n)$; also $a_n = \omega(b_n)$ if $b_n = o(a_n)$. We denote $a_n \sim b_n$ if as $n \to \infty$, we have $|a_n/b_n| \to 1$.

### 5.1.1 Block Markov Chains (BMCs)

We have defined the BMC model in Section 1.5 but we briefly recall the definition here for convenience.

A BMC has $n$ labeled states, and $K$ clusters. By this we mean that the set of states $[n] = \{1, \ldots, n\}$ is partitioned so that $[n] = \cup_{k=1}^K \mathcal{V}_k$ with $\mathcal{V}_k \cap \mathcal{V}_l = \emptyset$ for all $k \neq l$. We let $\alpha = (\alpha_1, \ldots, \alpha_K)$ with $\sum_{k=1}^K \alpha_k = 1$ be the *cluster ratios* and let $q = (q_{kl})_{k,l \in [K]}$ with $\sum_{l=1}^K q_{kl} = 1$ for $k \in [K]$ be the *cluster transition matrix.* We will assume the following for the derivation of our results:

**Assumption 29.** *The cluster ratios $\alpha$ are strictly positive, i.e., $\min_{k \in [K]} \alpha_k > 0$. The cluster transition matrix $q$ is strictly positive and has full rank, i.e., $\min_{k,l \in [K]} q_{kl} > 0$ and $\operatorname{rank}(q) = K$. The parameters $\alpha, q, K$ are all independent of $n$.*

Given $n$ and $(\alpha, q)$ we construct a BMC $\{X_t\}_{t \geq 0}$ as follows. For $k = 2, \ldots, K$, assign $|\mathcal{V}_k| = \lfloor n\alpha_k \rfloor$ states to cluster $k$. Place all remaining states in cluster 1 so that $|\mathcal{V}_1| = n - \sum_{k=2}^K |\mathcal{V}_k|$. Notice that $|\mathcal{V}_1| - \lfloor n\alpha_1 \rfloor \leq K - 1$. The BMC $\{X_t\}_{t \geq 0}$ is a time-homogeneous Markov chain with transition matrix $P \in (0,1)^{n \times n}$ that satisfies element-wise

$$P_{x,y} = \mathbb{P}[X_{t+1} = y | X_t = x] = \frac{q_{\nu(x),\nu(y)}}{|\mathcal{V}_{\nu(y)}|} \quad \text{for} \quad x, y \in [n]. \tag{5.2}$$

Here, recall that $\nu : [n] \to [K]$ denotes the function that assigns to each state $x \in [n]$ its cluster $\nu(x) \in [K]$, that is, $\nu$ is the cluster assignment. Assumption 29 guarantees that $P$ is of rank $K$. The assumption that $q$ is strictly positive ensures that the BMC is irreducible and aperiodic. We can therefore let $\Pi \in (0,1)^n$ denote the stationary distribution of the BMC, which satisfies $\Pi^{\mathrm{T}} = \Pi^{\mathrm{T}} P$. It should be noted that a BMC is not necessarily reversible. The assumptions that $\alpha, q, K$ are independent of $n$ guarantee that the BMC has a mixing time of $\Theta(1)$, as we will discuss in Section 5.2. Distinct from the definition of a BMC in [27], we allow for self-jumps for a state. We anticipate that the results of this chapter also hold for the case when self-jumps are not allowed if the proofs are modified appropriately.

Given a BMC and some $T_n \in \mathbb{N}_+$, a sample path $X_0, X_1, \dots, X_{T_n}$ of length $T_n$ is obtained from $\{X_t\}_{t \geq 0}$. Let $\hat{N} \in \mathbb{N}_0^{n \times n}$ denote the random matrix that records the number of transitions that occurred between each pair of states within this sample path. Thus element-wise

$$\hat{N}_{xy} = \sum_{t=0}^{T_n - 1} \mathbb{1}[X_t = x, X_{t+1} = y] \quad \text{for} \quad x, y \in [n]. \tag{5.3}$$

We let $N$ denote $\hat{N}$'s expectation conditional on $X_0 \overset{(\mathrm{d})}{=} \mathrm{Unif}(\Pi)$. The previous definitions imply that $N = T_n \mathrm{Diag}(\Pi) P$.

### 5.1.2 Spectral norm in BMCs

Compared with the previous bound for the spectral norm in 1.19, it was also conjectured in [27] that the order of the spectral norm in BMCs was $\Theta_{\mathbb{P}}(\sqrt{T_n/n})$. This order was also suggested in Chapter 1 with the comparison Table 1.5 between the spectral norms of the ERRG and BMCs. Our first result is a new lower bound of the spectral norm in BMCs which confirms that the order is at least as expected. The proof uses the fact that if $T_n = o(n^2)$, then most transitions are seen only once or not at all. Hence, for any fixed row of $\hat{N} - N$, the $\ell_1$-norm is approximately the square of the $\ell_2$-norm. In particular, we can use a combinatorial argument that yields the correct bound (see Section 5.5).

**Proposition 17.** *If $\omega(n) = T_n = o(n^2)$, then there exist constants $\mathfrak{b}, \mathfrak{c}_{\mathfrak{b}} > 0$ independent of $n$ and an integer $n_0 \in \mathbb{N}_+$ such that for all $n \geq n_0$,*

$$\mathbb{P}\left[\|\hat{N} - N\| > \mathfrak{b}\sqrt{\frac{T_n}{n}}\right] \geq 1 - \mathrm{e}^{-\mathfrak{c}_{\mathfrak{b}} \frac{T_n}{n}}. \tag{5.4}$$

*In particular, $\|\hat{N} - N\| = \Omega_{\mathbb{P}}(\sqrt{T_n/n})$.*

Our second result is an order-wise matching upper bound for $\|\hat{N} - N\|$, which, together with the lower bound of Proposition 17, fully characterizes the order of the spectral norm in BMCs. Before we proceed, note that the asymptotic growth of $T_n$ determines the sparsity of $\hat{N}$. This will dictate the type of analysis that needs to be conducted. We will refer to the scenarios $T_n = \omega(n \log n)$, $T_n = o(n \log n)$ and $T_n = \Theta(n \log n)$ as the dense, sparse, and critical regimes [27], similar to the terminology for ERRGs.

Our analysis in the sparse regime requires that we remove states that are visited unusually often in the BMC, similarly to the technique in [133] used for the ERRG. We therefore consider *trimmed matrices*. For any subset $\Gamma \subseteq [n]$, possibly random, let $\hat{N}_\Gamma$ be the random matrix that remains after setting all entries on the rows and columns of $\hat{N}$ corresponding to states not in $\Gamma$ to zero. Thus element-wise

$$(\hat{N}_\Gamma)_{x,y} \triangleq \begin{cases} \hat{N}_{x,y} & \text{if } x, y \in \Gamma \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

In this chapter we improve firstly the bound in 1.19 in both the dense and sparse regimes by obtaining an upper bound for the spectral norm of order $\sqrt{T_n/n}$. This matches the lower bound asymptotically and proves that the order is asymptotically optimal. Secondly, we prove that this bound also holds in the dense regime $T_n = \Omega(n \log n)$ without trimming.

**Theorem 30.** *Under Assumption 29, the following holds:*

*(a) If $T_n = \Omega(n \log n)$, then*

$$\|\hat{N} - N\| = O_{\mathbb{P}}(\sqrt{T_n/n}). \tag{5.6}$$

*(b) If $T_n = \omega(n)$ and $\Gamma^{\mathrm{c}}$ is a set of size $\lfloor n \mathrm{e}^{-T_n/n} \rfloor$ containing the states with highest number of visits, i.e., with the property that $\min_{y \in \Gamma^{\mathrm{c}}} \hat{N}_{[n],y} \geq \max_{y \in \Gamma} \hat{N}_{[n],y}$, then*

$$\|\hat{N}_{\Gamma} - N\| = O_{\mathbb{P}}(\sqrt{T_n/n}). \tag{5.7}$$

Proposition 17 together with Theorem 30 yields $\sigma_1(\hat{N} - N) = \Theta_{\mathbb{P}}(\sqrt{T_n/n})$. As a corollary to Theorem 30 we also obtain asymptotic scalings and bounds on the singular values of $\hat{N}_{\Gamma}$:

**Corollary 7.** *Presume Assumption 29. If $T_n = \omega(n)$, then*

$$\sigma_i(\hat{N}_{\Gamma}) = \begin{cases} \Theta_{\mathbb{P}}(T_n/n) & \text{if } i \in [K], \\ O_{\mathbb{P}}(\sqrt{T_n/n}) & \text{otherwise.} \end{cases}. \tag{5.8}$$

The proof of Theorem 30 follows a similar strategy as the proof of [27, Prop. 7]. In turn, the proof in [27] was based on [133] which itself took inspiration from [157]. The idea is to upper bound spectral norms of random matrices using an $\epsilon$-net argument and then separate into contributions of so-called light and heavy pairs. While the application of this technique to random graphs and SBMs has become common [93, 36, 67], the distinct difficulty with BMCs is that $\hat{N}$ has dependent entries. Fortunately, the fast mixing time of the BMC can be exploited: using techniques from [98] we can obtain concentration inequalities that are sufficiently strong to argue that the dependencies are negligible asymptotically. Compared to [27], we simplify the estimate of the contributions of the light pairs by proving that the bound holds without the need for regularization whenever $T_n = \Omega_{\mathbb{P}}(n \log(n))$, and we improve upon the logarithmic term present in [27, Lemma 11] by strengthening the bounds for the discrepancy property in BMCs (see Section 5.2.4). We also briefly validate our results numerically in Section 5.6.

This chapter is structured as follows. In Section 5.2 we describe the main properties of BMCs. In Section 5.3 we prove Theorem 30 and its main steps. In Section 5.4 we prove Corollary 7 on the singular values of $\hat{N}$ and $\hat{N}_{\Gamma}$ and in Section 5.5 we prove the lower bound given in Proposition 17. Finally, we simulate BMCs and numerically validate the statement of Theorem 30 in Section 5.6.

## 5.2 Properties of block Markov chains

In this section we cover properties of BMCs that will be exploited to prove Theorem 30. First, we introduce some notation that we will use throughout this chapter.

**Notation.** Let $\mathbb{B}_r^n(x) \subseteq \mathbb{R}^n$ be the $n$-dimensional ball of radius $r$ centered around $x \in \mathbb{R}^n$. Similarly, let $\mathbb{S}_r^{n-1}(x) \subseteq \mathbb{R}^n$ be the $(n-1)$-dimensional sphere with radius $r$ centered around $x \in \mathbb{R}^n$. For any pair of subsets $(\mathcal{A}, \mathcal{B}) \subseteq [n]^2$, we also introduce the short-hand notation $A_{\mathcal{A},\mathcal{B}} \triangleq \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} A_{x,y}$. For $a, b \in \mathbb{R}$, we denote $a \vee b = \max\{a, b\}$.

### 5.2.1 Asymptotic properties of $N$

Let $\pi = (\pi_1, \ldots, \pi_K) \in \mathbb{R}^K$ denote the unique stationary distribution of $q$, which thus satisfies $\pi^T = q\pi^T$. By the Perron–Frobenius theorem, $\min_{k\in[K]} \pi_k = \pi_{\min} > 0$ due to the positivity assumption for $q$ under Assumption 29. We have, moreover, that $\sigma_K(q) = \sigma_{\min}(q) > 0$ due to the assumption $\operatorname{rank}(q) = K$. These properties of positivity have immediate implications for the asymptotic scaling of the entries of $N$. In particular, we prove the following in Appendix 5.A.1:

**Lemma 40.** *There exist constants $0 < \mathfrak{n}_1 < \mathfrak{n}_2 < \infty$, $0 < \mathfrak{p}_1 < \mathfrak{p}_2 < \infty$ independent of $n$ and an integer $m \in \mathbb{N}_+$ such that for all $n \geq m$ and all $x, y \in [n]$, $\mathfrak{n}_1 T_n/n^2 \leq N_{x,y} \leq \mathfrak{n}_2 T_n/n^2$ and $\mathfrak{p}_1/n \leq P_{x,y} \leq \mathfrak{p}_2/n$.*

Lemma 40 combined with the block structure of $P$ has implications for the asymptotic scalings of the singular values of $N$. Observe first that the block structure of $P$ defined in (5.2) and the assumption $\operatorname{rank}(q) = K$ imply that $P$ has $K$ non-zero and $n-K$ zero singular values respectively. In particular, for $i \in [n]$,

$$\sigma_i(P) = \begin{cases} \sigma_i(q) + o(1) = \Theta(1) & \text{if } i \in [K] \\ 0 & \text{otherwise.} \end{cases} \tag{5.9}$$

Furthermore, the unique stationary distribution $\Pi$ of $P$ is then given by

$$\Pi = \Big( \frac{\pi_1}{|\mathcal{V}_1|} u_1, \ldots, \frac{\pi_K}{|\mathcal{V}_K|} u_K \Big) \in (0,1)^n, \tag{5.10}$$

where for $k \in [K]$, $u_k = (1, \ldots, 1) \in (0,1)^{|\mathcal{V}_k|}$ is the all-one vector of its respective dimension. Observe now that Assumption 29, together with the fact that $|\mathcal{V}_i| \sim n\alpha_i$ for $i \in [n]$, implies that $\Pi_i = \Theta(1/n)$ for $i \in [n]$. Since $N = T_n \operatorname{Diag}(\Pi) P$, we can conclude that the singular values of $N$ satisfy

$$\sigma_i(N) = \begin{cases} \Theta(T_n/n) & \text{if } i \in [K], \\ 0 & \text{otherwise.} \end{cases} \tag{5.11}$$

The following is an example of the spectrum of $N$ for a given $q$ and $\alpha$:

**Example 31.** *Let $0 < a, b < 1$ such that $0 < a + b < 1$ and $a \neq 1/3, b \neq 1/3$. Suppose that*

$$q = \begin{pmatrix} a & b & 1-a-b \\ b & 1-a-b & a \\ 1-a-b & a & b \end{pmatrix} \quad \text{and} \quad \alpha = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}. \tag{5.12}$$

*In this symmetric case $q$ has full rank and*

$$\sigma_1(N) = \frac{T_n}{3n}, \quad \text{and} \quad \sigma_2(N) = \sigma_3(N) = \frac{T_n}{3n}\sqrt{1 + 3(a^2 - a + ab - b + b^2)}. \tag{5.13}$$

### 5.2.2 A mixing time of $\Theta(1)$

For $\varepsilon \in [0,1)$, the *$\varepsilon$-mixing time of a Markov chain* can be defined as

$$t_{\text{mix}}(\varepsilon) = \min\{t \geq 0 : d(t) \geq \varepsilon\}. \tag{5.14}$$

Here

$$d(t) = \sup_{x\in[n]} d_{\mathrm{TV}}\big(\mathbb{P}[X_t = \cdot|X_0 = x], \Pi\big) \quad \text{and} \quad d_{\mathrm{TV}}(\mu,\nu) = \tfrac{1}{2}\sum_{x\in[n]} |\mu_x - \nu_x|. \tag{5.15}$$

The mixing time of a BMC is relatively short and in fact $\Theta(1)$. This can be credited to the facts that the block structure is independent of $n$, and that the graph of a BMC is a complete graph [27, Prop. 2]: *For any BMC with $n \geq 4/\alpha_{\min}$ let $\eta > 0$ be such that $1 < \max_{a,b,c}\{p_{b,a}/p_{c,a}, p_{a,b}/p_{a,c}\} \leq \eta$. Then, we have $t_{\mathrm{mix}}(\varepsilon) \leq -c_{\mathrm{mix}} = -1/\log(1-1/2\eta)$.*

Note that the assumption $\eta > 1$ in [27, Prop. 2] follows from Assumption 29(i). Indeed, since $\operatorname{rank}(q) > 1$ there exists at least one $a \in [K]$ such that for some $c \neq b$ we have $q_{a,b}/q_{a,c} > 1$. Finally, positivity of $q$ allows us to find a finite $\eta$.

The relatively short mixing time of a Markov chain can be related to the *pseudo spectral gap*. Let $\lambda(A)$ be the second largest eigenvalue of a symmetric matrix $A$. Then define

$$\gamma_{\mathrm{ps}} = \max_{i\geq 1} \frac{1-\lambda((P^*)^i P^i)}{i} \quad \text{where} \quad P^*_{x,y} = \frac{P_{y,x}}{\Pi_x}\Pi_y. \tag{5.16}$$

The pseudo spectral gap $\gamma_{\mathrm{ps}}$ plays a role in bounding the mixing time, as shown in[98, Prop. 3.4]: *For $\varepsilon \in [0,1)$, $\gamma_{\mathrm{ps}} \geq (1-\varepsilon)/t_{\mathrm{mix}}(\varepsilon/2)$.* For BMCs in particular, this implies that $\gamma_{\mathrm{ps}} \geq 1/(2(1+4\eta))$; see the paragraph preceding [27, SM1(26)] for a proof.

With the pseudo spectral gap, we can then prove sharp concentration inequalities for different quantities pertaining to the BMC using the following result from [98, Thm. 3.4]:

**Lemma 41** ([98, Thm. 3.4])**.** *Let $X_0, X_1, \ldots, X_{T_n-1}$ be a stationary Markov chain with pseudo spectral gap $\gamma_{\mathrm{ps}}$. Let $f \in L^2(\Pi)$, with $|f(x) - \mathbb{E}_\Pi(f)| \leq C$ for every $x \in \Omega$. Let $V_f = \mathrm{Var}_\Pi(f)$. Then, for any $z > 0$,*

$$\mathbb{P}\Big[\Big|\sum_{t=0}^{T_n-1} f(X_t) - \mathbb{E}_\Pi\Big[f(X_t)\Big]\Big| \geq z\Big] \leq 2\exp\Big(-\frac{z^2\gamma_{\mathrm{ps}}}{8(T_n+1/\gamma_{\mathrm{ps}})V_f + 20zC}\Big). \tag{5.17}$$

### 5.2.3 Bounded degrees

Using Lemma 41 we can prove for example that if we were to picture a sample path $X_0, X_1, \ldots, X_{T_n}$ as a directed graph, then the in- and outdegree of all states (vertex) are $O_{\mathbb{P}}(T_n/n)$. Recall the notation that $\hat{N}_{\mathcal{A},\mathcal{B}} = \sum_{x\in\mathcal{A}}\sum_{y\in\mathcal{B}} \hat{N}_{x,y}$ for any two subsets $\mathcal{A}, \mathcal{B} \subseteq [n]$. The out- and indegree of a state $y \in [n]$ are then given by $\hat{N}_{y,[n]}$, $\hat{N}_{[n],y}$, respectively. We prove the following in Appendix 5.A.2:

**Lemma 42.** *The following holds for any BMC:*

*(a) If $T_n = \Omega(n\log n)$, then there exists a constant $\mathfrak{b}_1 > 0$ independent of $n$ such that for sufficient large $n$*

$$\max_{y\in[n]}\big\{\hat{N}_{[n],y} \vee \hat{N}_{y,[n]}\big\} \leq \mathfrak{b}_1\frac{T_n}{n} \quad \text{at least with probability} \quad 1-\frac{2}{n}. \tag{5.18}$$

*(b) If $T_n = \omega(n)$ and $\Gamma^{\mathrm{c}}$ is a set of size $\lfloor n\mathrm{e}^{-T_n/n} \rfloor$ containing the states with highest number of visits, then there exists a constant $\mathfrak{b}_2 > 0$ independent of $n$ such that for sufficiently large $n$*

$$\max_{y\in\Gamma}\left\{\hat{N}_{\Gamma,y} \vee \hat{N}_{y,\Gamma}\right\} \leq \mathfrak{b}_2 \frac{T_n}{n} \quad \text{at least with probability} \quad 1 - 2\mathrm{e}^{-\frac{T_n}{n}}. \tag{5.19}$$

With Lemma 42(b) we can see that, whenever $T = \omega(n)$, the trimming of a fixed number of largest-degree states controls the degrees with high probability just as with the usual trimming of states with degrees above a threshold in [133]. Whenever one of the events in Lemma 42 hold, we say that the bounded degree property holds.

### 5.2.4 Discrepancy property

For $\mathcal{A},\mathcal{B} \subseteq V$, let

$$e(\mathcal{A},\mathcal{B}) = \sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}} \hat{N}_{ij} \tag{5.20}$$

and $\mu(\mathcal{A},\mathcal{B}) = \mathbb{E}[e(\mathcal{A},\mathcal{B})]$. A similar definition will be used when trimming: for $\mathcal{A},\mathcal{B} \subseteq V$, let $e_\Gamma(\mathcal{A},\mathcal{B}) = \sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}}(\hat{N}_\Gamma)_{ij}$. Note that for any fixed $\mathcal{A},\mathcal{B} \subset [n]$, $e_\Gamma(\mathcal{A},\mathcal{B}) \leq e(\mathcal{A},\mathcal{B})$. We define now the discrepancy property. For graphs, this property tells us that the graph has no denser subgraph compared to itself. This will help us in the bounding of the spectral norm later on.

**Definition 32.** *Let $\mathfrak{d}_1, \mathfrak{d}_2 > 0$ be two constants independent of $n$. We say that $\hat{N}$* is $(\mathfrak{d}_1,\mathfrak{d}_2)$-discrepant *if for every pair $(\mathcal{A},\mathcal{B}) \subseteq [n]^2$ one of the following holds:*

*(i)* $\frac{e(\mathcal{A},\mathcal{B})n^2}{|\mathcal{A}||\mathcal{B}|T_n} \leq \mathfrak{d}_1$,

*(ii)* $e(\mathcal{A},\mathcal{B}) \log \frac{e(\mathcal{A},\mathcal{B})n^2}{|\mathcal{A}||\mathcal{B}|T_n} \leq \mathfrak{d}_2(|\mathcal{A}| \vee |\mathcal{B}|) \log \frac{n}{|\mathcal{A}| \vee |\mathcal{B}|}$.

*Similarly, we say that $\hat{N}_\Gamma$* is $(\mathfrak{d}_1,\mathfrak{d}_2)$-discrepant *when the conditions hold with $e_\Gamma(\mathcal{A},\mathcal{B})$ replacing $e(\mathcal{A},\mathcal{B})$.*

We prove that if the bounded degree property holds, then the discrepancy property also holds with high probability. The constants $\mathfrak{d}_1$ and $\mathfrak{d}_2$ will be positive and dependent on $\alpha$ and $q$. The proof follows the method in [93] and is relegated to Appendix 5.A.3. A key step we prove is a uniform concentration inequality for the discrepancy in BMCs (see Lemma 47 in Appendix 5.A.3), which may be of independent interest.

**Proposition 18.** *For any BMC there exist sufficiently large constants $\mathfrak{b}_3, \mathfrak{b}_4, \mathfrak{d}_1, \mathfrak{d}_2 > 0$ independent of $n$ such that the following holds:*

*(a) If $T_n = \Omega(n \log n)$ and $\max_{y\in[n]}\left\{\hat{N}_{[n],y} \vee \hat{N}_{y,[n]}\right\} \leq \mathfrak{b}_3 T_n/n$, then for sufficiently large $n$, $\hat{N}$ is $(\mathfrak{d}_1,\mathfrak{d}_2)$-discrepant at least with probability $1 - 1/n$.*

*(b) If $T_n = \omega(n)$, $\Gamma^{\mathrm{c}}$ is a set of size $\lfloor n\mathrm{e}^{-T_n/n} \rfloor$ containing the states with highest number of visits, and moreover $\max_{y\in\Gamma}\left\{\hat{N}_{\Gamma,y} \vee \hat{N}_{y,\Gamma}\right\} \leq \mathfrak{b}_4 T_n/n$, then for sufficiently large $n$, $\hat{N}_\Gamma$ is $(\mathfrak{d}_1,\mathfrak{d}_2)$-discrepant at least with probability $1 - 1/n$.*

## 5.3 Bounding the spectral norm of $\hat{N}_\Gamma - N$

We will now prove Theorem 30 by bounding the *spectral norm* of $\hat{N}_\Gamma - N$, i.e., the *operator norm induced by the vector* 2*-norm*:

$$\|\hat{N}_\Gamma - N\| = \sup_{x\in\mathbb{R}^n\backslash\{0\}} \frac{\|(\hat{N}_\Gamma - N)x\|_2}{\|x\|_2}. \tag{5.21}$$

Recall that for any matrix $A \in \mathbb{R}^{n\times n}$ we have $\|A\| = \sigma_1(A)$. Instead of working with (5.21), we will use a rectangular quotient relation for convenience [112, (3.10)]: note that

$$\|\hat{N}_\Gamma - N\| = \sup_{x,y\in\mathbb{S}_1^{n-1}(0)} |x^{\mathrm{T}}(\hat{N}_\Gamma - N)y|. \tag{5.22}$$

The proof strategy is as follows. We first use an $\epsilon$-net argument to pass the supremum over the set $\mathbb{S}_1^{n-1}(0)$ in (5.22) to a maximization over a finite set $\mathcal{T}_\epsilon$ say. Next, for each $(x,y) \in \mathcal{T}_\epsilon$, we can bound the sum $|x^T(\hat{N}_\Gamma - N)y| \le L(x,y) + H(x,y)$ by the sum of $L(x,y)$ and $H(x,y)$. $L(x,y)$ is a sum over entries of $x$ and $y$ whose sizes are small, and $H(x,y)$ is a sum over entries whose sizes are large. These will be called the contributions of the light pairs and heavy pairs, respectively. For the light pairs, concentration results for sums of entries of $\hat{N}_\Gamma$ and using the fact that $\Gamma$ is of fixed size, although random in content, we can prove the bound $L(x,y) = O_{\mathbb{P}}(\sqrt{T_n/n})$. For the heavy pairs, concentration results for the entries of $\hat{N}_\Gamma$ are not enough in the sparse regime. Instead, we use the discrepancy property of $\hat{N}$ and $\hat{N}_\Gamma$. This property of graphs says roughly that the number of edges between two sets is not much larger than its average. We prove that $\hat{N}_\Gamma$ satisfies the discrepancy property with high probability and using this fact we can prove that $H(x,y) = O_{\mathbb{P}}(\sqrt{T_n/n})$.

### 5.3.1 Passing to a finite $\epsilon$-net

We start by defining $\epsilon$-nets. Recall that $\mathbb{B}_r^n(x) \subseteq \mathbb{R}^n$ is the $n$-dimensional ball of radius $r$ centered around $x \in \mathbb{R}^n$:

**Definition 33.** *Let $\epsilon \in (0,\infty)$. An* $\epsilon$-net *for $(\mathbb{B}_1^n(0), \|\cdot\|_2)$ is a finite subset $\mathcal{N}_\epsilon \subseteq \mathbb{B}_1^n(0)$ such that for any $x \in \mathbb{B}_1^n(0)$ there exists $y \in \mathcal{N}_\epsilon$ such that $\|x-y\|_2 \le \epsilon$.*

An $\epsilon$-net for $(\mathbb{B}_1^n(0), \|\cdot\|_2)$ has several useful properties, which we will exploit. The following properties are proven in Appendix 5.B.1. For any subset $\mathcal{A} \subseteq [n]$ and any vector $b \in \mathbb{R}^n$, we let $b^{\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$ denote the vector obtained by deleting the rows in the index set $\mathcal{A}$.

**Lemma 43.** *The following holds:*

*(a) Let $\epsilon \in (0,1/3)$. If $\mathcal{N}_\epsilon$ is an $\epsilon$-net for $(\mathbb{B}_1^n(0), \|\cdot\|_2)$, then for any matrix $A \in \mathbb{R}^{n\times n}$*

$$\|A\| = \sup_{x,y\in\mathbb{S}_1^{n-1}(0)} |x^{\mathrm{T}}Ay| \le \frac{1}{1-3\epsilon} \sup_{x,y\in\mathcal{N}_\epsilon} |x^{\mathrm{T}}Ay|. \tag{5.23}$$

*(b) Let $\epsilon \in (0,\infty)$. If $\mathcal{N}_\epsilon$ is an $\epsilon$-net for $(\mathbb{B}_1^n(0), \|\cdot\|_2)$, then for any subset $\mathcal{A} \subseteq [n]$, the subset $\mathcal{N}_\epsilon^{\mathcal{A}} = \{x^{\mathcal{A}} : x \in \mathcal{N}_\epsilon\}$ is an $\epsilon$-net for $(\mathbb{B}_1^{|\mathcal{A}|}(0), \|\cdot\|_2)$.*

In order to control the size of the entries of $x \in \mathcal{N}_\epsilon$, we will use the following specific set, which is also used in [157, 133, 93]:

$$\mathcal{T}_\epsilon = \Big\{ x \in \mathbb{R}^n : x \in \frac{\epsilon}{\sqrt{n}} \mathbb{Z}^n, \|x\|_2 \leq 1 \Big\}. \tag{5.24}$$

Observe that $\mathcal{T}_\epsilon$ is indeed an $\epsilon$-net for $(\mathbb{B}_1^n(0), \|\cdot\|_2)$. The properties in Lemma 43 thus apply to $\mathcal{T}_\epsilon$. Furthermore, the size of the set is bounded by $|\mathcal{T}_\epsilon| \leq (9/\epsilon)^n$ [133, Claim 2.9].

## 5.3.2 The sets of light- and heavy pairs

The next course of action will be to derive for every $x, y \in \mathcal{T}_\epsilon$ an upper bound of the kind $|x^{\mathrm{T}}(\hat{N}_\Gamma - N)y| \leq \mathfrak{c}\sqrt{T_n/n}$, where $\mathfrak{c}$ is a constant independent of $n$ which holds with probability $1 - O(1/n)$. For $x, y \in \mathbb{B}_1^n(0)$, define the *set of light pairs* by

$$\mathcal{L}(x,y) = \Big\{ (i,j) \in [n]^2 : |x_i y_j| \leq \frac{1}{n}\sqrt{\frac{T_n}{n}} \Big\}. \tag{5.25}$$

Similarly, we define the *set of heavy pairs* by

$$\mathcal{H}(x,y) = \mathcal{L}^{\mathrm{c}}(x,y) = \Big\{ (i,j) \in [n]^2 : |x_i y_j| > \frac{1}{n}\sqrt{\frac{T_n}{n}} \Big\}. \tag{5.26}$$

Using the triangle inequality we can then split the bounding by writing

$$\begin{aligned} |x^{\mathrm{T}}(\hat{N}_\Gamma - N)y| &\leq \Big| \sum_{(i,j)\in\mathcal{L}} x_i y_j \big((\hat{N}_\Gamma)_{ij} - N_{ij}\big) \Big| + \Big| \sum_{(i,j)\in\mathcal{L}^{\mathrm{c}}} x_i y_j \big((\hat{N}_\Gamma)_{ij} - N_{ij}\big) \Big| \\ &= L(x,y) + H(x,y) \end{aligned} \tag{5.27}$$

say, almost surely. Here, $L(x,y)$ and $H(x,y)$ denote the *contributions of the light* and *heavy pairs*, respectively. To simplify the exposition, we will omit the indication $(x,y)$ from the sets of light and heavy pairs whenever they appear in a subscript.

## 5.3.3 Bounding the contribution of the light pairs

We split the bounding of $L(x,y)$ into two parts. Let $\mathcal{K}^{\mathrm{c}} = (\Gamma^{\mathrm{c}} \times [n]) \cup ([n] \times \Gamma^{\mathrm{c}})$ denote the set of transitions that are trimmed (recall that $\Gamma^{\mathrm{c}}$ denotes the set of states that are trimmed). Using that (i) $(\hat{N}_\Gamma)_{ij} = 0$ whenever $i \notin \Gamma$ or $j \notin \Gamma$ by its definition in (5.5) and (ii) $\mathcal{K} = \Gamma^2$ as well as the triangle inequality, we obtain

$$\begin{aligned} L(x,y) &\overset{(5.27)}{=} \Big| \sum_{(i,j)\in\mathcal{L}} x_i y_j \big((\hat{N}_\Gamma)_{ij} - N_{ij}\big) \Big| \\ &\overset{\text{(i)}}{=} \Big| \sum_{(i,j)\in\mathcal{L}\cap\mathcal{K}} x_i y_j (\hat{N}_{ij} - N_{ij}) - \sum_{(i,j)\in\mathcal{L}\cap\mathcal{K}^{\mathrm{c}}} x_i y_j N_{ij} \Big| \\ &\overset{\text{(ii)}}{\leq} \Big| \sum_{(i,j)\in\mathcal{L}\cap\Gamma^2} x_i y_j (\hat{N}_{ij} - N_{ij}) \Big| + \Big| \sum_{(i,j)\in\mathcal{L}\cap\mathcal{K}^{\mathrm{c}}} x_i y_i N_{ij} \Big| \\ &= L_1(x,y) + L_2(x,y), \end{aligned} \tag{5.28}$$

say, almost surely.

**Bound for $L_1(x,y)$**

For any subset $\mathcal{A} \subseteq [n]$ and matrix $A \in \mathbb{R}^{n\times n}$, let $A^{\mathcal{A}}$ denote the submatrix obtained by deleting the rows and columns in the index set $\mathcal{A}$. Consequently $A \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$. Recall that we have adopted similar notation for vectors. Define for any subset $\mathcal{A} \subseteq [n]$, not necessarily random, and any $x,y \in \mathbb{B}_1^n(0)$,

$$\mathcal{L}^{\mathcal{A}}(x,y) = \mathcal{L}(x,y) \cap \mathcal{A}^2 = \left\{(i,j) \in \mathcal{A}^2 : |x_i y_j| \leq \frac{1}{n}\sqrt{\frac{T_n}{n}}\right\} \tag{5.29}$$

as well as

$$L^{\mathcal{A}}(x,y) = \Big| \sum_{(i,j)\in\mathcal{L}^{\mathcal{A}}} x_i y_j (\hat{N}_{ij}^{\mathcal{A}} - N_{ij}^{\mathcal{A}})\Big|. \tag{5.30}$$

Note that for any $x,y \in \mathbb{B}_1^n(0)$, $L_1(x,y) = L^{\Gamma}(x,y)$ almost surely.

We proceed in a manner similar as in [133]. We must deal, however, with the added difficulty that there are dependencies between the entries of $\hat{N}$ and that $\Gamma$ is random. Therefore, our first step will be to prove that for any deterministic minor, $\max_{x,y\in\mathcal{T}_\epsilon} L^{\mathcal{A}}(x,y) = O_{\mathbb{P}}(\sqrt{T_n/n})$. Our second step is to use a union bound argument and lift this result to $\max_{\mathcal{A}\in\mathcal{M}_{n,\delta}} \max_{x,y\in\mathcal{T}_\epsilon} L^{\mathcal{A}}(x,y) = O_{\mathbb{P}}(\sqrt{T_n/n})$, where $\mathcal{M}_{n,\delta}$ denotes the set of all subsets of size at least $(1-\delta)n$. In particular, because $|\Gamma| = n - \lfloor n e^{-T_n/n} \rfloor$ is deterministic, this implies the result.

**Step 1: Minors induced by deterministic sets** While we will ultimately use $\mathcal{T}_\epsilon$, the following arguments hold for any $\epsilon$-net for $(\mathbb{B}_1^n(0), \|\cdot\|_2)$ with a small enough number of points.

**Lemma 44.** *There exist a constant $\mathfrak{n}_2 > 0$ independent of $n$ and an integer $m \in \mathbb{N}_+$ such that for all $n \geq m$, any $\epsilon$-net $\mathcal{N}_\epsilon$ for $(\mathbb{B}_1^n(0), \|\cdot\|_2)$ with cardinality at most $(9/\epsilon)^n$, such as $\mathcal{T}_\epsilon$, any deterministic subset $\mathcal{A} \subseteq [n]$, and any $\mathfrak{f} \geq \max\{328(1+4\eta)\log(9/\epsilon), 8\mathfrak{n}_2(3+8\eta)\}$,*

$$\mathbb{P}\Big[\max_{x,y\in\mathcal{N}_\epsilon} |L^{\mathcal{A}}(x,y)| > \mathfrak{f}\sqrt{\frac{T_n}{n}}\Big] \leq 2\exp\Big(-\frac{\mathfrak{f}n}{164(1+4\eta)}\Big). \tag{5.31}$$

*Proof.* Recall from Lemma 40 that there exists a constant $\mathfrak{n}_2 > 0$ and integer $m \in \mathbb{N}_+$ such that for all $n \geq m$ and all $i,j \in [n]$, $N_{ij} \leq \mathfrak{n}_2 T_n/n^2$.

We are going to use Lemma 41 to prove Lemma 44. Observe that the two-dimensional stochastic process $\{(X_t, X_{t+1})\}_{t\geq 0}$ induced by the transitions of the BMC is in fact also a Markov chain. Moreover, the mixing time of $\{(X_t, X_{t+1})\}_{t\geq 0}$ requires just one more transition than the mixing time of $\{X_t\}_{t\geq 0}$. Consequently, for both of these Markov chains $\gamma_{ps} \geq 1/(2(4\eta+1))$ [27, SM1(26)].

Let $n \geq m$, $\mathcal{A} \subseteq [n]$ deterministic, and $x,y \in \mathbb{B}_1^n(0)$. Define for $t \in \{0,1,\ldots,T_n-1\}$

$$f_{x,y}^{\mathcal{A}}((X_t, X_{t+1})) = \sum_{(i,j)\in\mathcal{L}^{\mathcal{A}}} x_i y_j \mathbb{1}[X_t = i, X_{t+1} = j], \tag{5.32}$$

so that

$$\mathbb{E}[f_{x,y}^{\mathcal{A}}((X_t, X_{t+1}))] = \sum_{(i,j)\in\mathcal{L}^{\mathcal{A}}} x_i y_j \Pi_i P_{i,j}. \tag{5.33}$$

Observe that

$$
\begin{aligned}
L^{\mathcal{A}}(x,y) &\overset{(5.30)}{=} \Big| \sum_{(i,j)\in\mathcal{L}^{\mathcal{A}}} x_i y_j (\hat{N}_{ij}^{\mathcal{A}} - N_{ij}^{\mathcal{A}}) \Big| = \Big| \sum_{(i,j)\in\mathcal{L}^{\mathcal{A}}} x_i y_j \big(\hat{N}_{ij} - N_{ij}\big) \Big| \\
&= \Big| \sum_{t=0}^{T_n - 1} \sum_{(i,j)\in\mathcal{L}^{\mathcal{A}}} x_i y_j \Big( \mathbb{1}[X_t = i, X_{t+1} = j] - \Pi_i P_{i,j} \Big) \Big| \\
&= \Big| \sum_{t=0}^{T_n - 1} \big( f_{x,y}^{\mathcal{A}}((X_t, X_{t+1})) - \mathbb{E}[f_{x,y}^{\mathcal{A}}((X_t, X_{t+1}))] \big) \Big|. \qquad (5.34)
\end{aligned}
$$

This positions us to apply Lemma 41. All that remains is to provide bounds for the deviation of $f_{x,y}^{\mathcal{A}}((X_t, X_{t+1}))$ from its expectation. We claim that for all $t \in \{0, 1, \ldots, T_n - 1\}$, we have

$$
|f_{x,y}^{\mathcal{A}}((X_t, X_{t+1})) - \mathbb{E}[f_{x,y}^{\mathcal{A}}((X_t, X_{t+1}))]| \leq \frac{2}{n}\sqrt{\frac{T_n}{n}}, \qquad (5.35)
$$

$$
\mathrm{Var}[f_{x,y}^{\mathcal{A}}((X_t, X_{t+1}))] \leq \frac{\mathfrak{n}_2}{n^2}. \qquad (5.36)
$$

After having established these claims, the result will follow.

*Proof of Lemma 44, assuming* (5.35) *and* (5.36)*:* Applying Lemma 41 with the function $f$ replaced by $f_{x,y}^{\mathcal{A}}$ to the sample path $(X_0, X_1), \ldots, (X_{T_{n-1}}, X_{T_n})$ of the stationary two-dimensional Markov chain $\{(X_t, X_{t+1})\}_{t\geq 0}$ together with (5.35) and (5.36) implies that for any $\mathfrak{f} > 0$,

$$
\begin{aligned}
&\mathbb{P}\Big[ \Big| \sum_{t=0}^{T_n - 1} f_{x,y}^{\mathcal{A}}((X_t, X_{t+1})) - \mathbb{E}[f_{x,y}^{\mathcal{A}}((X_t, X_{t+1}))] \Big| > \mathfrak{f}\sqrt{\frac{T_n}{n}} \Big] \\
&\leq 2\exp\Big( -\frac{\mathfrak{f}^2 \gamma_{ps} \frac{T_n}{n}}{8(T_n + 1/\gamma_{ps})\frac{\mathfrak{n}_2}{n^2} + 40\mathfrak{f}\frac{T_n}{n^2}} \Big) = 2\exp\Big( -\frac{\mathfrak{f}^2 \gamma_{ps} n}{8\mathfrak{n}_2(1 + 1/(\gamma_{ps} T_n)) + 40\mathfrak{f}} \Big). \qquad (5.37)
\end{aligned}
$$

Note that $\gamma_{ps}$ may depend on $n$. Recall therefore that $T_n \geq 1$ and (i) $\gamma_{ps} \geq 1/(2(1+4\eta))$. Consequently $1 + 1/(\gamma_{ps} T_n) \leq 1 + 1/\gamma_{ps} \leq 3 + 8\eta$. The lower bound on the right-hand side is independent of $n$. We find that (ii) for any $\mathfrak{f} \geq 8\mathfrak{n}_2(3+8\eta)$,

$$
\begin{aligned}
&\mathbb{P}\Big[ L^{\mathcal{A}}(x,y) > \mathfrak{f}\sqrt{\frac{T_n}{n}} \Big] \\
&\leq 2\exp\Big( -\frac{\mathfrak{f}^2 \gamma_{ps} n}{8\mathfrak{n}_2(3+8\eta) + 40\mathfrak{f}} \Big) \overset{\text{(ii)}}{\leq} 2\exp\Big( -\frac{\mathfrak{f}\gamma_{ps} n}{41} \Big) \overset{\text{(i)}}{\leq} 2\exp\Big( -\frac{\mathfrak{f} n}{82(1+4\eta)} \Big). \qquad (5.38)
\end{aligned}
$$

Finally use (iii) Boole's inequality together with (5.38), in combination with (iv) Lemma 43(b) with the assumption $|\mathcal{N}_\epsilon| \leq (9/\epsilon)^n$ to conclude that (v) for any $\mathfrak{f} \geq \max\{328(1+4\eta)\log(9/\epsilon), 8\mathfrak{n}_2(3+8\eta)\}$,

$$
\begin{aligned}
&\mathbb{P}\Big[ \max_{x,y\in\mathcal{N}_\epsilon} |L^{\mathcal{A}}(x,y)| > \mathfrak{f}\sqrt{\frac{T_n}{n}} \Big] \overset{\text{(iii)}}{\leq} |\mathcal{N}_\epsilon|^2 \cdot 2\exp\Big( -\frac{\mathfrak{f} n}{82(1+4\eta)} \Big) \\
&\overset{\text{(iv)}}{\leq} 2\exp\Big( \Big( 2\log\frac{9}{\epsilon} - \frac{\mathfrak{f}}{82(1+4\eta)} \Big) n \Big) \overset{\text{(v)}}{\leq} 2\exp\Big( -\frac{\mathfrak{f} n}{164(1+4\eta)} \Big). \qquad (5.39)
\end{aligned}
$$

This establishes Lemma 44 under the assumption of (5.35) and (5.36). All that remains is to prove (5.35) and (5.36).

*Proof of* (5.35)*:* Let $t \in \{0,1,\dots,T_n-1\}$. We have that

$$\big|f^{\mathcal{A}}_{x,y}((X_t,X_{t+1})) - \mathbb{E}[f^{\mathcal{A}}_{x,y}((X_t,X_{t+1}))]\big| \overset{(5.32)}{=} \Big|\sum_{(i,j)\in\mathcal{L}^{\mathcal{A}}} x_i y_j \Big(\mathbb{1}[X_t=i,X_{t+1}=j] - \Pi_i P_{i,j}\Big)\Big|$$

$$\le \sup_{(i,j)\in\mathcal{L}^{\mathcal{A}}}\big\{|x_i y_j|\big\}\cdot\Big(1+\sum_{(i,j)\in[n]^2}\Pi_i P_{i,j}\Big) \overset{\text{(i)}}{\le} \frac{2}{n}\sqrt{\frac{T_n}{n}} \tag{5.40}$$

almost surely. Here, we (i) used the facts that $(i,j)\in\mathcal{L}^{\mathcal{A}}$ and $\sum_{(i,j)\in[n]^2}\Pi_i P_{i,j}=1$. This establishes (5.35).

*Proof of* (5.36)*:* Let $t \in \{0,1,\dots,T_n-1\}$. Observe that

$$\begin{aligned}
\operatorname{Var}[f^{\mathcal{A}}_{x,y}((X_t,X_{t+1}))] &\le \mathbb{E}\Big[\Big(\sum_{(i,j)\in\mathcal{L}^{\mathcal{A}}} x_i y_j \mathbb{1}[X_t=i,X_{t+1}=j]\Big)^2\Big] \\
&= \mathbb{E}\Big[\sum_{\substack{(i,j)\in\mathcal{L}^{\mathcal{A}}\\(k,l)\in\mathcal{L}^{\mathcal{A}}}} x_i y_j x_k y_l \mathbb{1}[X_t=i,X_{t+1}=j]\mathbb{1}[X_t=k,X_{t+1}=l]\Big] \\
&= \mathbb{E}\Big[\sum_{(i,j)\in\mathcal{L}^{\mathcal{A}}} |x_i y_j|^2 \mathbb{1}[X_t=i,X_{t+1}=j]\Big] \qquad (5.41) \\
&= \sum_{(i,j)\in\mathcal{L}^{\mathcal{A}}} |x_i y_j|^2 \Pi_i P_{ij} \le \max_{(i,j)\in[n]^2}\big\{\Pi_i P_{ij}\big\} \sum_{(i,j)\in[n]^2} |x_i y_j|^2.
\end{aligned}$$

Recall now that $x,y\in\mathbb{B}^n_1(0)$. This implies that $\sum_{(i,j)\in[n]^2}|x_i y_j|^2 = \sum_{(i,j)\in[n]^2} x_i^2 y_j^2 = \sum_{i\in[n]} x_i^2 \cdot \sum_{j\in[n]} y_j^2 \le 1$. Furthermore, observe that Lemma 40 implies $\Pi_i P_{ij} = N_{ij}/T_n \le \mathfrak{n}_2/n^2$. Use these two facts to bound (5.41). This proves (5.36).

This completes the proof of Lemma 44. □

**Step 2: Passing to a random minor** We have proven in Lemma 44 that the contribution of light pairs of a minor induced by a deterministic subset $\mathcal{A}$ contributes at most $O_{\mathbb{P}}(\sqrt{T_n/n})$ with high probability. We will now prove that this is also the case for all subsets of size $|\Gamma| = n - \lfloor n\mathrm{e}^{-T_n/n}\rfloor$ simultaneously:

**Proposition 19.** *There exists a constant $\mathfrak{n}_2>0$ independent of $n$ and an integer $m\in\mathbb{N}_+$ such that for all $n\ge m$, any $\epsilon$-net $\mathcal{N}_\epsilon$ for $(\mathbb{B}^n_1(0),\|\cdot\|_2)$ with cardinality at most $(9/\epsilon)^n$, such as $\mathcal{T}_\epsilon$, and all $\mathfrak{l}_1 \ge \max\{656(1+4\eta)\log 2, 328(1+4\eta)\log(9/\epsilon), 8\mathfrak{n}_2(3+8\eta)\}$,*

$$\mathbb{P}\Big[\max_{x,y\in\mathcal{N}_\epsilon}|L_1(x,y)| \ge \mathfrak{l}_1\sqrt{\frac{T_n}{n}}\Big] \le 2\exp\Big(-\frac{\mathfrak{l}_1 n}{328(1+4\eta)}\Big). \tag{5.42}$$

*Proof.* Recall Lemma 44: there exist a constant $\mathfrak{n}_2>0$ independent of $n$ and an integer $m\in\mathbb{N}_+$ such that its clauses hold.

Let $n\ge m$. To prove Proposition 19, we proceed as in [133]. Define for $\delta\in(0,1)$,

$$\mathcal{M}_{n,\delta} = \big\{\mathcal{A}\in\mathcal{P}([n]) \,\big|\, (1-\delta)n \le |\mathcal{A}| \le n\big\}. \tag{5.43}$$

Here, $\mathcal{P}([n])$ denotes the power set of $[n]$. Using (i) Boole's inequality, (ii) Lemma 44, and (iii) $|\mathcal{M}_{n,\delta}| \leq 2^n$ independently of $\delta$ yields that (iv) for all $\mathfrak{l}_1 \geq \max\{656(1+4\eta)\log 2, 328(1+4\eta)\log(9/\epsilon), 8\mathfrak{n}_2(3+8\eta)\}$,

$$\begin{aligned}
\mathbb{P}\Big[\max_{\mathcal{A}\in M_{n,\delta}} \max_{x,y\in\mathcal{N}_\epsilon^{\mathcal{A}}} |L^{\mathcal{A}}(x,y)| > \mathfrak{l}_1\sqrt{\frac{T_n}{n}}\Big] &\overset{\text{(i)}}{\leq} \sum_{\mathcal{A}\in M_{n,\delta}} \mathbb{P}\Big[\max_{x,y\in\mathcal{N}_\epsilon^{\mathcal{A}}} |L^{\mathcal{A}}(x,y)| > \mathfrak{l}_1\sqrt{\frac{T_n}{n}}\Big] \\
&\overset{\text{(ii)}}{\leq} |M_{n,\delta}| \cdot 2\exp\Big(-\frac{\mathfrak{l}_1 n}{164(1+4\eta)}\Big) \qquad (5.44) \\
&\overset{\text{(iii)}}{\leq} 2\exp\Big(\Big(\log 2 - \frac{\mathfrak{l}_1 n}{164(1+4\eta)}\Big)n\Big) \overset{\text{(iv)}}{\leq} 2\exp\Big(-\frac{\mathfrak{l}_1 n}{328(1+4\eta)}\Big).
\end{aligned}$$

Recalling (v) that $L_1(x,y) = L^{\Gamma}(x,y)$ almost surely completes the proof, because (vi) there exists a $\delta \in (0,1)$ such that the event

$$\begin{aligned}
\Big\{\max_{x,y\in\mathcal{N}_\epsilon} L_1(x,y) > \mathfrak{l}_1\sqrt{\frac{T_n}{n}}\Big\} &\overset{\text{(v)}}{=} \Big\{\max_{x,y\in\mathcal{N}_\epsilon} L^{\Gamma}(x,y) > \mathfrak{l}_1\sqrt{\frac{T_n}{n}}\Big\} \\
&\overset{\text{(vi)}}{\subseteq} \Big\{\max_{\mathcal{A}\in M_{n,\delta}} \max_{x,y\in\mathcal{N}_\epsilon} |L^{\mathcal{A}}(x,y)| > \mathfrak{l}_1\sqrt{\frac{T_n}{n}}\Big\}. \qquad (5.45)
\end{aligned}$$

This is because $|\Gamma| = n - \lfloor n\mathrm{e}^{-T_n/n}\rfloor$. This proves the proposition. □

### Bounding $L_2(x,y)$

**Proposition 20.** *There exist a constant $\mathfrak{l}_2 > 0$ independent of $n$ and an integer $m \in \mathbb{N}_+$ such that for all $n \geq m$, and any $\epsilon$-net $\mathcal{N}_\epsilon$ for $(\mathbb{B}_1^n(0), \|\cdot\|_2)$ with cardinality at most $(9/\epsilon)^n$, such as $\mathcal{T}_\epsilon$, $\mathbb{P}[\max_{x,y\in\mathcal{N}_\epsilon} L_2(x,y) \geq \mathfrak{l}_2\sqrt{T_n/n}] = 0$.*

*Proof.* Recall Lemma 40: there exists a constant $\mathfrak{n}_2 > 0$ and integer $m \in \mathbb{N}_+$ such that for all $n \geq m$ and all $i,j \in [n]$, $N_{ij} \leq \mathfrak{n}_2 T_n/n^2$.

Let $n \geq m$ and $x,y \in \mathbb{B}_1^n(0)$. By (i) the triangle inequality, (ii) $\mathcal{L}$'s definition in (5.25) and the fact that $\mathcal{L}\cap\mathcal{K}^{\mathrm{c}} \subseteq \mathcal{K}^{\mathrm{c}}$, (iii) expanding the maximization range from $\mathcal{K}^{\mathrm{c}}$ to $[n]^2$, (iv) the bound $|\mathcal{K}^{\mathrm{c}}| \leq 2n|\Gamma^{\mathrm{c}}|$ and Lemma 40, (v) $|\Gamma^{\mathrm{c}}| = \lfloor n\mathrm{e}^{-T_n/n}\rfloor \leq n\mathrm{e}^{-T_n/n}$, and finally (vi) for $z \geq 0$, $z\mathrm{e}^{-z} \leq 1$, we obtain that

$$\begin{aligned}
\Big|\sum_{(i,j)\in\mathcal{L}\cap\mathcal{K}^{\mathrm{c}}} x_i y_j N_{ij}\Big| &\overset{\text{(i)}}{\leq} \sum_{(i,j)\in\mathcal{L}\cap\mathcal{K}^{\mathrm{c}}} |x_i y_j| N_{ij} \overset{\text{(ii)}}{\leq} \frac{1}{n}\sqrt{\frac{T_n}{n}} \cdot \sum_{(i,j)\in\mathcal{K}^{\mathrm{c}}} N_{ij} \\
&\overset{\text{(iii)}}{\leq} \frac{1}{n}\sqrt{\frac{T_n}{n}} \cdot |\mathcal{K}^{\mathrm{c}}| \max_{(i,j)\in[n]}\{N_{ij}\} \overset{\text{(iv)}}{\leq} \frac{1}{n}\sqrt{\frac{T_n}{n}} \cdot 2\mathfrak{n}_2|\Gamma^{\mathrm{c}}|\frac{T_n}{n} \\
&\overset{\text{(v)}}{\leq} 2\mathfrak{n}_2\sqrt{\frac{T_n}{n}} \cdot \frac{T_n}{n}\mathrm{e}^{-\frac{T_n}{n}} \overset{\text{(vi)}}{\leq} 2\mathfrak{n}_2\sqrt{\frac{T_n}{n}} \qquad (5.46)
\end{aligned}$$

almost surely.

The proposition follows after an application of Boole's inequality: for any $\mathfrak{l}_2 > 2\mathfrak{n}_2$ independent of $n$,

$$\mathbb{P}\Big[\max_{x,y\in\mathcal{N}_\epsilon} L_2(x,y) \geq \mathfrak{l}_2\sqrt{\frac{T_n}{n}}\Big] \leq \sum_{x,y\in\mathcal{N}_\epsilon} \mathbb{P}\Big[L_2(x,y) \geq \mathfrak{l}_2\sqrt{\frac{T_n}{n}}\Big] \overset{(5.46)}{=} \Big(\frac{4}{\epsilon}\Big)^n \cdot 0 = 0. \qquad (5.47)$$

□

### 5.3.4 Bounding the contribution of the heavy pairs

We split the bounding of $H(x,y)$ into two parts too, using the triangle inequality: let

$$\begin{aligned} H(x,y) &\overset{(5.27)}{=} \Big| \sum_{(i,j)\in\mathcal{L}^c} x_i y_j \big((\hat{N}_\Gamma)_{ij} - N_{ij}\big)\Big| \\ &\leq \Big| \sum_{(i,j)\in\mathcal{L}^c} x_i y_j (\hat{N}_\Gamma)_{ij}\Big| + \Big| \sum_{(i,j)\in\mathcal{L}^c} x_i y_j N_{ij}\Big| \\ &= H_1(x,y) + H_2(x,y). \end{aligned} \tag{5.48}$$

**Bound for $H_1(x,y)$**

To bound the contribution of heavy pairs, we will follow the proof approaches in [133, 124] and specifically adapt [124, Appendix C]. Our primary modifications consist of proving the right asymptotic scalings for the discrepancy property and bounded degree property. In this manner, the bounds can be applied to $\hat{N}_\Gamma$, which enumerates the visits of a Markov chain in contrast to a random graphs which are the common setting when using these bounds. The discrepancy property and bounded degree properties will ultimately be guaranteed using a concentration inequality for Markov chains; recall also Lemma 42 and Proposition 18. Because the proof of the following proposition follows similar arguments as in [133, 124], we relegate the proof to Appendix 5.B.2.

**Proposition 21.** *If* $\max_{y\in\Gamma}\big\{\hat{N}_{\Gamma,y} \vee \hat{N}_{y,\Gamma}\big\} \leq \mathfrak{b}_2 T_n/n$ *and* $\hat{N}_\Gamma$ *satisfies the discrepancy property in Definition 32, then there exists a constant* $\mathfrak{h}_1 > 0$ *independent of* $n$ *and an integer* $m \in \mathbb{N}_+$ *such that for all* $n \geq m$,

$$\max_{x,y\in\mathcal{T}_\epsilon} H_1(x,y) \leq \mathfrak{h}_1 \sqrt{\frac{T_n}{n}}. \tag{5.49}$$

**Bound for $H_2(x,y)$**

**Proposition 22.** *There exists a constant* $\mathfrak{h}_2 > 0$ *independent of* $n$ *and an integer* $m \in \mathbb{N}_+$ *such that for all* $n \geq m$, *and any* $\epsilon$*-net* $\mathcal{N}_\epsilon$ *for* $(\mathbb{B}_1^n(0), \|\cdot\|_2)$ *with cardinality at most* $(9/\epsilon)^n$, *such as* $\mathcal{T}_\epsilon$, $\mathbb{P}[\max_{x,y\in\mathcal{N}_\epsilon} H_2(x,y) \geq \mathfrak{h}_2\sqrt{T_n/n}] = 0$.

*Proof.* Recall from Lemma 40 that there exists a constant $\mathfrak{n}_2 > 0$ and integer $m \in \mathbb{N}_+$ such that for all $n \geq m$ and all $i,j \in [n]$, $N_{ij} \leq \mathfrak{n}_2 T_n/n^2$. Let $n \geq m$ and $x,y \in \mathbb{B}_1^n(0)$. Observe that

$$H_2(x,y) = \Big| \sum_{(i,j)\in\mathcal{L}^c} x_i y_j N_{ij}\Big| \leq \max_{(i,j)\in[n]^2}\{N_{ij}\} \sum_{(i,j)\in\mathcal{L}^c} |x_i y_j| \leq \mathfrak{n}_2 \frac{T_n}{n^2} \sum_{(i,j)\in\mathcal{L}^c} |x_i y_j| \tag{5.50}$$

almost surely. Because (i) $x,y \in \mathbb{B}_1^n(0)$, and (ii) $x,y \in \mathcal{L}^c$, the inequalities

$$1 \overset{(i)}{\geq} \sum_{(i,j)\in[n]^2} x_i^2 y_j^2 = \Big( \sum_{(i,j)\in\mathcal{L}} + \sum_{(i,j)\in\mathcal{L}^c} \Big) x_i^2 y_j^2 \geq \sum_{(i,j)\in\mathcal{L}^c} |x_i y_j||x_i y_j| \overset{(ii)}{>} \frac{1}{n}\sqrt{\frac{T_n}{n}} \sum_{(i,j)\in\mathcal{L}^c} |x_i y_j|, \tag{5.51}$$

are satisfied. This yields

$$\sum_{(i,j)\in\mathcal{L}^{\mathrm{c}}} |x_i y_j| < n\sqrt{\frac{n}{T_n}}. \tag{5.52}$$

Bound (5.50) using (5.52) to obtain that

$$\Big|\sum_{(i,j)\in\mathcal{L}^{\mathrm{c}}} x_i y_j N_{ij}\Big| \leq \mathfrak{n}_2\sqrt{\frac{T_n}{n}} \tag{5.53}$$

almost surely.

The proposition follows after an application of a union bound. For any $\mathfrak{h}_2 > \mathfrak{n}_2$ independent of $n$ and any $n \geq m$,

$$\mathbb{P}\Big[\max_{x,y\in\mathcal{N}_\epsilon} H_2(x,y) \geq \mathfrak{h}_2\sqrt{\frac{T_n}{n}}\Big] \leq \sum_{x,y\in\mathcal{N}_\epsilon} \mathbb{P}[H_2(x,y) \geq \mathfrak{h}_2\sqrt{\frac{T_n}{n}}] \overset{(5.53)}{=} \Big(\frac{4}{\epsilon}\Big)^n \cdot 0 = 0. \tag{5.54}$$

□

### 5.3.5 Proof of Theorem 30

We will now combine the results and prove Theorem 30. We will bound the spectral norm of $\hat{N}_\Gamma - N$ and obtain Theorem 30(b); the proof of Theorem 30(a) follows the same steps and will therefore be skipped. The only difference is that the discrepancy property used in the first step with Proposition 18 requires trimming to hold when $\omega(n) = T_n = o(n\log n)$. We will remark when this is the case during the proof.

In order to use Lemma 43, we will assume from now on that $\epsilon = 1/4 \in (0,1/3)$. We thus consider the $\epsilon$-net $\mathcal{T}_{1/4}$. Let $\mathfrak{d} > 0$ be a constant independent of $n$ that we will choose sufficiently large later. Using (i) Lemma 43(a), we can then bound for each $\mathfrak{d} > 0$:

$$\begin{aligned}
\mathbb{P}\Big[\|\hat{N}_\Gamma - N\| \geq \mathfrak{d}\sqrt{\frac{T_n}{n}}\Big] &\overset{(\mathrm{i})}{\leq} \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} |x^{\mathrm{T}}(\hat{N}_\Gamma - N)y| \geq \frac{\mathfrak{d}}{4}\sqrt{\frac{T_n}{n}}\Big] \\
&\overset{(5.27)}{\leq} \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} L(x,y) \geq \frac{\mathfrak{d}}{8}\sqrt{\frac{T_n}{n}}\Big] + \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H(x,y) \geq \frac{\mathfrak{d}}{8}\sqrt{\frac{T_n}{n}}\Big] \\
&\overset{(5.28,\ 5.48)}{\leq} \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} L_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}}\Big] + \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} L_2(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}}\Big] \\
&\quad + \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}}\Big] + \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_2(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}}\Big].
\end{aligned} \tag{5.55}$$

From Propositions 19–22 it follows that there exist constants $\mathfrak{l}_1, \mathfrak{l}_2, \mathfrak{h}_1, \mathfrak{h}_2 > 0$ independent of $n$ and integers $m_1, \ldots, m_4 \in \mathbb{N}_+$, such that for any $\mathfrak{d}/16 > \max\{\mathfrak{l}_1, \mathfrak{l}_2, \mathfrak{h}_1, \mathfrak{h}_2\}$ and all $n \geq \max\{m_1, \ldots, m_4\}$:

$$\mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} L_2(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}}\Big] = 0, \quad \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_2(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}}\Big] = 0, \tag{5.56}$$

and furthermore

$$\mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} L_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}}\Big] \leq 2\mathrm{e}^{-\frac{\mathfrak{d}n}{5248(1+4\eta)}}. \tag{5.57}$$

In order to bound the probability of the event $\{\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq (\mathfrak{d}/16)\sqrt{T_n/n}\}$ using Proposition 21, we must condition on the events

$$\mathcal{D}_{\mathfrak{d}_1,\mathfrak{d}_2} = \big\{\hat{N}_\Gamma \text{ is } (\mathfrak{d}_1,\mathfrak{d}_2)\text{-discrepant}\big\}, \quad \mathcal{B}_{\mathfrak{b}_2} = \Big\{\max_{y\in\Gamma}\big\{\hat{N}_{\Gamma,y} \vee \hat{N}_{y,\Gamma}\big\} \leq \mathfrak{b}_2 \frac{T_n}{n}\Big\}, \tag{5.58}$$

with $\mathfrak{b}_2 > 0$ a sufficiently large constant independent of $n$. By the law of total probability,

$$\begin{aligned}
&\mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}}\Big] \\
&= \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}} \;\Big|\; \mathcal{B}_{\mathfrak{b}_2}\Big]\mathbb{P}[\mathcal{B}_{\mathfrak{b}_2}] \\
&\quad + \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}} \;\Big|\; \mathcal{B}^{\mathrm{c}}_{\mathfrak{b}_2}\Big]\mathbb{P}[\mathcal{B}^{\mathrm{c}}_{\mathfrak{b}_2}] \\
&\leq \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}} \;\Big|\; \mathcal{B}_{\mathfrak{b}_2}\Big] + \mathbb{P}[\mathcal{B}^{\mathrm{c}}_{\mathfrak{b}_2}].
\end{aligned} \tag{5.59}$$

Lemma 42(b) implies that there exists a constant $\mathfrak{b}_2 > 0$ independent of $n$ (which we now will specifically use) such that for sufficiently large $n$,

$$\mathbb{P}[\mathcal{B}^{\mathrm{c}}_{\mathfrak{b}_2}] \leq 2\mathrm{e}^{-\frac{T_n}{n}}. \tag{5.60}$$

To bound the remaining term in (5.59) we can again use the law of total probability:

$$\begin{aligned}
&\mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}} \;\Big|\; \mathcal{B}_{\mathfrak{b}_2}\Big] \\
&= \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}} \;\Big|\; \mathcal{B}_{\mathfrak{b}_2} \cap \mathcal{D}_{\mathfrak{d}_1,\mathfrak{d}_2}\Big]\mathbb{P}[\mathcal{D}_{\mathfrak{d}_1,\mathfrak{d}_2}|\mathcal{B}_{\mathfrak{b}_2}] \\
&\quad + \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}} \;\Big|\; \mathcal{B}_{\mathfrak{b}_2} \cap \mathcal{D}^{\mathrm{c}}_{\mathfrak{d}_1,\mathfrak{d}_2}\Big]\mathbb{P}[\mathcal{D}^{\mathrm{c}}_{\mathfrak{d}_1,\mathfrak{d}_2}|\mathcal{B}_{\mathfrak{b}_2}] \\
&\leq \mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}} \;\Big|\; \mathcal{B}_{\mathfrak{b}_2} \cap \mathcal{D}_{\mathfrak{d}_1,\mathfrak{d}_2}\Big] + \mathbb{P}[\mathcal{D}^{\mathrm{c}}_{\mathfrak{d}_1,\mathfrak{d}_2}|\mathcal{B}_{\mathfrak{b}_2}].
\end{aligned} \tag{5.61}$$

By Proposition 18(a), we find that for sufficiently large $n$,

$$\mathbb{P}[\mathcal{D}^{\mathrm{c}}_{\mathfrak{d}_1,\mathfrak{d}_2}|\mathcal{B}_{\mathfrak{b}_2}] \leq \frac{1}{n}. \tag{5.62}$$

Finally, Proposition 21 tells us that for sufficiently large constants $\mathfrak{b}_2, \mathfrak{d} > 0$ independent of $n$, and sufficiently large $n$,

$$\mathbb{P}\Big[\max_{x,y\in\mathcal{T}_{1/4}} H_1(x,y) \geq \frac{\mathfrak{d}}{16}\sqrt{\frac{T_n}{n}} \;\Big|\; \mathcal{B}_{\mathfrak{b}_2} \cap \mathcal{D}_{\mathfrak{d}_1,\mathfrak{d}_2}\Big] = 0. \tag{5.63}$$

Combining (5.55)–(5.63) yields that there exist a constant $\mathfrak{d} > 0$ independent of $n$ and an integer $m \in \mathbb{N}_+$ such that for all $n \geq m$,

$$\mathbb{P}\Big[\|\hat{N}_\Gamma - N\| \geq \mathfrak{d}\sqrt{\frac{T_n}{n}}\Big] \leq 2\mathrm{e}^{-\frac{\mathfrak{d}n}{5248(1+4\eta)}} + 2\mathrm{e}^{-\frac{T_n}{n}} + \frac{1}{n}. \tag{5.64}$$

We can conclude that if $T_n = \omega(n)$, then $\|\hat{N}_\Gamma - N\| = O_{\mathbb{P}}(\sqrt{T_n/n})$.

In case $T_n = \Omega(n \log n)$, we could have avoided trimming by using Proposition 18(a) and Lemma 42(a) instead. This requires a repetition of the arguments above but with $\hat{N}$ replacing $\hat{N}_\Gamma$. Ultimately, the right-hand side of (5.60) would be replaced by $O(1/n)$. This completes the proof.

## 5.4 Proof of Corollary 7

Now that we have established the tight bound $\sigma_1(\hat{N}_\Gamma - N) = O_{\mathbb{P}}(\sqrt{T_n/n})$ in Theorem 30, we can investigate the singular values of $\hat{N}_\Gamma$. Because we furthermore know the asymptotic order of the singular values of $N$ in (5.11), we can combine these two facts in a perturbative argument using Weyl's inequality:

**Lemma 45** (Weyl's inequality)**.** *Let $A, B \in \mathbb{R}^{s\times m}$ with $s \geq m$, and $\sigma_1(A) \geq \ldots \geq \sigma_m(A)$ and $\sigma_1(B) \geq \ldots \geq \sigma_m(B)$ be the singular values of $A$ and $B$, respectively. If $\|A - B\| \leq \epsilon$, then for all $i = 1, \ldots, m$, $|\sigma_i(A) - \sigma_i(B)| \leq \epsilon$.*

*Proof.* It follows from [151, Theorem 3.3.16] that for $i = 1, \ldots, m$, $\sigma_i(A) \leq \sigma_i(B) + \sigma_1(A - B)$ and $\sigma_i(B) \leq \sigma_i(A) + \sigma_1(B - A)$. The claim follows by noting that $\|A - B\| = \sigma_1(A - B) = \sigma_1(B - A)$. □

For any $\epsilon > 0$, there exists then $\delta_\varepsilon, m_\epsilon > 0$ such that,

$$\begin{aligned}\mathbb{P}\Big[\sigma_{K+1}(\hat{N}_\Gamma) \geq \delta_\varepsilon \sqrt{\frac{T_n}{n}}\Big] &= \mathbb{P}\Big[\sigma_{K+1}(\hat{N}_\Gamma) - \sigma_{K+1}(N) \geq \delta_\varepsilon \sqrt{\frac{T_n}{n}}\Big] \\ &\leq \mathbb{P}\Big[\sigma_1(\hat{N}_\Gamma - N) \geq \delta_\varepsilon \sqrt{\frac{T_n}{n}}\Big] \leq \epsilon \end{aligned} \tag{5.65}$$

for any $n \geq m_\epsilon$. This implies that $\sigma_{K+1}(\hat{N}_\Gamma) = O_{\mathbb{P}}(\sqrt{T_n/n})$ and so $\sigma_i(\hat{N}_\Gamma) = O_{\mathbb{P}}(\sqrt{T_n/n})$ for any $i = K+1, \ldots, n$. Similarly there exist $\kappa_\epsilon, l_\epsilon$ such that for any $n \geq l_\epsilon$ we have

$$\mathbb{P}\Big[|\sigma_K(\hat{N}_\Gamma) - \sigma_K(N)| \geq \kappa_\varepsilon \sqrt{\frac{T_n}{n}}\Big] \leq \mathbb{P}\Big[\sigma_1(\hat{N}_\Gamma - N) \geq \kappa_\varepsilon \sqrt{\frac{T_n}{n}}\Big] \leq \epsilon \tag{5.66}$$

and thus $\sigma_K(\hat{N}_\Gamma) - \sigma_K(N) = O_{\mathbb{P}}(\sqrt{T_n/n})$ also. Recall (5.11), which implies that there exist constants $\mathfrak{a}_1, \mathfrak{a}_2 > 0$ independent of $n$ such that for large $n$ we have $\mathfrak{a}_1 T_n/n \leq \sigma_K(N) \leq \mathfrak{a}_2 T_n/n$. Since $T_n = \omega(n)$, we have $\sqrt{T_n/n} \to \infty$ as $n \to \infty$. Let now $n_0$ be large enough such that for any $n \geq n_0$ we have $\mathfrak{e}_1 \geq \mathfrak{a}_1 - \kappa_\epsilon(\sqrt{T_n/n})^{-1}$ and $\mathfrak{e}_2 \leq \mathfrak{a}_2 + \kappa_\epsilon(\sqrt{T_n/n})^{-1}$ for some $\mathfrak{e}_1, \mathfrak{e}_2 > 0$. Then for any $n \geq \max\{m_\epsilon, l_\epsilon, n_0\}$,

$$\begin{aligned}&\mathbb{P}\Big[\mathfrak{e}_1 T_n/n \leq \sigma_K(\hat{N}_\Gamma) \leq \mathfrak{e}_2 T_n/n\Big] \geq \mathbb{P}\Big[\mathfrak{a}_1 T/n - \kappa_\varepsilon \sqrt{\frac{T_n}{n}} \leq \sigma_K(\hat{N}_\Gamma) \leq \mathfrak{a}_2 T/n + \kappa_\varepsilon \sqrt{\frac{T_n}{n}}\Big] \\ &\geq \mathbb{P}\Big[\{-\kappa_\varepsilon \sqrt{\frac{T_n}{n}} \leq \sigma_K(\hat{N}_\Gamma) - \mathfrak{a}_1 T/n\} \cap \{\sigma_K(\hat{N}_\Gamma) - \mathfrak{a}_2 T/n \leq \kappa_\varepsilon \sqrt{\frac{T_n}{n}}\}\Big] \qquad (5.67) \\ &\geq \mathbb{P}\Big[\big\{-\kappa_\varepsilon \sqrt{\frac{T_n}{n}} \leq \sigma_K(\hat{N}_\Gamma) - \sigma_K(N)\big\} \cap \big\{\sigma_K(\hat{N}_\Gamma) - \sigma_K(N) \leq \kappa_\varepsilon \sqrt{\frac{T_n}{n}}\big\}\Big] \\ &\geq 1 - \epsilon. \end{aligned}$$

The same argument holds for $\sigma_i(\hat{N}_\Gamma)$ for $i = 1, \ldots, K-1$. Hence, we obtain $\sigma_i(\hat{N}_\Gamma) = \Theta_{\mathbb{P}}(T/n)$.

## 5.5 Proof of Proposition 17

We now prove that if $\omega(n) = T_n = o(n^2)$, then there exist constants $\mathfrak{b}, \mathfrak{e}_{\mathfrak{b}} > 0$ independent of $n$ and an integer $n_0 \in \mathbb{N}_+$ such that for any $n \geq n_0$,

$$\mathbb{P}\Big[\|\hat{N} - N\| > \mathfrak{b}\sqrt{\frac{T_n}{n}}\Big] \geq 1 - \mathrm{e}^{-\mathfrak{e}_{\mathfrak{b}}\frac{T_n}{n}}. \tag{5.68}$$

Note from the definition of the spectral norm in (5.21) that

$$\|\hat{N} - N\|^2 \geq \|(1,0,\dots,0)^{\mathrm{T}}(\hat{N} - N)\|_2^2 = \sum_{j=1}^{n} |\hat{N}_{1j} - N_{1j}|^2 \tag{5.69}$$

almost surely. Therefore

$$\mathbb{P}\Big[\|\hat{N} - N\| > \mathfrak{b}\sqrt{\frac{T_n}{n}}\Big] \geq \mathbb{P}\Big[\sum_{j=1}^{n} |\hat{N}_{1j} - N_{1j}|^2 > \mathfrak{b}^2\frac{T_n}{n}\Big] = 1 - \mathbb{P}\Big[\sum_{j=1}^{n} |\hat{N}_{1j} - N_{1j}|^2 \leq \mathfrak{b}^2\frac{T_n}{n}\Big]. \tag{5.70}$$

It is thus enough to prove that there exist constants $\mathfrak{b}, \mathfrak{e}_{\mathfrak{b}} > 0$ independent of $n$ and an integer $n_0 \in \mathbb{N}_+$ such that for all $n \geq n_0$

$$\mathbb{P}\Big[\sum_{j=1}^{n} |\hat{N}_{1j} - N_{1j}|^2 \leq \mathfrak{b}^2\frac{T_n}{n}\Big] \leq \mathrm{e}^{-\mathfrak{e}_{\mathfrak{b}}\frac{T_n}{n}}. \tag{5.71}$$

To prove (5.71), we rely on the following lemma:

**Lemma 46.** *Suppose that $T_n = \omega(n)$.*

*(a) For any constant $\mathfrak{b} \in (0, \frac{1}{4}\pi_{\nu(1)}/\alpha_{\nu(1)})$ independent of $n$, there exist constants $\mathfrak{c}_{\mathfrak{b}} \in [\frac{1}{2}\pi_{\nu(1)}/\alpha_{\nu(1)}, \frac{1}{2}\pi_{\nu(1)}/\alpha_{\nu(1)} + \mathfrak{b}]$, $\mathfrak{d}_{\mathfrak{b}} \in (\pi_{\nu(1)}/\alpha_{\nu(1)}, \pi_{\nu(1)}/\alpha_{\nu(1)} + \mathfrak{b}]$ independent of $n$ and an integer $n_0 \in \mathbb{N}_+$, such that for all $n \geq n_0$,*

$$\mathfrak{c}_{\mathfrak{b}}\frac{T_n}{n} \leq N_{1,[n]} - \mathfrak{b}\frac{T_n}{n} \quad \text{and} \quad N_{1,[n]} \leq \mathfrak{d}_{\mathfrak{b}}\frac{T_n}{n} \leq N_{1,[n]} + \mathfrak{b}\frac{T_n}{n}. \tag{5.72}$$

*In particular*

$$\mathfrak{c}_{\mathfrak{b}}\frac{T_n}{n} \leq N_{1,[n]} - \mathfrak{b}\frac{T_n}{n} \leq \big(\mathfrak{d}_{\mathfrak{b}} - \mathfrak{b}\big)\frac{T_n}{n} \quad \text{and} \quad \mathfrak{c}_{\mathfrak{b}}\frac{T_n}{n} \leq N_{1,[n]} + \mathfrak{b}\frac{T_n}{n} \leq \big(\mathfrak{d}_{\mathfrak{b}} + \mathfrak{b}\big)\frac{T_n}{n}. \tag{5.73}$$

*(b) For any constant $\mathfrak{b} > 0$ independent of $n$, there exists a constant $\mathfrak{e}_{\mathfrak{b}} > 0$ independent of $n$ and an integer $n_1 \in \mathbb{N}_+$, such that for all $n \geq n_1$,*

$$\mathbb{P}\Big[\big|\hat{N}_{1,[n]} - N_{1,[n]}\big| \geq \mathfrak{b}\frac{T_n}{n}\Big] \leq 2\mathrm{e}^{-\mathfrak{e}_{\mathfrak{b}}\frac{T_n}{n}}. \tag{5.74}$$

*Proof.* The first claim follows because $N_{1,[n]} = \Theta(T_n/n)$; *cf.* Lemma 40.

The second claim can be proven using the same strategy as for (5.86), see the argument at (5.92)–(5.94) and equation (5.94) in particular. The only difference is that we need to keep track of the different constants $\mathfrak{b}, \mathfrak{e}_{\mathfrak{b}}$, and that we do not choose $\mathfrak{b}$ to be large. □

Let $\mathfrak{b} \in (0, \frac{1}{4}\pi_{\nu(1)}/\alpha_{\nu(1)})$. By Lemma 46(a) there exist constants $\mathfrak{c}_{\mathfrak{b}} \in [\frac{1}{2}\pi_{\nu(1)}/\alpha_{\nu(1)}, \frac{1}{2}\pi_{\nu(1)}/\alpha_{\nu(1)} + \mathfrak{b}]$, $\mathfrak{d}_{\mathfrak{b}} \in (\pi_{\nu(1)}/\alpha_{\nu(1)}, \pi_{\nu(1)}/\alpha_{\nu(1)} + \mathfrak{b}]$ and an integer $n_0 \in \mathbb{N}_+$ such that for all $n \geq n_0$, the event

$$\Bigl\{ |\hat{N}_{1,[n]} - N_{1,[n]}| \leq \mathfrak{b}\frac{T_n}{n} \Bigr\} \overset{(5.73)}{\subseteq} \Bigl\{ \hat{N}_{1,[n]} \in \Bigl[ \mathfrak{c}_{\mathfrak{b}}\frac{T_n}{n}, (\mathfrak{d}_{\mathfrak{b}} + \mathfrak{b})\frac{T_n}{n} \Bigr] \Bigr\} = \mathcal{B}_{\mathfrak{b}}, \tag{5.75}$$

say. Observe that its complement

$$\mathcal{B}_{\mathfrak{b}}^{\mathrm{c}} \subseteq \Bigl\{ |\hat{N}_{1,[n]} - N_{1,[n]}| > \mathfrak{b}\frac{T_n}{n} \Bigr\}. \tag{5.76}$$

Furthermore, by the law of total probability,

$$\begin{aligned}
&\mathbb{P}\Bigl[ \sum_{j=1}^{n} |\hat{N}_{1,j} - N_{1,j}|^2 \leq \mathfrak{b}\frac{T_n}{n} \Bigr] \\
&= \mathbb{P}\Bigl[ \sum_{j=1}^{n} |\hat{N}_{1,j} - N_{1,j}|^2 \leq \mathfrak{b}\frac{T_n}{n} \Bigm| \mathcal{B}_{\mathfrak{b}} \Bigr] \mathbb{P}[\mathcal{B}_{\mathfrak{b}}] + \mathbb{P}\Bigl[ \sum_{j=1}^{n} |\hat{N}_{1,j} - N_{1,j}|^2 \leq \mathfrak{b}\frac{T_n}{n} \Bigm| \mathcal{B}_{\mathfrak{b}}^{\mathrm{c}} \Bigr] \mathbb{P}[\mathcal{B}_{\mathfrak{b}}^{\mathrm{c}}] \\
&\overset{(5.76)}{\leq} \mathbb{P}\Bigl[ \sum_{j=1}^{n} |\hat{N}_{1,j} - N_{1,j}|^2 \leq \mathfrak{b}\frac{T_n}{n} \Bigm| \mathcal{B}_{\mathfrak{b}} \Bigr] + \mathbb{P}\Bigl[ |\hat{N}_{1,[n]} - N_{1,[n]}| > \mathfrak{b}\frac{T_n}{n} \Bigr] \\
&\overset{\text{(i)}}{\leq} \mathbb{P}\Bigl[ \sum_{j=1}^{n} |\hat{N}_{1,j} - N_{1,j}|^2 \leq \mathfrak{b}\frac{T_n}{n} \Bigm| \mathcal{B}_{\mathfrak{b}} \Bigr] + 2\mathrm{e}^{-\mathfrak{c}_{\mathfrak{b}}\frac{T_n}{n}}.
\end{aligned} \tag{5.77}$$

We (i) used Lemma 46(b) to establish the last step. What remains is to bound the second-to-last term.

Recall that by assumption, $\omega(n) = T_n = o(n^2)$. By Lemma 40 there exist constants $\mathfrak{n}_1 \in (0, \min_{k,l\in[K]} \pi_k q_{k,l}/(\alpha_k\alpha_l))$, $\mathfrak{n}_2 \in (\max_{k\in[K]} \pi_k/(\alpha_k\alpha_l), \infty)$ such that for all sufficiently large $n$, $\mathfrak{n}_1 T_n/n^2 \leq \min_{j\in[n]} N_{1,j} \leq \mathfrak{n}_2 T_n/n^2$. There thus also exists an integer $n_2 \in \mathbb{N}_+$ such that for all $n > n_2$, $\mathfrak{q}_1 T_n/n^2 \leq \min_{j\in[n]} N_{1,j} \leq 1/4$. Let $n \geq n_2$ so that if moreover $\hat{N}_{1,j} \geq 1$, then

$$|\hat{N}_{1,j} - N_{1,j}|^2 \geq \tfrac{1}{2}\hat{N}_{1,j}^2 + N_{1,j}^2; \tag{5.78}$$

and if instead $\hat{N}_{1,j} = 0$, then (5.78) still holds. Thus for any $n \geq n_2$,

$$\sum_{i=1}^{n} |\hat{N}_{1,j} - N_{1,j}|^2 \geq \tfrac{1}{2}\sum_{i=1}^{n} |N_{1,j}|^2 + n \min_{j\in[n]} N_{1,j}^2 \geq \tfrac{1}{2}\sum_{i=1}^{n} |N_{1,j}|^2 + \mathfrak{n}_1^2 \frac{T_n^2}{n^3}. \tag{5.79}$$

By assumption $T_n^2/n^3 = o(T_n/n)$. We can therefore write, for sufficiently large $n$,

$$\mathbb{P}\Bigl[ \sum_{i=1}^{n} |\hat{N}_{1,j} - N_{1,j}|^2 \leq \mathfrak{b}\frac{T_n}{n} \Bigm| \mathcal{B}_{\mathfrak{b}} \Bigr] \overset{(5.79)}{\leq} \mathbb{P}\Bigl[ \sum_{i=1}^{n} |\hat{N}_{1,j}|^2 \leq \bigl(2\mathfrak{b} - o(1)\bigr)\frac{T_n}{n} \Bigm| \mathcal{B}_{\mathfrak{b}} \Bigr]. \tag{5.80}$$

Observe now that if $\sum_{j=1}^{n} |\hat{N}_{1,j}|^2 \leq 2\mathfrak{b}T_n/n$, then at most $m \leq \lfloor 2\mathfrak{b}T_n/n \rfloor$ elements of the row vector $\hat{N}_{1,\cdot}$ can be strictly positive; $\hat{N}_{1,j_1} > 0, \ldots, \hat{N}_{1,j_m} > 0$ say. The reason is that for $j \in [n]$, $\hat{N}_{1,j} \in \mathbb{N}_0$. Using (i) the arithmetic–quadratic–mean inequality, we then obtain

$$\sum_{i=1}^{n} |\hat{N}_{1,j}|^2 = \sum_{k=1}^{m} |\hat{N}_{1,j_k}|^2 \overset{\text{(i)}}{\geq} \frac{1}{m}\Bigl( \sum_{k=1}^{m} \hat{N}_{1,j_k} \Bigr)^2 \geq \frac{n}{2\mathfrak{b}T_n} \hat{N}_{1,[n]}^2. \tag{5.81}$$

Now (i) bound (5.80) by (5.81), to find that (ii) for sufficiently large $n$,

$$\mathbb{P}\Big[\sum_{i=1}^{n}|\hat{N}_{1,j}-N_{1,j}|^2 \leq \mathfrak{b}\frac{T_n}{n}\ \Big|\ \mathcal{B}_{\mathfrak{b}}\Big] \overset{\text{(i)}}{\leq} \mathbb{P}\Big[\frac{n}{2\mathfrak{b}T_n}\hat{N}_{1,[n]}^2 \leq \big(2\mathfrak{b}-o(1)\big)\frac{T_n}{n}\ \Big|\ \mathcal{B}_{\mathfrak{b}}\Big]$$

$$=\mathbb{P}\Big[\hat{N}_{1,[n]} \leq 2\sqrt{\mathfrak{b}(\mathfrak{b}-o(1))}\frac{T_n}{n}\ \Big|\ \mathcal{B}_{\mathfrak{b}}\Big] \overset{\text{(ii)}}{\leq} \mathbb{P}\Big[\hat{N}_{1,[n]} \leq 2\mathfrak{b}\frac{T_n}{n}\ \Big|\ \mathcal{B}_{\mathfrak{b}}\Big] \overset{\text{(iii)}}{=} 0 \tag{5.82}$$

because of (iii) the definition of the event $\mathcal{B}_{\mathfrak{b}}$ in (5.75) combined with the fact that $2\mathfrak{b}T_n/n < \mathfrak{c}_{\mathfrak{b}}T_n/n$ by construction. This completes the proof. □

## 5.6 Numerical validation

We now briefly validate the asymptotics proven in Theorem 30 numerically.[1] We consider a regime that is as sparse as possible. This regime is the most interesting to examine and the asymptotics become challenging to observe due to proximity to the detectability threshold for BMCs.

Figure 5.6.1 shows the scaled spectral norm $\sqrt{n/T_n}\|\hat{N}-N\|$ as a function of $n$ for different asymptotic scalings of $T_n$. We can expect from Theorem 30 that whenever $T_n/n=\omega(1)$, this scaled singular value gap should be $\Theta_{\mathbb{P}}(1)$. Observe that both in the dense regime $T_n=\Omega(n\log n)$ (bottom black curve) and in the sparse regime $\Theta(n)=T_n=o(n\log(\log n))$ (middle red, yellow curves) this holds true. For the even sparser regime $T_n=o(n)$ (top blue curve), it looks like the scaled spectral norm may grow. Such a sparse regime is not covered by our analysis.

# Appendix

## 5.A Proofs of Section 5.2

### 5.A.1 Proof of Lemma 40

Start by noting that $\min_{k,l\in[K]}q_{k,l}>0$ independently of $n$. Consequently, by the Perron–Frobenious theorem, there exists an invariant distribution $\pi=(\pi_1,\dots,\pi_K)$ that also satisfies $\min_{k\in[K]}\pi_k>0$ independently of $n$, since $\pi^{\mathrm{T}}q=\pi^{\mathrm{T}}$ [27, Prop. 1].

Now let $\mathfrak{n}_1,\mathfrak{n}_2$ be independent of $n$ and satisfy

$$0<\mathfrak{n}_1<\min_{k,l\in[K]}\frac{\pi_k q_{k,l}}{\alpha_k\alpha_l}\leq\max_{k,l\in[K]}\frac{\pi_k q_{k,l}}{\alpha_k\alpha_l}<\mathfrak{n}_2<\infty. \tag{5.83}$$

[1] The simulation utilizes the *GNU Scientific Library (GSL)* for random number generation; *Eigen*, a high-level library for linear algebra, matrix, and vector operations; and the *Sparse Eigenvalue Computation Toolkit as a Redesigned ARPACK (SPECTRA)*, a library for large-scale eigenvalue problems built on top of Eigen. The mathematical components of our BMC simulator were furthermore unit tested to ensure validity. Finally, we instructed the Microsoft Visual C++ (MSVC) compiler to activate the *OpenMP* extension to parallelize the simulation across CPUs and so that Eigen could parallelize matrix multiplications (/openmp); to apply maximum optimization (/O2); to enable enhanced CPU instruction sets (/arch:AVX2); and to explicitly target 64-bit x64 hardware. The code can be found at https://gitlab.tue.nl/acss/public/spectral-norm-bounds-for-block-markov-chain-random-matrices.

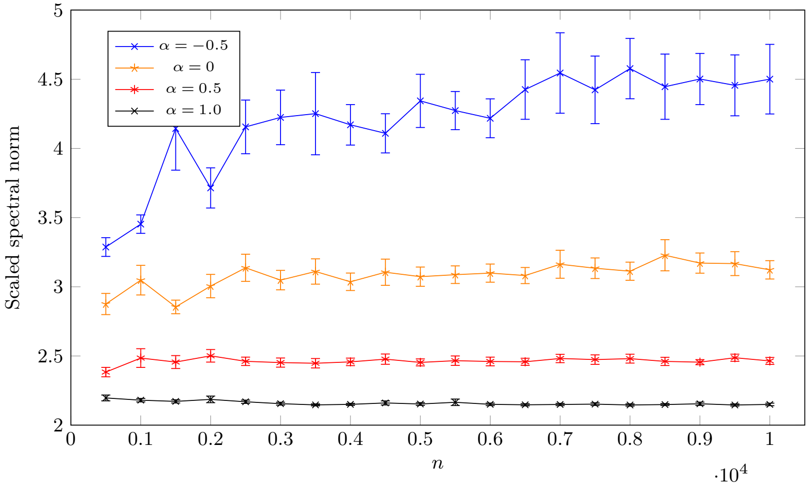


*Figure 5.6.1: Plots of the scaled spectral norm* $\sqrt{n/T_n}\|\hat{N}-N\|$ *for different asymptotic scalings of* $T_n$ *and* $n = 500, 1\,000, \ldots, 10\,000$*. The choice of the BMC parameters is* $q = \frac{1}{10}((2,3,5);(3,5,2);(5,2,3))$ *and* $\alpha = \frac{1}{3}(1,1,1)$*. Each* 95%*-confidence interval was the result of approximately* 48 *independent replications. The trajectory length was set to* $T_n = [n(\log n)^{\alpha}]$*, and* $|\Gamma^{\mathrm{c}}| = 0$ *states were trimmed. The curves correspond to* $\alpha = -0.5, 0, 0.5, 1$ *from top to bottom.*

Let $x, y \in [n]$. Observe that

$$\lim_{n\to\infty} \frac{N_{x,y}}{T_n/n^2} = \lim_{n\to\infty} n^2 \Pi_x P_{x,y} = \lim_{n\to\infty} n^2 \frac{\pi_{\nu(x)} q_{\nu(x),\nu(y)}}{|\mathcal{V}_{\nu(x)}||\mathcal{V}_{\nu(y)}|} = \frac{\pi_{\nu(x)} q_{\nu(x),\nu(y)}}{\alpha_{\nu(x)}\alpha_{\nu(y)}}. \tag{5.84}$$

Consequently

$$\mathfrak{n}_1 < \lim_{n\to\infty} \frac{N_{x,y}}{T_n/n^2} < \mathfrak{n}_2. \tag{5.85}$$

The conclusions pertaining to $P_{x,y}$ follow *mutatis mutandis*. This completes the proof.
□

### 5.A.2 Proof of Lemma 42

Proving this next claim is sufficient: if $T_n = \omega(n)$, then there exists a constant $\mathfrak{b}_3 > 0$ independent of $n$ such that for any $x \in [n]$ and all sufficiently large $n$,

$$\mathbb{P}\Big[\big|\hat{N}_{[n],x} - N_{[n],x}\big| \geq \mathfrak{b}_3 \frac{T_n}{n}\Big] \leq 2\mathrm{e}^{-2\frac{T_n}{n}}. \tag{5.86}$$

Lemma 42(a) would namely follow almost immediately. If $T_n = \Omega(n\log n)$, then

$$\begin{aligned} \mathbb{P}\Big[\max_{y\in[n]}\big\{\hat{N}_{y,[n]} \vee \hat{N}_{[n],y}\big\} \geq \mathfrak{b}_1 \frac{T_n}{n}\Big] &\overset{\text{(i)}}{\leq} \mathbb{P}\Big[\max_{y\in[n]} \hat{N}_{[n],y} \geq \mathfrak{b}_1 \frac{T_n}{n} - 1\Big] \\ &\overset{\text{(ii)}}{\leq} n \max_{x\in[n]} \mathbb{P}\Big[\hat{N}_{[n],x} \geq \mathfrak{b}_1 \frac{T_n}{n} - 1\Big] \end{aligned} \tag{5.87}$$

$$\overset{\text{(iii)}}{\le} n \max_{x\in[n]} \mathbb{P}\Big[|\hat{N}_{[n],x} - N_{[n],x}| \ge \mathfrak{b}_4 \frac{T_n}{n}\Big] \overset{(5.86)}{\le} 2n\mathrm{e}^{-2\log n}$$

for sufficiently large $n$ and some $\mathfrak{b}_4 > 0$ if $\mathfrak{b}_1$ is large enough. Here, we used (i) that for $y \in [n]\backslash\{X_0, X_{T_n}\}$, $\hat{N}_{y,[n]} = \hat{N}_{[n],y}$ and for $y \in \{X_0, X_{T_n}\}$, $|\hat{N}_{y,[n]} - \hat{N}_{[n],y}| \le 1$; (ii) Boole's inequality; and (iii) that for all $x \in [n]$, $N_{[n],x} = \Theta(T_n/n)$.

We prove now Lemma 42(b). We may assume without loss of generality that $|\Gamma^{\mathrm{c}}| = \lfloor n\mathrm{e}^{-T_n/n}\rfloor \ge 1$ since otherwise we would be in the previous case. Equivalently, $T_n \le n\log n$. Let

$$\mathcal{H} = \Big\{y \in [n] : \hat{N}_{[n],y} \ge \mathfrak{b}_2 \frac{T_n}{n} - 1\Big\} \tag{5.88}$$

be the set of states of at least the indicated degree. By (iv) construction of the trimming procedure and the set $\mathcal{H}$,

$$\mathbb{P}\big(\max_{y\in\Gamma}\{\hat{N}_{y,\Gamma} \vee \hat{N}_{\Gamma,y}\} \ge \mathfrak{b}_2 \frac{T_n}{n}\big) \overset{\text{(i)}}{\le} \mathbb{P}\big(\max_{y\in\Gamma} \hat{N}_{y,[n]} \ge \mathfrak{b}_2 \frac{T_n}{n} - 1\big) \overset{\text{(iv)}}{=} \mathbb{P}[|\mathcal{H}| > |\Gamma^{\mathrm{c}}|]. \tag{5.89}$$

By (v) Markov's inequality,

$$\begin{aligned}\mathbb{P}[|\mathcal{H}| > |\Gamma^{\mathrm{c}}|] &= \mathbb{P}\Big[\sum_{y\in[n]} \mathbb{1}\Big[\hat{N}_{[n],y} \ge \mathfrak{b}_2 \frac{T_n}{n} - 1\Big] \ge |\Gamma^{\mathrm{c}}| + 1\Big] \\ &\overset{\text{(v)}}{\le} \frac{1}{|\Gamma^{\mathrm{c}}|+1} \sum_{x\in[n]} \mathbb{P}\Big[\hat{N}_{[n],x} \ge \mathfrak{b}_2 \frac{T_n}{n} - 1\Big] \\ &\le \frac{1}{|\Gamma^{\mathrm{c}}|+1} \sum_{x\in[n]} \mathbb{P}\Big[\big|\hat{N}_{[n],x} - N_{[n],x}\big| \ge \mathfrak{b}_5 \frac{T_n}{n}\Big],\end{aligned} \tag{5.90}$$

for some $\mathfrak{b}_5 > 0$ if $\mathfrak{b}_2$ is large enough. Finally apply (5.86) and lower bound $|\Gamma^{\mathrm{c}}| \ge n\mathrm{e}^{-T_n/n} - 1$ to find that

$$\mathbb{P}\Big[\max_{y\in\Gamma}\{\hat{N}_{y,\Gamma} \vee \hat{N}_{\Gamma,y}\} \ge \mathfrak{b}_2 \frac{T_n}{n}\Big] \le 2\mathrm{e}^{-\frac{T_n}{n}}. \tag{5.91}$$

The final inequality follows because $T_n \le n\log n$. What remains is to prove (5.86).

*Proof of* (5.86). This is a tightening of [27, SM1(20)] by a logarithmic term, and the argument is a straightforward modification. Let $f(\cdot) = \mathbb{1}[\cdot = x]$ be such that $\sum_{t=1}^{T_n} f(X_t) = \hat{N}_{[n],x}$. Clearly for $x \in [n]$, $|f(X_t) - \mathbb{E}_\Pi[f(X_t)]| = |\mathbb{1}[X_t = x] - \Pi_x| \le 1 = C$ say. Moreover, for $x \in [n]$,

$$\begin{aligned}V_f = \mathrm{Var}[f(X_t)] &= \mathbb{E}[(\mathbb{1}[X_t = x] - \Pi_x)^2] \\ &= \mathbb{E}[\mathbb{1}[X_t = x]] - \Pi_x^2 = \Pi_x(1-\Pi_x) \le \frac{\pi_{\nu(x)}}{|\mathcal{V}_\nu(x)|} \le \max_{k\in[K]} \frac{\pi_k}{\alpha_k n} + o\Big(\frac{1}{n}\Big).\end{aligned} \tag{5.92}$$

By [98, Thm 3.4],

$$\mathbb{P}\Big[|\hat{N}_{[n],x} - N_{[n],x}| \ge z\Big] \le 2\exp\Big(-\frac{z^2\gamma_{\mathrm{ps}}}{8(T_n + 1/\gamma_{\mathrm{ps}})V_f + 20zC}\Big). \tag{5.93}$$

Let $\mathfrak{b}_5 > 0$ and specify $z = \mathfrak{b}_5 T_n/n$. Recall that $\gamma_{\mathrm{ps}} \ge 1/(8\eta + 2)$ [27, (26)]. Therefore

$$\mathbb{P}\Big[|\hat{N}_{[n],x} - N_{[n],x}| \ge \mathfrak{b}_5 \frac{T_n}{n}\Big] \le 2\exp\Big(-\frac{T_n}{n} \cdot \frac{\mathfrak{b}_5^2/(8\eta+2)}{20\mathfrak{b}_5 + 8\max_{k\in[K]} \pi_k/\alpha_k + o(1) + O(1/T_n)}\Big). \tag{5.94}$$

Choosing $\mathfrak{b}_5$ sufficiently large completes the proof. □

### 5.A.3 Proof of Proposition 18

We will prove Proposition 18 by modifying the proof approach of [93, Lem. 4.2]. The key difference is that in the present setting the entries of $\hat{N}$ are not independent. A similar argument can also be found in [27, Lem. 12] for a different definition of the discrepancy property. The discrepancy property differs in the present chapter so as to provide a tighter bound.

Observe that

$$\mathbb{E}[e(\mathcal{A},\mathcal{B})] = \sum_{x\in\mathcal{A}}\sum_{y\in\mathcal{B}}\mathbb{E}[\hat{N}_{x,y}] = \sum_{x\in\mathcal{A}}\sum_{y\in\mathcal{B}} N_{x,y}. \tag{5.95}$$

We therefore have as an immediate corollary of Lemma 40:

**Corollary 8.** *There exist constants $0 < \mathfrak{m}_1 < \mathfrak{m}_2 < \infty$ independent of $n$ and an integer $m \in \mathbb{N}_+$ such that for all $n \geq m$ and all $\mathcal{A},\mathcal{B} \subseteq [n]$, $\mathfrak{m}_1|\mathcal{A}||\mathcal{B}|T_n/n^2 \leq \mu(\mathcal{A},\mathcal{B}) \leq \mathfrak{m}_2|\mathcal{A}||\mathcal{B}|T_n/n^2$.*

We will prove the following lemma:

**Lemma 47.** *There exist a constant $k_0 > 0$ independent of $n$ and an integer $m \in \mathbb{N}_+$ such that for all $n \geq m$, all $k \geq k_0$, and all $\mathcal{A},\mathcal{B} \subseteq [n]$,*

$$\mathbb{P}[e(\mathcal{A},\mathcal{B}) \geq k\mu(\mathcal{A},\mathcal{B})] \leq 2\exp\Big(-\tfrac{1}{4}\mu(\mathcal{A},\mathcal{B})k\log k\Big). \tag{5.96}$$

The fact that the discrepancy property holds with high probability for both $\hat{N}$ and $\hat{N}_\Gamma$ follows from Corollary 8, Lemma 47, whenever $T_n = \Omega(n\log(n))$ and $T_n = \omega(n)$ respectively:

**Proof of Proposition 18 — Case $\hat{N}$, i.e., without trimming.**

We consider first the case without trimming. For convenience, let $\Delta = \max_{y\in[n]}\big\{\hat{N}_{[n],y} \vee \hat{N}_{y,[n]}\big\}$. Also, we assume that $|\mathcal{A}| \leq |\mathcal{B}|$ without loss of generality.

**Subcases $|\mathcal{B}| > n/\mathrm{e}$** For this subcase, the discrepancy property is satisfied since Definition 32(i) holds. Observe first that for all $\mathcal{A},\mathcal{B} \subseteq [n]$,

$$e(\mathcal{A},\mathcal{B}) \overset{(5.20)}{=} \sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}} \hat{N}_{ij} \leq \Delta\min\{|\mathcal{A}|,|\mathcal{B}|\} = \Delta|\mathcal{A}|. \tag{5.97}$$

We therefore have in particular that for all $\mathcal{A},\mathcal{B} \subseteq [n]$ such that $|\mathcal{B}| > n/\mathrm{e}$,

$$\frac{e(\mathcal{A},\mathcal{B})n^2}{|\mathcal{A}||\mathcal{B}|T_n} \leq \frac{\mathrm{e}e(\mathcal{A},\mathcal{B})n}{|\mathcal{A}|T_n} \leq \frac{\mathrm{e}\Delta|\mathcal{A}|n}{|\mathcal{A}|T_n} \leq \mathfrak{b}_3\mathrm{e}, \tag{5.98}$$

since $\Delta \leq \mathfrak{b}_3 T_n/n$ by assumption.

**Subcases $0 < |\mathcal{B}| \leq n/\mathrm{e}$** Lemma 47 tells us that there exists a constant $k_0 > 0$ independent of $n$ such that for all sufficiently large $n$, and all $k \geq k_0$, $\mathbb{P}[e(\mathcal{A},\mathcal{B}) > k\mu(\mathcal{A},\mathcal{B})] \leq 2\exp(-k\log(k)\mu(\mathcal{A},\mathcal{B})/4)$. We may presume that $k_0 \geq \max(1, 1/\mathfrak{m}_2, \mathfrak{m}_2)$, where we let $0 < \mathfrak{m}_1 < \mathfrak{m}_2 < \infty$ be as in Corollary 8. The reason for this choice will become apparent later on.

Let $n$ be sufficiently large, and let $\mathfrak{c}_1,\mathfrak{c}_2>0$ be constants independent of $n$ that we will choose later. Denote by $t^\star(\mathcal{A},\mathcal{B})>0$ the unique solution to

$$t\log t=\mathfrak{m}_1\mathfrak{c}_2|\mathcal{B}|\log(n/|\mathcal{B}|)/(2\mathfrak{m}_2\mu(\mathcal{A},\mathcal{B}))>0, \tag{5.99}$$

which exists and is unique because the function $t\log t$ is monotonically increasing for $t\geq 1$. Define $k^\star(\mathcal{A},\mathcal{B})=\max\{k_0,t^\star(\mathcal{A},\mathcal{B})\}$ and consider the event

$$\mathcal{E}=\bigcap_{\mathcal{A},\mathcal{B}\subseteq[n]:|\mathcal{A}|\leq|\mathcal{B}|\leq n/\mathrm{e}}\big\{e(\mathcal{A},\mathcal{B})\leq\mu(\mathcal{A},\mathcal{B})k^\star(\mathcal{A},\mathcal{B})\big\}. \tag{5.100}$$

We claim that if $\mathcal{E}$ holds, then $\hat{N}$ is discrepant. Specifically, for all pairs $(\mathcal{A},\mathcal{B})$ such that $k^\star(\mathcal{A},\mathcal{B})=k_0$, Definition 32(i) holds. Indeed, for any such pair, on event $\mathcal{E}$,

$$e(\mathcal{A},\mathcal{B})\leq k_0\mu(\mathcal{A},\mathcal{B})\leq k_0\mathfrak{m}_2\frac{T_n|\mathcal{A}||\mathcal{B}|}{n^2}\leq\mathfrak{c}_1\frac{T_n|\mathcal{A}||\mathcal{B}|}{n^2} \tag{5.101}$$

by Corollary 8 if $\mathfrak{c}_1\geq k_0\mathfrak{m}_2$. Furthermore: for all pairs $(\mathcal{A},\mathcal{B})$ such that $k^\star(\mathcal{A},\mathcal{B})=t^\star(\mathcal{A},\mathcal{B})>k_0\geq\max(1,1/\mathfrak{m}_2)$, Definition 32(ii) holds. To see this, consider that for any such pair $e(\mathcal{A},\mathcal{B})\leq\mu(\mathcal{A},\mathcal{B})t^\star(\mathcal{A},\mathcal{B})$ by event $\mathcal{E}$. Therefore, again by Corollary 8,

$$\frac{n^2e(\mathcal{A},\mathcal{B})}{T_n|\mathcal{A}||\mathcal{B}|}\leq\mathfrak{m}_2t^\star(\mathcal{A},\mathcal{B}) \tag{5.102}$$

on event $\mathcal{E}$. Remark now that for $1\leq t\leq\mathfrak{m}_2t^\star$, $t\log t\leq(\mathfrak{m}_2t^\star)\log(\mathfrak{m}_2t^\star)$ by monotonicity. Furthermore from the lower bound on $t^\star$, $\mathfrak{m}_2t^\star\geq\mathfrak{m}_2\max(1,1/\mathfrak{m}_2,\mathfrak{m}_2)\geq 1$; hence, for $0\leq t<1$, $t\log t\leq(\mathfrak{m}_2t^\star)\log(\mathfrak{m}_2t^\star)$ also (observe that the left-hand side of the inequality is nonpositive and the right-hand side nonnegative). Therefore, on the event $\mathcal{E}$,

$$\frac{n^2e(\mathcal{A},\mathcal{B})}{T_n|\mathcal{A}||\mathcal{B}|}\log\frac{n^2e(\mathcal{A},\mathcal{B})}{T_n|\mathcal{A}||\mathcal{B}|}\overset{(5.102)}{\leq}(\mathfrak{m}_2t^\star)\log(\mathfrak{m}_2t^\star)\overset{\text{(iii)}}{\leq}2\mathfrak{m}_2t^\star\log t^\star\overset{\text{(iv)}}{=}\frac{\mathfrak{m}_1\mathfrak{c}_2|\mathcal{B}|}{\mu(\mathcal{A},\mathcal{B})}\log\frac{n}{|\mathcal{B}|} \tag{5.103}$$

because (iii) $\log(\mathfrak{m}_2t^\star)\leq 2\log t^\star$ since $t^\star\geq\max(1,1/\mathfrak{m}_2,\mathfrak{m}_2)\geq\mathfrak{m}_2$ and (iv) of the definition of $t^\star(\mathcal{A},\mathcal{B})$. After (v) multiplying (5.103) by $(T_n/n^2)|\mathcal{A}||\mathcal{B}|$ and (vi) utilizing Corollary 8 one final time, observe that

$$e(\mathcal{A},\mathcal{B})\log\frac{n^2e(\mathcal{A},\mathcal{B})}{T_n|\mathcal{A}||\mathcal{B}|}\overset{\text{(v)}}{\leq}\mathfrak{m}_1\mathfrak{c}_2|\mathcal{B}|\frac{(T_n/n^2)|\mathcal{A}||\mathcal{B}|}{\mu(\mathcal{A},\mathcal{B})}\log\frac{n}{|\mathcal{B}|}\overset{\text{(vi)}}{\leq}\mathfrak{c}_2|\mathcal{B}|\log\frac{n}{|\mathcal{B}|}. \tag{5.104}$$

What remains is to prove that the event $\mathcal{E}$ holds at least with probability $1-1/n$. We can do so using the De Morgan identities, which imply that

$$\mathbb{P}[\mathcal{E}]=1-\mathbb{P}\Big[\bigcup_{\mathcal{A},\mathcal{B}\subseteq[n]:|\mathcal{A}|\leq|\mathcal{B}|\leq n/\mathrm{e}}\big\{e(\mathcal{A},\mathcal{B})>\mu(\mathcal{A},\mathcal{B})k^\star\big\}\Big], \tag{5.105}$$

and then upper bounding the right term by $1/n$. By (i) Boole's inequality, (ii) Lemma 47, and (iii) the definition of $k^\star(\mathcal{A},\mathcal{B})$ and Corollary 8, for sufficiently large $n$,

$$\begin{aligned}
&\mathbb{P}\Big[\bigcup_{\mathcal{A},\mathcal{B}\subseteq[n]:|\mathcal{A}|\leq|\mathcal{B}|\leq n/\mathrm{e}}\big\{e(\mathcal{A},\mathcal{B})>\mu(\mathcal{A},\mathcal{B})k^\star(\mathcal{A},\mathcal{B})\big\}\Big]\\
&\overset{\text{(i)}}{\leq}\sum_{\mathcal{A},\mathcal{B}\subseteq[n]:|\mathcal{A}|\leq|\mathcal{B}|\leq n/\mathrm{e}}\mathbb{P}\Big[e(\mathcal{A},\mathcal{B})>\mu(\mathcal{A},\mathcal{B})k^\star(\mathcal{A},\mathcal{B})\Big]\\
&\overset{\text{(ii)}}{\leq}\sum_{\mathcal{A},\mathcal{B}\subseteq[n]:|\mathcal{A}|\leq|\mathcal{B}|\leq n/\mathrm{e}}2\exp\Big(-\tfrac{1}{4}\mu(\mathcal{A},\mathcal{B})k^\star(\mathcal{A},\mathcal{B})\log k^\star(\mathcal{A},\mathcal{B})\Big)\\
&\overset{\text{(iii)}}{\leq}\sum_{\mathcal{A},\mathcal{B}\subseteq[n]:|\mathcal{A}|\leq|\mathcal{B}|\leq n/\mathrm{e}}2\exp\Big(-\frac{\mathfrak{m}_1\mathfrak{c}_2}{8\mathfrak{m}_2}|\mathcal{B}|\log\frac{n}{|\mathcal{B}|}\Big).
\end{aligned}\tag{5.106}$$

Finally, by (iv) collecting terms and upper bounding their numbers, and utilizing (v) $\binom{n}{s}\leq(n\mathrm{e}/s)^s$ and (vi) for $t\in[1,n/\mathrm{e}]$, $t\leq t\log(n/t)$, we find that for sufficiently large $n$,

$$\begin{aligned}
(5.106)&\overset{\text{(iv)}}{\leq}\sum_{1\leq a\leq b\leq n/\mathrm{e}}2\binom{n}{a}\binom{n}{b}\exp\Big(-\frac{\mathfrak{m}_1\mathfrak{c}_2}{8\mathfrak{m}_2}b\log\frac{n}{b}\Big)\\
&\overset{\text{(v)}}{\leq}\sum_{1\leq a\leq b\leq n/\mathrm{e}}2\Big(\frac{n\mathrm{e}}{a}\Big)^a\Big(\frac{n\mathrm{e}}{b}\Big)^b\exp\Big(-\frac{\mathfrak{m}_1\mathfrak{c}_2}{8\mathfrak{m}_2}b\log\frac{n}{b}\Big)\\
&\leq\sum_{1\leq a\leq b\leq n/\mathrm{e}}2\exp\Big(a+a\log\frac{n}{a}+b+b\log\frac{n}{b}-\frac{\mathfrak{m}_1\mathfrak{c}_2}{8\mathfrak{m}_2}b\log\frac{n}{b}\Big)\\
&\leq\sum_{1\leq a\leq b\leq n/\mathrm{e}}2\exp\Big(2b+2b\log\frac{n}{b}-\frac{\mathfrak{m}_1\mathfrak{c}_2}{8\mathfrak{m}_2}b\log\frac{n}{b}\Big)\\
&\overset{\text{(vi)}}{\leq}\sum_{1\leq a\leq b\leq n/\mathrm{e}}2\exp\Big(4b\log\frac{n}{b}-\frac{\mathfrak{m}_1\mathfrak{c}_2}{8\mathfrak{m}_2}b\log\frac{n}{b}\Big)\\
&\leq\sum_{1\leq a\leq b\leq n/\mathrm{e}}2\exp\Big(-\Big(\frac{\mathfrak{m}_1\mathfrak{c}_2}{8\mathfrak{m}_2}-4\Big)b\log\frac{n}{b}\Big)\\
&\leq\sum_{1\leq a\leq b\leq n/\mathrm{e}}2n^{-\frac{\mathfrak{m}_1\mathfrak{c}_2}{8\mathfrak{m}_2}+4}\leq n^{-\frac{\mathfrak{m}_1\mathfrak{c}_2}{8\mathfrak{m}_2}+7}.
\end{aligned}\tag{5.107}$$

The event $\mathcal{E}$ thus holds at least with probability $1-1/n$ for sufficiently large $n$ if we choose the constants in Definition 32 to be $\mathfrak{d}_2=\mathfrak{c}_2\geq 64\mathfrak{m}_2/\mathfrak{m}_1$. Finally, from (5.98) and (5.101) we also need to choose $\mathfrak{d}_1=\mathfrak{c}_1\geq\max\{\mathfrak{b}_3\mathrm{e},k_0\mathfrak{m}_2\}$. This completes the proof. □

**Proof of Proposition 18 — Case $\hat{N}_\Gamma$, i.e., with trimming.**

We consider second the case with trimming. For notational convenience, let now $\Delta_\Gamma=\max_{y\in[n]}\big\{\hat{N}_{\Gamma,y}\vee\hat{N}_{y,\Gamma}\big\}$. Again, we assume that $|\mathcal{A}|\leq|\mathcal{B}|$ without loss of generality.

**Subcases $|\mathcal{B}|>n/\mathrm{e}$** Equations (5.97)–(5.98) hold *mutatis mutandis* after replacing $e(\mathcal{A},\mathcal{B})$ by $e_\Gamma(\mathcal{A},\mathcal{B})$, $\hat{N}$ by $\hat{N}_\Gamma$, $\Delta$ by $\Delta_\Gamma$ and $\mathfrak{b}_3$ by $\mathfrak{b}_4$.

**Subcases $\mathcal{B} \leq n/\mathrm{e}$** Equations (5.100)–(5.107) hold *mutatis mutandis* after replacing $e(\mathcal{A},\mathcal{B})$ by $e_\Gamma(\mathcal{A},\mathcal{B})$. This is because $e_\Gamma(\mathcal{A},\mathcal{B}) \leq e(\mathcal{A},\mathcal{B})$. In particular, utilize the monotonicity of $t \log t$ for proving counterparts to (5.101)–(5.104), and the inequality

$$\mathbb{P}[e_\Gamma(\mathcal{A},\mathcal{B}) > \mu(\mathcal{A},\mathcal{B})k^\star] \leq \mathbb{P}[e(\mathcal{A},\mathcal{B}) > \mu(\mathcal{A},\mathcal{B})k^\star] \tag{5.108}$$

for showing replacements of (5.105)–(5.106). The constant $\mathfrak{d}_1$ will now depend on $\mathfrak{b}_4$.

This completes the first part. □

What remains is to prove Lemma 47. This is done in Appendix 5.A.3.

### Proof of Lemma 47

Let $a \in \mathbb{R}$ and $n \in \mathbb{N}_+$ for now be arbitrary. We will first bound

$$\mathbb{P}[e(\mathcal{A},\mathcal{B}) - \mu(\mathcal{A},\mathcal{B}) \geq a]. \tag{5.109}$$

To do so, we are going to split the sum $e(\mathcal{A},\mathcal{B})$ into two parts. Let $E(T_n) = \{t \in \{0,1,\ldots,T_n - 1\} : t \equiv 0 \mod 2\}$ denote the even numbers up to $T_n - 1$, and $O(T_n) = \{0,1,\ldots,T_n - 1\} \backslash E(T_n)$ denote the odd numbers up to $T_n - 1$. Write

$$\begin{aligned}
e(\mathcal{A},\mathcal{B}) - \mu(\mathcal{A},\mathcal{B}) &= \sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}}\bigl(\hat{N}_{ij} - N_{ij}\bigr) \\
&= \sum_{t=0}^{T_n-1}\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}}\bigl(\mathbb{1}[X_t = i, X_{t+1} = j] - \tfrac{1}{T_n}N_{ij}\bigr) \\
&= \sum_{t\in E(T_n)}\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}}\bigl(\mathbb{1}[X_t = i, X_{t+1} = j] - \tfrac{1}{T_n}N_{ij}\bigr) \\
&\qquad + \sum_{t\in O(T_n)}\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}}\bigl(\mathbb{1}[X_t = i, X_{t+1} = j] - \tfrac{1}{T_n}N_{ij}\bigr) \\
&= e_0(\mathcal{A},\mathcal{B}) - \tfrac{1}{2}\mu(\mathcal{A},\mathcal{B}) + e_1(\mathcal{A},\mathcal{B}) - \tfrac{1}{2}\mu(\mathcal{A},\mathcal{B}),
\end{aligned} \tag{5.110}$$

say. Using a union bound we obtain

$$\mathbb{P}[e(\mathcal{A},\mathcal{B}) - \mu(\mathcal{A},\mathcal{B}) \geq a] \leq \mathbb{P}[2e_0(\mathcal{A},\mathcal{B}) - \mu(\mathcal{A},\mathcal{B}) \geq a] + \mathbb{P}[2e_1(\mathcal{A},\mathcal{B}) - \mu(\mathcal{A},\mathcal{B}) \geq a]. \tag{5.111}$$

It suffices to bound either of the right members by symmetry.

Suppose therefore that, without loss of generality, the probability pertaining to $e_0(\mathcal{A},\mathcal{B})$ is larger. By Markov's inequality,

$$\begin{aligned}
\mathbb{P}[2e_0(\mathcal{A},\mathcal{B}) - \mu(\mathcal{A},\mathcal{B}) \geq a] &\leq \inf_{h>0}\Bigl\{\mathrm{e}^{-ha}\mathbb{E}\Bigl[\mathrm{e}^{h(2e_0(\mathcal{A},\mathcal{B}) - \mu(\mathcal{A},\mathcal{B}))}\Bigr]\Bigr\} \\
&= \inf_{h>0}\Bigl\{\mathrm{e}^{-h(a+\mu(\mathcal{A},\mathcal{B}))}\mathbb{E}[\mathrm{e}^{2he_0(\mathcal{A},\mathcal{B})}]\Bigr\}.
\end{aligned} \tag{5.112}$$

For $t \in \{0,1,\ldots,T_n\}$, let $\mathcal{F}_t$ be the $\sigma$-algebra generated by $\{X_0,\ldots,X_t\}$. By the law of total expectation,

$$\begin{aligned}
\mathbb{E}[\mathrm{e}^{2he_0(\mathcal{A},\mathcal{B})}] &= \mathbb{E}\Bigl[\mathbb{E}\Bigl[\mathrm{e}^{2he_0(\mathcal{A},\mathcal{B})} \Bigm| \mathcal{F}_{T_n-2}\Bigr]\Bigr] \qquad (5.113) \\
&= \mathbb{E}\Bigl[\mathbb{E}\Bigl[\mathrm{e}^{2h\sum_{t\in E(T_n)}\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}}\mathbb{1}[X_t=i,X_{t+1}=j])} \Bigm| \mathcal{F}_{T_n-2}\Bigr]\Bigr] \\
&= \mathbb{E}\Bigl[\mathrm{e}^{2h\sum_{t\in E(T_n-2)}\sum_{i\in\mathcal{A},j\in\mathcal{B}}\mathbb{1}[X_t=i,X_{t+1}=j]}\mathbb{E}\Bigl[\mathrm{e}^{2h\sum_{i\in\mathcal{A},j\in\mathcal{B}}\mathbb{1}[X_{T_n-1}=i,X_{T_n}=j]} \Bigm| \mathcal{F}_{T_n-2}\Bigr]\Bigr].
\end{aligned}$$

We can in principle calculate the inner conditional expectation. An upper bound suffices however, which we will derive next.

Let $h > 0$. By Lemma 40, there exist a constant $\mathfrak{p}_2 > 0$ and integer $m \in \mathbb{N}_+$ such that for all $n \geq m$,

$$\begin{aligned}
&\mathbb{E}\Big[\mathrm{e}^{2h\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}}\hat{N}_{ij}(T_n)}\,\Big|\,\mathcal{F}_{T_n-2}\Big]\\
&=\mathbb{E}\Big[\mathbb{1}[(X_{T_n-1},X_{T_n})\in[n]^2\backslash(\mathcal{A}\times\mathcal{B})]+\mathrm{e}^{2h}\mathbb{1}[(X_{T_n-1},X_{T_n})\in\mathcal{A}\times\mathcal{B}]\,\Big|\,\mathcal{F}_{T_n-2}\Big]\\
&\leq\mathbb{E}\Big[1+\mathrm{e}^{2h}\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}}\mathbb{1}[(X_{T_n-1},X_{T_n})=(i,j)]\,\Big|\,\mathcal{F}_{T_n-2}\Big]\\
&\leq 1+\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}}\mathrm{e}^{2h}P_{X_{T_n-2},i}P_{i,j}\leq 1+\mathrm{e}^{2h}\mathfrak{p}_2\frac{|\mathcal{A}||\mathcal{B}|}{n^2}.
\end{aligned}\tag{5.114}$$

Bounding (5.113) by (5.114), iterating the argument $T_n/2$ times, and using the elementary bound $1+z \leq \mathrm{e}^z$ for $z \geq 0$, we obtain

$$\mathbb{E}[\mathrm{e}^{2he_0(\mathcal{A},\mathcal{B})}]\leq\Big(1+\mathrm{e}^{2h}\mathfrak{p}_2\frac{|\mathcal{A}||\mathcal{B}|}{n^2}\Big)^{T_n/2}\leq\exp\Big(\frac{T_n}{2}\mathrm{e}^{2h}\mathfrak{p}_2\frac{|\mathcal{A}||\mathcal{B}|}{n^2}\Big).\tag{5.115}$$

Hence, for all $n \geq m$,

$$\mathbb{P}[2e_0(\mathcal{A},\mathcal{B})-\mu(\mathcal{A},\mathcal{B})\geq a]\leq\inf_{h>0}\Big\{\exp\Big(-h(a+\mu(\mathcal{A},\mathcal{B}))+\frac{T_n}{2}\mathrm{e}^{2h}\mathfrak{p}_2\frac{|\mathcal{A}||\mathcal{B}|}{n^2}\Big)\Big\}.\tag{5.116}$$

Finally, we specify $a=(k-1)\mu(\mathcal{A},\mathcal{B})$. Observe that $a$ is $n$-dependent. The infimum in (5.116) then occurs at

$$h_n^{\mathrm{opt}}=\tfrac{1}{2}\log\frac{k\mu(\mathcal{A},\mathcal{B})n^2}{T_n\mathfrak{p}_2|\mathcal{A}||\mathcal{B}|}.\tag{5.117}$$

Substituting (5.117) into (5.116) we find that for all $n \geq m$,

$$\mathbb{P}[2e_0(\mathcal{A},\mathcal{B})-\mu(\mathcal{A},\mathcal{B})\geq(k-1)\mu(\mathcal{A},\mathcal{B})]\leq\exp\Big(\tfrac{1}{2}k\mu(\mathcal{A},\mathcal{B})\Big(1-\log\frac{k\mu(\mathcal{A},\mathcal{B})n^2}{T_n\mathfrak{p}_2|\mathcal{A}||\mathcal{B}|}\Big)\Big).\tag{5.118}$$

By rearranging the left-hand side (5.118) and applying Corollary 8, we find that for all $n \geq m$ and (i) all $k \geq \exp(2-2\log(\mathfrak{m}_1/\mathfrak{p}_2)) > 0$,

$$\mathbb{P}[2e_0(\mathcal{A},\mathcal{B})\geq k\mu(\mathcal{A},\mathcal{B})]\leq\exp\Big(\tfrac{1}{2}k\mu(\mathcal{A},\mathcal{B})\Big(1-\log\frac{\mathfrak{m}_1k}{\mathfrak{p}_2}\Big)\Big)\overset{\text{(i)}}{\leq}\exp\Big(-\tfrac{1}{4}k\log k\mu(\mathcal{A},\mathcal{B})\Big).\tag{5.119}$$

We obtain the same bound for $\mathbb{P}[2e_1(\mathcal{A},\mathcal{B})\geq k\mu(\mathcal{A},\mathcal{B})]$ in (5.111) *mutatis mutandis.* Together with (5.111) this yields that for all $n \geq m$, and all $k \geq k_0$,

$$\mathbb{P}[e(\mathcal{A},\mathcal{B})\geq k\mu(\mathcal{A},\mathcal{B})]\leq 2\exp\Big(-\tfrac{1}{4}\mu(\mathcal{A},\mathcal{B})k\log k\Big).\tag{5.120}$$

This completes the proof. □

# 5.B Proofs of Section 5.3

## 5.B.1 Proof of Lemma 43

*Proof.* We prove Lemma 43(a) first. Let $x^{\mathrm{opt}}, y^{\mathrm{opt}} \in \mathbb{S}_1^{n-1}(0)$ be such that $\|A\| = |(x^{\mathrm{opt}})^{\mathrm{T}} A\, y^{\mathrm{opt}}|$. Choose $x_*, y_* \in \mathcal{N}_\epsilon$ such that $\|x^{\mathrm{opt}} - x_*\|_2 < \epsilon$ and $\|y^{\mathrm{opt}} - y_*\|_2 < \epsilon$. This is possible by construction of the $\epsilon$-net, and because $\mathbb{S}_1^{n-1}(0) \subset \mathbb{B}_1^n(0)$ thus implying that $x^{\mathrm{opt}}, y^{\mathrm{opt}} \in \mathbb{B}_1^n(0)$ also. Using the triangle inequality, we find that

$$\begin{aligned}\|A\| = |(x^{\mathrm{opt}})^{\mathrm{T}} A y^{\mathrm{opt}}| &\leq |(x^{\mathrm{opt}} - x_*)^{\mathrm{T}} A y^{\mathrm{opt}}| + |(x^{\mathrm{opt}})^{\mathrm{T}} A (y^{\mathrm{opt}} - y_*)| \\ &\quad + |x_*^{\mathrm{T}} A y_*| + |(x^{\mathrm{opt}} - x_*)^{\mathrm{T}} A (y^{\mathrm{opt}} - y_*)| \leq (2\epsilon + \epsilon^2)\|A\| + |x_*^{\mathrm{T}} A y_*|.\end{aligned} \tag{5.121}$$

Rearranging terms, it follows that

$$\|A\| \leq \frac{1}{1 - 2\epsilon - \epsilon^2} |x_*^{\mathrm{T}} A y_*| \leq \frac{1}{1 - 3\epsilon} \max_{x,y \in \mathcal{N}_\epsilon} |x^{\mathrm{T}} A y|. \tag{5.122}$$

This proves Lemma 43(a).

Finally, we prove Lemma 43(b). Observe that for any $a \in \mathbb{B}_0^{|\mathcal{A}|}(0)$, there exists a point $b \in \mathbb{B}_1^n(0)$ such that $b^{\mathcal{A}} = a$. Note furthermore that for any $b \in \mathbb{B}_1^n(0)$, there exists a point $c \in \mathcal{N}_\epsilon$ such that $\|b - c\|_2 \leq \epsilon$. Thus: for any $a \in \mathbb{B}_0^{|\mathcal{A}|}(0)$ there exists a point $c^{\mathcal{A}} \in \mathcal{N}_\epsilon^{\mathcal{A}}$ such that $\|a - c^{\mathcal{A}}\|_2^2 = \|b^{\mathcal{A}} - c^{\mathcal{A}}\|_2^2 \leq \|b - c\|_2^2 \leq \epsilon^2$. Observe finally that $\mathcal{N}_\epsilon^{\mathcal{A}} \subseteq \mathbb{B}_0^{|\mathcal{A}|}(0)$. This proves that $\mathcal{N}_\epsilon^{\mathcal{A}}$ is an $\epsilon$-net for $(\mathbb{B}_0^{|\mathcal{A}|}(0), \|\cdot\|_2)$. □

## 5.B.2 Proof of Proposition 21

Observe that $\mathcal{T}_\epsilon$ is finite. It is therefore sufficient to prove that there exists a constant $\mathfrak{h}_1 > 0$ independent of $n$ and $x, y \in \mathcal{T}_\epsilon$ such that for sufficiently large $n$

$$H_1(x,y) \leq \mathfrak{h}_1 \sqrt{\frac{T_n}{n}} \tag{5.123}$$

almost surely.

Consider any pair $x, y \in \mathcal{T}_\epsilon$. For $i, j \in \{1, 2, \ldots, \lceil \log(\sqrt{n}/\epsilon)/\log 2 \rceil\}$, define

$$\mathcal{A}_i(x) = \Bigl\{ v \in [n] : \frac{\epsilon}{\sqrt{n}} 2^{i-1} \leq |x_v| < \frac{\epsilon}{\sqrt{n}} 2^i \Bigr\}, \tag{5.124}$$

$$\mathcal{B}_j(y) = \Bigl\{ w \in [n] : \frac{\epsilon}{\sqrt{n}} 2^{j-1} \leq |y_w| < \frac{\epsilon}{\sqrt{n}} 2^j \Bigr\}. \tag{5.125}$$

Remark now firstly that by definition of the set of heavy pairs in (5.26): for all $(v,w) \in \mathcal{L}^{\mathrm{c}}(x,y)$, $|x_v y_w| > (1/n)\sqrt{T_n/n}$. Thus if any component of either $x$ or $y$ is zero, for example $x_{v^\star} = 0$ and/or $y_{w^\star}$ say, then for any $v, w \in [n]$, $(v^\star, w) \notin \mathcal{L}^{\mathrm{c}}(x,y)$ and/or $(v, w^\star) \notin \mathcal{L}^{\mathrm{c}}(x,y)$. Secondly, take note of the definition of $\mathcal{T}_\epsilon$ in (5.24): if $x, y$ are such that no component at all equals zero, then it must be that for all $(v,w) \in [n]^2$, $|x_v| \geq \epsilon/\sqrt{n}$ and $|y_w| \geq \epsilon/\sqrt{n}$. Consider these facts and now examine the definitions of $\mathcal{A}_i(x)$, $\mathcal{B}_j(y)$ in (5.124), (5.125): by construction for any $(v,w) \in \mathcal{L}^{\mathrm{c}}(x,y)$, there exist a unique index pair $(i^\star, j^\star)$ such that $(v,w) \in \mathcal{A}_{i^\star}(x) \times \mathcal{B}_{j^\star}(y)$. Furthermore, for any index pair $(i,j)$, if $(v,w) \in \mathcal{A}_i(x) \times \mathcal{B}_j(y)$, then $|x_v y_w| > (1/n)\sqrt{T_n/n}$ if $2^{i+j} \geq 4\sqrt{T_n/n}/\epsilon^2$. We therefore have the set equality

$$\mathcal{L}^{\mathrm{c}}(x,y) = \bigcup_{(i,j): 2^{i+j} > 4\sqrt{T_n/n}/\epsilon^2} \bigl( \mathcal{A}_i(x) \times \mathcal{B}_j(y) \bigr). \tag{5.126}$$

Next, we apply the triangle inequality:

$$H_1(x,y) = \Bigl| \sum_{(v,w)\in\mathcal{L}^{\mathrm{c}}} x_v(\hat{N}_\Gamma)_{vw} y_w \Bigr| \leq \sum_{(v,w)\in\mathcal{L}^{\mathrm{c}}} |\hat{N}_\Gamma|_{vw}|x_v y_w|. \tag{5.127}$$

Observe that

$$\begin{aligned} H_1(x,y) &\overset{(5.127)}{\leq} \sum_{(v,w)\in\mathcal{L}^{\mathrm{c}}} |\hat{N}_\Gamma|_{vw}|x_v y_w| \overset{(5.126)}{=} \sum_{(i,j):2^{i+j}>4\sqrt{T_n/n}/\epsilon^2} \sum_{(v,w)\in\mathcal{A}_i\times\mathcal{B}_j} |\hat{N}_\Gamma|_{vw}|x_v y_w| \\ &\overset{(5.124,\ 5.125)}{<} \sum_{(i,j):2^{i+j}>4\sqrt{T_n/n}/\epsilon^2} \sum_{(v,w)\in\mathcal{A}_i\times\mathcal{B}_j} |\hat{N}_\Gamma|_{vw}\epsilon 2^i \epsilon 2^j \frac{1}{n} \\ &\overset{(5.20)}{=} \sum_{(i,j):2^{i+j}>4\sqrt{T_n/n}/\epsilon^2} \epsilon 2^i \epsilon 2^j \frac{1}{n} e_\Gamma(\mathcal{A}_i,\mathcal{B}_j). \end{aligned} \tag{5.128}$$

Substitute $\mu_{ij} \triangleq |\mathcal{A}_i||\mathcal{B}_j|T/n^2$ (not to be confused by the actual mean of $e_\Gamma(\mathcal{A}_i,\mathcal{B}_j)$) into (5.128) and collect terms as follows:

$$H_1(x,y) \leq \sqrt{\frac{T_n}{n}}\epsilon^2 \sum_{(i,j):2^{i+j}>\frac{4\sqrt{T_n/n}}{\epsilon^2}} \underbrace{|\mathcal{A}_i|2^{2i}\frac{1}{n}}_{\alpha_i} \cdot \underbrace{|\mathcal{B}_j|2^{2j}\frac{1}{n}}_{\beta_j} \cdot \underbrace{\frac{e_\Gamma(\mathcal{A}_i,\mathcal{B}_j)}{\mu_{ij}2^{i+j}}\sqrt{\frac{T_n}{n}}}_{\sigma_{ij}}. \tag{5.129}$$

We will separate the sum in (5.129) in two parts. Define

$$\mathcal{C}_1 = \Bigl\{(i,j) : 2^{i+j} \geq \frac{4\sqrt{\frac{T_n}{n}}}{\epsilon^2}, (\mathcal{A}_i,\mathcal{B}_j) \text{ satisfies } (i) \text{ in Definition 32}\Bigr\}, \quad \text{and} \tag{5.130}$$

$$\mathcal{C}_2 = \Bigl\{(i,j) : 2^{i+j} \geq \frac{4\sqrt{T_n/n}}{\epsilon^2}, (\mathcal{A}_i,\mathcal{B}_j) \text{ satisfies } (ii) \text{ in Definition 32}\Bigr\}\backslash\mathcal{C}_1. \tag{5.131}$$

Note that $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$ by definition and moreover, $\mathcal{C}_1 \cup \mathcal{C}_2 = \{(i,j) : 2^{i+j} \geq 4\sqrt{T_n/n}/\epsilon^2\}$ since by assumption the discrepancy property holds. With the definitions in (5.129)–(5.131) it thus suffices to show that there exists a constant $\mathfrak{c} > 0$ independent of $n$ such that for sufficiently large $n$,

$$\sum_{(i,j)\in\mathcal{C}_1\cup\mathcal{C}_2} \alpha_i\beta_j\sigma_{ij} = \Bigl(\sum_{(i,j)\in\mathcal{C}_1} + \sum_{(i,j)\in\mathcal{C}_2}\Bigr)\alpha_i\beta_j\sigma_{ij} \leq \mathfrak{c}. \tag{5.132}$$

Note first that by (5.124) and (5.125),

$$\begin{aligned} &\sum_i \alpha_i = \sum_i |\mathcal{A}_i|\frac{4}{\epsilon^2}\Bigl(2^{i-1}\frac{\epsilon}{\sqrt{n}}\Bigr)^2 \overset{(5.124)}{\leq} \frac{4}{\epsilon^2}\sum_{v\in[n]}|x_v|^2 = \frac{4\|x\|_2^2}{\epsilon^2} \leq \frac{4}{\epsilon^2}, \\ &\text{and similarly} \quad \sum_i \beta_i \overset{(5.125)}{\leq} \frac{4\|y\|_2^2}{\epsilon^2}. \end{aligned} \tag{5.133}$$

Also define $e_{ij} \triangleq e_\Gamma(\mathcal{A}_i,\mathcal{B}_j)$ to declutter notation.

**Case** $(i,j) \in \mathcal{C}_1$ For $(i,j) \in \mathcal{C}_1$, Property (i) in Definition 32 is satisfied. This implies that

$$\sigma_{ij} = \frac{e_{ij}}{\mu_{ij} 2^{i+j}} \sqrt{\frac{T_n}{n}} \leq \frac{\eth_1}{2^{i+j}} \sqrt{\frac{T_n}{n}} \overset{(5.130)}{\leq} \frac{\eth_1 \epsilon^2}{4}. \tag{5.134}$$

Together with (5.133), (5.134) implies

$$\sum_{(i,j)\in\mathcal{C}_1} \alpha_i \beta_j \sigma_{ij} \overset{(5.134)}{\leq} \sum_{i,j} \alpha_i \beta_j \frac{\eth_1 \epsilon^2}{4} = \Big(\sum_i \alpha_i\Big)\Big(\sum_j \beta_j\Big) \frac{\eth_1 \epsilon^2}{4} \overset{(5.133)}{\leq} \frac{4\eth_1}{\epsilon^2}. \tag{5.135}$$

**Case** $(i,j) \in \mathcal{C}_2$ For $(i,j) \in \mathcal{C}_2$, bounding is more complicated. Presume that $|\mathcal{A}_i| \leq |\mathcal{B}_i|$ without loss of generality. Property (ii) in Definition 32 then reduces to

$$e_{ij} \log \frac{e_{ij}}{\mu_{ij}} \leq \eth_2 |\mathcal{B}_j| \log \frac{n}{|\mathcal{B}_j|}. \tag{5.136}$$

Substituting $\mu_{ij} = |\mathcal{A}_i||\mathcal{B}_j|T/n^2$ and $|\mathcal{B}_j| = \beta_j 2^{-2j} n$ in (5.136), we find that Property (ii) is equivalent to

$$\frac{e_{ij}|\mathcal{A}_i|T_n}{\mu_{ij} n^2} \log \frac{e_{ij}}{\mu_{ij}} \leq \eth_2 \log\Big(\frac{2^{2j}}{\beta_j}\Big). \tag{5.137}$$

Multiply the left- and right-hand sides by $2^{-(i+j)}$ to identify $\sigma_{ij} = e_{ij}\mu_{ij}^{-1} 2^{-(i+j)} \sqrt{T_n/n}$ and write:

$$\sigma_{ij} \frac{|\mathcal{A}_i|}{n} \sqrt{\frac{T_n}{n}} \log \frac{e_{ij}}{\mu_{ij}} \leq \eth_2 2^{-(i+j)} \log\Big(\frac{2^{2j}}{\beta_j}\Big). \tag{5.138}$$

Recall that $|\mathcal{A}_i| = \alpha_i 2^{-2i} n$. Therefore,

$$\alpha_i \sigma_{ij} \sqrt{\frac{T_n}{n}} \log \frac{e_{ij}}{\mu_{ij}} \leq \eth_2 \frac{2^i}{2^j} \Big(\log 2^{2j} - \log \beta_j\Big). \tag{5.139}$$

Knowing that (5.139) holds, let us go back to $\mathcal{C}_2$ and separate this set into disjoint subsets. Define

$$\begin{aligned}
\mathcal{D}_1 &= \big\{(i,j) \in \mathcal{C}_2 : \sigma_{ij} \leq 1\big\},\\
\mathcal{D}_2 &= \big\{(i,j) \in \mathcal{C}_2 \backslash \mathcal{D}_1 : 2^i > 2^j \sqrt{T_n/n}\big\},\\
\mathcal{D}_3 &= \big\{(i,j) \in \mathcal{C}_2 \backslash (\mathcal{D}_1 \cup \mathcal{D}_2) : \log(e_{ij}/\mu_{ij}) > \tfrac{1}{4}(\log 2^{2j} - \log \beta_j)\big\},\\
\mathcal{D}_4 &= \big\{(i,j) \in \mathcal{C}_2 \backslash (\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3) : \log 2^{2j} \geq -\log \beta_j\big\},\\
\mathcal{D}_5 &= \mathcal{C}_2 \backslash (\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3 \cup \mathcal{D}_4).
\end{aligned} \tag{5.140}$$

Notice from (5.140) that $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ for $i \neq j$ and moreover, $\mathcal{D}_1 \cup \cdots \cup \mathcal{D}_5 = \mathcal{C}_2$. We go subcase by subcase and check that in each subcase we obtain the right bound of order $O(\sqrt{T_n/n})$.

**Subcase** $(i,j) \in \mathcal{D}_1$ According to the subcase, (i) $\sigma_{ij} \leq 1$. By (ii) expanding the summation range, it follows that

$$\sum_{(i,j)\in\mathcal{D}_1} \alpha_i \beta_j \sigma_{ij} \overset{\text{(i)}}{\leq} \sum_{(i,j)\in\mathcal{D}_1} \alpha_i \beta_j \overset{\text{(ii)}}{\leq} \sum_{i,j} \alpha_i \beta_j = \Big(\sum_i \alpha_i\Big)\Big(\sum_j \beta_j\Big) \overset{(5.133)}{\leq} \frac{2^4}{\epsilon^4}. \tag{5.141}$$

Notice that this did not yet require the calculation in (5.139). We will use it from subcase $\mathcal{D}_3$ onward.

**Subcase** $(i,j) \in \mathcal{D}_2$ We have

$$e_{ij} = e_\Gamma(\mathcal{A}_i, \mathcal{B}_j) \overset{(5.20)}{=} \sum_{x \in \mathcal{A}_i} \sum_{y \in \mathcal{B}_j} (\hat{N}_\Gamma)_{x,y}. \tag{5.142}$$

Eq. (5.19) holds by assumption, i.e., we have a bounded degree. That is, for some $\mathfrak{b}_2 > 0$,

$$e_{ij} \le |\mathcal{A}_i| \mathfrak{b}_2 \frac{T_n}{n}. \tag{5.143}$$

Dividing by $\mu_{ij}$ and using its definition before (5.129), this implies

$$\frac{e_{ij}}{\mu_{ij}} \le \mathfrak{b}_2 \frac{n}{|\mathcal{B}_j|}. \tag{5.144}$$

Fix $i \in [n]$. Recall that (i) the subcase implies that $2^i > 2^j \sqrt{T_n/n}$. Therefore,

$$\begin{aligned} \sum_j \beta_j \sigma_{ij} \mathbb{1}[(i,j) \in \mathcal{D}_2] &\overset{(5.129)}{=} \sum_j |\mathcal{B}_j| 2^{j-i} \frac{1}{n} \frac{e_{ij}}{\mu_{ij}} \sqrt{\frac{T_n}{n}} \mathbb{1}[(i,j) \in \mathcal{D}_2] \\ &\overset{(5.144)}{\le} \sum_j 2^{j-i} \sqrt{\frac{T_n}{n}} \mathfrak{b}_2 \mathbb{1}[(i,j) \in \mathcal{D}_2] \\ &\overset{\text{(i)}}{\le} \sum_j 2^{j-i} \sqrt{\frac{T_n}{n}} \mathfrak{b}_2 \mathbb{1}\Big[2^{j-i} < 1/\sqrt{\frac{T_n}{n}}\Big] \overset{\text{(ii)}}{\le} \sqrt{\frac{T_n}{n}} \frac{2\mathfrak{b}_2}{\sqrt{T_n/n}} \le 2\mathfrak{b}_2. \end{aligned} \tag{5.145}$$

Here, we have (ii) used that for $r > 1$, $a > 0$,

$$\sum_{m=-\infty}^{\infty} r^m \mathbb{1}[r^m < a] = \cdots + r^{m^\star - 2} + r^{m^\star - 1} + r^{m^\star} = r^{m^\star}\big(1 + r^{-1} + r^{-2} + \cdots\big) \le \frac{a}{1 - 1/r}, \tag{5.146}$$

where $m^\star = \max\{m \in \mathbb{Z} : r^m < a\}$.

Therefore, after also expanding the summation range, for certain constants $\mathfrak{c}_1, \mathfrak{c}_2 > 0$ independent of $n$ and for sufficiently large $n$,

$$\sum_{(i,j) \in \mathcal{D}_2} \alpha_i \beta_j \sigma_{ij} = \sum_i \alpha_i \sum_j \beta_j \sigma_{ij} 1[(i,j) \in \mathcal{D}_2] \overset{(5.145)}{\le} \mathfrak{c}_1 \sum_i \alpha_i \overset{(5.133)}{\le} \mathfrak{c}_2. \tag{5.147}$$

**Subcase** $(i,j) \in \mathcal{D}_3$ The subcase implies (i) that $\sigma_{ij} > 1$, (ii) that $2^i \le 2^j \sqrt{T_n/n}$, and (iii) that $\log(e_{ij}/\mu_{ij}) > \frac{1}{4}(\log 2^{2j} - \log \beta_j)$. Bounding (5.139) directly with these facts, we find that

$$\alpha_i \sigma_{ij} \overset{(5.139)}{\le} \mathfrak{d}_2 \frac{2^i}{2^j} \sqrt{\frac{n}{T_n}} \frac{\log 2^{2j} - \log \beta_j}{\log(e_{ij}/\mu_{ij})} \overset{\text{(iii)}}{<} 4\mathfrak{d}_2 \frac{2^i}{2^j} \sqrt{\frac{n}{T_n}}. \tag{5.148}$$

Fix $j$. Then,

$$\begin{aligned} \sum_i \alpha_i \sigma_{ij} \mathbb{1}[(i,j) \in \mathcal{D}_3] &\overset{(5.148)}{\le} \sum_i 4\mathfrak{d}_2 \frac{2^i}{2^j} \sqrt{\frac{n}{T_n}} \mathbb{1}[(i,j) \in \mathcal{D}_3] \\ &\overset{\text{(ii)}}{\le} \sum_i 4\mathfrak{d}_2 2^{i-j} \sqrt{\frac{n}{T_n}} \mathbb{1}\Big[2^{i-j} \le \sqrt{\frac{T_n}{n}}\Big] \overset{(5.146)}{\le} 8\mathfrak{d}_2. \end{aligned} \tag{5.149}$$

It follows immediately that for some constant $\mathfrak{c}_3 > 0$ independent of $n$ and for sufficiently large $n$,

$$\sum_{(i,j)\in\mathcal{D}_3} \alpha_i\beta_j\sigma_{ij} = \sum_j \beta_j \sum_i \alpha_i\sigma_{ij}\mathbb{1}[(i,j)\in\mathcal{D}_3] \overset{(5.149)}{\leq} 8\mathfrak{d}_2 \sum_j \beta_j \overset{(5.133)}{\leq} \mathfrak{c}_3. \tag{5.150}$$

**Subcase** $(i,j) \in \mathcal{D}_4$ Recall that the subcase implies (i) that $\sigma_{ij} > 1$, (ii) that $2^i \leq 2^j\sqrt{T_n/n}$, (iii) that $\log(e_{ij}/\mu_{ij}) \leq \frac{1}{4}(\log 2^{2j} - \log\beta_j)$, and (iv) that $\log 2^{2j} \geq -\log\beta_j$. Therefore, in this subcase,

$$\log(e_{ij}/\mu_{ij}) \overset{\text{(iii)}}{\leq} \tfrac{1}{4}(\log 2^{2j} - \log\beta_j) \overset{\text{(iv)}}{\leq} \tfrac{1}{2}\log 2^{2j} = \log 2^j. \tag{5.151}$$

Furthermore,

$$0 \overset{\text{(i)}}{<} \log\sigma_{ij} \overset{(5.129)}{=} \log(e_{ij}/\mu_{ij}) - \log 2^i - \log 2^j + \log\sqrt{\frac{T_n}{n}} \overset{(5.151)}{\leq} -\log 2^i + \log\sqrt{\frac{T_n}{n}}. \tag{5.152}$$

Consequently, because (5.152) is strictly positive, the subcase implies that

$$2^i < \sqrt{\frac{T_n}{n}}. \tag{5.153}$$

If $(i,j)\in\mathcal{C}_2$ and thus $(i,j)\notin\mathcal{C}_1$ from the definition in (5.130), we have that $\log(e_{ij}/\mu_{ij}) > \mathfrak{d}_2$ by the discrepancy property. Thus

$$\begin{aligned}\alpha_i\sigma_{ij}\mathfrak{d}_2 < \alpha_i\sigma_{ij}\log\frac{e_{ij}}{\mu_{ij}} &\overset{(5.139)}{\leq} \mathfrak{c}_4\frac{2^i}{2^j}\sqrt{\frac{n}{T_n}}\left(\log 2^{2j} - \log\beta_j\right) \\ &\overset{\text{(iv)}}{\leq} 4\mathfrak{c}_4 2^i\sqrt{\frac{n}{T_n}}\cdot 2^{-j}\log 2^j \overset{\text{(v)}}{\leq} 4\mathfrak{c}_4 2^i\sqrt{\frac{n}{T_n}}.\end{aligned} \tag{5.154}$$

Here, (v) followed because $z^{-1}\log z \leq 1$ for $z \geq 0$. It follows after also expanding the summation range that

$$\begin{aligned}\sum_{(i,j)\in\mathcal{D}_4} \alpha_i\beta_j\sigma_{ij} = \sum_j\beta_j\sum_i\alpha_i\sigma_{ij}\mathbb{1}[(i,j)\in\mathcal{D}_4] &\overset{(5.154)}{\leq} \sqrt{\frac{n}{T_n}}\sum_j\beta_j\sum_i\mathfrak{c}_5 2^i\mathbb{1}[(i,j)\in\mathcal{D}_4] \\ &\overset{(5.153)}{\leq} \sqrt{\frac{n}{T_n}}\sum_j\beta_j\sum_i\mathfrak{c}_5 2^i\mathbb{1}\left[2^i < \sqrt{\frac{T_n}{n}}\right] \overset{(5.146)}{\leq} \mathfrak{c}_6\sum_j\beta_j \leq \mathfrak{c}_7.\end{aligned} \tag{5.155}$$

**Subcase** $(i,j) \in \mathcal{D}_5$ Recall that this subcase implies (i) that $\sigma_{ij} > 1$, (ii) that $2^i \leq 2^j\sqrt{T_n/n}$, (iii) that $\log(e_{ij}/\mu_{ij}) \leq \frac{1}{4}(\log 2^{2j} - \log\beta_j)$, and that (iv) $\log 2^{2j} < -\log\beta_j$. Similar to the previous subcase,

$$\log\frac{e_{ij}}{\mu_{ij}} \overset{\text{(iii)}}{\leq} \tfrac{1}{4}(\log 2^{2j} - \log\beta_j) \overset{\text{(iv)}}{\leq} \tfrac{1}{2}(-\log\beta_j) \overset{\text{(v)}}{\leq} -\log\beta_j, \tag{5.156}$$

where (v) since $j \geq 1$ we have $-\log(\beta_j) > \log 2^{2j} > 0$. This implies that

$$\frac{e_{ij}}{\mu_{ij}} \leq \frac{1}{\beta_j} \tag{5.157}$$

or equivalently

$$\beta_j \sigma_{ij} \overset{(5.129)}{\leq} \beta_j \frac{e_{ij}}{\mu_{ij} 2^{i+j}} \sqrt{\frac{T_n}{n}} \leq \frac{1}{2^{i+j}} \sqrt{\frac{T_n}{n}}. \tag{5.158}$$

Recall (5.131): for all $(i,j) \in \mathcal{C}_2$, $2^{i+j} \geq 4\sqrt{T_n/n}/\epsilon^2$. Therefore

$$\begin{aligned} \sum_{(i,j)\in\mathcal{D}_5} \alpha_i \beta_j \sigma_{ij} &= \sum_i \alpha_i \sum_j \beta_j \sigma_{ij} \mathbb{1}[(i,j) \in \mathcal{D}_5] \overset{(5.158)}{\leq} \sum_i \alpha_i \sum_j \frac{1}{2^{i+j}} \sqrt{\frac{T_n}{n}} \mathbb{1}[(i,j) \in \mathcal{D}_5] \\ &\overset{(5.131)}{\leq} \sum_i \alpha_i \sum_j \frac{1}{2^{i+j}} \sqrt{\frac{T_n}{n}} \mathbf{1}\Big[2^{i+j} \geq \frac{4}{\epsilon^2} \sqrt{\frac{T_n}{n}}\Big] \overset{\text{(vi)}}{\leq} \sum_i \alpha_i \frac{\epsilon^2}{4} \leq \mathfrak{c}_8. \end{aligned} \tag{5.159}$$

Here, we have (vi) used that for $r > 1$, $a > 0$,

$$\sum_{m=0}^{\infty} \frac{1}{r^m} \mathbb{1}[r^m \geq a] = \frac{1}{r^{m^\star}} + \frac{1}{r^{m^\star+1}} + \frac{1}{r^{m^\star+2}} + \cdots = \frac{1}{r^{m^\star}} \Big(1 + \frac{1}{r} + \frac{1}{r^2} + \cdots\Big) \leq \frac{1/a}{1-1/r}, \tag{5.160}$$

where $m^* = \min\{m \in \mathbb{Z} : r^m \geq a\}$. This completes the proof.

# Chapter 6

# Experimental evaluation of the BMC model in sequential data

Based on [9]:  
*"Detection and evaluation of clusters within sequential data"*  
by A. Van Werde, A. Senen–Cerda, G. Kosmella, and J. Sanders

## 6.1 Introduction

In Chapter 5, we derived order-sharp spectral bounds for the Block Markov Chain (BMC) model that characterize its empirical spectral error. Such bounds, while asymptotic in nature, are required to guarantee consistency of a spectral clustering algorithm [93, 27]. We have verified in Corollary 7 of Chapter 5 that the count matrix $\hat{N}$ has $K$ singular values of order $T_n/n$, while all other singular values are of order $\sqrt{T_n/n}$. Hence, for a scaling of the length of the path $T_n = \omega(n)$, we would be able to see the difference, albeit asymptotically, in the form of a spectral gap in the count matrix.

Real-world data, however, is not expected to strictly satisfy the theoretical properties assumed in the model. In practice, networks are not homogeneous or dense. Examples can be found in social networks, for example, where there are large hubs that are commonly centered around a few people with a high number of connections. Even with an approximate block-uniform community structure present, a single vertex with a high degree in such a dense network may still induce a large spectral error. Despite these issues, meaningful clusters can still be discovered. We show a concrete example of clusters found within animal movement data from [9].

A group of bisons have Global Positioning System (GPS) tags that record their position across time as they roam in a landscape. After a sufficiently long time, we assign the GPS coordinates to states on a grid and the movement trajectory of the animals is depicted in Figure 6.1.1. The raw data is not very meaningful to a human if we only look at transitions. However, after using the clustering algorithm for BMCs, the resulting clusters fit very

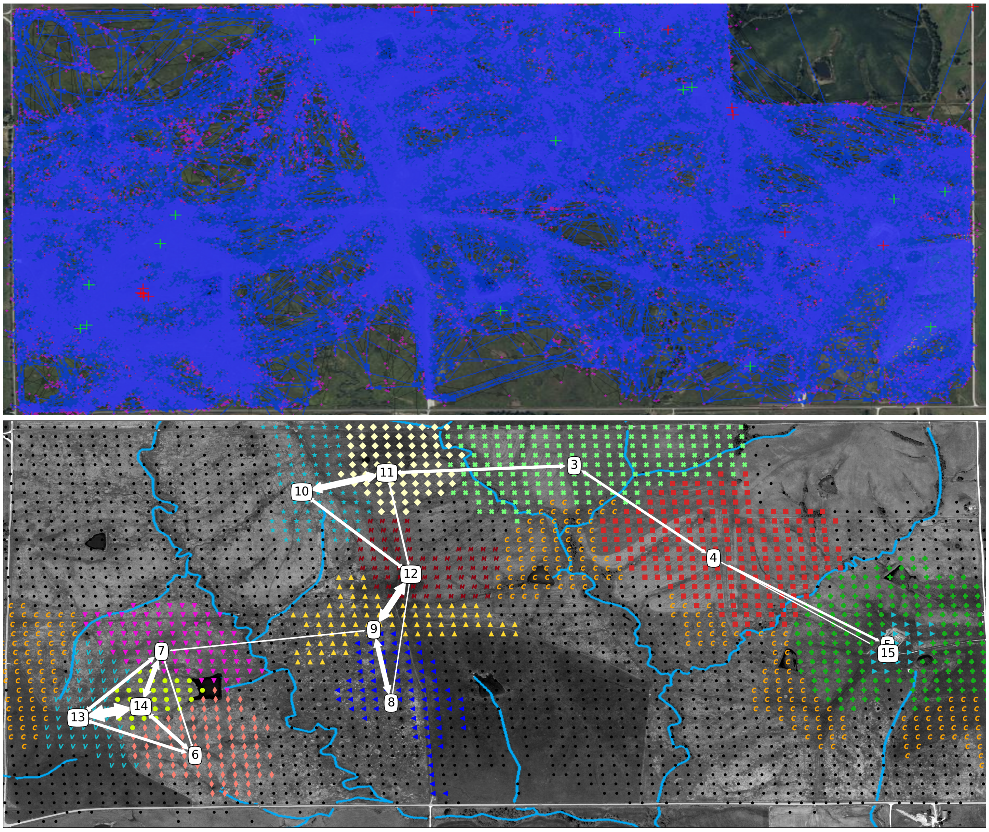


*Figure 6.1.1: (Above) A screenshot from movebank.org [71] displaying the raw GPS data of bisons. (Below) Plot of the recovered clusters from [9]. The clusters are obtained by using the clustering algorithm for BMCs superimposed with geographical features. Note that rivers separate the found clusters well even though no distance based information is used in the clustering algorithm. Other concrete examples can be also found in [9].*

well with geographical features like rivers and fields, and the inferred cluster transition probabilities tell us about the dynamics of the movement of the herd.

In Figure 6.1.1 the clustering algorithm for BMCs seems to yield clusters that are tied to an underlying structure of the data, namely the geographical features that drive the sequential process. In other examples, however, it may not be so clear if the clustering algorithm for BMCs can still yield useful clusters and recover part of the low-dimensional structure of the data. Thus, a practical study is warranted.

In this chapter we will therefore take a more practical approach to clustering with BMCs and use real-world sequential data to test the clustering algorithm. Firstly, we compare the spectral gap in data to the one predicted in Chapter 5 and the robustness of the clustering algorithm for BMCs to perturbations is examined. Secondly, we will test if the model is suitable for describing the clusters and the transition dynamics of several datasets of sequential data compared to other models with less complexity. Specifically, the following questions will be investigated:

- Can we see a spectral gap of $\hat{N}$ in real-world sequential data?
- Is the clustering algorithm robust to violations of the BMC assumption?
- How can it be decided whether a simpler model than a BMC would suffice for explaining the dynamics of the transitions, or that a richer model is required?

This chapter is based on [9], where a broad study on the performance of the clustering algorithm for BMCs in real-world sequential data is conducted with different tools. The clustering algorithm is applied to recover clusters in selected datasets with different characteristics coming from genetics, finance, texts, and animal movement. A variety of tools are used to evaluate its performance, including spectral analysis, benchmarking as well as statistical tools for model selection. We will restrict ourselves in this chapter, however, only to a part of the evaluation of the clusters and the clustering algorithm. We assume, therefore, that the inferred cluster assignment $\hat{\nu}_n$ and number of clusters $K$ for these datasets is provided. For the full practical study we refer to [9]. There, the preprocessing of the datasets, the shortcomings and process for obtaining these clusters as well as the choices for $K$ are also discussed.

In this chapter, we will denote the length of a sample path by $\ell$ instead of $T_n$. The latter term assumes an asymptotic relationship between $n$ and the length of the path, which is not the case in real data.

*Datasets.* The first dataset comes from sequences of codons of human Deoxyribonucleic Acid (DNA) of a gene obtained from [16]. The dataset has been preprocessed and the state space constitutes all possible 3-letter codon triplets. This yields a state space of $n = 64$ states. The length of the dataset is around $\ell \simeq 1.64 \cdot 10^5$. The second dataset consists of the sequences of the daily best performing stocks from the Standard and Poor's 500 (S&P500) index between the years 2001 and 2021 obtained from [11]. The state space consists of $n = 300$ different stocks and the length is the number of suitable operation days, namely $\ell \simeq 4900$. Our third dataset consists of the GPS coordinates of bisons in a landscape, which we have seen in Figure 6.1.1 and have been obtained from [71].[1] In this case, the state space has $n = 3155$ states and the length of the path is approximately $\ell \simeq 1.93 \cdot 10^5$. All datasets have been preprocessed and cleaned in order to have a single sequence of states to which the clustering algorithm can be applied. The frequency matrix $\hat{N}$ encodes the information of the clusters in all these datasets. For the case of the DNA and S&P500 data, we represent the inferred clusters in Figure 6.1.2.

## Summary of results

We conduct several experiments with the datasets. In Section 6.2 we first analyze the spectral gap for the DNA and S&P500 datasets and qualitatively compare the gap with the expected orders for BMCs obtained in Chapter 5. The comparison for both datasets

[1] We thank the Nature Conservancy and Dr. Stephen Blake of Saint Louis University for permission to use the bison movement data. The Max Planck Institute for Animal Behavior and the National Geographic Society Committee for Research and Exploration (Grant: 9385-13 ) funded the bison GPS collars.

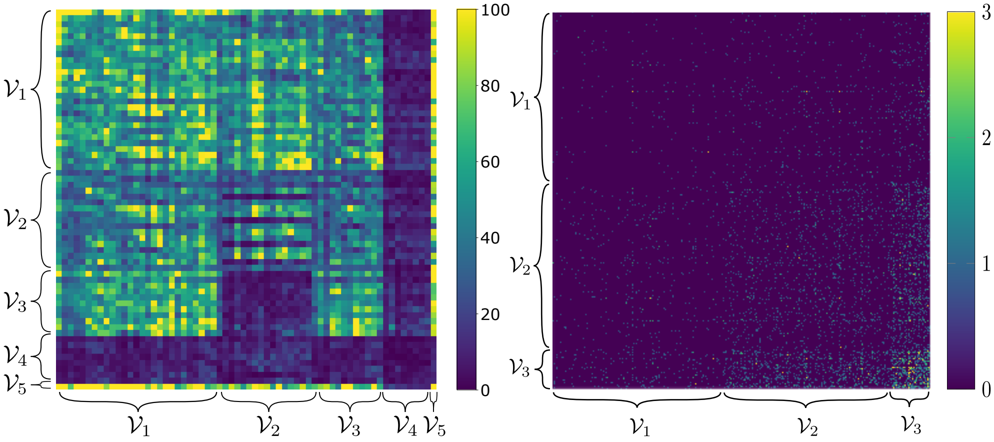


*Figure 6.1.2: (Left) The frequency matrix $\hat{\mathrm{N}}$ of the DNA dataset when the codons are sorted by the five detected clusters. The maximum value has been capped for better visualization. (Right) The frequency matrix $\hat{\mathrm{N}}$ of the Stock Market dataset when the companies are sorted by the three clusters detected. The maximum value has been capped for better visualization. Note that in this case, the matrix is very sparse.*

implies that the spectral norm of the counting matrix appears to be too sensitive to violations of the BMC model. We justify this conclusion and motivate the use of other spectral approaches to quantify the agreement between the BMC model and data.

The previous result on the spectral gap raises concerns about the robustness of the clustering algorithm when data does not follow a BMC model. In Section 6.3 the robustness of the spectral clustering algorithm is evaluated by using generative models of perturbed BMCs. In particular, models where the block transition matrix $P_{\mathrm{BMC}}$ of a BMC is perturbed by a Markov kernel $\Delta$ are considered. As a measure of robustness the number of misclassified states, and and the estimation error of the parameters compared to those of the generative model are considered. The dependence of these two terms on the perturbation strength is the target measure for different types of perturbations. The results from [9] indicate that the BMC model is robust to small to moderate perturbations and can be used to approximate perturbed BMCs for sample paths of reasonable length $\ell$.

Finally, in Section 6.4 we will perform model selection with the BMC model. We will consider if the model is suitable for explaining the transition dynamics of the data or if less or more complex models are better suited. Our focus will be on the Markovian assumption of the model. In particular, we will consider different models where dependence can be further in the past than the immediate previous state or where no dependence exists. That is, we will consider models of the data that follow an $r$th-order Markov chain with $r \in \{0, 1, 2, 3, 4\}$. The main difficulty here is that model selection with a full state space with $n$ states becomes unfeasible due to the large number of free parameters and the comparably short sample path length $\ell$. To avoid this issue, we use the sequence of clusters induced by the cluster assignment $\hat{\nu}_n$ to exploit the dimensionality reduction and conduct model selection. In the procedure, we use information criteria to select the best order that would explain the DNA, GPS and S&P500 datasets. The criteria point to nonzero order Markovian dependence for the DNA and GPS datasets but for the S&P500 dataset, there is not enough evidence to select an order with enough certainty. We support these conclusion with additional simulations and with a robustness analysis of the information criteria.

Analysis of real-world sequential data is challenging and even with a well-grounded model like the BMC model at hand, the shortcomings of the model assumptions quickly become apparent in the data analysis. Among the limitations, we can mention that data can be highly inhomogeneous, clusters can consist of dissimilar states only loosely associated, and data can be just too sparse to analyze. However, by carefully considering these limitations, the BMC model can still be used to obtain insight into the cluster transition dynamics of real sequential data, as it seems to be the case in e.g., Figure 6.1.1.

## 6.2 Spectral norm from detected clusters

From the clusters obtained in [9], we can study the spectrum and spectral gap $\|\hat{N} - N\|$ for the different datasets. Due to the large state space of the GPS dataset, we will restrict to the DNA and S&P500 datasets instead. For both datasets, the $\hat{N}$ matrices are depicted in Figure 6.1.2 with labeled clusters.

For a cluster assignment $\nu_n : [n] \to [K]$ and a frequency matrix $\hat{N}$, assuming that $\hat{N}$ follows a BMC model, we can obtain an unbiased estimate of the cluster transition matrix $q \in [0,1]^{K\times K}$ as well as the cluster invariant distribution $\pi \in [0,1]^K$. Since the distribution of the data is unknown, the underlying model will be assumed to be a BMC with cluster assignment $\nu_n : [n] \to [K]$ and cluster transition probabilities $q$ recovered from the dataset. In this case, we denote by $N$ the expected value of $\hat{N}$ assuming that the trajectory is sampled from a BMC with those inferred parameters and with the same length $\ell$ as the dataset. An example of a sampling of this model using these assumptions is shown in Chapter 1 with Figure 1.8. With $N$ at hand we can compute the spectral norm for each dataset. We denote the $i$th singular value of $L$ by $\sigma_i(L)$. The spectral norm of $\hat{N}$ will thus be $\sigma_1(\hat{N})$.

Figure 6.2.1 depicts the spectrum for the DNA and S&P500 datasets. The first clear observation is that the order of the spectral gap $\sigma_1(\hat{N} - N)$ is comparable to that of $\sigma_1(N)$ and $\sigma_1(\hat{N})$ for both datasets. From the depiction of the count matrix $\hat{N}$ in Figure 6.1.2, there seem to be large inhomogeneities in the transition count for some states compared to those of a BMC model. These inhomogeneities, as a byproduct of the underlying unknown distribution of the data, yield a large difference in terms of spectral norm between the estimation of $N$ assuming a BMC model and the sample $\hat{N}$. Specifically, the order of the bulk of the singular values of $\hat{N}$, which should have order $O(\sqrt{\ell/n})$ asymptotically, is comparable to that of the nonzero singular values of the block matrix $N$, which should have a larger order $\Theta(\ell/n)$ asymptotically. While in the S&P500 data the sparsity may explain the discrepancy of the orders of the singular values, in the DNA dataset it does not.

We note that the setup of the experiment assumes that the number of states $n$ is large enough to already see discrepancies between a BMC model and the data-generating process. To fully support these conclusions, statistical tests with guarantees for finite number of states $n$ and length of sample path $\ell$ would still be required.

The previous observations show the shortcomings of using the spectral norm and spectral error as a measure of comparison between a BMC model and real data. In particular, the spectral error can capture features of the data which are beyond the BMC model and so the spectral gap does not appear to be very robust when data does not exactly follow a BMC. This occurs similarly when there is a large sampling noise with $n$ and $\ell$ small. For example, if one state is visited much more often than all others, say a positive fraction of the total length of the path, then this state will contribute to the spectral error in a commensurable manner. The fact that the spectral norm is so sensitive to model violations could then be used for a statistical test to determine if data is generated from a BMC model with a certain block structure. A similar result in this direction has already been obtained for the Stochastic Block Model (SBM) [81].

A different approach to analyze the cluster structure of the data as shown in [9] is to use the bulk of the distribution that contains the spectral noise, as it avoids the large signal part of the spectrum. In Figure 6.2.1, for example, we can already see that the location of the bulk of the spectrum is similar for $\hat{N}$ and $\hat{N} - N$. Functions that characterize this similarity could thus become useful. A theoretical study of the bulk of the spectrum for BMC has been conducted in [2] and it is one of the tools used in [9] to study the disagreement between data and the BMC model.

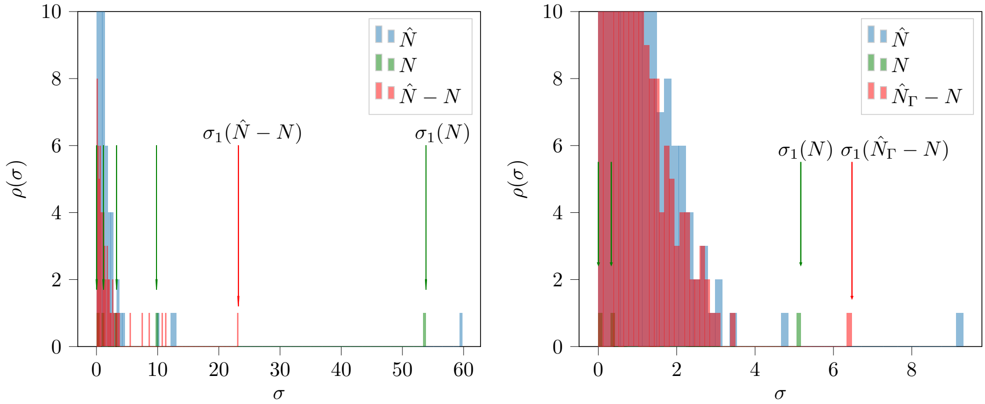


*Figure 6.2.1: Normalized spectral density of $\hat{N}, N$ and either $\hat{N} - N$ or $\hat{N}_\Gamma - N$ in blue, green and red for the DNA and Stock Market datasets respectively. The green arrows mark the location of the spectrum of $N$ in decreasing order and in red the spectral norm of $\hat{N}_\Gamma - N$.* (Left) *DNA dataset normalized spectral density. We see that while the spectral norm of $\hat{N}$ and $N$ are similar, the spectral error is still large. This suggests that there are inhomogeneous structures in the data that the BMC model cannot fully capture.* (Right) *Stock Market dataset normalized spectral density. In this case, the spectral norm of $\hat{N}_\Gamma - N$ is even larger than that of $N$, likely due to sparsity and in-cluster inhomogeneity of $\hat{N}$.*

## 6.3 Robustness of the clustering procedure to model violations

In Section 6.2 we have seen that the spectral gap does not appear to be robust to model violations. This was expected, however, and some other measure for comparison is thus needed. A natural choice in community detection is the number of misclassified states, that is, the number of states that have been incorrectly assigned to the different clusters. In real data, however, we do not have access to this information. In order to still be able to study the number of misclassified states, the robustness of the clustering procedure to violations of the model assumption can be examined with synthetically generated data. In this section, which follows [9], the performance of the clustering procedure is investigated when the data-generating process is not actually a BMC, but it is known exactly what it is.

Two main measures of performance using synthetic data are studied. Firstly, in Section 6.3.1, the number of misclassified states is considered. Secondly, in Section 6.3.2, the approximation error is examined in a parameter estimation problem where the objective is to estimate the true transition matrix $P$ of a Markovian data-generating process which need not be a BMC.

The first measure of performance requires that the notion of misclassification is sensible even though the data-generating process is not a BMC. To this end, models where communities are well-defined can be especially useful for comparison. In this section, a *perturbed BMC* model is considered. Let $\{B_t\}_{t\geq 0}$ denote a sequence of independent, identically distributed Bernoulli random variables, each taking the value 1 with probability $\varepsilon$ and 0 with probability $1-\varepsilon$. Let $\Delta$ be the transition matrix of a generic 1st-order Markov Chain (MC) on $[n]$ and $P_{\text{BMC}}$ be the transition matrix of a generic BMC. Then, the perturbed BMC, denoted by $\{X_t^\varepsilon\}_{t\geq 0}$, has conditional transition probabilities given by

$$\mathbb{P}[X_{t+1}^\varepsilon = j \mid X_t^\varepsilon = i, B_t = b] = \begin{cases} P_{\text{BMC},ij} & \text{if } b = 0, \\ \Delta_{ij} & \text{otherwise.} \end{cases} \tag{6.1}$$

In other words, a sequence $X_0^\varepsilon \to \cdots \to X_\ell^\varepsilon$ from the perturbed BMC is generated by randomly selecting either the transition matrix $P_{\text{BMC}}$ of a BMC, or the transition matrix $\Delta$ of some other 1st-order MC, for each transition. A perturbed BMC has then the distribution of a MC with transition matrix

$$P_{\text{Perturbed}} := (1-\varepsilon)P_{\text{BMC}} + \varepsilon\Delta, \tag{6.2}$$

where parameter $\varepsilon \in [0,1]$ can be understood on average, as the proportion of transitions that are affected by the non-BMC part $\Delta$.

For the tests, ground-truth communities correspond to those of the BMC-part of the perturbed model in (6.2). The definition of a perturbed BMC requires one to also specify the nature of the perturbation kernel $\Delta$. The following kernels are used for this purpose to model different types of model violations:

(i) *Uniform stochastic*: The matrix $\Delta$ is sampled uniformly at random in the set of stochastic matrices. This is accomplished by sampling each row independently from a Dirichlet$(1/n,\dots,1/n)$ distribution.

(ii) *Degree* 0: First, $\pi_i$ for $i \in [n]$ is constructed as follows: independent exponential random variables $e_1, \ldots, e_n \sim \text{Exponential}(1)$ are sampled and $\pi_i = e_i / (\sum_{j=1}^{n} e_j) > 0$ is defined. Then, $\Delta_{ij} = \pi_j$ is defined for all $i, j \in [n]$.

(iii) *Heavy-tailed*: Let $X$ be a random matrix whose entries $X_{ij}$ are i.i.d. positive random variables with a heavy-tailed distribution. The kernel $\Delta$ is then found by normalizing the rows in order to obtain a stochastic matrix $\Delta := \operatorname{diag}\big((\sum_j X_{ij})^{-1}\big)_{i=1}^{n} X$. The heavy-tailed entries $X_{ij}$ are sampled from a Zipf distribution with exponent $s = 3/2$.

(iv) *Sparse*: Consider constants $d > 0$ and $c > 0$, and construct a random matrix $X = A + cJ$ where $A$ is the adjacency matrix from a directed Erdös–Rényi random graph with average outgoing degree $d$ and $J$ is a constant matrix $J_{ij} = 1/n$. The kernel $\Delta$ is then found by rescaling the rows in order to obtain a stochastic matrix $\Delta = \operatorname{diag}\big((\sum_j X_{ij})^{-1}\big)_{i=1}^{n} X$. In this case, $d = 5$ and $c = 0.1$ are taken.

In the subsequent experiments, let $n = 2m$ be an even integer. The BMC which is perturbed is chosen to have two equally-sized clusters ($K = 2$) and a cluster transition matrix given by

$$q = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}.$$

### 6.3.1 Misclassification ratio for perturbed BMCs

This section concerns the number of misclassified states when a perturbed BMC model is clustered. Recall that the BMC model is chosen to have two equally-sized clusters which means that the cluster assignment map may be picked to be given by $\nu_n(i) = 1 + \mathbb{1}[i > n/2]$. Let $\hat{\nu}_n : [n] \to \{1, 2\}$ be an estimated cluster assignment which is output by the clustering procedure. Then, the missclassification ratio $\mathcal{E}$ is defined as

$$\mathcal{E} := \frac{1}{n} \min_{\rho \in S_2} \#\{v \in [n] : \nu_n(v) \neq (\rho \circ \hat{\nu}_n)(v)\}. \tag{6.3}$$

Here $S_2$ denotes the set of permutations of $\{1, 2\}$.

Recall from (6.1) that the parameter $\varepsilon$ of the perturbed BMC measures the fraction of transitions which are affected by the perturbation. In other words, $\varepsilon$ measures the strength of the perturbation. The estimated average misclassification ratio $\hat{\mathcal{E}}$ for a numerical experiment is displayed as a function of the perturbation level $\varepsilon$ in Figure 6.3.1. Up to $\varepsilon \approx 0.1$ the algorithm succeeds in recovering the exact cluster assignment for all four models. The exact number will naturally depend on the parameters of the BMC which was perturbed and will consequently be different in different contexts. From this experiment it is concluded that the algorithm appears to be robust with regard to small to moderate model violations.

### 6.3.2 Bias–variance trade-off for parameter estimation in a perturbed BMC

It may occur in some cases that one is not interested in the clusters themselves but rather views them as a means to an end. Consider the scenario where one wants to

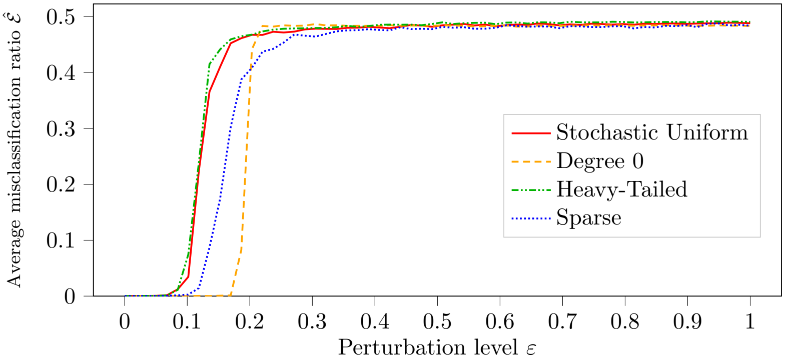


*Figure 6.3.1: The expected proportion of states which are misclassified in terms of the perturbation level $\varepsilon$ for four different perturbation models $\Delta$ and a sample path of length $\ell_n = \lfloor 30n\log(n)\rfloor$ with a state space of and of size $n = 500$. Note that the values depend on the precise parameters of the BMC which was perturbed.*

estimate the transition kernel of a MC which need not be a BMC. It may be suspected, however, that there could be some underlying clusters in the data but also that there could be parts of the dynamics which do not respect the cluster structure. In such a case a perturbed BMC would be a suitable model for the data. Notably, one is here not intrinsically interested in the BMC-component $P_{\mathrm{BMC}}$, but rather in estimating the ground-truth $P_{\mathrm{True}} := (1-\varepsilon)P_{\mathrm{BMC}} + \varepsilon\Delta$, which is not a BMC. It can still be the case that one can exploit the underlying clusters to improve the performance of estimation nonetheless.

Assume that the number of underlying clusters $K$ is known and a sample path $X_0^{\varepsilon}, \ldots, X_{\ell}^{\varepsilon}$ of length $\ell$ of a perturbed BMC is provided. Let $\hat{N}$ denote the associated empirical frequency matrix. A natural general-purpose estimator for the transition matrix, which does not rely on the existence of clusters, is given by the empirical transition matrix $\hat{P}(\ell)$. The entries of the empirical transition matrix for a length $\ell > 0$ are given by

$$\hat{P}_{\mathrm{Empirical}}(\ell)_{ij} := \begin{cases} \frac{\hat{N}_{ij}}{\sum_{k=1}^{n} \hat{N}_{ik}}, & \text{if } \hat{N}_{ij} \neq 0 \\ 0, & \text{if } \hat{N}_{ij} = 0. \end{cases} \tag{6.4}$$

Another estimator may be found by using the clusters $\hat{\mathcal{V}}_1, \ldots, \hat{\mathcal{V}}_K$ that the clustering algorithm for BMCs outputs. One can then hope that, since $P_{\mathrm{True}} \approx P_{\mathrm{BMC}}$ for $\varepsilon \approx 0$, it would be sufficient to consider an estimator $\hat{P}_{\mathrm{BMC}}$ for $P_{\mathrm{BMC}}$ whose entries for a given length $\ell > 0$ of the trajectory are given by

$$\hat{P}_{\mathrm{BMC}}(\ell)_{ij} := \begin{cases} \frac{1}{\#\hat{\mathcal{V}}_{\hat{\nu}_n(j)}} \frac{\sum_{x \in \hat{\mathcal{V}}_{\hat{\nu}_n(i)}, y \in \hat{\mathcal{V}}_{\hat{\nu}_n(j)}} \hat{N}_{x,y}}{\sum_{m=1}^{K} \sum_{x \in \hat{\mathcal{V}}_{\hat{\nu}_n(i)}, y \in \hat{\mathcal{V}}_m} \hat{N}_{x,y}}, & \text{if } \sum_{x \in \hat{\mathcal{V}}_{\hat{\nu}_n(i)}, y \in \hat{\mathcal{V}}_{\hat{\nu}_n(j)}} \hat{N}_{x,y} \neq 0 \\ 0, & \text{if } \sum_{x \in \hat{\mathcal{V}}_{\hat{\nu}_n(i)}, y \in \hat{\mathcal{V}}_{\hat{\nu}_n(j)}} \hat{N}_{x,y} = 0. \end{cases} \tag{6.5}$$

Finally, the following trivial estimator which does not use any data is also considered:

$$\hat{P}_{\mathrm{Uniform}}(\ell)_{ij} = \frac{1}{n}.$$

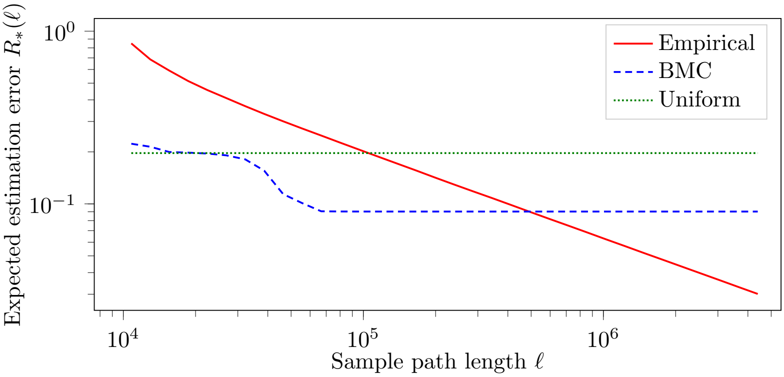


*Figure 6.3.2: Estimated expected estimation error* $R_*(\ell) := \mathbb{E}[\|P - \hat{P}_*(\ell)\|]$ *for* $* \in \{Empirical, BMC, Uniform\}$*. The ground-truth model in this experiment was a perturbed BMC with a heavy-tailed perturbation of strength* $\varepsilon = 0.05$ *on a state space of size* $n = 1000$*. In red: the empirical estimator* $\hat{P}_{Empirical}$ *which is the maximum likelihood estimator for a MC with no additional assumptions. In blue: the BMC estimator* $\hat{P}_{BMC}$*. In green: the trivial estimator* $\hat{P}_{Uniform,ij} := 1/n$ *which does not even use the data. Let us remark that the values depicted depend on the precise parameters of the BMC which was perturbed.*

The performance of these estimators will be measured as a function of the length of the sample path $\ell$ using the expected estimation error:

$$R_*(\ell) := \mathbb{E}[\|P_{\text{True}} - \hat{P}_*(\ell)\|] \quad \text{where } * \in \{\text{Empirical}, \text{BMC}, \text{Uniform}\}. \tag{6.6}$$

Here, $\|\cdot\|$ denotes the operator norm $\|M\| = \sup_{\|v\|_2=1} \|Mv\|_2$.

To ascertain the expected estimation error, a numerical experiments is conducted with a state space of size $n = 1000$ and a heavy-tailed perturbation model of perturbation strength $\varepsilon = 0.05$. Figure 6.3.2 displays estimated values of the expected estimation error $R_*(\cdot)$ as a function of the length $\ell$ of the sample path. A number of different regimes may be identified. First, the regime where the sample path is short, specifically $\ell \approx 10^4$. Here the empirical estimator $\hat{P}_{\text{Empirical}}$ and the BMC estimator $\hat{P}_{\text{BMC}}$ are both unable to outperform the trivial estimator $\hat{P}_{\text{Uniform}}$. The empirical estimator even performs significantly worse than the trivial estimator in this regime. Second, a regime where the sample path is of medium length $\ell \approx 10^5$. Here the clustering procedure succeeds and $\hat{P}_{\text{BMC}}$ becomes the best-performing. Finally, a regime where the sample path is long and $\ell > 10^6$. Here the empirical estimator becomes the best-performing estimator.

These different regimes can be understood in terms of a bias–variance tradeoff. Indeed, for short- to medium-sized sample paths the BMC estimator $\hat{P}_{\text{BMC}}$ has significantly less variance than the empirical estimator $\hat{P}_{\text{Empirical}}$ since it depends on fewer parameters. This decreased variance is the dominant factor in the approximation error in this regime. On the other hand, for long sample paths both estimators $\hat{P}_{\text{BCM}}$ and $\hat{P}$ have low variance and the bias incurred by the approximation $P_{\text{True}} \approx P_{\text{BMC}}$ becomes dominant.

From Sections 6.3.1 and 6.3.2, it can be concluded that the clustering algorithm is robust to small to moderate perturbations. Thus, the clusters obtained with the clustering algorithm may still yield useful insights when the transition kernel has an approximate block-structure.

## 6.4 Model selection for the order of detected clusters

In Section 6.3, the robustness of the clustering algorithm has been tested for BMCs in case the data does not exactly follow a BMC model. Synthetic data coming from perturbed BMCs was used for that purpose.

As we saw from Figure 6.1.2, we do not expect a BMC to be the true data generating process in real-life data. In fact, a BMC with more clusters may fit the data better at the cost of more parameters. Hence, asking which model fits the data is ill-posed. What we can ask instead is if the model is appropriate to describe part of the dynamics of the sequential data, given its complexity, where by complexity we mean the number of free parameters used for the model. A way to resolve these issues is by using model selection.

### 6.4.1 Model selection with information criteria

In model selection, different candidate models for the process generating the data are considered and their ability to explain the observations is compared taking also into account their complexity. A simple way of doing model selection is by using training and validation sets, which is equivalent to obtaining an empirical risk minimizer and then validating the model with data that was not in the training set—recall the setting of statistical learning in Section 1.3. Other similar schemes like cross-validation can also provide a good scheme for assessing what models are most useful. In the context of BMCs, in [9], which this chapter is based on, training and validation sets are used for model selection with Kullback–Leibler (KL)-divergences as a metric of comparison.

Splitting the data can be sometimes undesirable however. Namely, if the data is sparse, the estimated models will become even less accurate. Similarly, when data is inhomogeneous, it may not be clear how to split the data appropriately. Conducting cross-validation may also become too expensive computationally. Information criteria, that we use in this section, can constitute then a suitable alternative.

For a collection of models $\mathbb{Q}(\cdot|\theta)$ parametrized by $\theta \in \Theta$ in some space, a natural metric to compute the goodness-of-fit from given data samples $y$ is to use the log-likelihood

$$\log \mathcal{L}(y|\theta), \tag{6.7}$$

where $\mathcal{L}(y|\theta)$ is the likelihood of the model $\mathbb{Q}(\cdot|\theta)$ given samples $y$. Given the data $y$, we could compute the best $\theta \in \Theta$ that maximizes the log-likelihood, that is, we would obtain the maximum log-likelihood estimator $\hat{\theta}(y) \in \Theta$. The value of the log-likelihood at this maximizer $\log \mathcal{L}(y|\hat{\theta}(y))$ may seem to provide an estimator for the goodness-of-fit of the best model selected in $(\mathbb{Q}(\cdot|\theta))_{\theta \in \Theta}$ that explains the data. This simultaneous estimation and selection with $y$, however, adds bias to the estimator of the log-likelihood compared to a case when independent data for estimation and selection are used.

The bias of the estimator was characterized in [164] under some conditions, and an unbiased estimator was given by the AIC (Akaike Information Criterion)

$$\mathrm{AIC}(\mathbb{Q}) = -2\log\mathcal{L}(y|\hat{\theta}(y)) + 2n_{\mathbb{Q}}, \tag{6.8}$$

where $n_{\mathbb{Q}}$ denotes the number of degrees of freedom in the collection $(\mathbb{Q}(\cdot|\theta))_{\theta\in\Theta}$. Many other criteria with different properties have been proposed in the literature: Bayesian Information Criterion (BIC), Corrected AIC (AICc), etc [161, 135, 52]. For comparing models with sparse data, however, the penalization term in the AIC is insensitive. Moreover, if the true model is among the candidates, the criterion is not consistent. That is, the probability of choosing the correct model as the sample size increases [161] does not converge to one. For our setting we will instead use the Consistent Akaike Information Criterion (CAIC) [161]:

$$\mathrm{CAIC}(\mathbb{Q}) = -2\log\mathcal{L}(y|\hat{\theta}(y)) + n_{\mathbb{Q}}(\log(\ell)+1), \tag{6.9}$$

where $\ell$ is the number of independent samples in $y$. The CAIC is consistent [161] and also penalizes the criterion when the amount of data is sparse—when $\ell$ is small. We remark that CAIC is similar to the BIC (the penalization is $n_{\mathbb{Q}}\log(\ell)$ instead).

Information criteria are used in parametric models, where taking the number of degrees of freedom into account is important for assessing the performance of different models. However, they usually only provide asymptotic consistency, and for finite sample sizes their performance is unknown. Furthermore, in our case, data is not expected to be fully independent and so we should expect some unknown bias in the estimation procedure.

We will nonetheless use information criteria in Section 6.4.2 to look at a specific problem: selecting the order of a BMC.

We compare the BMC model to other similar models. Namely, generalized BMCs models where the dependence of the cluster transitions is not just a 1st-order MC, but can have lower- or higher-order Markovian dependence. The models that we will use are defined as follows:

### 0th-order BMCs

Let $K \in [n]$ and consider an arbitrary probability distribution $\eta : [K] \to [0,1]$. A 0*th-order BMC* is then a BMC with cluster transition matrix $q_{k,l} := \eta_l$ for all $k,l \in [K]$. The 0th-order BMC will serve as a benchmark to assess whether the structures we find are actually due to the sequential nature of the process and do not admit time-independent explanation.

In a 0th-order BMC, each next sample $X_{t+1}$ is independent of the previous sample $X_t$. A 0th-order BMC therefore generates sequences of independent and identically distributed random variables. This is contrary to a 1st-order BMC, which generates a sequence of dependent random variables. Thus, the probability of a specific observation depends on the cluster of the observation, and specifically is identical for every observation within that cluster.

### $r$th-order BMCs

Conversely, one can also consider models with higher-order dependencies than a 1st-order BMC has. Consider a discrete-time stochastic process $\{Y_t\}_{t=1}^{\ell}$ (not necessarily a MC) that satisfies $Y_t \in [n]$ for some $n \in \mathbb{N}_+$. We say that $\{Y_t\}_{t\geq 1}$ is an $r$th-order MC if and only if for all $t \in [\ell - r]$, all multi-indices $\boldsymbol{i}^r = (i_1, \ldots, i_r) \in [n]^r$, states $s_{t-r}, \ldots, s_1$, and $j \in [n]$,

$$\begin{aligned} &\mathbb{P}[Y_{t+1} = j \mid Y_t = i_r, Y_{t-1} = i_{r-1}, \ldots, Y_{t-r+1} = i_1, \\ &\qquad\qquad Y_{t-r} = s_{t-r}, , \ldots, Y_1 = s_1] \\ &= \mathbb{P}[Y_{t+1} = j \mid Y_t = i_r, Y_{t-1} = i_{r-1}, \ldots, Y_{t-r+1} = i_1] =: Q^r_{\boldsymbol{i}^r, j} \end{aligned} \tag{6.10}$$

for some transition matrix $Q^r \in [0,1]^{n^r \times n}$.

Then, we define the process $\{X^r_t\}_{t\geq 1}$ to be an *rth-order* BMC if $\{X^r_t\}_{t\geq 1}$ is an MC with state transition probabilities given by

$$Q^r_{i^r,j} = \frac{q^r_{\nu(\boldsymbol{i}^r),\nu(j)}}{|\mathcal{V}_{\nu(j)}|}, \tag{6.11}$$

where $q^r_{\nu(i^r),\nu(j)}$ are now cluster transition probabilities of an $r$th-order MC and we have the multi-indices of clusters $\nu(\boldsymbol{i}^r) = (\nu(i_1), \ldots, \nu(i_r)) \in [K]^r$. Hence an $r$th-order BMC $X^r_t$ is a BMC, where the clustered process $Y^r_t = \nu(X^r_t)$ follows an $r$th-order MC.

We will use these models to conduct model selection in the next section.

## 6.4.2 Model selection for the order of a BMC

Suppose that a sequence $X_{1:\ell} = \{X_t\}_{t=1}^{\ell}$ was in fact generated by some $r$th-order BMC, but that the order $r \in \{0, 1, \ldots\}$ is unknown. We will use techniques for model selection to try and determine $r$ from the cluster sequence $Y_{1:\ell} = \nu_n(X_{1:\ell})$.

There are two reasons for using $Y_{1:\ell}$ instead of $X_{1:\ell}$. First, the parametric models for higher-order MCs without clusters have a comparable number of free parameters as the sequence length $\ell$ itself, so estimators for the order will behave poorly. If we look at the cluster sequence instead, the number of degrees of freedom will depend on the cluster number $K$ instead of the number of states $n$, and fortunately $K \ll n$. Secondly, we can study the robustness of the model selection procedure depending on the clustering algorithm, since instead of the chain of states we are only interested in the chain of clusters, which may be more robust to errors.

### Order selection by minimizing an information criterion

The parameter that determines the $r$th-order BMC model for $Y_{1:\ell}$ is a transition matrix $Q^r$; recall (6.10). Note here that the chain $Y^r_{1:\ell-r}$ will be constructed from the chain of clusters $Y_{1:\ell} = \nu_n(X_{1:\ell})$ for a fixed cluster assignment $\nu_n$, which we are provided by the clustering algorithm. We assume $\nu_n$ fixed for the time being.

To estimate $Q^r$ one can consider the log-likelihood

$$\log \mathcal{L}(Y_{1:\ell} \mid Q^r) = \sum_{t=r}^{\ell-r-1} \log(Q^r_{Y_{t-r+1:t}, Y_{t+1}}). \tag{6.12}$$

The maximum-likelihood estimator associated with (6.12) is given by

$$(\hat{Q}^{r,\mathrm{MLE}})_{\boldsymbol{i}^r,j} := \begin{cases} \dfrac{\sum_{t=r}^{\ell-r-1} \mathbb{1}[Y_{t-r+1:t} = \boldsymbol{i}^r, Y_{t+1} = j]}{\sum_{t=r}^{\ell-r-1} \mathbb{1}[Y_{t-r+1:t} = \boldsymbol{i}^r]} & \text{if } \sum\limits_{t=r}^{\ell-r-1} \mathbb{1}[Y_{t-r+1:t} = \boldsymbol{i}^r] > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{6.13}$$

Here $\boldsymbol{i}^r, j$ run over all possible sequences in $[K]^r$ and $[K]$ respectively. We denote

$\hat{\mathbb{Q}}^{r,\mathrm{MLE}}$ : The Maximum-Likelihood Estimator (MLE) of an $r$th-order MC estimated from the observation sequence $Y_{1:\ell}$.

That is, $\hat{\mathbb{Q}}^{r,\mathrm{MLE}}$ is the law of an $r$th-order MC with $K$ states and transition matrix $\hat{Q}^{r,\mathrm{MLE}}$.

To determine what order $r$ is the true underlying order of the data we would like to compare $\hat{\mathbb{Q}}^{r,\mathrm{MLE}}$ and $\hat{\mathbb{Q}}^{s,\mathrm{MLE}}$ for some $s \neq r$. As remarked in Section 6.4, using the log-likelihood for this purpose would give a biased estimator. Therefore, we use instead the CAIC in (6.14): for the model $\hat{\mathbb{Q}}^{r,\mathrm{MLE}}$,

$$\mathrm{CAIC}(\hat{Q}^{r,\mathrm{MLE}}) := -2\log\mathcal{L}(Y_{1:\ell} \mid \hat{Q}^{r,\mathrm{MLE}}) + \mathrm{DF}(K,r)\big(1+\log(\ell-r)\big). \tag{6.14}$$

Here, $\mathrm{DF}(K,r)$ is the number of degrees of freedom in an $r$th-order MC constrained to have fixed parameters $K$ and $r$. Specifically,

$$\mathrm{DF}(K,r) = K^r(K-1), \tag{6.15}$$

where the factor $(K-1)$ is due to the fact that the rows of $Q^r$ sum to one.

We will utilize the CAIC to select the right order as follows. From the collection of models $\hat{\mathbb{Q}}^{0,\mathrm{MLE}}$, $\hat{\mathbb{Q}}^{1,\mathrm{MLE}}$, $\hat{\mathbb{Q}}^{2,\mathrm{MLE}}$, …, we may determine the order $r^{\mathrm{CAIC}}$ that minimizes the CAIC:

$$r^{\mathrm{CAIC}} := \mathrm{argmin}_{r\in\{0,1,2,\dots\}}\, \mathrm{CAIC}(\hat{Q}^{r,\mathrm{MLE}}). \tag{6.16}$$

Note that lower-dimensional models are favored since the degrees of freedom $\mathrm{DF}(K,r)$, and thus the penalty terms in (6.14), increase exponentially in $K,r$.

In order to evaluate the robustness of the CAIC criterion, we will furthermore estimate the over- and underfit error probabilities with tailor-made error models.

### 6.4.3 Selected orders

We now examine the results of the selection of the order of the chain using an information criterion as described in Section 6.4.2. To do so, we compute information criteria for all datasets. In Section 6.4.4, we will furthermore ascertain the robustness of the criterion. For simplicity, in this latter task we focus just on the DNA and the S&P500 datasets.

#### Results

We compute (6.14) for $r = 0,1,2,3,4$. The results are shown in Table 6.4.1. The magnitude of the CAIC in Table 6.4.1 depends strongly on the observation sequence and the number of clusters. In particular, for the GPS dataset the the criterion selects $r = 2$ and the differences between the criterion values are also notably large for most orders $r \in \{0,1,2,3,4\}$. This is mostly due to the large number of clusters $K = 15$, for which higher orders become highly penalized. For the DNA dataset, the criterion suggests that orders $r \in \{1,2\}$ best approximate the data. For the S&P500 dataset, on the other hand, orders $r \in \{0,1\}$ are selected as the best candidates. For the latter, a possible explanation for this result is that the number of clusters $K = 3$ is small. In this case, the transitions between individual states become aggregated in the clusters thereby making the chain closer to stationarity. Note, however, that more clusters would also not guarantee different values for $r$, as the number of transitions of a cluster for larger $K$ would be even sparser. For model selection with models with different number of clusters $K$, we refer to [9, Section 8.4].

| $r$ | DNA | incr. (%) | GPS($\times 10^3$) | incr. (%) | S&P500 | incr. (%) |
|---|---|---|---|---|---|---|
| 0 | 432650 | n.a. | 960.63 | n.a. | **9853** | **n.a.** |
| 1 | 431502 | -0.27 | 626.54 | -34.8 | 9860 | +0.07 |
| 2 | **431263** | **-0.32** | **571.49** | **-40.5** | 9940 | +0.81 |
| 3 | 435228 | +0.69 | 1121.90 | +16.8 | 10253 | +3.1 |
| 4 | 458512 | +5.3 | 9789.27 | +1019 | 11162 | +8.9 |

*Table 6.4.1: The CAIC in* (6.14) *for the different datasets. Note that the relative difference between the values pertaining to different orders is often small. For example, the differences are less than* 0.1% *between orders* 1, 2 *for the DNA data, and between orders* 0, 1 *for the S&P500 data. This is not the case, however, with the GPS dataset.*

In Table 6.4.1, we expect that there is a large variance in the CAIC values and some over- or underfitting of the order is possible. The criterion indicates nonetheless that the transitions of the found clusters, except maybe for the S&P500 dataset, can be better approximated by a MC of order $r \geq 1$. We will now support this conclusion empirically with the error models for the DNA and S&P500 datasets.

### 6.4.4 Experimental evaluation of the CAIC criterion

In order to ascertain how significant the information criteria are, we will use the original data $X_{1:\ell}$ and use the parameters of the maximum-likelyhood estimator that best suit an $r$th-order BMC. Let $\hat{\mathbb{P}}^{r,\mathrm{MLE}}$ be the law of the best $r$th-order BMC that fits the data. Then, for each $r$ we will use the model consisting of $\hat{\mathbb{P}}^{r,\mathrm{MLE}}$ together with a perturbation coming from a MC of different order that $r$. We remark that all these models will use the full state space $[n]$. The clustered process generated by these error models will be analyzed. For $r \in \{0,1\}$ we will consider two data-generating models together with errors determined by a parameter $\varepsilon \in [0,1)$. The models are:

$\mathbb{W}^1_\varepsilon$: A perturbed 1st-order BMC with probability distribution $\hat{\mathbb{P}}^{1,\mathrm{MLE}}$ and a perturbation given by a heavy-tailed 0th-order perturbation. This model is defined similarly to the pertubation model in Section 6.3. In this case, however, the perturbation here is assumed to be a 0th-order MC as defined in Section 6.4.1 instead of a 1st-order MC.

$\mathbb{W}^0_\varepsilon$: A perturbed 0th-order BMC with probability distribution $\hat{\mathbb{P}}^{0,\mathrm{MLE}}$ and a perturbation given by a heavy-tailed 1st-order MC as defined in Section 6.3.

Denote $Y^{r,\varepsilon}_{1:\ell} = \nu_n(X^\varepsilon_{1:\ell})$ the cluster process if $X^\varepsilon_{1:\ell} \sim \mathbb{W}^r_\varepsilon$. We will study the robustness of the CAIC criterion by examining how often it over- and underfits when selecting $s$ for the models $\hat{\mathbb{Q}}^{s,\mathrm{MLE}}$ with the clustered sequence $Y^{r,\varepsilon}_{1:\ell}$. To study this aspect, we will consider two targets for the CAIC and restrict to the orders $r \in \{0,1\}$. The first target is the overfit error probability

$$e_{\mathrm{over}}(\varepsilon) := \mathbb{P}_{X^\varepsilon_{1:\ell} \sim \mathbb{W}^0_\varepsilon}(\mathrm{argmin}_{r\in\{0,1\}}\mathrm{CAIC}(Y^{r,\varepsilon}_{1:\ell}) = 1), \tag{6.17}$$

that is, the probability that the criterion selects a 1st-order process for the chain of cluster transitions when the underlying generating process is $\hat{\mathbb{P}}^{0,\mathrm{MLE}}$ and the only higher order contributions come from perturbations. The second target is the underfit error probability defined as

$$e_{\text{under}}(\varepsilon) := \mathbb{P}_{X^{\varepsilon}_{1:\ell}\sim\mathbb{W}^1_{\varepsilon}}(\text{argmin}_{r\in\{0,1\}}\text{CAIC}(Y^{\varepsilon}_{1:\ell}) = 0), \tag{6.18}$$

that is, the probability that the criterion selects a 0th-order process as the best-candidate while $\hat{\mathbb{P}}^{1,\text{MLE}}$ is the actual underlying data-generating process.

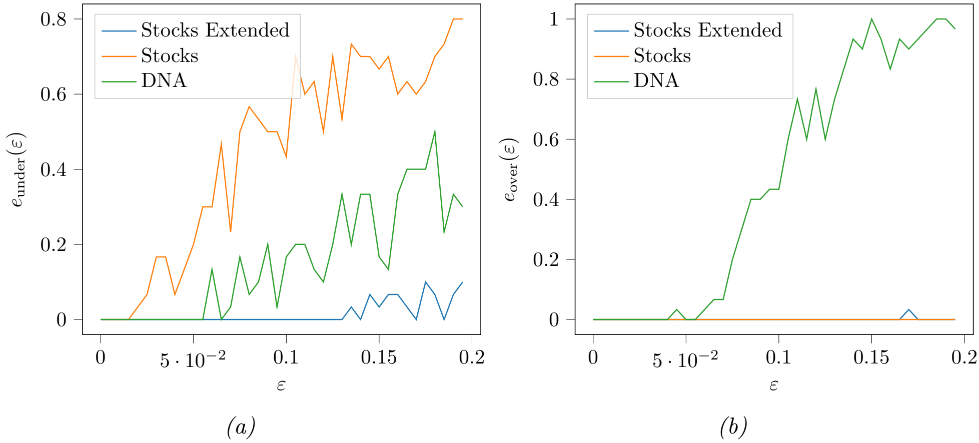


*Figure 6.4.1:* (a) *Underfit error probability* $e_{\text{under}}(\varepsilon)$ *in* (6.18) *depending on the perturbation* $\varepsilon$ *for the DNA, S&P500 dataset, and extended S&P500 datasets assuming the data-generating process is* $\mathbb{W}^1_{\varepsilon}$ *for their respective datasets. Note the effect of reducing sparsity for the extended S&P500 dataset in reducing the underfit error probability. The plot suggests that the probability of underfitting for small* $\varepsilon$*, while nonzero, is less than the probability of selecting the correct order.* (b) *Overfit error probability* $e_{\text{over}}(\varepsilon)$ *depending on the perturbation* $\varepsilon$ *for the DNA, S&P500 dataset, and extended S&P500 datasets assuming the data-generating process is* $\mathbb{W}^0_{\varepsilon}$*. Compared to the underfit error probability, the CAIC is robust at selecting a model with lower order, provided there is enough information. In particular, for the DNA, the* 1*st-order perturbation becomes dominant in the criterion fairly quickly after* $\varepsilon > 0.05$*. In all tests the number of repetitions was* $R = 30$*.*

We focus now on the DNA and S&P500 datasets. We will compute the CAIC and the error probabilities obtained by using a BMCs model inferred from the data together with an error model of a different order. Because the S&P500 dataset is the least clear dataset, we also consider a synthetic observation sequence. The sequence is generated using the same model $\mathbb{W}^r_{\varepsilon}$ as is obtained for the S&P500 dataset, but will be five times as long: $5\ell_{\text{SM}}$, where $\ell_{\text{SM}}$ is the length of the path of the dataset. We will refer to this synthetic observation sequence as the “extended stock market model.” In this manner we can see the effect of sparsity on the criterion robustness *as if* we had access to more data (albeit from a BMC).

Figure 6.4.1 shows the error probabilities as well as centered CAIC values. We see that both the underfit $e_1(\varepsilon)$ and overfit error $e_2(\varepsilon)$ are usually small for small $\varepsilon$. The overfit error is, however, considerably larger for the DNA dataset than for the S&P500

dataset. This supports the claim that the CAIC chooses the model with fewest parameters for the same amount of information, that is, the criterion is less prone to overfit when the data is sparse. The underfit error is on the contrary small for the DNA dataset, also for $\varepsilon \in [0.1, 0.2]$. This suggests that order selection via information criteria is robust to misclassification error.

The case of the S&P500 dataset is especially interesting. In Table 6.4.1, the criterion is just slightly lower for $r = 0$ than $r = 1$ whereas in the $\mathbb{W}^1_\varepsilon$ model in Figure 6.4.1(a), the criterion selects $r = 1$ up to $\varepsilon \sim 0.1$. Afterwards, deviating from the BMC model by just 1 out of 10 transitions will make the value of the criterion for $r = 0$ very close to that of $r = 1$, similarly as in Table 6.4.1. This is also supported by Figure 6.4.2, where the difference between the criterion for $r = 0$ and $r = 1$ in the S&P500 dataset takes values in $[0, 10]$, which we coincidentally also see in Table 6.4.1. This suggests that there may be a 1st-order Markovian structure in the S&P500 dataset but also a strong 0th-order process. Alternatively, the data may simply be too sparse for the CAIC to select a suitable order. This hypothesis is also supported by the stock market extended dataset, where model selection with five times more data has fewer such problems.

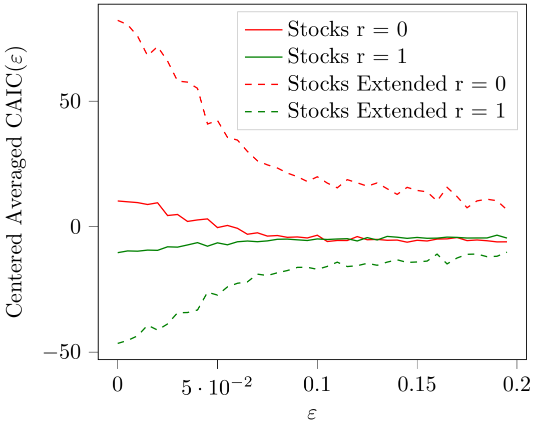


*Figure 6.4.2: Centered average of CAIC for the S&P500 dataset and S&P500 dataset extended datasets assuming the data-generating process is $\mathbb{W}^1_\varepsilon$. Note that the criterion is not predictive of the true order of the data-generating process after $\varepsilon \sim 0.08$. However, by increasing the dataset size 5-fold, it can very robustly select the correct order even for larger error $\varepsilon$. We remark that considered individually, the empirical standard deviation of the CAIC average—note, not centered—for each r is an order of magnitude too large to be represented in the plot $(\mathrm{Std}(\mathrm{CAIC}(Y_{1:\ell}(\varepsilon)) \simeq O(10^2))$. Despite this large standard deviation, after centering the criteria on the center of mass of the CAIC averages of different orders r obtained from the same sample, the relative difference is small and the selection process is robust for small error $\varepsilon$.*

We finally remark that looking at information criteria for the unclustered observation sequences $X_{1:\ell}$ provides no useful insights due to the large dimensionality of the models. In particular, the CAIC criteria for the unclustered observation sequences for order $r \in \{0, 1\}$ can be seen in Table 6.4.2. As the data shows, the CAIC criterion just picks the model with the smallest number of parameters. This is even more extreme in the GPS and S&P500 datasets, where on top of a large model dimension we have sparse data.

| $r$ | DNA | GPS | S&P500 |
|---|---|---|---|
| 0 | **1339.5** $\times 10^3$ | **2943** $\times 10^3$ | **54.27** $\times 10^3$ |
| 1 | 1361.9 $\times 10^3$ | $\approx$ 1 $\times 10^8$ | 882 $\times 10^3$ |

*Table 6.4.2: The CAIC in* (6.14) *for the sequence* $X_{1:\ell}$ *for different datasets.*

### 6.4.5 Conclusion

The main takeaways from the previous results address the last question posed in Section 6.1:

- Model selection is feasible if we use the clustered sequence $Y_{1:\ell} = \nu_n(X_{1:\ell})$ obtained after the clustering algorithm. This namely reduces the amount of free parameters of the models considerably.
- For the DNA and GPS datasets, the CAIC selects a nonzero-order MC for the cluster dynamics.
- For the S&P500 dataset the CAIC shows that the data is too sparse for selecting a specific order with certainty. However, there are indications that the values obtained in the CAIC for the S&P500 dataset are consistent with a 1st-order BMC model with a strong 0th-order MC baseline.

Hence, while uncertainty is still high, there is evidence that the BMC is a better suited model for describing the transition dynamics on sequential data than models with less complexity which do not assume a Markovian structure of the dynamics.

# Bibliography

[1] J. Sanders and A. Senen-Cerda. “Spectral norm bounds for block Markov chain random matrices.” In: *Stochastic Processes and their Applications* 158 (2023).

[2] J. Sanders and A. Van Werde. “Singular value distribution of dense random matrices with block Markovian dependence.” In: *Stochastic Processes and their Applications* 158 (2023).

[3] B. Bah, H. Rauhut, U. Terstiege, and M. Westdickenberg. “Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers.” In: *Information and Inference: A Journal of the IMA* 11 (2022).

[4] Y. Jedra, J. Lee, A. Proutière, and S.-Y. Yun. “Nearly optimal latent state decoding in block MDPs.” In: *arXiv preprint arXiv:2208.08480* (2022).

[5] O. A. Manita, M. A. Peletier, J. W. Portegies, J. Sanders, and A. Senen–Cerda. “Universal approximation in dropout neural networks.” In: *Journal of Machine Learning Research* 23 (2022).

[6] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. “High-resolution image synthesis with latent diffusion models.” In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[7] A. Senen–Cerda and J. Sanders. “Asymptotic convergence rate of Dropout on shallow linear neural networks.” In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6 (2022).

[8] R. Szeliski. *Computer vision: Algorithms and applications*. Springer Nature, 2022.

[9] A. Van Werde, A. Senen–Cerda, G. Kosmella, and J. Sanders. “Detection and evaluation of clusters within sequential data.” In: *arXiv preprint arXiv:2210.01679* (2022).

[10] D. Zhang, N. Maslej, E. Brynjolfsson, J. Etchemendy, T. L. James, M. Helen Ngo, J. C. Niebles, M. Sellitto, E. Sakhaee, Y. Shoham, J. Clarck, and R. Perrault. *The AI index 2022 annual report*. Tech. rep. Stanford University, 2022.

[11] Alpha Vantage Co. *Stock Data API*. 2021. URL: https://www.alphavantage.co/.

[12] R. Arora, P. Bartlett, P. Mianjy, and N. Srebro. “Dropout: Explicit forms and capacity control.” In: *International Conference on Machine Learning*. 2021.

[13] D. Elbrächter, D. Perekrestenko, P. Grohs, and H. Bölcskei. “Deep neural network approximation theory.” In: *IEEE Transactions on Information Theory* 67 (2021).

[14] T. Gao, H. Liu, J. Liu, H. Rajan, and H. Gao. "A global convergence theory for deep ReLU implicit networks via over-parameterization." In: *International Conference on Learning Representations*. 2021.

[15] A. Jentzen and A. Riekert. "On the existence of global minima and convergence analyses for gradient descent methods in the training of deep neural networks." In: *arXiv preprint arXiv:2112.09684* (2021).

[16] National Library of Medicine. *OCA2 melanosomal transmembrane protein Homo sapiens (human)*. https://www.ncbi.nlm.nih.gov/gene/4948. Accessed in October 2021, RefSeq Accession NC_000015.10. 2021.

[17] S. Tarmoun, G. Franca, B. D. Haeffele, and R. Vidal. "Understanding the Dynamics of Gradient Flow in Overparameterized Linear models." In: *International Conference on Machine Learning*. 2021.

[18] Z. Zhu, X. Li, M. Wang, and A. Zhang. "Learning Markov models via low-rank optimization." In: *Operations Research* 70 (2021).

[19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. "Language models are few-shot learners." In: *Advances in Neural Information Processing Systems*. 2020.

[20] Y. Drori and O. Shamir. "The complexity of finding stationary points with stochastic gradient descent." In: *International Conference on Machine Learning*. 2020.

[21] B. Fehrman, B. Gess, and A. Jentzen. "Convergence rates for the stochastic gradient descent method for non-convex objective functions." In: *Journal of Machine Learning Research* 21 (2020).

[22] A. Foong, D. Burt, Y. Li, and R. Turner. "On the expressiveness of approximate inference in Bayesian neural networks." In: *Advances in Neural Information Processing Systems*. 2020.

[23] C. Gallicchio and S. Scardapane. "Deep randomized neural networks." In: *Recent Trends in Learning From Data*. Springer, 2020.

[24] G. Garbi, E. Incerto, and M. Tribastone. "Learning queuing networks by recurrent neural networks." In: *ACM/SPEC International Conference on Performance Engineering*. 2020.

[25] P. Mianjy and R. Arora. "On convergence and generalization of Dropout training." In: *Advances in Neural Information Processing Systems*. 2020.

[26] A. Pal, C. Lane, R. Vidal, and B. D. Haeffele. "On the regularization properties of structured dropout." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[27] J. Sanders, A. Proutière, and S.-Y. Yun. "Clustering in block Markov chains." In: *The Annals of Statistics* 48 (2020).

[28] J. Schmidt–Hieber. "Nonparametric regression using deep neural networks with ReLU activation function." In: *The Annals of Statistics* 48 (2020).

[29] A. Senen–Cerda and J. Sanders. "Almost sure convergence of Dropout algorithms for neural networks." In: *arXiv preprint arXiv:2002.02247* (2020).

[30] Y. Tian, J. Qian, and S. Sra. "Towards minimax optimal reinforcement learning in factored Markov decision processes." In: *Advances in Neural Information Processing Systems*. 2020.

[31] C. Wei, S. Kakade, and T. Ma. "The implicit and explicit regularization effects of Dropout." In: *International Conference on Machine Learning*. 2020.

[32] Y. Zhou, A. R. Zhang, L. Zheng, and Y. Wang. "Optimal high-order tensor SVD via tensor-train orthogonal iteration." In: *arXiv preprint arXiv:2010.02482* (2020).

[33] D. Zou, Y. Cao, D. Zhou, and Q. Gu. "Gradient descent optimizes over-parameterized deep ReLU networks." In: *Machine Learning* 109 (2020).

[34] Z. Allen-Zhu, Y. Li, and Z. Song. "A convergence theory for deep learning via over-parameterization." In: *International Conference on Machine Learning*. 2019.

[35] S. Arora, N. Golowich, N. Cohen, and W. Hu. "A convergence analysis of gradient descent for deep linear neural networks." In: *International Conference on Learning Representations*. 2019.

[36] F. Benaych-Georges, C. Bordenave, and A. Knowles. "Largest eigenvalues of sparse inhomogeneous Erdös–Rényi graphs." In: *The Annals of Probability* 47 (2019).

[37] G. De Bie, G. Peyré, and M. Cuturi. "Stochastic deep networks." In: *International Conference on Machine Learning*. 2019.

[38] Z. Du, N. Ozay, and L. Balzano. "Mode clustering for Markov jump systems." In: *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*. 2019.

[39] Y. Duan, T. Ke, and M. Wang. "State aggregation learning from Markov transition data." In: *Advances in Neural Information Processing Systems*. 2019.

[40] D. P. Kingma, M. Welling, et al. "An introduction to variational autoencoders." In: *Foundations and Trends in Machine Learning* 12 (2019).

[41] A. Labach, H. Salehinejad, and S. Valaee. "Survey of dropout methods for deep neural networks." In: *arXiv preprint arXiv:1904.13310* (2019).

[42] P. Mianjy and R. Arora. "On Dropout and nuclear norm regularization." In: *International Conference on Machine Learning*. 2019.

[43] M. Sewak. "Deep Q network (DQN), double DQN, and dueling DQN." In: *Deep Reinforcement Learning*. Springer, 2019.

[44] O. Shamir. "Exponential convergence time of gradient descent for one-dimensional deep linear neural networks." In: *Conference on Learning Theory*. 2019.

[45] M. Szczepański. *Economic impacts of artificial intelligence*. Tech. rep. European Parliamentary Research Service (EPRS), 2019.

[46] Y. Yin. "Random neural network methods and deep learning." In: *Probability in the Engineering and Informational Sciences* 35 (2019).

[47] A. Zhang and M. Wang. "Spectral state compression of Markov processes." In: *IEEE Transactions on Information Theory* 66 (2019).

[48] A. Al-Kaff, D. Martin, F. Garcia, A. de la Escalera, and J. M. Armingol. "Survey of computer vision algorithms and applications for unmanned aerial vehicles." In: *Expert Systems with Applications* 92 (2018).

[49] P. L. Bartlett, D. P. Helmbold, and P. M. Long. “Gradient descent with identity initialization efficiently learns positive-definite linear transformations by deep residual networks.” In: *Neural Computation* 31 (2018).

[50] L. Bottou, F. E. Curtis, and J. Nocedal. “Optimization methods for large-scale machine learning.” In: *Siam Review* 60 (2018).

[51] J. Cavazza, P. Morerio, B. Haeffele, C. Lane, V. Murino, and R. Vidal. “Dropout as a low-rank regularizer for matrix factorization.” In: *International Conference on Artificial Intelligence and Statistics.* 2018.

[52] J. Ding, V. Tarokh, and Y. Yang. “Model selection techniques: An overview.” In: *IEEE Signal Processing Magazine* 35 (2018).

[53] J. Gillham, L. Rimmington, H. Dance, G. Verweij, A. Rao, K. B. Roberts, and M. Paich. *The macroeconomic impact of artificial intelligence.* Tech. rep. PricewaterhouseCoopers (PwC), 2018.

[54] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al. “Recent advances in convolutional neural networks.” In: *Pattern Recognition* 77 (2018).

[55] K. Hamidieh. “A data-driven statistical model for predicting the critical temperature of a superconductor.” In: *Computational Materials Science* 154 (2018).

[56] A. Jacot, F. Gabriel, and C. Hongler. “Neural tangent kernel: Convergence and generalization in neural networks.” In: *Advances in Neural Information Processing Systems.* 2018.

[57] W. Kirsch and T. Kriecherbauer. “Semicircle law for generalized Curie–Weiss matrix ensembles at subcritical temperature.” In: *Journal of Theoretical Probability* 31 (2018).

[58] W. Kirsch and T. Kriecherbauer. “Sixty years of moments for random matrices.” In: *Non-linear partial differential equations, mathematical physics, and stochastic analysis.* European Mathematical Society, 2018.

[59] C. M. Le, E. Levina, and R. Vershynin. “Concentration of random graphs and application to community detection.” In: *International Congress of Mathematicians.* 2018.

[60] P. Mianjy, R. Arora, and R. Vidal. “On the implicit bias of Dropout.” In: *International Conference on Machine Learning.* 2018.

[61] S. Oymak. “Learning compact neural networks with regularization.” In: *International Conference on Machine Learning.* 2018.

[62] G. Urban, K. Bache, D. T. Phan, A. Sobrino, A. K. Shmakov, S. J. Hachey, C. C. Hughes, and P. Baldi. “Deep learning for drug discovery and cancer research: Automated analysis of vascularization images.” In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16 (2018).

[63] J. Cavazza, C. Lane, B. D. Haeffele, V. Murino, and R. Vidal. “An analysis of Dropout for matrix factorization.” In: *arXiv preprint arXiv:1710.03487* (2017).

[64] T. DeVries and G. W. Taylor. “Improved regularization of convolutional neural networks with cutout.” In: *arXiv preprint arXiv:1708.04552* (2017).

[65] C. Gao, Z. Ma, A. Y. Zhang, and H. H. Zhou. "Achieving optimal misclassification proportion in stochastic block models." In: *The Journal of Machine Learning Research* 18 (2017).

[66] S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro. "Implicit regularization in matrix factorization." In: *Advances in Neural Information Processing Systems*. 2017.

[67] C. M. Le, E. Levina, and R. Vershynin. "Concentration and regularization of random graphs." In: *Random Structures & Algorithms* 51 (2017).

[68] D. Molchanov, A. Ashukha, and D. Vetrov. "Variational dropout sparsifies deep neural networks." In: *International Conference on Machine Learning*. 2017.

[69] H. Nguyen. "A universal approximation theorem for Gaussian-gated mixture of experts models." In: *SSRN Electronic Journal* (2017).

[70] Q. Nguyen and M. Hein. "The loss surface of deep and wide neural networks." In: *International Conference on Machine Learning*. 2017.

[71] D. L. Stephen Blake Randy Arndt. *Movebank*. https://www.movebank.org/cms/webapp?gwt_fragment=page=studies,path=study8019591. Accessed: 2022-08-16. 2017.

[72] E. Abbe, A. S. Bandeira, and G. Hall. "Exact recovery in the Stochastic Block Model." In: *IEEE Transactions on Information Theory* 62 (2016).

[73] S. Bhojanapalli, B. Neyshabur, and N. Srebro. "Global optimality of local search for low rank matrix recovery." In: *Advances in Neural Information Processing Systems*. 2016.

[74] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[75] B. Hajek, Y. Wu, and J. Xu. "Achieving exact cluster recovery threshold via semidefinite programming." In: *IEEE Transactions on Information Theory* 62 (2016).

[76] W. Hochstättler, W. Kirsch, and S. Warzel. "Semicircle law for a matrix ensemble with dependent entries." In: *Journal of Theoretical Probability* 29 (2016).

[77] H. Karimi, J. Nutini, and M. Schmidt. "Linear convergence of gradient and proximal–gradient methods under the Polyak-Łojasiewicz condition." In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 2016.

[78] K. Kawaguchi. "Deep learning without poor local minima." In: *Advances in Neural Information Processing Systems*. 2016.

[79] E. Kay and A. Agarwal. "Dropconnected neural network trained with diverse features for classifying heart sounds." In: *Computing in Cardiology Conference*. 2016.

[80] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. "Gradient descent converges to minimizers." In: *arXiv preprint arXiv:1602.04915* (2016).

[81] J. Lei. "A goodness-of-fit test for Stochastic Block Models." In: *The Annals of Statistics* 44 (2016).

[82] Z. Li, B. Gong, and T. Yang. "Improved Dropout for shallow and deep learning." In: *Advances in Neural Information Processing Systems*. 2016.

[83] H. D. Nguyen, L. R. Lloyd-Jones, and G. J. McLachlan. "A universal approximation theorem for mixture-of-experts models." In: *Neural Computation* 28 (2016).

[84] S. Semeniuta, A. Severyn, and E. Barth. “Recurrent Dropout without memory loss.” In: *International Conference on Computational Linguistics: Technical Papers*. 2016.

[85] S.-Y. Yun and A. Proutière. “Optimal cluster recovery in the labeled Stochastic Block Model.” In: *Advances in Neural Information Processing Systems*. 2016.

[86] E. Abbe and C. Sandon. “Community detection in general Stochastic Block Models: Fundamental limits and efficient algorithms for recovery.” In: *IEEE Annual Symposium on Foundations of Computer Science*. 2015.

[87] E. Abbe and C. Sandon. “Recovering communities in the general Stochastic Block Model without knowing the parameters.” In: *Advances in Neural Information Processing Systems*. 2015.

[88] K. Avrachenkov, L. Cottatellucci, and A. Kadavankandy. “Spectral properties of random matrices for Stochastic Block Model.” In: *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*. 2015.

[89] S. Bubeck et al. “Convex optimization: Algorithms and complexity.” In: *Foundations and Trends in Machine Learning* 8 (2015).

[90] S. Ioffe and C. Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift.” In: *International Conference on Machine Learning*. 2015.

[91] V. Jog and P.-L. Loh. “Information-theoretic bounds for exact recovery in weighted stochastic block models using the Rényi divergence.” In: *arXiv preprint arXiv:1509.-06418* (2015).

[92] D. P. Kingma, T. Salimans, and M. Welling. “Variational Dropout and the local reparameterization trick.” In: *Advances in Neural Information Processing Systems*. 2015.

[93] J. Lei and A. Rinaldo. “Consistency of spectral clustering in Stochastic Block Models.” In: *The Annals of Statistics* 43 (2015).

[94] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. “Human-level control through deep reinforcement learning.” In: *Nature* 518 (2015).

[95] E. Mossel, J. Neeman, and A. Sly. “Consistency thresholds for the planted bisection model.” In: *ACM Symposium on Theory of Computing*. 2015.

[96] E. Mossel, J. Neeman, and A. Sly. “Reconstruction and estimation in the Planted Partition Model.” In: *Probability Theory and Related Fields* 162 (2015).

[97] B. Neyshabur, R. Tomioka, and N. Srebro. “Norm-based capacity control in neural networks.” In: *Conference on Learning Theory*. 2015.

[98] D. Paulin. “Concentration inequalities for Markov chains by Marton couplings and spectral methods.” In: *Electronic Journal of Probability* 20 (2015).

[99] J. A. Tropp. “An introduction to matrix concentration inequalities.” In: *Foundations and Trends in Machine Learning* 8 (2015).

[100] P. Baldi and P. Sadowski. “The Dropout learning algorithm.” In: *Artificial intelligence* 210 (2014).

[101] D. P. Kingma and J. Ba. “ADAM: A method for stochastic optimization.” In: *arXiv preprint arXiv:1412.6980* (2014).

[102] L. Massoulié. “Community detection thresholds and the weak Ramanujan property.” In: *ACM Symposium on Theory of Computing.* 2014.

[103] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour. “Dropout improves recurrent neural networks for handwriting recognition.” In: *International Conference on Frontiers in Handwriting Recognition.* 2014.

[104] K. Simonyan and A. Zisserman. “Very deep convolutional networks for large-scale image recognition.” In: *arXiv preprint arXiv:1409.1556* (2014).

[105] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: a simple way to prevent neural networks from overfitting.” In: *The Journal of Machine Learning Research* 15 (2014).

[106] S.-Y. Yun and A. Proutière. “Accurate community detection in the Stochastic Block Model via spectral algorithms.” In: *arXiv preprint arXiv:1412.7335* (2014).

[107] S.-Y. Yun and A. Proutière. “Community Detection via random and adaptive sampling.” In: *Conference on Learning Theory.* 2014.

[108] W. Zaremba, I. Sutskever, and O. Vinyals. “Recurrent neural network regularization.” In: *arXiv preprint arXiv:1409.2329* (2014).

[109] J. Ba and B. Frey. “Adaptive Dropout for training deep neural networks.” In: *Advances in Neural Information Processing Systems.* 2013.

[110] P. Baldi and P. J. Sadowski. “Understanding Dropout.” In: *Advances in Neural Information Processing Systems.* 2013.

[111] J. Bochnak, M. Coste, and M.-F. Roy. *Real algebraic geometry.* Springer Science & Business Media, 2013.

[112] A. Dax. “From eigenvalues to singular values: a review.” In: *Advances in Pure Mathematics* 3 (2013).

[113] J. M. Lee. “Smooth manifolds.” In: *Introduction to Smooth Manifolds.* Springer, 2013.

[114] S. Wager, S. Wang, and P. S. Liang. “Dropout training as adaptive regularization.” In: *Advances in Neural Information Processing Systems.* 2013.

[115] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. “Regularization of neural networks using Dropconnect.” In: *International Conference on Machine Learning.* 2013.

[116] C. Bordenave, P. Caputo, and D. Chafaï. “Circular law theorem for random Markov matrices.” In: *Probability Theory and Related Fields* 152 (2012).

[117] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. “Improving neural networks by preventing co-adaptation of feature detectors.” In: *arXiv preprint arXiv:1207.0580* (2012).

[118] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks.” In: *Advances in Neural Information Processing Systems.* 2012.

[119] T. Tao. *Topics in random matrix theory.* American Mathematical Society, 2012.

[120] C. Bordenave, P. Caputo, and D. Chafaï. “Spectrum of large random reversible Markov chains: heavy-tailed weights on the complete graph.” In: *The Annals of Probability* 39 (2011).

[121] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. “Inference and phase transitions in the detection of modules in sparse networks.” In: *Physical Review Letters* 107 (2011).

[122] R. Tibshirani. “Regression shrinkage and selection via the lasso: a retrospective.” In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73 (2011).

[123] C. Bordenave, P. Caputo, and D. Chafaï. “Spectrum of large random reversible Markov chains: two examples.” In: *ALEA: Latin American Journal of Probability and Mathematical Statistics* 7 (2010).

[124] R. H. Keshavan, A. Montanari, and S. Oh. “Matrix completion from a few entries.” In: *IEEE Transactions on Information Theory* 56 (2010).

[125] Y. LeCun, C. Cortes, and C. Burges. “MNIST handwritten digit database.” In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010).

[126] S. Timotheou. “The random neural network: a survey.” In: *The Computer Journal* 53 (2010).

[127] V. S. Borkar. *Stochastic approximation: a dynamical systems viewpoint.* Springer, 2009.

[128] A. Krizhevsky. “Learning multiple layers of features from tiny images.” 2009.

[129] A. Rahimi and B. Recht. “Uniform approximation of functions with random bases.” In: *Allerton Conference on Communication, Control, and Computing.* 2008.

[130] A. Rahimi and B. Recht. “Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning.” In: *Advances in Neural Information Processing Systems.* 2008.

[131] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. “The graph neural network model.” In: *IEEE Transactions on Neural Networks* 20 (2008).

[132] J. Bolte, A. Daniilidis, A. S. Lewis, and M. Shiota. “Clarke critical values of subanalytic Lipschitz continuous functions.” In: *Annales Polonici Mathematici* 87 (2005).

[133] U. Feige and E. Ofek. “Spectral techniques applied to sparse random graphs.” In: *Random Structures & Algorithms* 27 (2005).

[134] S. Robin, F. Rodolphe, and S. Schbath. *DNA, words and models: statistics of exceptional words.* Cambridge University Press, 2005.

[135] D. Anderson and K. Burnham. “Model selection and multi-model inference.” In: *Second. NY: Springer-Verlag* (2004).

[136] K. Smith, L. Kahanpää, P. Kekäläinen, and W. Traves. *An invitation to algebraic geometry.* Springer Science & Business Media, 2004.

[137] A. Arvanitogeōrgos. *An introduction to Lie groups and the geometry of homogeneous spaces.* American Mathematical Society, 2003.

[138] P. Hajłasz. "Whitney's example by way of Assouad's embedding." In: *Proceedings of the American Mathematical Society* 131 (2003).

[139] M. Krivelevich and B. Sudakov. "The largest eigenvalue of sparse random graphs." In: *Combinatorics, Probability and Computing* 12 (2003).

[140] H. Kushner and G. G. Yin. *Stochastic approximation and recursive algorithms and applications.* Springer Science & Business Media, 2003.

[141] E. Gelenbe, Z.-H. Mao, and Y.-D. Li. "Function approximation with spiked random networks." In: *IEEE Transactions on Neural Networks* 10 (1999).

[142] E. Gelenbe, Z.-W. Mao, and Y.-D. Li. "Approximation by random networks with bounded number of layers." In: *IEEE Signal Processing Society Workshop.* IEEE. 1999.

[143] C. Manning and H. Schutze. *Foundations of statistical natural language processing.* MIT Press, 1999.

[144] A. Pinkus. "Approximation theory of the MLP model in neural networks." In: *Acta Numerica* 8 (1999).

[145] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86 (1998).

[146] Y. Makovoz. "Uniform approximation by neural networks." In: *Journal of Approximation Theory* 95 (1998).

[147] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-dynamic programming.* Athena Scientific, 1996.

[148] Y. Makovoz. "Random approximants and neural networks." In: *Journal of Approximation Theory* 85 (1996).

[149] D. P. Bertsekas and J. N. Tsitsiklis. "Neuro-dynamic programming: an overview." In: *IEEE Conference on Decision and Control.* 1995.

[150] B. Igelnik and Y.-H. Pao. "Stochastic choice of basis functions in adaptive function approximation and the Functional-Link Net." In: *IEEE Transactions on Neural Networks* 6 (1995).

[151] R. A. Horn and C. R. Johnson. *Topics in matrix analysis.* Cambridge university press, 1994.

[152] Y.-H. Pao, G.-H. Park, and D. J. Sobajic. "Learning and generalization characteristics of the random vector functional-link net." In: *Neurocomputing* 6 (1994).

[153] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function." In: *Neural Networks* 6 (1993).

[154] K. Hornik. "Approximation capabilities of multilayer feedforward networks." In: *Neural Networks* 4 (1991).

[155] T. Ando. "Majorization, doubly stochastic matrices, and comparison of eigenvalues." In: *Linear Algebra and its Applications* 118 (1989).

[156] G. Cybenko. "Approximation by superpositions of a sigmoidal function." In: *Mathematics of Control, Signals and Systems* 2 (1989).

[157] J. Friedman, J. Kahn, and E. Szemeredi. “On the second eigenvalue of random regular graphs.” In: *ACM Symposium on Theory of Computing*. 1989.

[158] L. Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition.” In: *Proceedings of the IEEE* 77 (1989).

[159] H. White. “An additional hidden unit test for neglected nonlinearity in multilayer feedforward networks.” In: *International Joint Conference on Neural Networks*. 1989.

[160] S. Hanson and L. Pratt. “Comparing biases for minimal network construction with back-propagation.” In: *Advances in Neural Information Processing Systems*. 1988.

[161] H. Bozdogan. “Model selection and Akaike’s Information Criterion (AIC): The general theory and its analytical extensions.” In: *Psychometrika* 52 (1987).

[162] H. Almagor. “A Markov analysis of DNA sequences.” In: *Journal of Theoretical Biology* 104 (1983).

[163] A. W. Marshall, I. Olkin, and B. C. Arnold. *Inequalities: Theory of majorization and its applications*. Springer, 1979.

[164] H. Akaike. “A new look at the statistical model identification.” In: *IEEE Transactions on Automatic Control* 19 (1974).

[165] V. A. Marčenko and L. A. Pastur. “Distribution of eigenvalues for some sets of random matrices.” In: *Mathematics of the USSR-Sbornik* 1 (1967).

[166] P. Erdos and A. Rényi. “On the evolution of random graphs.” In: *Publ. Math. Inst. Hung. Acad. Sci* 5 (1960).

[167] P. Erdös and A. Rényi. “On random graphs. I.” In: *Publ. Math. Debrecen* 6 (1959).

[168] E. P. Wigner. “On the distribution of the roots of certain symmetric matrices.” In: *Annals of Mathematics* (1958).

[169] T. W. Anderson and L. A. Goodman. “Statistical inference about Markov chains.” In: *The Annals of Mathematical Statistics* 28 (1957).

[170] J. Kiefer and J. Wolfowitz. “Stochastic estimation of the maximum of a regression function.” In: *The Annals of Mathematical Statistics* 23 (1952).

[171] H. Robbins and S. Monro. “A stochastic approximation method.” In: *The Annals of Mathematical Statistics* (1951).

[172] A. Sard. “The measure of the critical values of differentiable maps.” In: *Bulletin of the American Mathematical Society* 48 (1942).

[173] A. P. Morse. “The behavior of a function on its critical set.” In: *Annals of Mathematics* 40 (1939).

# Summary

In this thesis we investigate asymptotic properties of machine learning algorithms that are based on the use of structured random networks. We investigate *dropout*, an algorithm to avoid overfitting during the training of Neural Networks (NNs), and Block Markov Chains (BMCs), a type of models used for cluster detection in sequential data. Our main approach for studying both of these algorithms is theoretical, but an underlying aim of our work is to also provide results that may prove useful in practice. The thesis is divided into two parts.

The first part of this thesis focuses on *dropout*. Dropout is a well-known method for avoiding overfitting during the training of NNs that temporarily 'drops' nodes of the network during training with stochastic gradient descent. A *dropout* probability determines the amount of stochastic change of the NN and penalizes overfitting. Despite its ample use, precise statistical and convergence properties of dropout are not yet understood and a good choice of parameters is usually unknown.

In this thesis, we use stochastic approximation techniques to study and understand the convergence properties of dropout and its variants. We examine convergence guarantees for dropout and how the convergence rate depends on the choice of parameters of the NN, like depth, width, as well as the dropout probability. We obtain general convergence guarantees when training NNs with dropout as well as estimates on its sample complexity depending on the dropout probability. We investigate how the configuration of the NN depends on the convergence rate of training with dropout; theoretically, and with simulations. We obtain explicit descriptions of the convergence rate of dropout in different simplified models for deep and shallow NNs by analyzing an associated Ordinary Differential Equation (ODE) induced by dropout and by using different techniques from nonconvex optimization. Furthermore, we also study random NNs that have dropout as their source of randomness and show that they can approximate functions arbitrarily well, despite the additional randomness.

The second part of this thesis focuses on clustering with BMCs. A BMC is a model for clusters in sequential data where clusters are defined by states in a Markov chain that have a block structure of its transition probabilities. In this model, a trajectory of the Markov chain of a certain length is observed and the goal is to infer the underlying cluster structure. The transition probabilities in a BMC depend only on the cluster of origin and the target cluster of a transition, different to other well-known models in community detection where edges between nodes are sampled independently. Hence, in the random trajectory of a BMC, there are *dependencies* between consecutive transitions across different times. A clustering algorithm with theoretical guarantees for exact recovery of the clusters has

recently been obtained by Sanders, Proutière, and Yun [27]. A key condition in the recovery guarantee is to ensure that the spectral norm of a matrix describing the dynamics of the trajectory is small enough.

In this thesis, we use tools from random matrix theory and sparse random graphs to obtain order-sharp bounds of the spectral norm of BMCs. This involves using spectral concentration for matrices with dependent entries coming from a Markov chain. Moreover, we also tackle the case when the observed trajectory is short and the associated random matrix is sparse.

While clustering is theoretically possible on data generated by a BMCs, no experimental study of the model existed in the literature. We therefore test the clustering algorithm for BMCs in real-world sequential data and compare the bounds for the spectral norm from datasets in finance, genetics and geography. Furthermore, we evaluate the robustness of the clustering algorithm and the merit of BMCs compared to simpler models for each dataset.

# About the author

Albert Senén Cerdà was born in Benicarló in 1992. He obtained his high-school diploma at I.E.S. Ramón Cid in Benicarló in 2010. Albert obtained both the Bachelor degree in Mathematics as well as in Physics at Autonomous University of Barcelona in 2015. During his bachelor studies he received the Pere Menal Fellowship, and he also spent time at Max Planck Institute for Physics in Munich as an intern. After his bachelor studies, he went on to pursue a Master degree in Mathematics at the University of Göttingen, Germany. He graduated in 2018 with the thesis *Topological aspects of Nahm's equation.*

In 2018, Albert started a PhD in applied mathematics in the department of mathematics, computer science and electrical engineering at Delft University of Technology (TU Delft) in the Netherlands, under the supervision of dr.ir. Jaron Sanders. In 2019, Albert joined his supervisor now at the Stochastic Operations Research (SOR) group in the mathematics and computer science department at Eindhoven University of Technology (TU/e). His PhD deals with stochastic optimization applied to problems in machine learning, operations research, and probability. During his PhD, Albert visited the *Laboratoire d'analyse et d'architecture des systèmes* (LAAS-CNRS) in Toulouse, France to work with Dr. Matthieu Jonckheere.

During his PhD, Albert has presented his work in several workshops and conferences in the Netherlands and abroad; for example, in the YEQT workshop at EURANDOM and the ACM SIGMETRICS conference. He has also been teaching assistant for courses in Data Communications Networks, Calculus, and Statistical Learning Theory, for which he was recipient of an excellent teaching evaluation award. Albert has also been supervisor of Rens Hoogendorp, a BSc student, and has co-organized the 8th SOR PhD Colloquium at TU/e.

Albert will defend his PhD thesis at TU/e on May 15, 2023.