Imperial College London

BioMedIA

Advancing probabilistic and causal deep learning in medical image analysis

Author Miguel Monteiro

Submitted December 7th 2022

Supervisors Prof. Ben Glocker[1]

Prof. Daniel Rueckert[1]

[1]Department of Computing, Imperial College London, UK.

A thesis submitted in fulfilment of the requirements for the degree of *Philosophiae Doctor*

Biomedical Image Analysis Group, Department of Computing,

Imperial College London

Biomedical Image Analysis Group,
Department of Computing,
Huxley Building,
Imperial College London,
180 Queens Gate,
South Kensington,
SW7 2AZ

To my family, friends, and those who may find this document helpful.

Declaration

I, Miguel Monteiro, hereby declare that this thesis is my own work unless otherwise specified. All published papers that are reproduced in this thesis are appropriately referenced and the reproduction adheres to the relevant copyright licenses. The thesis template has been adapted from Rob Robinson (Robinson 2020), who made his template available on his Github https://github.com/mlnotebook.

Copyright

Acknowledgements

First and foremost, I would like to thank my parents, Raquel and Gabriel, and my siblings, Vera and Vasco, for all the love and support, without which I would not have been able to complete or even start this journey.

I want to express my gratitude to my supervisor, Professor Ben Glocker, who believed in me and offered me the incredible opportunity to work on cutting-edge research projects in such a creative and dynamic environment. His support, guidance and labour have been endless and instrumental throughout the PhD.

I would like to thank all of my prodigiously-talented co-authors, Virginia Newcombe, Francois Mathieu, David K Menon, Konstantinos Kamnitsas, Loïc Le Folgoc, Daniel Coelho de Castro, Nick Pawlowski, Mark van der Wilk, Fabio Ribeiro de Sousa, whose contributions and guidance were indispensable for the work here presented. I would also like to acknowledge all my colleagues at BioMedIA, old and new, for all the memories and for making coming into the office something to look forward to.

I also want to thank Professor Arlindo Oliveira for giving the opportunity to do machine learning research for the first time and putting me on track to be able to pursue a PhD.

Lastly, I would like to thank all my friends and family at home and in London for being there in the good times and the bad, making getting a PhD a lot less painful and a lot more enjoyable.

During my research I have received support by the European Union 7th Framework Programme (EC grant 602150) as part of the CENTER-TBI study. I have also been supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 757173, Project MIRA).

Abstract

The power and flexibility of deep learning have made it an indispensable tool for tackling modern machine learning problems. However, this flexibility comes at the cost of robustness and interpretability, which can lead to undesirable or even harmful outcomes. Deep learning models often fail to generalise to real-world conditions and produce unforeseen errors that hinder wide adoption in safety-critical critical domains such as healthcare. This thesis presents multiple works that address the reliability problems of deep learning in safety-critical domains by being aware of its vulnerabilities and incorporating more domain knowledge when designing and evaluating our algorithms. We start by showing how close collaboration with domain experts is necessary to achieve good results in a real-world clinical task - the multiclass semantic segmentation of traumatic brain injuries (TBI) lesions in head CT. We continue by proposing an algorithm that models spatially coherent aleatoric uncertainty in segmentation tasks by considering the dependencies between pixels. The lack of proper uncertainty quantification is a robustness issue which is ubiquitous in deep learning. Tackling this issue is of the utmost importance if we want to deploy these systems in the real world. Lastly, we present a general framework for evaluating image counterfactual inference models in the absence of ground-truth counterfactuals. Counterfactuals are extremely useful to reason about models and data and to probe models for explanations or mistakes. As a result, their evaluation is critical for improving the interpretability of deep learning models.

Contents

1 Introduction **15**
- 1.1 Motivation . . . 15
- 1.2 Research aims and thesis outline . . . 17
- 1.3 List of publications . . . 19

2 Background **21**
- 2.1 Probabilistic Modelling . . . 21
  - 2.1.1 Unconditional Models . . . 21
  - 2.1.2 Conditional Models . . . 24
- 2.2 Deep Learning . . . 26
- 2.3 Deep Latent Variable Generative Models . . . 29
- 2.4 Structural Causal Models . . . 31

3 Multiclass semantic segmentation and quantification of traumatic brain injury lesions on head CT using deep learning **33**
- 3.1 Introduction . . . 33
- 3.2 Materials and Methods . . . 35
  - 3.2.1 Study design and participants . . . 35
  - 3.2.2 Procedures . . . 35
  - 3.2.3 Image pre-processing, model and training . . . 36
  - 3.2.4 Statistical analysis . . . 37
- 3.3 Results . . . 40
- 3.4 Discussion . . . 43

4 Stochastic Segmentation Networks: modelling spatially correlated aleatoric uncertainty **48**
- 4.1 Introduction . . . 48
- 4.2 Related Work . . . 49
- 4.3 Methods . . . 51
  - 4.3.1 Background . . . 51
  - 4.3.2 Stochastic segmentation networks . . . 52
- 4.4 Results . . . 53
  - 4.4.1 Toy problem . . . 53
  - 4.4.2 Lung nodule segmentation in 2D . . . 54
  - 4.4.3 Brain tumour segmentation in 3D . . . 57
  - 4.4.4 Rank ablation study . . . 60
  - 4.4.5 Application to generative models . . . 61
- 4.5 Discussion . . . 63

5 Measuring axiomatic soundness of counterfactual image models **65**
- 5.1 Introduction . . . 65
- 5.2 Related Work . . . 66

5.3 Methods . . . 67
5.3.1 Counterfactual Functions . . . 68
5.3.1.1 Function properties from axiomatic definition of counterfactuals 69
5.3.1.2 Partial counterfactual functions . . . 69
5.3.1.3 Partial counterfactual function decomposition . . . 70
5.3.1.4 Learning partial counterfactual functions from data . . . 70
5.3.2 Measuring soundness of counterfactuals . . . 71
5.3.3 Simulated Interventions . . . 72
5.3.4 Deep generative models as approximate counterfactual functions . . . 73
5.4 Results . . . 74
5.4.1 Colour MNIST . . . 74
5.4.2 3D Shapes . . . 78
5.4.3 CelebA-HQ . . . 81
5.5 Discussion . . . 88

6 Conclusion **89**
6.1 Summary of contributions . . . 89
6.2 Limitations and Future Research . . . 90
6.2.1 Granularity and accuracy of automatic traumatic brain injury segmentation 90
6.2.2 Local and global spatially correlated uncertainty . . . 91
6.2.3 Effectiveness metric depends on data and lack of full support for the observational distribution . . . 92

Bibliography **93**

A Supplementary material for segmentation of traumatic brain injury **104**
A.1 Absolute volume error . . . 104
A.2 Lesion localisation . . . 104

B Supplementary material for Stochastic Segmentation Networks **111**

C Supplementary material for axiomatic quality of counterfactuals **116**
C.1 Reversibility proof . . . 116
C.2 Distance metrics . . . 116

List of Figures

2.1 Example of a Markovian SCM. . . . 32

3.1 Qualitative multiclass segmentation results. . . . 40
3.2 Per-class boxplots of DSC stratified by volume threshold. . . . 41
3.3 Bland-Altman plots for lesion volume estimation. . . . 43
3.4 Bland-Altman plots for lesion progression. . . . 44
3.5 Receiver operating characteristic curves for lesion detection and classification. . . . 45

4.1 Probabilistic graphical model for a two-pixel segmentation problem. . . . 52
4.2 Toy problem. . . . 54
4.3 Qualitative results on the LIDC-IDRI dataset. . . . 55
4.4 Qualitative results on the BraTS 2017 dataset. . . . 58
4.5 Distribution of sample average class DSC per case. . . . 59
4.6 Sample manipulation after inference. . . . 59
4.7 Impact of rank on different performance metrics for the BraTS dataset. . . . 60
4.8 Visual impact of rank on samples for one case. . . . 61
4.9 Samples from a standard VAE and from a VAE with a spatially coherent observational distribution. . . . 63

5.1 Process of making a parent independent of the remainder using twin network SCMs and interventions. . . . 70
5.2 Colour MNIST joint distribution of digit and hue for different SCMs. . . . 75
5.3 Colour MNIST samples. . . . 76
5.4 Soundness tests on colour MNIST for a disentangled model and a entangled model. 77
5.5 Effectiveness/Composition tests on 3D shapes for VAE and GAN. . . . 80
5.6 Measuring CelebA-HQ test set effectiveness, composition and reversibility as a function of the number of latent variables abducted. . . . 83
5.7 Effectiveness/Composition plots for CelebA-HQ . . . 85
5.8 Evolution of 'smiling' CelebA-HQ counterfactuals with number of latents abducted. 86
5.9 Evolution of 'eyeglasses' CelebA-HQ counterfactuals with number of latents abducted. . . . 87

A.1 Data flow diagram for model development and validation. . . . 105
A.2 Per-class Bland-Altman plots for intra-rater reproducibility. . . . 107
A.3 Per-class Bland-Altman plot inter-rater variability. . . . 107
A.4 Per-class boxplots of the absolute error for different volume groups. . . . 108
A.5 Per-class boxplots of the absolute error between the true lesion volume change and the predicted lesion volume change for different volume groups. . . . 108
A.6 Lesion localisation example. . . . 110

B.1 Results of sampling from the proposed stochastic model (1/4). . . . 112
B.2 Results of sampling from the proposed stochastic model (2/4). . . . 113
B.3 Results of sampling from the proposed stochastic model (3/4). . . . 114

B.4 Results of sampling from the proposed stochastic model (4/4). 115

List of Tables

3.1 Cohort details for both datasets. . . . 39
3.2 Evaluation Metrics stratified by volume threshold. . . . 42
3.3 Multiclass detection and classification results for three volume thresholds and detection results for the external validation dataset CQ500. . . . 46

4.1 Quantitative results on the LIDC-IDRI dataset. . . . 56
4.2 Quantitative results on the BraTS 2017 dataset. . . . 57

5.1 Soundness metrics on colour MNIST. . . . 76

A.1 Distribution of lesion presence and volume for the train and test set. . . . 106
A.2 Intra-class correlation coefficient (ICC) for intra-rater reproducibility and inter-rater variability. . . . 106
A.3 Evaluation metrics stratified by volume group. . . . 109

C.1 Effectiveness results on colour MNIST when using pseudo oracles trained from biased data without using a simulated intervention. . . . 123
C.2 Effectiveness results on colour MNIST when using pseudo oracles trained from biased data using a simulated intervention. . . . 123
C.3 Effectiveness on colour MNIST when using linear/logistic regression as pseudo oracles trained from unbiased data. . . . 124
C.4 Results of quality tests on 3D shapes. . . . 124
C.5 Quality metrics on the CelebA-HQ. . . . 125

List of Source Code

C.1 VAE architecture for the colour MNIST dataset. 118
C.2 GAN architecture for the colour MNIST dataset. 119
C.3 Pseudo-oracle architecture for the colour MNIST and 3D shapes datasets. 120
C.4 VAE architecture for the 3D shapes dataset. 120
C.5 GAN architecture for the 3D shapes dataset. 121
C.6 Pseudo-oracle architecture for the CelebA-HQ dataset. 122

Chapter 1

Introduction

1.1 Motivation

Machine learning provides a collection of algorithms that can learn to perform a task from training data and make predictions on unseen test data without being explicitly programmed. Among these algorithms, deep learning has seen an explosion in commercial and research applications due to its incredible power and flexibility. Some notable applications include natural image classification of large and diverse datasets (Krizhevsky et al. 2017; Simonyan and Zisserman 2014), playing board and video games at superhuman levels (V. Mnih et al. 2015; Silver et al. 2018), voice assistants based on speech recognition (Sainath et al. 2015; Saon et al. 2013), machine translation (Bahdanau et al. 2014; Y. Wu et al. 2016), object detection for vision of autonomous vehicles (Redmon et al. 2016), predicting the structure of proteins for drug discovery (Jumper et al. 2021), among others.

In the healthcare space, machine learning models have the potential to address important societal challenges, such as the deterioration of patient care due to ageing populations and medical workforce shortages, lack of access to healthcare experts due to poor socio-economic conditions, the diagnosis of under-studied rare diseases, drug discovery, or the development of surgical robots and tools (Esteva, Robicquet, et al. 2019). By leveraging large medical datasets, we can combine medical knowledge from countless medical professionals into an autonomous algorithm capable of leveraging its large knowledge base without requiring human intervention. This autonomy can be used to pre-screen patients, give fast second opinions, and generally free up doctors' time, thus expanding access to healthcare overall. In medical imaging, deep learning algorithms have achieved physician-level accuracy in various diagnostic tasks, such as diagnosing diabetic retinopathy from iris fundus images (Gulshan et al. 2016) and skin lesions from photographs (Esteva, Kuprel, et al. 2017). Furthermore, they can significantly alleviate a radiologist's workload by automating time-consuming and repetitive tasks such as semantic segmentation of 3D images (Bakas, Akbari, et al. 2017; Bakas, Reyes, et al. 2018).

Despite its potential, if misused, deep learning can have unintended consequences, often aggravating the problems it proposes to solve. First, a close multi-disciplinary collaboration between clinical and computer science experts is necessary to ensure that we tackle the correct problems and do not waste valuable resources (Roberts et al. 2021; Wiens et al. 2019). Second, we must be

aware of a critical vulnerability in deep learning that can exacerbate undesirable biases and create fragile systems with multiple ethical problems (Wiens et al. 2019).

Deep learning often behaves like an evil genie, giving us exactly what we asked for instead of what we wanted. At the core of the problem is the fact that deep learning is a greedy pattern-matching algorithm which always tries to find the easiest solution to a problem, which seldomly matches the human or even logical solution. Deep learning methods often pick up on spurious correlations in the training data as the basis for their decisions. When leveraging these spurious correlations is the quickest solution to the problem, the algorithm will use them even though they do not match the true causal nature of the system we are trying to model. This problem is known as shortcut learning (Geirhos et al. 2020). For example, suppose we want to detect a specific pathology in medical scans, but our healthy and pathological cases come from different scanners. A deep learning algorithm would likely learn to separate the cases using the scanner type as a proxy for pathology since these properties provide an easier path towards the goal we have set. While this is not what we wanted, it is, perhaps without realising, what we asked for. In the medical domain, this problem can manifest itself in other extremely nefarious ways, such as amplifying racial or gender biases (Adamson and Smith 2018; Buolamwini and Gebru 2018; Obermeyer et al. 2019), lack of generalisation between different hospitals due to dataset shift (Quinonero-Candela et al. 2008), and over-reliance on opaque and flawed algorithms for critical decisions on patient care (Roberts et al. 2021).

In addition to the problem of shortcut learning, deep learning also suffers from a lack of interpretability and over-confident predictions. These issues often lead to obvious yet hard-to-diagnose and correct errors, which are unacceptable in high-risk applications such as medicine. We argue that the solution to the problem is incorporating as much domain knowledge as possible into our deep learning models instead of expecting them to learn from data. This knowledge, also known as inductive biases or priors, can improve robustness and interpretability and mitigate the effect of nefarious biases. Essentially, we want our model to match the data-generation process as closely as possible. We do not expect the data to contain all the information necessary, and we are happy to build part of the solution into the algorithm if we are confident that such a prior is correct.

One crucial component of any good probabilistic model is proper uncertainty quantification. A model must give us not only an accurate prediction but also an accurate estimate of its uncertainty about the prediction. This ability allows us to know when the model is unsure of itself and use that information accordingly when using the model's prediction to make decisions. Therefore, reliable uncertainty estimates are necessary for any serious deep learning application. In reality, good uncertainty quantification is achieved by simply choosing a probabilistic model whose assumptions closely match the data-generation process. However, convenience is often put above rigour, resulting in poor and sometimes incorrect uncertainty estimates.

Another beneficial way of incorporating domain knowledge into our models is incorporating our

causal knowledge about the system into our algorithms (Pearl 2009). By telling a model which variables are causes and which are effects from the start instead of expecting it to be learnt from data, we bias our model towards a more accurate understanding of the world. These strong priors mitigate the effect of spurious correlations and nefarious biases, which can often be present in observational data. In summary, we are telling our models that there is more to a system than the empirical correlations observed in the training data. There is an underlying causal mechanism we want to model, and our observations are potentially contaminated by unwanted confounding and spurious correlations. Additionally, causal models allow us to ask counterfactual queries, aka 'what-if' questions. These questions allow us to gain a deeper understanding of our model's inner workings or even of the world that it is modelling.

1.2 Research aims and thesis outline

Motivated by the above, this thesis presents multiple works that address the reliability problems we face when deploying deep learning systems in safety-critical domains by being aware of their vulnerabilities and incorporating more domain knowledge into the design and evaluation of our algorithms. Specifically, we touch on the following topics:

- **Collaboration with domain-experts.** We show how close collaboration with domain experts is necessary to successfully deploy deep learning tools in a challenging medical application;
- **Uncertainty quantification.** We demonstrate how more accurate assumptions about the data-generation process can lead to better uncertainty quantification and hence better models;
- **Evaluation of counterfactual models.** We develop a framework to measure how closely our algorithms are modelling the true causal nature of the data-generation process as opposed to simply taking advantage of spurious correlations.

In Chapter 2, we briefly introduce the topics necessary to understand the following chapters and frame the research into a larger context. We start by reviewing classic unconditional and conditional probabilistic models. We continue by touching upon the basic building blocks of deep learning algorithms. After, we revisit the use of deep learning for latent variable generative models. We finish by reviewing structural causal models.

In Chapter 3, we develop an algorithm for the multiclass semantic segmentation of traumatic brain injuries (TBI) lesions in head CT. We demonstrate how close collaboration with medical experts and meticulous development is necessary to successfully deploy deep learning tools in clinical applications. TBI is one of the leading causes of mortality worldwide, and CT is the imaging modality of choice for quick assessment of lesion presence and load. However, its conventional use requires expert clinical interpretation and does not provide detailed quantitative outputs. Our algorithm has the ability to separately segment, quantify, and detect multiclass haemorrhagic lesions and perile-

sional oedema. These volumetric lesion estimates allow clinically relevant quantification of lesion burden and progression, with potential applications for personalised treatment strategies.

- **Monteiro, M.**, Newcombe, V. F., Mathieu, F., Adatia, K., Kamnitsas, K., Ferrante, E., Das, T., Whitehouse, D., Rueckert, D., Menon, D. K., et al. (2020). "Multiclass semantic segmentation and quantification of traumatic brain injury lesions on head CT using deep learning: an algorithm development and multicentre validation study". In: *The Lancet Digital Health* 2.6, e314–e322

In Chapter 4, we propose stochastic segmentation networks (SSNs) – an algorithm that models spatially coherent aleatoric uncertainty in segmentation tasks by taking into account the dependencies between pixels. Proper uncertainty quantification is essential for improving the interpretability and robustness of deep learning algorithms. In medical imaging, for example, experts will often disagree about the presence of an object or the exact location of its boundaries. An accurate model of the data-generation process would therefore be able to represent this inherent uncertainty and predict multiple plausible hypotheses for the same input. In sensitive applications, having good uncertainty estimates is not only a sign of a good model of the data but also a strict requirement for informing critical decision-making. In contrast to approaches that produce pixel-wise estimates, SSNs model joint distributions over entire label maps and thus can generate multiple spatially coherent hypotheses for a single image – more accurately modelling the data-generation process. We eliminate the necessity for the independence assumption between pixels by using a low-rank multivariate normal distribution over the logit space to model the probability of the label map given the image. We obtain a spatially consistent probability distribution that can be efficiently computed by a neural network without any changes to the underlying architecture. We tested our method on the segmentation of real-world medical data, including lung nodules in 2D CT and brain tumours in 3D multimodal MRI scans. SSNs outperform state-of-the-art for modelling correlated uncertainty in ambiguous images while being much simpler, more flexible, and more efficient.

- **Monteiro, M.**, Le Folgoc, L., Coelho de Castro, D., Pawlowski, N., Marques, B., Kamnitsas, K., Wilk, M. van der, and Glocker, B. (2020). "Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty". In: *Advances in Neural Information Processing Systems* 33, pp. 12756–12767

In Chapter 5, we present a general framework for evaluating image counterfactuals. Counterfactuals are 'what-if' questions that allow us to better understand the world by observing fictitious branches of reality. In the context of deep learning, we can use counterfactuals to probe a model for interpretable explanations or as data-augmentation in downstream tasks to correct for nefarious biases. The power and flexibility of deep generative models make them valuable tools for learning mechanisms in structural causal models. However, their flexibility makes counterfactual identifiability impossible in the general case. Motivated by these issues, we revisit Pearl's axiomatic definition of counterfactuals to determine the necessary constraints of any counterfactual infer-

ence model: composition, reversibility, and effectiveness (Galles and Pearl 1998; Pearl 2009). We frame counterfactuals as functions of an input variable, its parents, and counterfactual parents and use the axiomatic constraints to restrict the set of functions that could represent the counterfactual, thus deriving distance metrics between the approximate and ideal functions. We demonstrate how these metrics can be used to compare and choose between different approximate models and to provide insight into a model's shortcomings and trade-offs.

Lastly, Chapter 6 summarises the contributions of the thesis, points out their limitations and suggests avenues for future work.

1.3 List of publications

During my PhD I have led or contributed to the following publications:

- **Monteiro, M.**, Kamnitsas, K., Ferrante, E., Mathieu, F., McDonagh, S., Cook, S., Stevenson, S., Das, T., Khetani, A., Newman, T., et al. (2019). "Tbi lesion segmentation in head ct: Impact of preprocessing and data augmentation". In: *International MICCAI Brainlesion Workshop*, pp. 13–22
- **Monteiro, M.**, Newcombe, V. F., Mathieu, F., Adatia, K., Kamnitsas, K., Ferrante, E., Das, T., Whitehouse, D., Rueckert, D., Menon, D. K., et al. (2020). "Multiclass semantic segmentation and quantification of traumatic brain injury lesions on head CT using deep learning: an algorithm development and multicentre validation study". In: *The Lancet Digital Health* 2.6, e314–e322
- Mathieu, F., Güting, H., Gravesteijn, B., **Monteiro, M.**, Glocker, B., Kornaropoulos, E. N., Kamnistas, K., Robertson, C. S., Levin, H., Whitehouse, D. P., et al. (2020). "Impact of antithrombotic agents on radiological lesion progression in acute traumatic brain injury: a CENTER-TBI propensity-matched cohort analysis". In: *Journal of neurotrauma* 37.19, pp. 2069–2080
- Mathieu, F., Zeiler, F. A., Ercole, A., **Monteiro, M.**, Kamnitsas, K., Glocker, B., Whitehouse, D. P., Das, T., Smielewski, P., Czosnyka, M., et al. (2020). "Relationship between measures of cerebrovascular reactivity and intracranial lesion progression in acute traumatic brain injury patients: a CENTER-TBI study". In: *Journal of Neurotrauma* 37.13, pp. 1556–1565
- Zeiler, F. A., Mathieu, F., **Monteiro, M.**, Glocker, B., Ercole, A., Beqiri, E., Cabeleira, M., Stocchetti, N., Smielewski, P., Czosnyka, M., et al. (2020). "Diffuse intracranial injury patterns are associated with impaired cerebrovascular reactivity in adult traumatic brain injury: a CENTER-TBI Validation Study". In: *Journal of Neurotrauma* 37.14, pp. 1597–1608
- **Monteiro, M.**, Le Folgoc, L., Coelho de Castro, D., Pawlowski, N., Marques, B., Kamnitsas, K., Wilk, M. van der, and Glocker, B. (2020). "Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty". In: *Advances in Neural Information Processing Systems*

33, pp. 12756–12767

- Zeiler, F. A., Mathieu, F., **Monteiro, M.**, Glocker, B., Ercole, A., Cabeleira, M., Stocchetti, N., Smielewski, P., Czosnyka, M., Newcombe, V., et al. (2021). "Systemic Markers of Injury and Injury Response Are Not Associated with Impaired Cerebrovascular Reactivity in Adult Traumatic Brain Injury: A Collaborative European Neurotrauma Effectiveness Research in Traumatic Brain Injury (CENTER-TBI) Study". In: *Journal of Neurotrauma* 38.7, pp. 870–878
- Whitehouse, D. P., **Monteiro, M.**, Czeiter, E., Vyvere, T. V., Valerio, F., Ye, Z., Amrein, K., Kamnitsas, K., Xu, H., Yang, Z., et al. (2022). "Relationship of admission blood proteomic biomarkers levels to lesion type and lesion burden in traumatic brain injury: A CENTER-TBI study". In: *EBioMedicine* 75, p. 103777
- Bernhardt, M., Castro, D. C., Tanno, R., Schwaighofer, A., Tezcan, K. C., **Monteiro, M.**, Bannur, S., Lungren, M. P., Nori, A., Glocker, B., et al. (2022). "Active label cleaning for improved dataset quality under resource constraints". In: *Nature communications* 13.1, pp. 1–11
- Rosnati, M., Soreq, E., **Monteiro, M.**, Li, L., Graham, N. S., Zimmerman, K., Rossi, C., Carrara, G., Bertolini, G., Sharp, D. J., et al. (2022). "Automatic lesion analysis for increased efficiency in outcome prediction of traumatic brain injury". In: *International Workshop on Machine Learning in Clinical Neuroimaging*, pp. 135–146
- Langley, J., **Monteiro, M.**, Jones, C., Pawlowski, N., and Glocker, B. (2022). "Structured Uncertainty in the Observation Space of Variational Autoencoders". In: *Transactions on Machine Learning Research*

Chapter 2

Background

2.1 Probabilistic Modelling

Probability theory stands at the core of all machine learning algorithms. This framework allows us to model the probability of random events, given our assumptions about the world and observational data. We start by looking at unconditional models using parametric distributions from the exponential family. We will use these as building blocks for more complex algorithms such as conditional linear models and deep neural networks. In this section, we give a brief overview of the topic. For an in-depth treatment see Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. Vol. 4. 4. Springer.

2.1.1 Unconditional Models

Coin flip. Suppose we have a coin and wish to determine whether it is a fair coin – meaning the probability of getting the two faces is the same. To build a probabilistic model of this event, we introduce a random variable X representing the outcome of the coin flip, taking the value 0 for tails or 1 for heads. Next, we must decide what class of probability distributions best describes the data-generating process. The Bernoulli distribution is well-suited since the models of the outcome of a discrete variable with only two possible outcomes. The probability mass function of the Bernoulli distribution is given by:

$$P(X = x) = p^x(1 - p)^{(1-x)}, \tag{2.1}$$

where the parameter p represents the probability of the outcome being heads ($X = 1$).

Having decided on a model, we proceed by collecting a set of N observations $x_1, \ldots, x_N$ corresponding to the outcomes of independent coin flips. Our aim is to estimate the parameter p given the observations. To do this, we can use maximum likelihood estimation (MLE), which aims to find the value of the parameter that maximises the likelihood of all observations:

$$\arg\max_p \prod_{n=1}^{N} P(X = x_n). \tag{2.2}$$

Using the log-likelihood for convenience of calculation, we obtain the following optimisation objec-

tive:

$$\arg\max_p \log \prod_{n=1}^{N} P(X = x_n) = \arg\max_p \sum_{n=1}^{N} \log P(X = x_n), \tag{2.3}$$

which we can solve by taking the derivative with respect to the parameter p and finding the value of p for which the derivative is zero – the point of maximum log-likelihood:

$$\begin{aligned} \frac{\partial}{\partial p} \sum_{n=1}^{N} \log P(X = x_n) &= \frac{d}{d\pi} \sum_{n=1}^{N} (x_n \log(p) + (1 - x_n) \log(1 - p)) \\ &= \frac{\sum_{n=1}^{N} x_n}{p} - \frac{\sum_{n=1}^{N} 1 - x_n}{1 - p} \\ p &= \frac{1}{N} \sum_{n=1}^{N} x_n. \end{aligned} \tag{2.4}$$

Now that we have estimated the probability of obtaining heads, we can determine whether the coin is fair ($p = 0.5$). Note that p matches the expected value of the observed values $\mathbb{E}_{x \sim X}[x]$. Thus, calculating p is equivalent to counting the number of heads and dividing by the total number of flips.

Loaded dice. Suppose we have an M face dice and wish to determine whether it is loaded – the probability of all faces is not the same. We introduce a random variable X representing the number on the face of the dice taking discrete integer values from 1 to M. We use a categorical distribution since it models the probability distribution of a discrete variable with multiple possible outcomes. The probability mass function of the categorical distribution is:

$$P(X = x) = \prod_{i}^{M} p_i^{[x=i]}, \tag{2.5}$$

where p_i are the parameters of the distribution corresponding to the probability of obtaining a specific outcome.

Given a set of N observations $x_1, \ldots, x_N$, using maximum log-likelihood estimation, we obtain the following optimisation objective:

$$\arg\max_{p_1, \ldots, p_M} \sum_{n=1}^{N} \log P(X = x_n). \tag{2.6}$$

Taking the partial derivative with respect to a specific parameter p_j and noting that the categorical

probabilities must sum to 1, we obtain.

$$\begin{aligned}\frac{\partial}{\partial p_j}\sum_{n=1}^{N}\log P(x=x_n) &= \sum_{n=1}^{N}\log\left(p_j^{[x_n=j]}(1-p_j)^{[x_n\neq j]}\right)\\ &= \frac{\sum_{n=1}^{N}[x_n=j]}{p_j} - \frac{\sum_{n=1}^{N}x_n\neq j}{1-p_j}\\ p_j &= \frac{1}{N}\sum_{n=1}^{N}[x_n=j].\end{aligned} \tag{2.7}$$

We can now calculate the probability of obtaining a given face and determine whether the dice is fair ($p_i = 1/M, \forall i \in [1, \dots M]$). This result is familiar from the coin flip example because the categorical distribution is merely an extension of the Bernoulli distribution for the case where there are more than two possible outcomes.

Javelin throw. Suppose we want a model for the distance of javelin throw amongst Olympic athletes. Unlike the coin flip scenario, now we have a continuous variable X which can take values in $\mathbb{R}^+$. Thus, we opt to model the distribution using a normal distribution, where the probability density function is given by:

$$P(X=x) = \frac{1}{\sqrt{(2\pi\sigma^2)}}\exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right), \tag{2.8}$$

where μ is the mean of the distribution, and σ^2 is its variance.

Give a set of observations $x_1, \dots, x_N$, using maximum likelihood estimation, we obtain the following optimisation problem:

$$\arg\max_{\mu,\sigma^2}\sum_{n=1}^{N}\log P(X=x_n). \tag{2.9}$$

Taking the partial derivative with respect to mean μ we obtain the first-order condition:

$$\frac{\partial}{\partial\mu}\sum_{n=1}^{N}\log P(X=x_n) = -\frac{1}{\sigma^2}\sum_{n=1}^{N}(x_n+\mu), \tag{2.10}$$

which is equal to zero when:

$$\mu = \frac{1}{N}\sum_{n=1}^{N}x_n. \tag{2.11}$$

Following the same procedure for the variance σ^2, we obtain:

$$\frac{\partial}{\partial\sigma^2}\sum_{n=1}^{N}\log P(X=x_n) = \frac{1}{2\sigma^2}\left(\frac{1}{\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - N\right), \tag{2.12}$$

which is, assuming $\sigma^2 > 0$, is zero when:

$$\sigma^2 = \frac{1}{N}\sum(x_n - \mu)^2. \tag{2.13}$$

Having estimated the parameters of the normal distribution, we can now use the cumulative probability distribution to estimate the probability that a throw comes from an Olympic athlete, a normal person or a demi-god.

2.1.2 Conditional Models

So far, we have dealt with examples where the variable of interest does not depend on any other observed variables. We now consider the case where there is a target variable Y, which has a linear dependency on a variable X. We also consider generalised linear models, where a (non-parameterised) link function provides the relationship between the linear predictor and the parameters of the likelihood.

Linear regression. Going back to the javelin throw example, suppose we measure a set of M bio-mechanical features $x = (1, x_1, \ldots, x_M)$ for each athlete in addition to the distance of their throw y. Notice that the subscript now indexes features and not data samples and that we have introduced a constant bias feature in x. We want to obtain the conditional model $P(Y = y | X = x)$ of the distance of a throw given said features. One simple way to do this, is to assume a linear relationship between the features and the target variable, and a zero-mean additive Gaussian noise model. The predictor $\hat{y}(x;\theta)$ is given by:

$$\hat{y}(x;\theta) = x^T\theta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{2.14}$$

where $\theta = (\theta_0, \theta_1, \ldots, \theta_M)$ are the parameters of the model and σ^2 is the variance of the additive noise. Using the Gaussian noise model we obtain a Gaussian likelihood for the observations:

$$P(Y = y | X = x) = \mathcal{N}(\hat{y}(x;\theta), \sigma^2). \tag{2.15}$$

This allows us to use maximum likelihood estimation to obtain the values of the parameters θ. Given a matrix of observations $\boldsymbol{X}$ where each row is a sample and each column corresponds to a feature, and a column matrix $\boldsymbol{Y}$ with the target values for each sample, optimising the log-likelihood with respect to θ yields the following solution:

$$\arg\max_{\theta} P(Y = \boldsymbol{Y} | X = \boldsymbol{X}) = (\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}^T\boldsymbol{Y}. \tag{2.16}$$

We see how we can obtain a close-form solution of the parameters of the conditional model, allowing us to make predictions about Y when given observations of X.

Logistic Regression and Binary Classification. Linear regression and the Gaussian likelihood are not appropriate for discrete variables. For a discrete binary variable, we would like to *classify* a data-point into one of two classes, given its observed features. To do this, we use logistic regression, the generalised linear model for discrete binary variables.

As we have seen in Section 2.1.1, the Bernoulli likelihood is well-suited for discrete binary variables. However, the unbound codomain of the affine transformation in a linear regression model is incompatible with the constraint that the probability parameter of the Bernoulli distribution must be in the range of 0 to 1. To solve this problem, we use the sigmoid link function to bound the output of the affine transformation, which gives us the following predictor:

$$\hat{y}(x;\theta) = \text{sigmoid}(x^T\theta) = \frac{1}{1+e^{-x^T\theta}} \tag{2.17}$$

We can now use the Bernoulli log-likelihood to obtain the conditional model:

$$\begin{aligned}\log P(Y=y|X=x) &= \log \text{Bernoulli}(\hat{y}(x;\theta)) \\ &= y\log(\hat{y}(x;\theta)) + (1-y)\log(1-\hat{y}(x;\theta))\end{aligned} \tag{2.18}$$

Since $\hat{y}(x;\theta)$ is bound between 0 and 1, we can interpret it as the conditional probability of a positive outcome given the input features. We can then threshold its value to separate the positive and negative classes. Unlike in linear regression, there is no closed-form expression for the maximum likelihood estimate of the parameters. Hence, we must resort to numerical optimisation methods such as Newton's method.

Multinomial Logistic Regression Multi-Class Classification The extension of logistic regression to the multi-class problem is called multinomial logistic regression. The predictor $\hat{y}(x;\theta)$ is now a vector of class-probabilities given by a softmax link function:

$$\hat{y}(x;\theta) = \left(\frac{e^{x^T\theta_1}}{\sum_{i=1}^C e^{x^T\theta_i}}, \frac{e^{x^T\theta_2}}{\sum_{i=1}^C e^{x^T\theta_i}}, \ldots, \frac{e^{x^T\theta_C}}{\sum_{i=1}^C e^{x^T\theta_i}}\right), \tag{2.19}$$

where C is the number of classes and $\theta = (\theta_1, \theta_2, \ldots, \theta_C)$ resulting in one affine transformation per class.

Using the categorical log-likelihood for the observations, we obtain the following conditional model:

$$\begin{aligned}\log P(Y=y|X=x) &= \log \text{Cat}(\hat{y}(x;\theta)) \\ &= \sum_{i=1}^C y_i \log(\hat{y}(x;\theta)_i).\end{aligned} \tag{2.20}$$

2.2 Deep Learning

For complex non-linear relationships and high-dimensional data, linear models tend to underfit the data and have poor predictive performance. The core idea of deep learning is to create a stack of linear layers intertwined with non-linearities to obtain flexible models capable of efficiently fitting non-linear functions to high-dimensional data. In this section, we give a brief overview of the main components of deep learning. For a more complete treatment of the topic, see Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Deep learning models, also called neural networks, can be seen as a composition of functions f_k, each parameterised by its own set of parameters[1] θ_k, mapping the input features to the target variable:

$$\hat{y}(x;\theta) = f_1(\,\cdot\,;\theta_1) \circ f_2(\,\cdot\,;\theta_2) \circ \dots f_K(\,\cdot\,;\theta_K)(x). \tag{2.21}$$

The nature and order of these functions make up the neural network's architecture – a topic of intense research where researchers try to find which architecture best suits their application.

The conditional negative log-likelihood of Y given X is referred to as the loss function:

$$-\log P(Y = y|X = x) = \mathcal{L}(y, \hat{y}(x;\theta)) \tag{2.22}$$

Minimising the loss function is equivalent to maximising the likelihood with respect to the neural network's parameters. Maximum likelihood estimation is not tractable in deep models, there are usually no closed-form expressions, and second-order numeric optimisation methods fall to the curse of dimensionality. One of the major keys to the success of deep learning is the gradient descent and back-propagation algorithms (LeCun, Boser, et al. 1989; Rumelhart et al. 1986).

Gradient descent. The gradient descent algorithm is an iterative process to find a maximum or minimum of a function. At each time-step t, we calculate the gradient of the function with respect to its parameters and take a small step in the direction which maximises the likelihood:

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial \mathcal{L}(y, \hat{y}(x;\theta_t))}{\partial \theta_t}, \tag{2.23}$$

where α is a hyper-parameter called *learning rate*. The algorithm is usually applied to batches of data as opposed to the entire dataset resulting in what is known as stochastic gradient descent. Note that gradient descent is not guaranteed to find the global minimum of the likelihood function, as there are many local optima and the one we converge to depends on the initial values of the parameters.

[1] Without loss of generality, we admit the existence of functions with an empty set of parameters, such as the Relu non-linearity.

Back-propagation. To calculate the gradients needed for gradient descent, we use the back-propagation algorithm, which allows us to recursively calculate gradients starting at the last layer of the network. We begin by introducing the intermediate outputs of the network, which are computed in the forward pass:

$$o_k = f_1(\,\cdot\,;\theta_1) \circ \dots f_k(\,\cdot\,;\theta_k)(x), \tag{2.24}$$

where each intermediate output can be obtained from the previous as such: $o_k = f_k(o_{k-1}, \theta_k)$. Next, using the chain rule for derivatives for composite functions, we can obtain the gradient of the loss function with respect to an arbitrary parameter θ_k as follows:

$$\begin{aligned} \frac{\mathcal{L}(y, \hat{y}(x;\theta))}{\partial \theta_k} &= \frac{\partial \mathcal{L}(y, \hat{y}(x;\theta))}{\partial o_k} \frac{\partial o_k}{\theta_k} \\ &= \frac{\partial \mathcal{L}(y, \hat{y}(x;\theta))}{\partial o_k} \frac{\partial f_k(o_{k-1}, \theta_k)}{\theta_k}. \end{aligned} \tag{2.25}$$

This simple rule allows us to compute the gradient for a specific parameter given the local gradient of the function pertaining to the parameter and the gradient of the loss function with respect to the output of the same function. With this in mind, we can create an iterative algorithm to *back-propagate* the gradient through the network starting at the last layer and obtain the gradients for all parameters. For the last layer of the network we have a $k = K$ and thus $o_k = \mathcal{L}(y, \hat{y}(x;\theta))$, thus the first term in equation 2.25 is equal to 1 and hence:

$$\frac{\partial \mathcal{L}(y, \hat{y}(x;\theta))}{\partial \theta_K} = \frac{\partial f_K(o_{K-1}, \theta_K)}{\partial \theta_K}. \tag{2.26}$$

For the second to last layer:

$$\frac{\mathcal{L}(y, \hat{y}(x;\theta))}{\partial \theta_{K-1}} = \frac{\partial \mathcal{L}(y, \hat{y}(x;\theta))}{\partial o_K} \frac{\partial o_K}{\partial o_{K-1}} \frac{\partial f_{K-1}(o_{K-2}, \theta_{K-1})}{\partial \theta_{K-1}}, \tag{2.27}$$

where we have already computed the first term, and thus we need only to compute the two last terms. For the third to last layer:

$$\frac{\mathcal{L}(y, \hat{y}(x;\theta))}{\partial \theta_{K-2}} = \frac{\partial \mathcal{L}(y, \hat{y}(x;\theta))}{\partial o_K} \frac{o_K}{\partial o_{K-1}} \frac{\partial o_{K-1}}{\partial o_{K-2}} \frac{\partial f_{K-2}(o_{K-3}, \theta_{K-2})}{\partial \theta_{K-2}}, \tag{2.28}$$

where again, we have already computed the two first terms and need only to compute the last two. We can continue the process recursively until we obtain all of the parameter's gradients.

Image Classification and Segmentation. One of the main applications of deep learning is the multi-class classification of images. Images are high-dimensional dimensional and complex data points living on a grid, making simple machine-learning methods unsuitable models in this case. In this setting, the input X is a 2D or 3D grid of pixels/voxels, and the target variable Y is a class label. Following the same procedure as we did for multinomial logistic regression, we obtain the

following predictor:

$$\hat{y}(x;\theta) = \left(\frac{e^{f(x;\theta)_1}}{\sum_{i=1}^{C} e^{f(x;\theta)_i}}, \frac{e^{f(x;\theta)_2}}{\sum_{i=1}^{C} e^{f(x;\theta)_i}}, \ldots, \frac{e^{f(x;\theta)_C}}{\sum_{i=1}^{C} e^{f(x;\theta)_i}} \right), \tag{2.29}$$

with the only difference being that instead of using a linear transformation on the input data, we use a neural network f parameterised by θ, which outputs a vector of *logits*, which are then fed to the softmax activation function. Using the categorical log-likelihood, we obtain the famous cross-entropy loss function:

$$\mathcal{L}(y, \hat{y}(x;\theta)) = -\sum_{i=1}^{C} y_i \log(\hat{y}(x;\theta)_i). \tag{2.30}$$

Fully-connected linear layers have one parameter per pixel and ignore the spatial structure of the data, incurring a high computational cost. In contrast, image convolutions share parameters between pixels and allow for much more efficient use of computational resources. For this reason, convolutions were first introduced in neural networks in 1989 for hand-digit recognition (LeCun, Bottou, et al. 1998) at a time when computational power was a significant limitation. More recently, advancements in graphics processing units (GPUs) which provide substantial increases in computational power over central processing units (CPUs), allowed convolutional neural networks to thrive at natural image classification tasks with large and complex datasets (Krizhevsky et al. 2017). Later developments cemented convolutional neural networks for image classification by using deeper models (Simonyan and Zisserman 2014) and residual connections (He et al. 2016a).

Image segmentation is a particular case of image classification where instead of predicting a class for the whole image, we predict a class for each pixel/voxel. Using a shared neural network backbone, we can efficiently obtain a logit map with the same spatial dimensions as the input image. Fully-convolutional neural networks are used to produce an output with the same spatial dimensions as the input image (Long et al. 2015). When the image size becomes prohibitively large, it is common to use multi-resolution architectures which down-sample the image and perform most of the computation at lower image resolutions (Kamnitsas, Ledig, et al. 2017; Ronneberger et al. 2015). Like in classification, we compute the log-likelihood of one pixel using a softmax activation and a categorical distribution. Assuming that the distribution of each pixel is independent of the distributions of other pixels when given the input image, we can compute the log-likelihood of the entire label map by simply summing the individual contributions of each pixel. In Chapter 2, we show an application of this process by performing multi-class segmentation of traumatic brain injury in 3-dimensional head CT images. In Chapter 3, we show that the pixel-wise independence assumption used to arrive at the log-likelihood for the whole image is somewhat strong and propose a better way of computing the likelihood by taking into account correlations between pixels.

2.3 Deep Latent Variable Generative Models

The unconditional models of Section 2.1.1 are generative models of the data since we can use them to simulate the data-generating process and obtain new samples from the distribution. However, these simple models quickly reach their limit when fitting to high-dimensional structured data such as images. To generate new samples from complex distributions, we must rely on more powerful models, such as the neural networks presented in Section 2.2. The deep models presented in Section 2.2 are known as discriminative models. These models aim to obtain a low-dimensional target variable by conditioning it on a high-dimensional structured variable, such as an image. In this section, we show how we can use neural networks and latent variable models to model complex probability distributions and generate high-dimensional structured data.

Latent variable models. Consider a simple probabilistic graphical model $Z \rightarrow X$, where X is the observed variable distributed according to an unknown distribution $p(x)$, and Z is a latent variable with a known prior distribution $p(z)$ amenable to sampling and tractable likelihood calculations. We wish to obtain the generative model $p(x)$ from which we can sample new instances of X. We assume a conditional likelihood $p_\theta(x \,|\, z)$ parameterised by parameters θ from which we can obtain samples of X by passing samples of Z through the model. The log-likelihood of X is given by:

$$\log p(x) = \int_z p_\theta(x, z) dz = \int_z p_\theta(x \,|\, z) p(z) dz. \tag{2.31}$$

We can use maximum likelihood estimation to obtain estimates for the parameters θ. However, for any moderately complex model, such as a neural network, the likelihood is intractable. Noting that the integral is simply the expectation $\mathbb{E}_{p(z)}\, p_\theta(x \,|\, z)$, we could naively compute it using Monte-Carlo integration. However, this estimator is known to have extremely high variance and result in poor parameter estimates (Kingma and Welling 2014).

Variational autoencoders (VAEs). In variational autoencoders (Kingma and Welling 2014; Rezende et al. 2014), we take a better approach to solving the integral by using variational inference. We introduce an auxiliary distribution $q(z)$ and use Jensen's inequality to obtain the following variational bound:

$$\begin{aligned}
\log p(x) &= \log \int_z p_\theta(x, z) dz \\
&= \log \int_z p_\theta(x, z) \frac{q(z)}{q(z)} dz \\
&= \log \mathbb{E}_{q(z)} \frac{p_\theta(x, z)}{q(z)} \\
&\geq \mathbb{E}_{q(z)} \log \frac{p_\theta(x, z)}{q(z)} = \mathbb{E}_{q(z)}[\log p_\theta(x \,|\, z)] - D_{\mathrm{KL}}[q(z) \| p(z)].
\end{aligned} \tag{2.32}$$

We can see that maximising the variational lower bound, also called the evidence lower bound, also maximises the log-likelihood. The key to successful variational inference is to choose the auxiliary distribution $q(z)$, which makes the estimation problem easier. In the VAE framework, we use an amortised variational approximation $q_\phi(z\,|\,x)$ which is modelled by a neural network encoder $z = g_\phi(x)$ with parameters ϕ. The likelihood $p_\theta(x\,|\,z)$ is modelled by a neural network decoder $x = f_\theta(z)$ with parameters θ. The resulting ELBO is given by the following:

$$ELBO = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x\,|\,z)] - D_{\mathrm{KL}}[q_\phi(z\,|\,x)\|p(z)]. \tag{2.33}$$

The first term of the equation is known as the reconstruction term, and the second term is the Kullback–Leibler (KL) divergence between the latent prior and the approximate latent posterior. The latent prior $p(z)$ and approximate latent posterior $q_\phi(z\,|\,x)$ are usually chosen to be Gaussian, enabling a closed-form expression for the KL divergence and its gradient. Optimising the reconstruction term using the naive Monte-Carlo estimate of the gradient is known to be of high variance (Paisley et al. 2012). For this reason, we re-parameterise the random variable $z \sim q_\phi(z\,|\,x)$ using an auxiliary noise variable $\epsilon \sim p(\epsilon)$. For example, for a Gaussian latent posterior $q_\phi(z\,|\,x) = \mathcal{N}(\mu_\phi(x), \sigma^2_\phi(x))$, we would use standard normal distribution as the auxiliary noise variable $p(\epsilon) = \mathcal{N}(0, 1)$, resulting in the following Monte-Carlo approximation of the reconstruction term:

$$\mathbb{E}_{q_\phi(z|x)} \log p_\theta(x\,|\,z) = \frac{1}{M}\sum_{m=1}^{M} f(\mu_\phi(x) + \sqrt{\sigma^2_\phi(x)}\epsilon^{(m)}),\ \epsilon^{(m)} \sim \mathcal{N}(0, 1). \tag{2.34}$$

Generative Adversarial Networks (GANs). Variational autoencoders are amortised explicit likelihood models because we directly optimise an approximation of the likelihood. In the case of generative adversarial networks (Goodfellow, Pouget-Abadie, et al. 2020), this objective is a minimax game between a generator and a discriminator. The generator attempts to produce samples from the observed distribution from random noise, and the discriminator tries to distinguish these 'fake' samples from the 'real' samples of the empirical distribution. In summary, we try to *match* the generated to the observed distribution. Formally, this objective is given by:

$$\min_\theta \max_\phi \mathbb{E}_{x\sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]. \tag{2.35}$$

$G_\theta(z)$ is a neural network generator parameterised by θ, which takes as input a sample from the auxiliary latent noise variable Z and tries to produce a 'fake' sample of the distribution $p(x)$. $D_\phi(x)$ is a neural network discriminator parameterised by ϕ, which tries to distinguish samples from the real distribution $p(x)$ from the 'fake' samples produced by the generator.

While the GAN objective might initially seem ad-hoc, Nowozin et al. (2016) have shown that this objective is equivalent to minimising an approximation of the Jensen–Shannon divergence via convex

risk minimisation. Furthermore, they showed the GAN objective is a particular case of an algorithm that allows the estimation of other f-divergences, such as the KL divergence. If the divergence between two distributions is zero, then they are the same distribution. Thus we can see how this distribution matching procedure results in a sensible generative model.

2.4 Structural Causal Models

This section gives a brief overview of Structural Causal Models (SCMs), also known as Structural Equation Models (SEMs). For a complete treatment of the theory, see Pearl, J. (2009). *Causality*. Cambridge University Press and Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

An SCM is a set of functional assignments (called *mechanisms*) that represent the relationship between a variable, its direct causes (called *parents*) and all other unaccounted sources of variation (called *exogenous noise*). Formally, an SCM is defined as:

$$X_j \coloneqq f_j(\mathbf{PA}_j, N_j) \quad j = 1, \ldots, d, \tag{2.36}$$

where a random variable X_j is a function f of its endogenous causes $\mathbf{PA}_j$ (parents) and its exogenous causes N_j (exogenous noise).

In SCMs, we assume that the mechanisms are algorithmically independent of each other. In the case of two variables, this reduces to the independence of cause and mechanism. An SCM is said to be semi-Markovian if it is acyclic, a variable cannot be its own cause. These models can be represented by directed acyclic graphs (DAGs). If in addition to acyclicity the the exogenous noise variables are jointly independent of each other the model is said to be Markovian. Figure 2.1 shows an example of a Markovian SCM represented in DAG form.

In Pearl's ladder of causation, we have, in increasing order of difficulty: association, intervention, and counterfactuals. Classic statistical models are sufficient to answer associative questions. However, for questions regarding interventions and counterfactuals, SCMs are powerful and convenient tools.

Intervening on a variable means setting it to a specific value. When we intervene on a variable, we stop the causal effects from its parents since we are forcing the variable to take a given value. This process results in a modified SCM where we remove the arrows from the parents to the intervened variable. We can calculate interventional distributions by intervening on one or more variables in the SCM, sampling from the exogenous noise distributions and using the modified SCM to compute the quantity of interest. Here we assume we have access to the equations that govern the SCM. Estimating these equations often involves collecting interventional data through randomised control trials.

Counterfactuals are what-if question about the system that allows us to ask how the system would have behaved had some variables taken different values. Counterfactuals differ from interventions since they are questions about a specific observation, not the entire distribution. As a result, answering counterfactual questions is only possible by knowing the complete state of the system before acting on a variable, which implies estimating the value of the exogenous noise variables, a process known as abduction. Conceptually, estimating counterfactuals is a three-step process: 1) Abduction: infer the exogenous noise from the observation and its parents; 2) Action: intervene on the targeted parents; 3) Prediction: propagate the effect of the intervention through the modified model to generate the counterfactual.

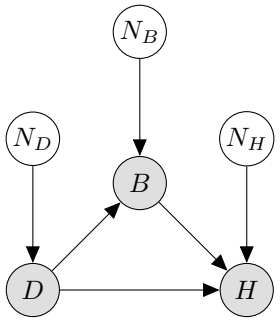


Figure 2.1: Example of a Markovian SCM represented in DAG form. D=Drug; B=Blood pressure; H=Heart attack. N_D, N_B, N_H are the exogenous noise variables. The arrows represent the direction of causation. The drug causes both blood pressure and heart attack, and blood pressure causes heart attack.

Chapter 3

Multiclass semantic segmentation and quantification of traumatic brain injury lesions on head CT using deep learning

This chapter is based on the following publication:

- **Monteiro, M.**, Newcombe, V. F., Mathieu, F., Adatia, K., Kamnitsas, K., Ferrante, E., Das, T., Whitehouse, D., Rueckert, D., Menon, D. K., et al. (2020). "Multiclass semantic segmentation and quantification of traumatic brain injury lesions on head CT using deep learning: an algorithm development and multicentre validation study". In: *The Lancet Digital Health* 2.6, e314–e322

MM, VFJN, DKM, and BG conceived and designed the study. MM did the implementation, analysed data, and cowrote the manuscript with VFJN. MM, VFJN, DKM, and BG revised and finalised the manuscript. VFJN, FM, KA, and DW did the manual and semi-automatic segmentation of the scans or provided broader clinical input, or both. KK, EF, and BG provided feedback on the development of the model. TD provided specialist neuroradiological oversight of image analysis. VFJN, DR, DKM, and BG secured the funding.

Code available at:

- https://github.com/biomedia-mira/blast-ct

3.1 Introduction

With an estimated global incidence of more than 60 million cases per year, traumatic brain injury (TBI) is the leading cause of mortality in young adults and a major cause of morbidity worldwide (Dewan et al. 2018; Maas, Menon, Adelson, et al. 2017). CT is the imaging modality of choice to assess the extent and distribution of injury, provide input to prognostic models, and assess the requirement for surgery (Amyot et al. 2015). Being able to automatically and accurately quantify lesion load and its progression would provide a more objective basis than qualitative assessment by visual inspection for medical and surgical treatment decision making.

A substantial focus of TBI research has been to refine the current classification schemes into more

therapeutically meaningful categories by incorporating information on a patient's genetic, blood, and cerebrospinal fluid biomarkers along with clinical and neuroimaging data (Carney et al. 2017; Maas, Menon, Adelson, et al. 2017). Hence, being able to reliably and efficiently differentiate lesion types and compute their spatial distribution, number, and volumes would enable optimised and more individualised treatment strategies. Such automated assessment would also facilitate the analysis of large imaging datasets, which are emerging as an essential research resource. Finally, by far the greatest burden of TBI is in low-income and middle-income countries (Dewan et al. 2018), where radiological expertise is likely to be be less easily available. Having automatic CT analysis algorithms would be of particular benefit in such contexts (Dewan et al. 2018).

Substantial inter-centre variability and discordance by radiologists exists when reporting CT scan results from patients with TBI (Vande Vyvere et al. 2019). Automating such quantitative measurements would, in theory, circumvent inter-observer variability and allow for analysis of large-scale imaging datasets. Until recently, attempts to automate acute intracranial haemorrhage segmentation on CT have relied on techniques such as intensity thresholding and active contouring, which still require some degree of manual input, and have only been applied to small datasets, raising concerns about the robustness and generalisability of these models (Bardera et al. 2009; Bhadauria et al. 2013; Roy et al. 2015; Zaki et al. 2011). Little past success in this context probably reflects two challenges in working with this patient population. First, the heterogeneity of radiographic phenotypes in TBI makes the development of accurate segmentation rules challenging. Second, the diffuse nature of the injury in a large proportion of patients with TBI renders the manual annotations required to establish a ground truth reference dataset difficult and time consuming.

Convolutional neural networks (CNNs) have emerged as a powerful tool for image segmentation, with the ability to learn complex non-linear mappings between the input image and segmentation (LeCun, Bottou, et al. 1998). Previous deep learning studies for segmentation of TBI lesions have focused on the segmentation of undifferentiated haemorrhagic lesions, with no attempts to differentiate pathoanatomical lesion types (Jain et al. 2019). Although such binary image-level detection of abnormalities might prove useful for triaging patients in need of urgent medical attention, it has little value in supporting precision medicine, quantifying lesion progression in trials of new therapies, or predictive modelling of clinical outcome. Other studies have focused on lesion detection at an image level with differentiation of intracranial haemorrhage types (Chilamkurthy et al. 2018; Kuo et al. 2019). In addition to detection, one study showed qualitative results for segmentation (Kuo et al. 2019). However, this study provided no quantitative metrics, did not specifically address TBI, and provided no assessment of oedema. Accurate quantification of lesion volumes can only be achieved when using voxel-wise labels (*i.e.*, for segmentation of lesions) as opposed to image-level labels (*i.e.*, for classification of images). Voxel-wise labels allow for both volume quantification and localisation of lesions, which may be important for improved understanding of the factors that lead to lesion progression and to more clinically relevant prognostic schemes.

We aimed to develop and validate a new, clinically relevant algorithm based on deep CNNs for multiclass, voxel-wise segmentation, volumetric quantification, and detection of TBI lesion types visible in CT.

3.2 Materials and Methods

3.2.1 Study design and participants

The data used in this study were from the Collaborative European Neuro Trauma Effectiveness Research in TBI study (CENTER-TBI, NCT02210221) (Maas, Menon, Steyerberg, et al. 2015; Steyerberg et al. 2019), accessed using the Neurobot platform (RRID/SCR_017004, core data version 2.0, release date May 15, 2019). Patients were recruited between Dec 9, 2014, and Dec 17, 2017, in 60 centres across Europe. Data collection, handling, and storage are described in detail elsewhere (Maas, Menon, Steyerberg, et al. 2015; Steyerberg et al. 2019). CT scans were collected as part of standard clinical practice, using various platforms and imaging parameters (Vande Vyvere et al. 2019).

Ethical approval was obtained in accordance with all relevant laws and regulations for each recruiting site, and informed consent by patients or their legal representative or next of kin was obtained according to local laws and regulations (Maas, Menon, Steyerberg, et al. 2015). A complete ethics statement, which contains a comprehensive list of sites, ethical committees, and approval numbers, is available online at https://www.center-tbi.eu/project/.

3.2.2 Procedures

For development and internal validation, we use two datasets from CENTER-TBI: dataset 1 and dataset 2. We used a two-step process to acquire a large number of annotated scans (see Figure A.1). The scans in dataset 1 were annotated manually in a bespoke segmentation tool (ImSeg, version 1.9, BioMedIA, London, UK) by trained personnel (FM and KA) and checked by two other experts (VFJN and TD). These segmentations were used to develop the initial segmentation model and then excluded from any subsequent training or evaluation to avoid skewing the analysis of results.

With the model developed on dataset 1, we did automatic lesion segmentation on dataset 2. These automatic segmentations were refined manually by trained personnel (FM and KA) using ITK-SNAP (version 3.8.0-beta), and the corrections were reviewed by two experts (VFJN and TD) to provide high-quality, accurate ground truth lesion segmentations. The refined segmentations contained four lesion types: intraparenchymal haemorrhage; extra-axial haemorrhage, which includes subdural haematoma, extradural haematoma, and traumatic subarachnoid haemorrhage; perilesional oedema (hereafter referred to as oedema); and intraventricular haemorrhage. Small petechial haemorrhages, which probably arise from diffuse vascular injury and are thought to be a surro-

gate for accompanying diffuse axonal injury (Figueira Rodrigues Vieira and Guedes Correa 2020; Haacke et al. 2010), were classified as intraparenchymal haemorrhage.

To establish whether the semi-automatic annotation procedure of dataset 2 provided adequate reproducibility, we did repeat manual segmentation on 20 scans by a single expert (FM) to assess intra-rater reproducibility, and on 25 scans by a second expert (DW) to assess inter-rater variability.

For the subsequent analyses, we split dataset 2 into a training and test set. Different scans from the same patient were placed together in either the training or the test set to avoid the correlation between repeat scans biasing the results. Only scans with more than 1 mL of lesion load were included in the training set to ensure that there was enough training signal for the CNN.

For external validation, we used the CQ500 dataset, a publicly available, anonymised, TBI CT dataset provided by the Centre for Advanced Research in Imaging, Neurosciences and Genomics, New Delhi, India available at http://headctstudy.qure.ai/dataset. This dataset provides image-level labels as opposed to voxel-wise segmentations. However, it is the largest labelled TBI cohort available publicly, and no other dataset provides voxel-wise segmentations.

3.2.3 Image pre-processing, model and training

For pre-processing, we resampled images to an isotropic resolution of $1 \times 1 \times 1$ mm and bounded the intensities between -15 and 100 Hounsfield units (HU) before scaling the range between -1 and 1. We do not perform skull-stripping (Monteiro, Kamnitsas, et al. 2019). We note that all evaluation and statistical analysis was done after mapping the output of the CNN back to the image's original native resolution.

The segmentation model used is called DeepMedic (Kamnitsas, Ferrante, et al. 2016; Kamnitsas, Ledig, et al. 2017), a convolutional neural network (CNN) designed for lesion segmentation on three-dimensional brain images. The network processes image patches at full, three-times down-sampled, and five-times down-sampled resolution to gather the necessary context for the segmentation. To keep the architecture up to date with the state-of-the-art we use residual connections (He et al. 2016a), batch-normalization (Ioffe and Szegedy 2015), and pre-activation blocks (He et al. 2016b).

The network was trained for 1200 epochs with an initial learning rate of 0.001. The learning rate was halved at the following milestones: 440, 640, 800, 900, 980, and 1050 epochs. We employed the RMSprop optimizer (Tieleman and Hinton 2012) with a momentum 0.6, alpha 0.9 and a l2 weight penalty of 0.0001. We used a standard cross-entropy loss and a batch size 10 for training. The patch sampling procedure is as follows. At each epoch, 50 images are randomly sampled from the training set. From each image, 20 patches of size $110 \times 110 \times 110$ mm are extracted according to a sampling scheme - a patch centred on a background voxel is sampled with a probability of 30%

and on one of the four lesion classes with a probability of 17.5% each (70% total for any lesion). Since no padding is used, the network outputs a prediction with size $30 \times 30 \times 30$ mm.

For data-augmentation, with a probability of 50%, we applied one or more of the following transformations to the input : random intensity histogram deformation (intensity shift 0.00 ± 0.05, intensity scale 1.00 ± 0.01); random flipping of the x and/or y-axis; random right-angle rotation of the of the x and/or y-axis; random elastic deformation using a procedural generated noise to create plausible distortions in the anatomy (Monteiro, Kamnitsas, et al. 2019).

To account for the stochasticity of neural network training, we trained twelve models with the same configuration but different random seeds. The final segmentation prediction for a test image was obtained by averaging over the individual predictions of each model.

3.2.4 Statistical analysis

Classic sample size calculation is not directly applicable to CNN-based segmentation. The sample sizes in this work followed the common principle in current deep learning research whereby more data tends to yield better results. Thus, we attempted to maximise the number of scans for training and testing under the constraint of finite resources for expert annotations.

Evaluation metrics were computed and stratified by lesion class and volume. A virtual lesion class (any lesion) consisting of the combined lesion map that merged all lesion types into one was created to allow for evaluation in terms of lesion versus non-lesion.

To assess the performance of the algorithm, we used the Dice similarity coefficient (DSC), which measures the agreement between manual and automatic segmentation. Since the mean DSC is sensitive to lesions with small volumes or scans on which lesions are not present, we report DSC scores for lesions above several volume thresholds. DSC is a well accepted metric for assessing accuracy in image segmentation (Menze et al. 2014). The Dice similarity coefficient is defined as $DSC = \frac{2TP}{2TP+FP+FN}$, where TP denotes true positives, FP denotes false positives and FN denotes false negatives. This is equivalent to the statistical measure F1-score which is the harmonic mean between sensitivity and positive predictive value. However, it is not meaningful when assessing performance with respect to clinical utility. In a multiclass setting, where most scans do not contain all lesion classes if a scan does not contain a certain lesion class (TP + FN = 0) and the algorithm mislabels one voxel as being from that class (FP = 1), then that scan will count as a zero towards the average DSC of that class. In the same scenario, if the algorithm does not make a mistake (FP = 0), then the DSC is not defined, and thus the scan will not increase the average even though the prediction is perfect. This property of the metric inevitably misrepresents the performance of the algorithm by disproportionally penalizing cases with small or non-existent lesions. For example, consider a scan where there is one large EAH but no other lesion types. If the algorithm perfectly segments the EAH lesion but also mislabels one voxel as oedema, then this scan will count as zero towards the average DSC of the oedema class, heavily penalizing the metric

even though the prediction for the whole scan was near perfect. As a result, DSC alone is not appropriate to assess the clinical utility of the algorithm. For a clinically relevant assessment, we have provided additional metrics such as lesion volume estimates and receiver operating characteristic (ROC) curves for lesion detection and lesion volume classification.

To assess the accuracy of the algorithm at estimating lesion volume, we extracted lesion volumes from the manual and predicted segmentations to calculate volume error, which we summarised in Bland-Altman plots. We also assessed the accuracy of the algorithm at quantifying lesion progression. To obtain the error in volume change, we calculated the true volume difference and predicted volume difference between repeat scans for patients in the test set who had repeat scans for which both timepoints could be established.

The output of the segmentation algorithm can be used for lesion detection and lesion volume classification. We used the true lesion volume to set a classification target (*e.g.*, target is positive if the true volume is greater than 1 mL and negative otherwise). We then used the predicted lesion volume as the score on which a threshold was varied to calculate ROC curves. We addressed three key lesion detection and lesion volume classification problems to assess the clinical applicability of the model: (1) ability to detect lesions, which is equivalent to classifying lesions with a volume greater than 0 mL; (2) classification of lesions with a volume greater than 1 mL, to enable comparison with findings from datasets that did not contain small lesions; and (3) classification of lesions with a volume greater than 25 mL, equivalent to Marshall grade V/VI (Marshall et al. 1991), which may indicate lesions requiring surgical intervention.

For each curve, we computed the area under the curve (AUC), its 95% CI using the Hanley and McNeil approach (Hanley and McNeil 1982), the sensitivity and specificity of the two operating points (sensitivity at a specificity of 0.90 and vice versa), and their 95% CIs using the Clopper-Pearson method (Clopper and Pearson 1934).

We used our algorithm to segment the scans in the CQ500 dataset and to calculate lesion volumes. These are used as the classification score to compare with the ground truth image-level labels provided. This dataset was used only at the end for final validation, never during development. This approach validated the lesion detection performance of our algorithm on an external, independent dataset from a different patient population. CQ500 was not annotated for oedema, and so instead of our summated any lesion class we report on intracranial haemorrhage, which includes all haemorrhage classes in our analysis: intraparenchymal haemorrhage, extra-axial haemorrhage, and intraventricular haemorrhage.

Table 3.1: Cohort details for both datasets.

	Dataset 1 (n=27)	Dataset 2 (n=512)
Age (years)	46 (16–77)	58 (6–89)
Sex		
Female	5 (19%)	163 (32%)
Male	22 (81%)	349 (68%)
Mechanism of injury		
Acceleration or deceleration	7 (26%)	111 (22%)
Blow to head or hit object	4 (15%)	77 (15%)
Fall from height	13 (48%)	208 (41%)
Multi-mechanistic	2 (7%)	99 (19%)
Unknown	1 (4%)	17 (3%)
Injury severity		
Mild (GCS 13–15)	7 (26%)	299 (58%)
Moderate (GCS 9–12)	2 (7%)	57 (11%)
Severe (GCS <9)	18 (67%)	136 (27%)
Missing	0	20 (4%)
Time from injury to first CT scan (h)	2.4 (1.2–8.0)	2.0 (0.2–77.0)
Repeat scan done	26 (96%)	412 (80%)
Time from injury to second CT scan (h)	16.0 (5.0–79.0)	19.0 (0.9–190.0)
Interval between CT scans (h)	14.0 (3.6–77.0)	16.0 (0.1–190.0)
Marshall score		
I	2 (7%)	120 (23%)
II	11 (41%)	234 (46%)
III	2 (7%)	29 (6%)
IV	0	6 (1%)
V	0	2 (<1%)
VI	12 (44%)	121 (24%)
Presence of:		
Epidural haematoma	10 (37%)	54 (11%)
Acute subdural haematoma	13 (48%)	223 (44%)
Traumatic subarachnoid haemorrhage	20 (74%)	313 (61%)
Intraventricular haemorrhage	6 (22%)	88 (17%)
Intraparenchymal haemorrhage	18 (67%)	224 (44%)
Cisternal compression	9 (33%)	99 (19%)
Midline shift >5 mm	8 (30%)	71 (14%)
Glasgow Outcome Score at 6 months		
1	6 (22%)	66 (13%)
2	0	0
3	9 (33%)	84 (16%)
4	7 (26%)	126 (25%)
5	2 (7%)	199 (39%)
Missing	3 (11%)	37 (7%)

3.3 Results

Dataset 1 consisted of 98 different CT scanning sessions from 27 patients from one centre (Cambridge University NHS Foundation Trust, Cambridge, UK). Data from this centre were available first as part of a preliminary proof-of-concept study. Dataset 2 consisted of 839 different CT scanning sessions from 512 patients and 38 different centres from which data were available at the time of the study, including Cambridge NHS Foudation Trust. The procedure of semi-automatic segmentation enabled the creation of a much larger dataset (839 vs 98 scans) without a commensurate increase in resource requirements. Table 3.1 shows the cohort characteristics of both datasets, representing the broad spectrum of TBI. From dataset 2, 184 scans were included in the training subset and 655 scans were included in the test subset. Consistent with the known heterogeneity of TBI, 744 (89%) of 839 scans did not contain all four lesion types (Maas, Menon, Adelson, et al. 2017; Steyerberg et al. 2019). The distribution of lesions is available in Table A.1.

Figure 3.1 shows qualitative results for five different cases from our test set, showing the visual agreement between the true and predicted segmentations. Figure 3.2 shows DSC boxplots. The median DSC for the any lesion class was 36.0% (IQR 0.0–63.4) when including all 599 scans (469 with lesions plus 130 with no lesions but where our model predicted a lesion). In addition to calculating DSCs using all the test scans, we chose the following preplanned thresholds to address different performance levels: 0 mL, 1 mL, and 5 mL (see Table 3.2). Limiting the analysis to the

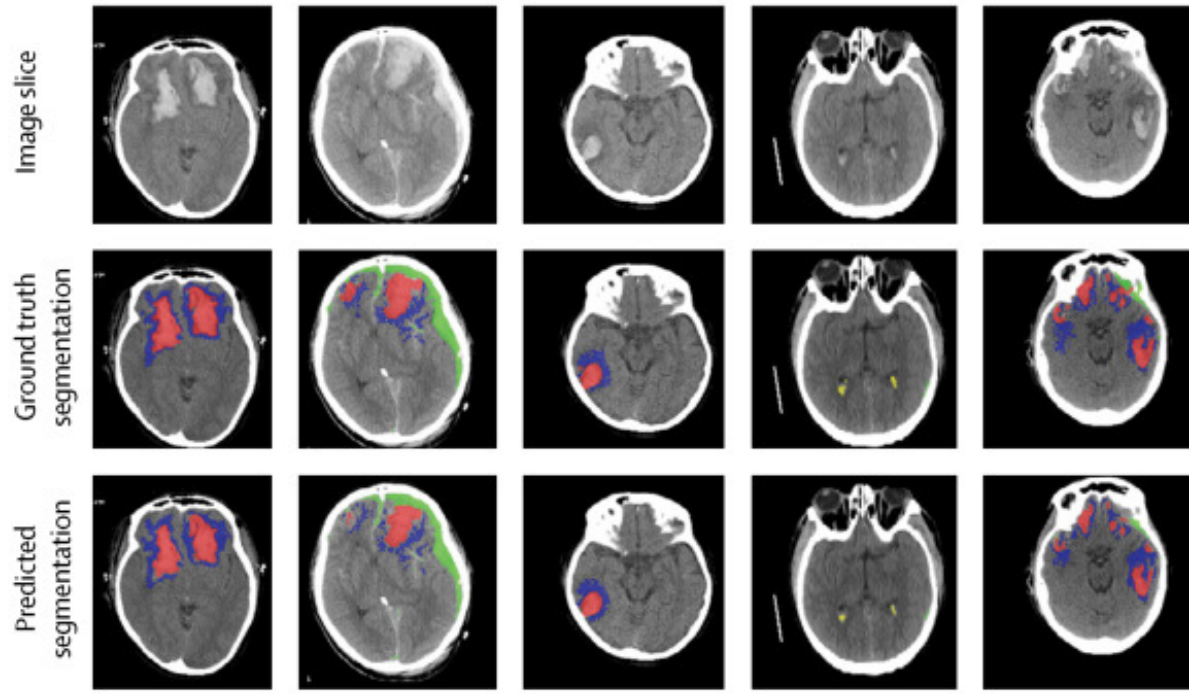


Figure 3.1: Qualitative multiclass segmentation results. IPH is shown in red, EAH in green, perilesional oedema in blue, and IVH in yellow.

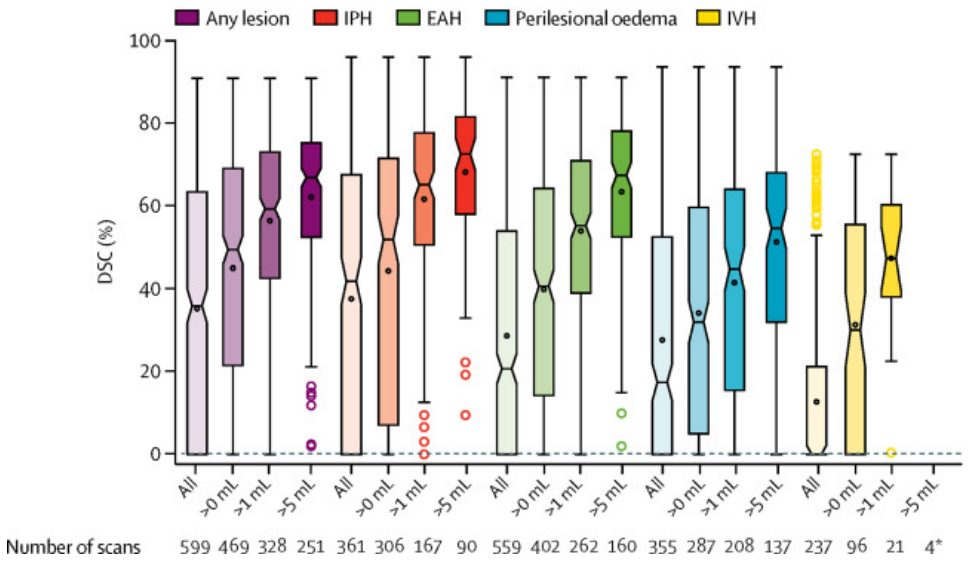


Figure 3.2: Per-class boxplots of DSC stratified by volume threshold. For each class, each boxplot progressively excludes lesions with volume smaller than a threshold. For each individual boxplot, the central line represents the median and the black circle the mean. The box shows the IQR and is indented to indicate the 95% CI of the median. Whiskers adjacent to the boxes represent 1.5 times the IQR. Coloured circles are outliers. Table 3.2 shows the same results in tabular format. DSC=Dice similarity coefficient. EAH=extra-axial haemorrhage. IPH=intraparenchymal haemorrhage. IVH=intraventricular haemorrhage. *Not plotted owing to insufficient data.

469 scans with lesions increased the median DSC to 49.4% (IQR 21.5–67.1), and the exclusion of lesions of 1 mL or smaller further increased the DSC to 59.3% (42.6–73.1, n=328). A similar relationship between lesion volume and DSC was noted for individual lesion classes (Figure 3.2). For lesions with a volume greater than 1 mL, the median DSC was 65.2% (IQR 50.6–77.8, n=167) for intraparenchymal haemorrhage, 55.3% (39.1–71.0, n=262) for extra-axial haemorrhage, 44.8% (15.5–64.1, n=208) for oedema, and 47.3% (38.1–60.3, n=21) for intraventricular haemorrhage; for lesion volumes greater than 5 mL, these numbers increased to 72.6% (58.1–81.6, n=90) for intraparenchymal haemorrhage, 67.5% (52.5–78.2, n=160) for extra-axial haemorrhage, and 54.6% (32.0–68.1, n=137) for oedema. To compare with previous literature, we combined intraparenchymal haemorrhage and extra-axial haemorrhage and obtained a median DSC of 72.0% (59.2–80.1, n=210) for lesion volume greater than 5 mL.

Figure 3.3 shows Bland-Altman plots of the agreement between the true and predicted lesion volumes. The mean difference was 0.86 mL (95% CI -5.23 to 6.94) for intraparenchymal haemorrhage, 1.83 mL (-12.01 to 15.66) for extra-axial haemorrhage, 2.09 mL (-9.38 to 13.56) for oedema, and 0.07 mL (-1.00 to 1.13) for intraventricular haemorrhage. For lesions with a volume greater than 5 mL, the median absolute error was 3.57 mL (IQR 1.96 to 7.97, n=90) for intraparenchymal haemorrhage and 4.57 mL (2.18 to 8.88, n=160) for extra-axial haemorrhage. For further discussion regarding absolute volume error see Section A.1. Regarding the reproducibility of the manual annotation procedure, for intra-rater reproducibility (n=20) and inter-rater variability (n=25), we obtained

Table 3.2: Evaluation Metrics stratified by volume threshold. Numbers presented as number of scans (n), mean (standard deviation), median (range). For lesion change the volume group is defined by the initial lesion volume.

class	thresh	DSC (%)			Absolute Error (mL)			Lesion Change Abs Error (mL)		
		count	mean	median	count	mean	median	count	mean	median
lesion (any)	all	599	35.3 (30.4)	36.0 (63.4)	655	5.32 (10.73)	0.74 (5.76)	98	3.57 (6.00)	0.47 (3.38)
	>0mL	469	45.0 (27.3)	49.4 (47.6)	469	7.26 (12.07)	2.32 (8.36)	73	4.45 (6.20)	1.46 (4.72)
	>1mL	328	56.5 (21.2)	59.3 (30.4)	328	10.24 (13.37)	5.72 (11.11)	45	5.92 (6.39)	3.10 (9.27)
	>5mL	251	62.2 (17.7)	66.9 (22.8)	251	12.95 (14.21)	8.27 (12.36)	34	7.31 (6.69)	3.94 (10.63)
	>10mL	210	65.1 (16.1)	69.0 (20.0)	210	14.83 (14.81)	9.33 (12.58)	26	7.47 (6.79)	3.94 (10.63)
	>25mL	134	68.2 (14.5)	70.7 (19.2)	134	19.84 (16.37)	15.68 (15.12)	13	11.08 (7.35)	12.73 (13.79)
IPH	all	361	37.6 (32.9)	41.9 (67.6)	655	1.03 (3.05)	0.03 (0.52)	98	0.81 (2.20)	0.03 (0.37)
	>0mL	306	44.4 (31.3)	51.9 (64.4)	306	2.15 (4.18)	0.54 (1.91)	40	1.73 (3.16)	0.31 (1.99)
	>1mL	167	61.7 (20.9)	65.2 (27.2)	167	3.77 (5.12)	1.82 (3.04)	17	3.71 (4.05)	2.24 (2.65)
	>5mL	90	68.2 (17.2)	72.6 (23.5)	90	5.97 (6.14)	3.57 (5.71)	13	3.30 (3.17)	2.24 (2.65)
	>10mL	57	71.0 (17.2)	76.0 (23.0)	57	8.20 (6.69)	6.13 (7.85)	8	3.25 (2.46)	2.65 (2.40)
	>25mL	19	70.7 (16.5)	73.7 (23.3)	19	13.27 (8.23)	14.41 (11.57)	2	3.01 (0.77)	3.01 (0.77)
EAH	all	559	28.7 (29.8)	20.7 (53.9)	655	2.38 (6.89)	0.34 (1.82)	98	1.25 (2.28)	0.31 (1.07)
	>0mL	402	39.9 (28.1)	40.7 (50.1)	402	3.72 (8.49)	1.09 (3.37)	67	1.80 (2.59)	0.85 (1.95)
	>1mL	262	54.0 (21.8)	55.3 (31.9)	262	5.49 (10.07)	2.43 (4.64)	36	2.29 (2.66)	1.36 (2.30)
	>5mL	160	63.5 (18.6)	67.5 (25.7)	160	8.10 (12.15)	4.57 (6.70)	24	2.98 (3.00)	2.32 (2.71)
	>10mL	127	67.0 (17.0)	70.4 (21.7)	127	9.49 (13.25)	5.59 (7.55)	15	3.26 (3.59)	2.16 (2.67)
	>25mL	61	71.9 (16.5)	75.7 (15.7)	61	14.63 (17.42)	8.85 (10.99)	4	3.80 (5.16)	0.97 (3.13)
oedema	all	355	27.7 (29.0)	17.5 (52.6)	655	2.34 (5.76)	0.02 (1.62)	98	2.21 (4.49)	0.12 (1.93)
	>0mL	287	34.2 (28.6)	32.0 (54.6)	287	5.17 (7.69)	2.20 (5.86)	35	4.59 (5.21)	3.03 (7.85)
	>1mL	208	41.6 (27.8)	44.8 (48.5)	208	7.05 (8.29)	3.93 (7.20)	25	6.33 (5.24)	4.18 (7.18)
	>5mL	137	51.3 (24.1)	54.6 (36.1)	137	9.72 (9.09)	6.87 (8.23)	14	7.49 (4.99)	7.26 (5.63)
	>10mL	95	55.2 (23.0)	59.2 (26.3)	95	12.11 (9.93)	9.46 (11.49)	9	8.85 (5.22)	8.73 (5.77)
	>25mL	36	61.2 (22.6)	64.4 (31.8)	36	18.45 (12.57)	18.54 (14.19)	4	10.86 (6.54)	8.88 (5.99)
IVH	all	237	12.7 (22.3)	0.0 (21.2)	655	0.09 (0.54)	0.00 (0.02)	98	0.15 (0.81)	0.00 (0.03)
	>0mL	96	31.4 (25.4)	30.1 (55.5)	96	0.56 (1.30)	0.19 (0.33)	11	0.63 (1.48)	0.11 (0.13)
	>1mL	21	47.4 (17.6)	47.3 (22.2)	21	1.92 (2.31)	0.97 (0.82)	2	0.11 (0.08)	0.11 (0.08)

agreements in the range of 0.90-1.00 for all lesion types (see Table A.2).

98 patients in the test set who had repeat scans for which both timepoints could be established (196 scans) were included in the calculations of true and predicted volume difference. Figure 3.4 presents Bland-Altman plots of the agreement between the true and predicted lesion volume change. The mean difference was 0.46 mL (95% CI -4.04 to 4.97) for intraparenchymal haemorrhage, -0.37 mL (-5.42 to 4.69) for extra-axial haemorrhage, 0.68 mL (-9.03 to 10.39) for oedema, and 0.12 mL (-1.48 to 1.71) for intraventricular haemorrhage. In Sectionn A.2, we show that our algorithm enables localisation of lesions (*i.e.*, the quantification of lesion volume by brain region).

Table 3.3 and Figure 3.5 show the results of lesion volume classification and lesion detection for external validation. For image-level detection of lesions, we obtained an AUC of 0.89 (95% CI 0.86–0.91) for the any lesion class, 0.87 (0.85–0.90) for the intraparenchymal haemorrhage class, 0.89 (0.86–0.91) for the extra-axial haemorrhage class, 0.89 (0.86–0.92) for the oedema class, and 0.89

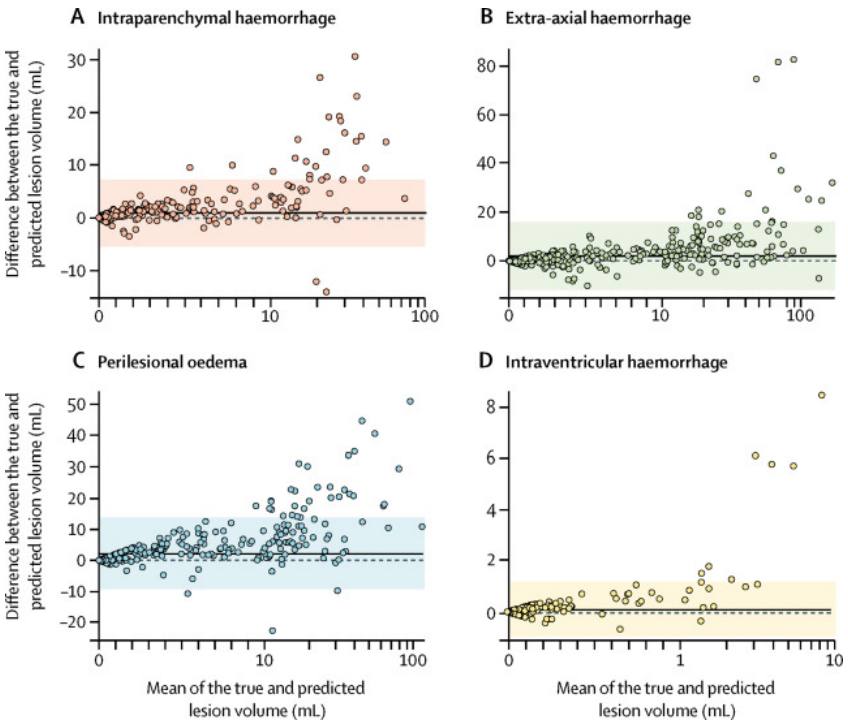


Figure 3.3: Bland-Altman plots for lesion volume estimation. The solid horizontal lines are means and the shaded regions are 95% CIs. The x-axes are on a logarithmic scale to improve visualisation. Axes are plotted on different scales across plots for clarity. Absolute volume errors are shown in Tables 3.2 and A.3.

(0.85–0.93) for the intraventricular haemorrhage class. For the 1 mL threshold, the AUCs increased to 0.96 (0.95–0.98), 0.99 (0.98–1.00), 0.97 (0.95–0.98), 0.94 (0.92–0.96), and 0.99 (0.95–1.00), indicating that most of the missed lesions are very small. For the classification of large lesions (> 25 mL), the AUCs were 0.99 (0.98–1.00) for any lesion, 0.99 (0.97–1.00) for intraparenchymal haemorrhage, 0.99 (0.98–1.00) for extra-axial haemorrhage, and 0.98 (0.95–1.00) for oedema. On the external validation set, we reported an AUC of 0.83 (95% CI 0.79–0.87) for the intracranial haemorrhage class, 0.90 (0.86–0.94) for the intraparenchymal haemorrhage class, 0.80 (0.75–0.85) for the extra-axial haemorrhage class, and 0.95 (0.89–1.00) for the intraventricular haemorrhage class.

3.4 Discussion

In this study, we found that the voxel-wise segmentation produced by a CNN can be used for volumetric quantification and detection and classification of multiclass TBI lesions in head CT, as well as for the assessment of lesion progression. We were able to accurately quantify and detect lesions on an external, independent dataset. To our knowledge, this is the largest study so far to use a ground truth reference of manually annotated and manually corrected automatic segmentations of CT scans. The size and diversity of this multicentre dataset provide insights into the performance of deep learning in a real-world clinical scenario. We extend findings from previous studies (Chil-

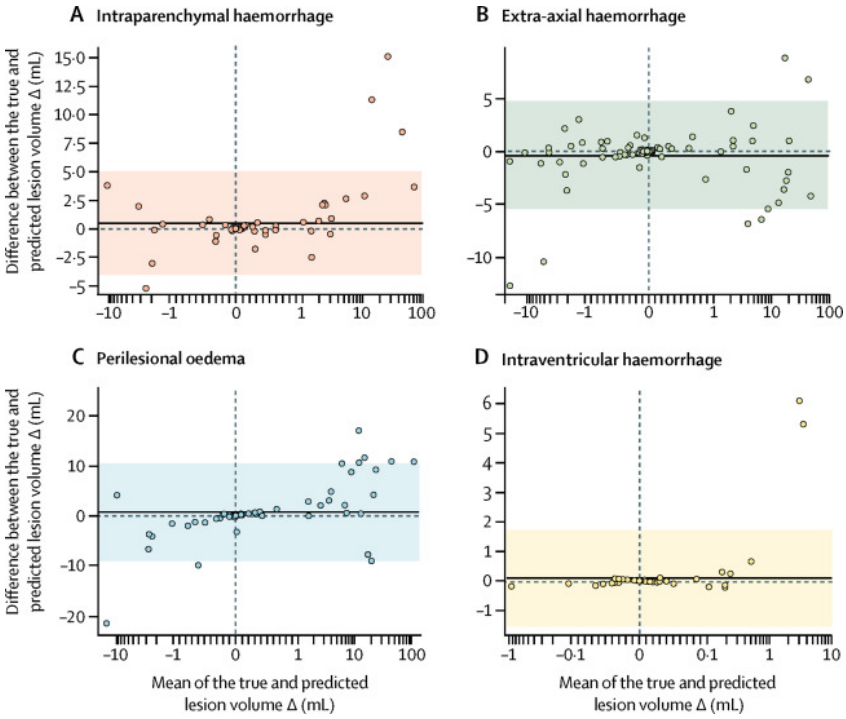


Figure 3.4: Bland-Altman plots for lesion progression. The solid horizontal lines are means and the shaded regions are 95% CIs. The x-axes are on a logarithmic scale to improve visualisation. Axes are plotted on different scales across plots for clarity. Absolute volume change errors are shown in Tables 3.2 and A.3.

amkurthy et al. 2018; Jain et al. 2019; Kuo et al. 2019) by providing quantitative volumetric results separately for intraparenchymal haemorrhage, extra-axial haemorrhage, intraventricular haemorrhage, and perilesional oedema.

The CNN provided a well calibrated prediction of lesion volume since differences between the true and predicted volumes were small when compared with the overall lesion volume. The funnelling observed can be explained by lesions being predicted where there were none and vice versa, which mostly occurs for smaller lesions. For comparison, previous work (Jain et al. 2019) reported a median absolute error of 8.83 mL (n=39) for intraparenchymal haemorrhage and extra-axial haemorrhage lesions combined while considering only lesions with a volume greater than 5.5 mL. In our analysis, we did fine-grained segmentation of these two classes individually and validated our CNN on a larger dataset. For lesions with a volume greater than 5 mL, our median absolute error was smaller than that reported previously (Jain et al. 2019) for intraparenchymal haemorrhage and extra-axial haemorrhage.

The potential clinical applicability of the volume estimates is further confirmed by our results on lesion progression. Such progression of intracranial lesions represents a major target for therapies in the acute phase. For example, cerebral contusions are common after TBI, occurring in up to two-thirds of patients admitted to hospital (Collaborators et al. 2004, 2005), and progression of such lesions is common, occurring in up to half of patients within the first 24-48 h (Kurland et al.

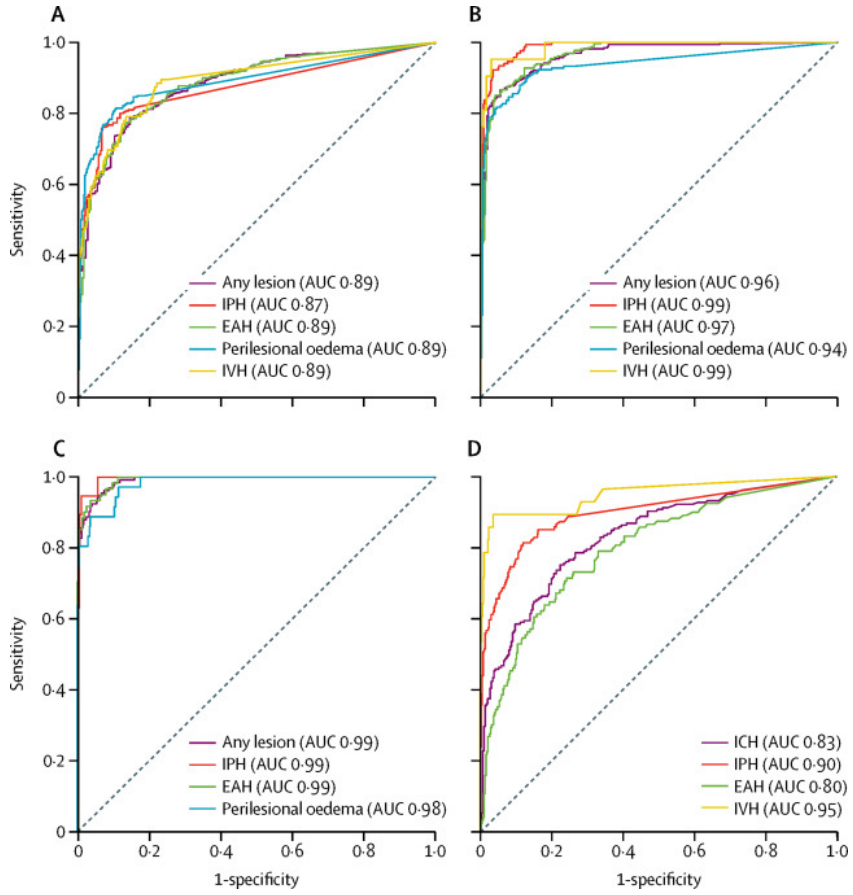


Figure 3.5: Receiver operating characteristic curves for lesion detection and classification. Classification of lesions with a volume greater than 0 mL (A), greater than 1 mL (B), and greater than 25 mL (C) on the internal validation set, and detection of lesions on the external validation set CQ500 (D). AUC=area under the curve. EAH=extra-axial haemorrhage. IPH=intraparenchymal haemorrhage. IVH=intraventricular haemorrhage.

2012; Narayan et al. 2008; Oertel et al. 2002). The ability to automatically monitor lesion progression offers key opportunities to improve patient stratification, guide and monitor management, and investigate potential causes and risk factors for lesion progression in large cohort studies such as CENTER-TBI (Steyerberg et al. 2019). Until now, the identification of factors that predict or cause contusion progression, or both, has been hampered by the need to estimate lesion volume and change manually, restricting analyses to small sample sizes (Kurland et al. 2012; Narayan et al. 2008; Oertel et al. 2002).

Regarding the underlying lesion segmentation, the DSC increased with lesion volume, illustrating that the DSC is sensitive to small or non-existent lesions, which is a limitation of the metric. The median DSC of 73.% (n=39) reported previously for large intraparenchymal haemorrhage and extra-axial haemorrhage lesions combined (lesion volume > 5.5 mL) (Jain et al. 2019) is similar to that found in our study.

The algorithm performed less well at quantifying perilesional oedema, and by extension mixed density lesions. However, the ability to undertake such quantification has not been reported pre-

Table 3.3: Multiclass detection and classification results for three volume thresholds and detection results for the external validation dataset CQ500. The high specificity and high sensitivity operating points were obtained using a cutoff of 0.90 or the closest possible available. The 0 mL threshold is equivalent to lesion detection. EAH=extra-axial haemorrhage. ICH=intracranial haemorrhage. IPH=intraparenchymal haemorrhage. IVH=intraventricular haemorrhage.

	Number of scans		**High-specificity operating point**		**High-sensitivity operating point**		**Area under the curve (95% CI)**
	Positives	Negatives	Mean sensitivity (95% CI)	Mean specificity (95% CI)	Mean sensitivity (95% CI)	Mean specificity (95% CI)	
>0 mL							
Any lesion	469	186	0.70 (0.66–0.74)	0.90 (0.85–0.94)	0.90 (0.87–0.93)	0.61 (0.54–0.68)	0.89 (0.86–0.91)
IPH	306	349	0.77 (0.72–0.82)	0.90 (0.87–0.93)	0.81 (0.76–0.85)	0.85 (0.80–0.88)	0.87 (0.85–0.90)
EAH	402	253	0.72 (0.67–0.76)	0.90 (0.86–0.94)	0.90 (0.87–0.93)	0.63 (0.57–0.69)	0.89 (0.86–0.91)
Perilesional oedema	287	368	0.80 (0.75–0.85)	0.90 (0.87–0.93)	0.85 (0.80–0.89)	0.82 (0.77–0.85)	0.89 (0.86–0.92)
IVH	96	559	0.70 (0.60–0.79)	0.90 (0.87–0.93)	0.90 (0.82–0.95)	0.75 (0.71–0.78)	0.89 (0.85–0.93)
>1 mL							
Any lesion	328	327	0.89 (0.85–0.92)	0.90 (0.86–0.93)	0.90 (0.87–0.93)	0.87 (0.83–0.91)	0.96 (0.95–0.98)
IPH	167	488	0.96 (0.92–0.98)	0.90 (0.87–0.93)	0.90 (0.85–0.94)	0.97 (0.94–0.98)	0.99 (0.98–1.00)
EAH	262	393	0.89 (0.85–0.93)	0.90 (0.87–0.93)	0.90 (0.86–0.93)	0.89 (0.85–0.92)	0.97 (0.95–0.98)
Perilesional oedema	208	447	0.86 (0.80–0.90)	0.90 (0.87–0.93)	0.90 (0.86–0.94)	0.86 (0.83–0.89)	0.94 (0.92–0.96)
IVH	21	634	0.95 (0.76–1.00)	0.90 (0.87–0.92)	0.90 (0.70–0.99)	0.97 (0.95–0.98)	0.99 (0.95–1.00)
>25 mL							
Any lesion	134	521	0.98 (0.94–1.00)	0.90 (0.87–0.92)	0.90 (0.84–0.95)	0.96 (0.94–0.98)	0.99 (0.98–1.00)
IPH	19	636	1.00 (0.82–1.00)	0.90 (0.88–0.92)	0.95 (0.74–1.00)	0.94 (0.92–0.96)	0.99 (0.97–1.00)
EAH	61	594	0.98 (0.91–1.00)	0.90 (0.87–0.92)	0.90 (0.80–0.96)	0.97 (0.96–0.99)	0.99 (0.98–1.00)
Perilesional oedema	36	619	0.89 (0.74–0.97)	0.90 (0.88–0.92)	0.92 (0.78–0.98)	0.89 (0.87–0.92)	0.98 (0.95–1.00)
External validation set CQ500							
ICH	205	285	0.59 (0.51–0.65)	0.90 (0.86–0.93)	0.90 (0.85–0.94)	0.51 (0.45–0.56)	0.83 (0.79–0.87)
IPH	134	356	0.76 (0.68–0.83)	0.90 (0.87–0.93)	0.89 (0.82–0.94)	0.74 (0.69–0.79)	0.90 (0.86–0.94)
EAH	119	371	0.49 (0.39–0.58)	0.90 (0.87–0.93)	0.91 (0.84–0.95)	0.38 (0.33–0.43)	0.80 (0.75–0.85)
IVH	28	462	0.89 (0.72–0.98)	0.90 (0.87–0.93)	0.93 (0.76–0.99)	0.68 (0.63–0.72)	0.95 (0.89–1.00)

viously; hence, we are unable to benchmark it against previous work. Although detection and delineation of high-intensity haemorrhagic lesions are straightforward, precise delineation of hypointense oedema can be challenging, even for radiologists. The ability of our algorithm to do this task, in addition to quantifying other lesion types, may be important for prognostication, aid detection and avoidance of secondary injury, the evaluation of neuroprotective measures, and as an intermediate biomarker for clinical trials aimed at the reduction of cerebral oedema and contusion growth (Mathieu, Zeiler, et al. 2020).

The accuracy of our CNN was lower in segmenting small haemorrhagic lesions. From a clinical perspective, however, this reduced accuracy is mitigated by the fact that the volume of these small lesions is less important in terms of prognostication or deciding on therapy. These small lesions are typically microhaemorrhages associated with diffuse vascular injury and are clinically used as a surrogate marker for diffuse axonal injury. Consequently, their clinical significance is dependent on number and distribution, rather than volume of individual lesions (Haacke et al. 2010).

Although our model was not designed for classification specifically, as a byproduct of the segmentation algorithm, it is able to do so with comparable performance to state-of-the-art methods

developed solely for detection (Chilamkurthy et al. 2018; Kuo et al. 2019). On the CQ500 dataset, previous work (Chilamkurthy et al. 2018) reported an AUC of 0.94 (95% CI 0.92–0.97) for intracranial haemorrhage, 0.95 (0.93–0.98) for intraparenchymal haemorrhage, 0.95 (0.91–0.99) for subdural haematoma, 0.97 (0.91–1.00) for extradural haematoma, 0.97 (0.92–0.99) for traumatic subarachnoid haemorrhage, and 0.93 (0.87–1.00) for intraventricular haemorrhage (Chilamkurthy et al. 2018). Apart from the intraventricular haemorrhage class, the AUCs we report on the same data are lower. However, our algorithm also has the ability to quantify lesion volume, shape, and location, which can be used to extract other radiological features of potential interest. Additionally, our results are not directly comparable with the previous work by Chilamkurthy et al. (2018) because they used certain rules to select the optimum scan per patient processed by their algorithm and we were not able to determine those rules for comparison. Instead, we processed all available scans for each patient (up to eight) and calculated the mean predicted volume for subsequent classification. Using a selected set of scans, as done in previous work, is likely to improve our results.

The ability to distinguish between different lesion types is important to aid understanding of pathophysiology and to implement personalised care. The heterogeneity of TBI is well described, encompassing a wide spectrum of pathologies, from axonal injury to focal contusions and extracranial bleeding. The large annotated dataset used in this study is representative of this clinical spectrum. The CENTER-TBI study (Ma et al. 2015; Steyerberg et al. 2019) allowed a large variety of vendors and acquisition protocols to be used. Images in this analysis were contributed from 38 centres. Consequently, the performance is not manufacturer or acquisition dependent. The ability to generalise is supported by validation on an external, independent dataset from a different continent, for which the results for lesion detection were comparable with the results obtained on internal data.

Adding the ability to distinguish the different types of extra-axial haemorrhage is important, particularly given that extradural haematomas portend a better prognosis, and the presence of traumatic subarachnoid haemorrhage is a marker for worse outcomes in prognostic models (Collaborators et al. 2004, 2005; Murray et al. 2007). Furthermore, expanding on the capability of lesion localisation may help answer key research questions and support clinical reporting of scans.

Future work needs to focus on the optimal incorporation of such algorithms into clinical practice, which must be accompanied by a rigorous assessment of performance, strengths, and weaknesses. Such algorithms will find clear research applications, and, if adequately validated, may be used to help facilitate radiology workflows by flagging scans that require urgent attention, aid reporting in resource-constrained environments, and detect pathoanatomically relevant features for prognostication and a better understanding of lesion progression.

Chapter 4

Stochastic Segmentation Networks: modelling spatially correlated aleatoric uncertainty

This chapter is based on the following publication:

- **Monteiro, M.**, Le Folgoc, L., Coelho de Castro, D., Pawlowski, N., Marques, B., Kamnitsas, K., Wilk, M. van der, and Glocker, B. (2020). "Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty". In: *Advances in Neural Information Processing Systems* 33, pp. 12756–12767

MM and LL developed the idea. MM did the implementation, experiments, and wrote the manuscript. MM, NP and MvdW conceived the toy experiment. BM prepared the data for the LIDC experiment. LL, DCC, NP, BM, KK, MvdW and BG provided useful discussion and feedback to MM on the method, experiments and manuscript. BG secured the funding.

Code available at:

- https://github.com/biomedia-mira/stochastic_segmentation_networks

4.1 Introduction

The task of semantic image segmentation is a highly structured prediction problem where the output label maps should capture the spatial consistency of the objects to be segmented. While casting image segmentation as a dense pixel-wise classification task is at the heart of most machine learning approaches (Criminisi et al. 2012; Everingham et al. 2015; Long et al. 2015), this paradigm largely ignores the underlying spatial structure. Methods will often rely on inductive biases to capture structure as opposed to modelling it directly. While this approach may yield reasonable, single deterministic predictions, it is insufficient to model the underlying distribution over multiple plausible outputs. In image segmentation, there is often more than one plausible solution for a given input. The exact location of object boundaries is often ambiguous, and ideally, the model should be able to capture this inherent uncertainty.

Uncertainty can be decomposed into aleatoric, which is inherent to the observations, and epistemic

uncertainty, which relates to the ambiguity about the model's parameters and can be explained away with more data (Kendall and Gal 2017), e.g., a noisy regression problem with many data points has low epistemic but high aleatoric uncertainty. In segmentation, aleatoric uncertainty is both spatially correlated and heteroscedastic, since an image can have both regions with higher and lower uncertainty. The ideal model should represent the joint probability distribution of the labels at every pixel given the image, enabling sampling multiple plausible label maps.

Because aleatoric uncertainty cannot be reduced by acquiring more data, modelling it explicitly is crucial in risk-sensitive applications. In medical imaging, the images are often noisy, and the boundaries between tissue types may not be well defined, which leads to disagreement even between experts. The ability to automatically generate multiple plausible hypotheses to choose from is of high value in applications such as radiotherapy, where trade-offs have to be made about which anatomical regions to include for invasive treatment. Additionally, providing confidence intervals alongside tumour boundaries would allow uncertainty to be taken into account when making critical decisions.

Fully convolutional neural networks (FCNNs) are the state-of-the-art for semantic segmentation (L. Chen et al. 2018; Long et al. 2015; Ronneberger et al. 2015). In principle, FCNNs are probabilistic models, since their output is a set of independent categorical distributions per pixel, parameterised by a softmax layer. Because these distributions are independent given the last layer's activations, sampling from this model would result in spatially incoherent segmentations (grainy label noise in the uncertain regions). We argue that any method that only produces independent pixel-wise uncertainty estimates is unable to generate spatially coherent label maps, and thus incapable of fully capturing the structured uncertainty.

Recent work extends FCNNs to model the joint distribution over labels given the image, allowing for multiple plausible segmentations (Baumgartner et al. 2019; Kohl, Romera-Paredes, et al. 2019, 2018). These methods have rigid, hierarchical, memory-intensive architectures, loss functions with manually tuned hyper-parameters, and require one partial forward pass per new sample. We introduce stochastic segmentation networks (SSNs), a lightweight and flexible alternative that efficiently captures correlations between pixels by modelling the logit map as a low-rank multivariate normal distribution. In contrast with previous approaches, our method is less complex, achieves higher predictive performance and can generate multiple samples from a single forward pass. In addition, it can be used with any existing architecture, and its efficiency makes it applicable to high-dimensional problems such as 3D imaging.

4.2 Related Work

In data constrained scenarios, Bayesian methods are useful for quantifying epistemic uncertainty for previously unseen examples. Seminal works by MacKay (1992) and Neal (1993) inspired infer-

ence methods in Bayesian deep learning such as Markov chain Monte-Carlo (Ma et al. 2015; Welling and Teh 2011) and variational inference methods (Blundell et al. 2015; Gal and Ghahramani 2016). These methods focus on estimating the posterior over the weights of a neural network which allows for estimating epistemic uncertainty independently of the task. Ensemble (Lakshminarayanan et al. 2017) and multi-head (Lee, Purushwalkam, et al. 2015; Lee, Purushwalkam Shiva Prakash, et al. 2016; Rupprecht et al. 2017) methods follow a frequentist approach to modelling the weight distributions. In the case of label disagreement or noise, defined as aleatoric uncertainty, the issue is not the lack of data. Still, both uncertainties are complementary. In classification, there is work on estimating aleatoric uncertainty by predicting Dirichlet distributions (Malinin and Gales 2018, 2019; Sensoy et al. 2018) as well as post-training calibration of the predicted class probabilities (Guo et al. 2017; Kull et al. 2019). In segmentation, attempts at quantifying aleatoric uncertainty on a pixel-wise level (Jungo et al. 2020; Kendall and Gal 2017; Tanno et al. 2017; Wang et al. 2019) ignore the joint distribution over labels.

Historically, probabilistic graphical models (PGMs) such as conditional random fields (CRFs) (Blake et al. 2011; Krähenbühl and Koltun 2011) have been used to explicitly model the joint probability distribution over labels. However, the inference was mostly limited to predicting the maximum *a posteriori* (MAP) estimate. Although there is work on obtaining the M-best diverse solutions for a given input image (Batra et al. 2012; Kirillov et al. 2015), these models are restricted to a fixed number of solutions and have computationally expensive inference. Work on combining PGMs and FCNNs to enforce label dependencies as a post-processing step (Arnab et al. 2018; L. Chen et al. 2018; Kamnitsas, Ledig, et al. 2017) or even within a single model (Zheng et al. 2015) suffers from the same limitations as classic PGMs when quantifying aleatoric uncertainty.

Recently, Kohl, Romera-Paredes, et al. (2019, 2018) and Baumgartner et al. (2019) have built on conditional variational auto-encoders (Kingma and Welling 2014; Sohn et al. 2015) to extend FCNNs for modelling spatially correlated aleatoric uncertainty. Hu et al. (2019) extend this framework by regressing the uncertainty maps in a supervised manner. Zhu, Zhang, et al. (2017) also make use of deep generative models for the related task of image-to-image translation with multiple possible outputs for a single input. These methods encode the image into one or more uncorrelated multivariate normal latent variables and rely on the decoder to translate the added uncorrelated stochasticity into meaningful spatial variation. Like variational auto-encoders, these models have the flexibility to transform the latent distributions into arbitrarily complex distributions with correlations between pixels. However, the placement of the latent variables within the network means that one partial forward pass is required for every new sample. Furthermore, this flexibility comes at the cost of having to use a cumbersome variational inference framework which makes use of a training-only posterior network and manually tuned hyper-parameters weighing the Kullback-Leibler divergence regularisation term of the loss. These overly expressive distributions might not justify their cost, a more constrained distribution could suffice and allow the use of simpler inference methods.

4.3 Methods

4.3.1 Background

We start by analysing the independence assumptions made to obtain the cross-entropy loss typically used in image segmentation. Consider a standard segmentation problem in which an image, $\boldsymbol{x}$, with K channels and S pixels, maps to a one-hot label map of the same size, $\boldsymbol{y}$, with C classes: $\boldsymbol{x}_i \in \mathbb{R}^K$ and $\boldsymbol{y}_i \in \{0,1\}^C$ for $i \in \{1,\dots,S\}$. In a classic CNN, the probability of one label, $p(\boldsymbol{y}_i|\boldsymbol{x})$, is the output of a softmax layer taking as input the logit, $\boldsymbol{\eta}_i$. Before any independence assumptions, the MAP estimate for the negative log-likelihood can be written as:

$$-\log p(\boldsymbol{y}|\boldsymbol{x}) = -\log \int p(\boldsymbol{y}|\boldsymbol{\eta}) p_\phi(\boldsymbol{\eta}|\boldsymbol{x}) d\boldsymbol{\eta}\,, \tag{4.1}$$

where $p_\phi(\boldsymbol{\eta}|\boldsymbol{x})$ is the probability of the logit map given the image under a model with parameters ϕ. To obtain the standard cross-entropy loss, we assume that the logit map is given by a deterministic function, $\boldsymbol{\eta} = f_\phi(\boldsymbol{x})$, which means $p_\phi(\boldsymbol{\eta}|\boldsymbol{x})$ can be written as:

$$p_\phi(\boldsymbol{\eta}|\boldsymbol{x}) = \delta_{f_\phi(\boldsymbol{x})}(\boldsymbol{\eta}) = \prod_{i=1}^{S} \delta_{[f_\phi(\boldsymbol{x})]_i}(\boldsymbol{\eta}_i)\,. \tag{4.2}$$

Due to this deterministic function, given the image and model, the logits, $\boldsymbol{\eta_i}$ for $i \in \{1,\dots,S\}$, are conditionally independent of each other, i.e., given the image and model, no new information can be gained about a single logit by observing its neighbours. Secondly, we must assume that the labels, $\boldsymbol{y}_i$ for $i \in \{1,\dots,S\}$, are independent of each other when given their respective logit:

$$p(\boldsymbol{y}|\boldsymbol{\eta}) = \prod_{i=1}^{S} p(\boldsymbol{y}_i|\boldsymbol{\eta}) = \prod_{i=1}^{S} p(\boldsymbol{y}_i|\boldsymbol{\eta}_i)\,. \tag{4.3}$$

This is a two-part assumption: first, it assumes that labels, $\boldsymbol{y}_i$, are independent of each other when given the full logit map, $\boldsymbol{\eta}$, and second, it assumes that each label, $\boldsymbol{y}_i$, only depends on its respective logit, $\boldsymbol{\eta}_i$, i.e., no new information can be gained about a label by observing the true values of its neighbours. Incorporating the assumptions of Equations 4.2 and 4.3 into Equation 4.1, and substituting $p(\boldsymbol{y}_i|\boldsymbol{\eta}_i)$ by a categorical distribution parameterised by the softmax transform of $\boldsymbol{\eta}_i$, we arrive at the familiar form for the cross-entropy:

$$-\log \prod_{i=1}^{S} p(\boldsymbol{y}_i|\boldsymbol{\eta}_i) = -\log \prod_{i=1}^{S}\prod_{c=1}^{C} (\mathrm{softmax}(\boldsymbol{\eta}_i)_c)^{y_{ic}} = -\sum_{i=1}^{S}\sum_{c=1}^{C} y_{ic} \log \mathrm{softmax}(\boldsymbol{\eta_i})_c\,. \tag{4.4}$$

Whereas in image-level classification these independence assumptions may be valid, in segmentation the labels at each pixel are clearly correlated, which should be taken into account.

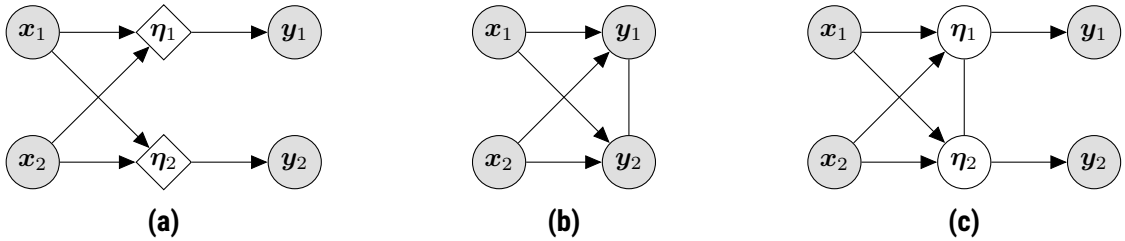


Figure 4.1: Probabilistic graphical model for a two-pixel segmentation problem: (a) neural network; (b) conditional random field; (c) proposed. $\boldsymbol{x}$ is the image, $\boldsymbol{y}$ the label map and $\boldsymbol{\eta}$ the logits. Circles represent random variables and rhombi represent deterministic variables. Shaded variables are observed and unshaded variables are unobserved.

4.3.2 Stochastic segmentation networks

In this work, we propose using weaker independence assumptions by using a more expressive distribution over logits. Specifically, we use a multivariate normal distribution whose parameters are the output of a neural network $\boldsymbol{\eta}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\Sigma}(\boldsymbol{x}))$, where $\boldsymbol{\mu}(\boldsymbol{x}) \in \mathbb{R}^{S\times C}$ and $\boldsymbol{\Sigma}(\boldsymbol{x}) \in \mathbb{R}^{(S\times C)^2}$. A non-diagonal multivariate normal distribution is the simplest distribution that models dependencies between pixels. However, the size of the full covariance matrix scales with the square of the number of pixels times the number of classes making it infeasible to compute for anything but very small images. For this reason, we use a low-rank parameterisation of the covariance matrix of the form:

$$\boldsymbol{\Sigma} = \boldsymbol{P}\boldsymbol{P}^T + \boldsymbol{D}\,, \tag{4.5}$$

where the covariance factor, $\boldsymbol{P}$, is a matrix of size $(S \times C) \times R$, where R is a hyper-parameter defining the rank of the parameterisation, and $\boldsymbol{D}$ is a diagonal matrix whose diagonal has $S \times C$ elements. Note that the covariance matrix dependencies are not only spatial but also class-wise. This low-rank parameterisation ensures that the three components describing the distribution: the mean, covariance factor, and covariance diagonal can be efficiently computed by a neural network.

By plugging this distribution into Equation 4.1, we no longer assume that logits, $\boldsymbol{\eta}_i$, are independent of each other. However, the integral also becomes intractable because of the softmax transform on the normal distribution. For this reason, we approximate the integral using Monte-Carlo integration:

$$-\log \int p(\boldsymbol{y}|\boldsymbol{\eta})p(\boldsymbol{\eta}|\boldsymbol{x})d\boldsymbol{\eta} \approx -\log \frac{1}{M}\sum_{m=1}^{M} p(\boldsymbol{y}|\boldsymbol{\eta}^{(m)}), \quad \boldsymbol{\eta}^{(m)}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\Sigma}(\boldsymbol{x}))\,, \tag{4.6}$$

where M is the number of Monte-Carlo samples used to approximate the integral. Because the distribution only has a few degrees of freedom, the Monte-Carlo integral has low variance. Making use of the assumptions in Equation 4.3 and using the $\mathrm{logsumexp}$ operator for numerical stability, we obtain a loss function which we can back-propagate through using the re-parameterisation

trick:

$$-\operatorname{logsumexp}_{m=1}^{M}\left(\sum_{i=1}^{S}\log(p(\boldsymbol{y}_i|\boldsymbol{\eta}_i^{(m)}))\right)+\log(M),\quad \boldsymbol{\eta}^{(m)}|\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}),\boldsymbol{\Sigma}(\boldsymbol{x})),\quad (4.7)$$

where $\log(p(\boldsymbol{y}_i|\boldsymbol{\eta}_i^{(m)}))$ can be solved as in Equation 4.4. For inference, with a single forward pass, we can sample from the distribution multiple times to obtain logit maps, which can be transformed into a probability or label maps. To obtain the most likely logit sample, we use the mean of the distribution.

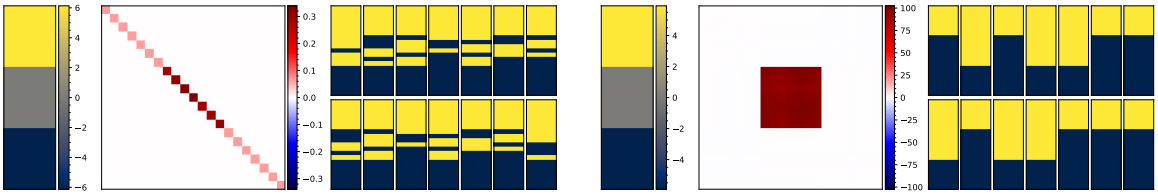
Figure 4.1 shows the probabilistic graphical models of a classic neural network, a CRF and the proposed model. While the neural network does not model dependencies between output labels, the CRF explicitly models these dependencies at the cost of an expensive inference procedure. In contrast, by implicitly modelling label dependencies in the logit space and then making independence assumptions, we can capture label dependencies while keeping the efficient inference of a neural network. The overhead of the proposed method is minimal: it involves using three maps instead of one at the end of the network, and sampling from a low-rank normal distribution to compute the loss, which is linear with the rank: $\mathcal{O}(rank)$. Thus, the overall cost is largely dominated by the underlying network.

4.4 Results

4.4.1 Toy problem

Consider a dataset on a one-dimensional 21-pixel line with one image for which there are two equiprobable label maps. For both label maps, the first third of the line is labelled 1 (on), and the last third is labelled 0 (off). However, the middle third is off for the first label map, and on for the second label map (see visual examples on the far right of Figure 4.2). In this setting, the labels of the middle third are uncertain but not independent. Since there is only one input, it is a constant and hence can be disregarded for further modelling. Thus, the goal of the problem becomes to find a generative model for the distribution of the two label maps.

A deterministic model would correctly learn the mean of the distribution but would yield implausible predictions. The first and last thirds would be correct, but the middle third would be arbitrarily fixed. For example, if the label maps were not equiprobable, the model would always generate the most probable one. Next, we consider two stochastic models where the distribution over logits is a multivariate normal distribution: one with a diagonal covariance matrix and one with a low-rank covariance matrix ($\mathrm{rank}=2$). We train these models with gradient descent and the loss function in Equation 4.7 using 200 Monte-Carlo samples for 10000 iterations. The results are shown in Figure 4.2. We observe that the diagonal model is able to learn the mean of the distribution and even which pixels have higher uncertainty. However, it cannot learn the structure of the noise and



(a) **(b)**

Figure 4.2: Toy problem results for the diagonal model (a) and a low-rank model (b). For both sub-figures, from left to right: mean, covariance matrix, and 14 random samples. The mean and samples are one-dimensional but expanded horizontally for improved visualisation. Colour bars indicate intensity values.

thus produces samples with uncorrelated noise. In contrast, the low-rank model is able to learn the correct noise structure and produce samples matching the desired distribution, yielding a higher log-likelihood, -0.93, when compared to the diagonal model, -4.87.

Caveat: Under our model, we can deduce that the true generative model is as follows: the mean is zero for the middle third, $+\infty$ for the first third, and $-\infty$ for the last the third. The covariance matrix is $+\infty$ for all entries regarding self and cross covariances of pixels in the middle third and zero elsewhere. This area of infinite covariance caused numerical stability issues since the covariance quickly grew to infinity producing overflow errors. Furthermore, we found that the covariance grew much faster than the mean causing the model to get stuck in suboptimal local minima. To address these issues, we pre-train the mean first and use early stopping to obtain the last model before an overflow error occurs. In the real data we used, the only area with infinite covariance is the air in the background of brain scans. We addressed the issue by masking out the background.

4.4.2 Lung nodule segmentation in 2D

To compare with previous work, we evaluated our model on the LIDC-IDRI dataset (Armato III et al. 2011) using the task defined by Kohl, Romera-Paredes, et al. (2018). The dataset consists of 1018 3D thorax CT scans where four radiologists have annotated multiple lung nodules in each scan. The dataset was annotated by 12 radiologists, and it is not possible to match an annotation to an expert. Thus, the four sets of annotations are not self-consistent in "style" across images. Regardless, this type of data is ideal for validating models which seek to capture the inherent uncertainty in the data — evident from the disagreement between experts. Kohl, Romera-Paredes, et al. (2018) preprocessed the data by extracting 2D slices centred around the annotated nodules. When at least one expert has segmented a nodule, a slice of the image and four expert segmentations were extracted. Empty segmentations were introduced when there were less than four annotations for a slice. This process yielded a dataset of 15096 slices each having four segmentations.

We compared with three baseline models: a deterministic U-Net (Ronneberger et al. 2015), a prob-

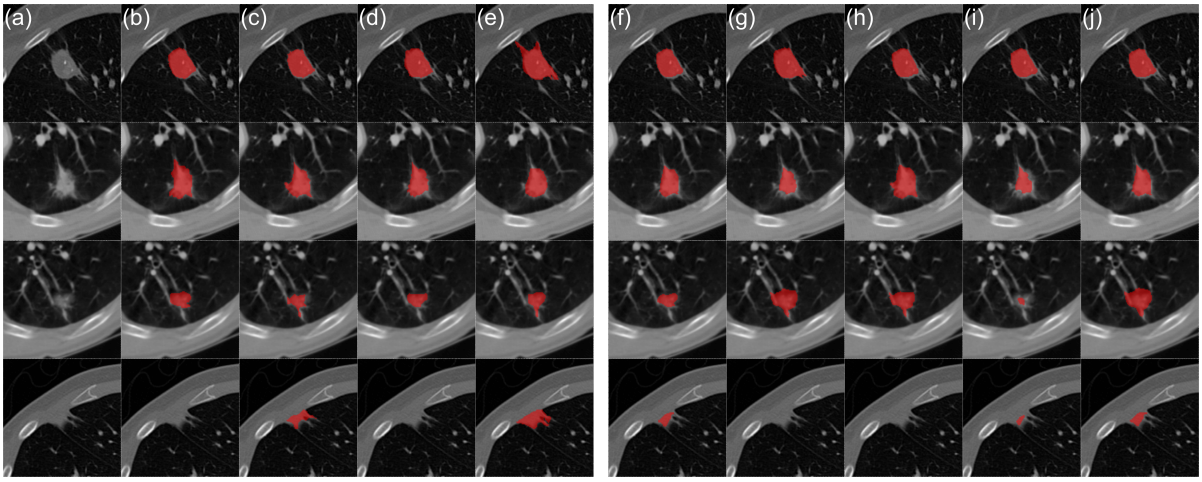


Figure 4.3: Qualitative results on the LIDC-IDRI dataset for the proposed model trained on four expert annotations: (a) CT image; (b-e) radiologist segmentations; (f) mean prediction; (g-j) samples.

abilistic U-Net (Kohl, Romera-Paredes, et al. 2018) and the PHiSeg model (Baumgartner et al. 2019) (the best performing variant reported). We used the pre-processed data provided by Kohl, Romera-Paredes, et al. (2018) and the code, configurations, and hyper-parameters provided by Baumgartner et al. (2019). The models were trained for 500000 iterations using the Adam optimiser (Kingma and Ba 2015) with a learning rate of 0.001 and a batch size of 12. The images were randomly augmented through flipping, rotating, and scaling. We implemented our algorithm on top of the provided deterministic U-Net with $\text{rank} = 10$ for the low-rank model, and, for comparison, we tested a model with a diagonal covariance matrix. By using the same backbone, code and hyper-parameters, we ensured a fair comparison with previous work.

We measured the predictive performance using the Dice Similarity Coefficient, $DSC = \frac{2TP}{2TP+FN+FP}$, where TP is true positives, FN is false negatives, and FP is false positives. Even if all four radiologists annotated a nodule, disagreements about its borders combined with the 3D to 2D preprocessing introduce several empty annotations (on average 1.6/4 = 40.4%). A non-empty prediction on an empty annotation results in a zero towards the average DSC, heavily penalising it. Therefore, we also report DSC_{nod} defined as the DSC computed only where the ground-truth annotations are not empty. Pixel-wise metrics for uncertainty quantification and calibration are not appropriate for spatially structured prediction such as segmentation. Hence, we used sample diversity to quantify the amount of uncertainty, and the distance between the expert and predicted distributions to quantify uncertainty calibration. Given the ground-truth distribution defined by the four expert segmentations, p, and the predicted distribution, $\hat{p}$, we measure the distance between the two using the generalised energy distance (Kohl, Romera-Paredes, et al. 2018; Székely and Rizzo 2013):

$$D^2_{\text{GED}}(p, \hat{p}) = 2 \mathop{\mathbb{E}}_{y\sim p, \hat{y}\sim\hat{p}}[d(y, \hat{y})] - \mathop{\mathbb{E}}_{y,y'\sim p}[d(y, y')] - \mathop{\mathbb{E}}_{\hat{y},\hat{y}'\sim\hat{p}}[d(\hat{y}, \hat{y}')], \tag{4.8}$$

where $d = 1 - \text{IoU}(\cdot, \cdot)$, if both segmentations are empty $d = 0$. In a multi-class setting, we

Table 4.1: Quantitative results on the LIDC-IDRI dataset for the five models trained on one set and four sets of annotations. Numbers are presented as mean ± standard error. Arrows in the column headers indicate the direction of increased performance.

model	trained on	DSC (%) ↑	DSC_{nod} (%) ↑	D^2_{GED} ↓	sample diversity
deterministic U-Net	set 0	37.5 ± 0.4	50.3 ± 0.4	0.698 ± 0.009	0.000 ± 0.000
probabilistic U-Net		38.4 ± 0.4	57.2 ± 0.4	0.516 ± 0.007	0.290 ± 0.004
PHiSeg		39.1 ± 0.4	51.3 ± 0.5	0.456 ± 0.008	0.215 ± 0.003
proposed (diagonal)		37.1 ± 0.4	51.2 ± 0.4	0.734 ± 0.009	0.001 ± 0.000
proposed (low-rank)		40.7 ± 0.4	58.6 ± 0.4	0.365 ± 0.005	0.399 ± 0.004
deterministic U-Net	all	35.9 ± 0.4	43.5 ± 0.5	0.607 ± 0.009	0.000 ± 0.000
probabilistic U-Net	sets	39.0 ± 0.4	50.6 ± 0.5	0.252 ± 0.004	0.469 ± 0.003
PHiSeg		33.8 ± 0.4	40.3 ± 0.5	0.224 ± 0.004	0.496 ± 0.003
proposed (diagonal)		37.0 ± 0.4	46.2 ± 0.5	0.622 ± 0.009	0.007 ± 0.001
proposed (low-rank)		43.6 ± 0.4	68.5 ± 0.3	0.225 ± 0.002	0.609 ± 0.002

average over the IoU of the individual classes, excluding the background class. We define sample diversity as $\mathbb{E}_{\hat{y},\hat{y}'\sim\hat{p}}[d(\hat{y},\hat{y}')]$. Note how both these metrics are bounded between zero and one.

To measure how models deal with increasing uncertainty in the labels, we trained each model using only one and all four annotations per image. We divided the data into train, validation and test sets (60/20/20%), and trained all models for 500k iterations with the same configuration described in Baumgartner et al. (2019). For the proposed loss function, we used 20 Monte-Carlo samples. We computed D^2_{GED} and sample diversity using 100 random samples. The prediction for the probabilistic baselines was obtained by averaging the probability maps of these samples (Baumgartner et al. 2019). For the proposed model, we used the mean of the logit map distribution. We computed the DSC between the prediction and the four ground-truths before averaging over sets of annotations and slices.

Table 4.1 shows the results for the five models and Figure 4.3 shows qualitative results for the proposed low-rank model trained on four sets of annotations. In terms of predictive performance, the proposed low-rank model outperformed the baselines for both settings. Of note, our model is the only method which benefits from the additional annotations yielding improved predictive performance. For uncertainty calibration, our model yielded the lowest D^2_{GED} except for the PHiSeg model with four annotations where their performance was comparable. In both settings, the proposed and baseline models obtained some measure of sample diversity, while the diagonal model nearly collapsed to a deterministic model, yielding very little sample diversity. For reference, the diversity between experts is 0.399 ± 0.002.

We note that the DSC reported for the baseline models is different from what is reported in the literature because we calculate DSC differently. The DSC (which is equivalent to the F1-score) reported in our work is lower than the results reported in PHiSeg (Baumgartner et al. 2019). The authors used a convention where the DSC is 1.0 if both the predicted and ground-truth slices are empty. We argue that this choice skews results since an algorithm that always predicts an empty label map would achieve an average DSC equal to the fraction of empty slices in the dataset, e.g. if the dataset has 40% of empty slices the average DSC is also 40%. In contrast, we used the standard definition of DSC, where these cases are undefined and thus excluded from the calculation of the average DSC. This changes the range of the numbers we report but not the underlying performance. When we calculated the DSC using the convention used in previous literature, we observed the baseline models performance to match that of what was previously reported in (Baumgartner et al. 2019).

4.4.3 Brain tumour segmentation in 3D

We also applied our method to the BraTS 2017 dataset (Bakas, Akbari, et al. 2017; Bakas, Reyes, et al. 2018; Menze et al. 2014). This dataset consists of 285 3D multimodal MRI images (four channels: T1, T1ce, T2 and Flair) where one radiologist has segmented four classes: background, non-enhancing/necrotic tumour core (NET), oedema (OD) and enhancing tumour core (ET). We implemented the proposed method on top of an implementation of DeepMedic (Kamnitsas, Ledig, et al. 2017; Kingma, Salimans, et al. 2016), a network specifically developed for brain segmentation. We use $\mathrm{rank} = 10$ for the low-rank model and omit the diagonal only model since it converged to a deterministic model. For an ablation study on the impact of the rank on the performance metrics see Section 4.4.4. The images have a resolution of $1\times1\times1$ mm and a size of $240\times240\times155$ voxels, making them too large to train on whole images. We trained the baseline and proposed models on image patches of 110 mm^3 (1mm^3 = 1 voxel), which, since no padding was used, result in label map patches of 30 mm^3. To test the effect of including longer distance dependencies between voxels, we also trained the proposed model on image patches of 140 mm^3 which result in label map patches of 60 mm^3. Note that, increasing the patch size of the baseline does not change its behaviour since the model is fully convolutional and its receptive field is 81 mm^3 (which is larger than 60 mm^3).

We split the data into training, validation and test sets (60/10/30%) and trained the models for

Table 4.2: Quantitative results on the BraTS 2017 dataset. Numbers are presented as mean ± standard error. Arrows in the column headers indicate the direction of increased performance.

model	DSC_{WT} (%) ↑	DSC_{NET} (%) ↑	DSC_{OD} (%) ↑	DSC_{ET} (%) ↑	D^2_{GED} ↓	sample diversity
Deepmedic	88.2 ± 1.3	60.5 ± 2.9	72.1 ± 2.3	67.3 ± 3.5	0.886 ± 0.043	0.000 ± 0.000
low-rank 30 mm	88.0 ± 1.3	59.3 ± 3.1	71.7 ± 2.3	68.7 ± 3.5	0.635 ± 0.029	0.312 ± 0.014
low-rank 60 mm	88.7 ± 1.3	59.6 ± 3.0	72.4 ± 2.2	69.2 ± 3.5	0.689 ± 0.031	0.217 ± 0.012

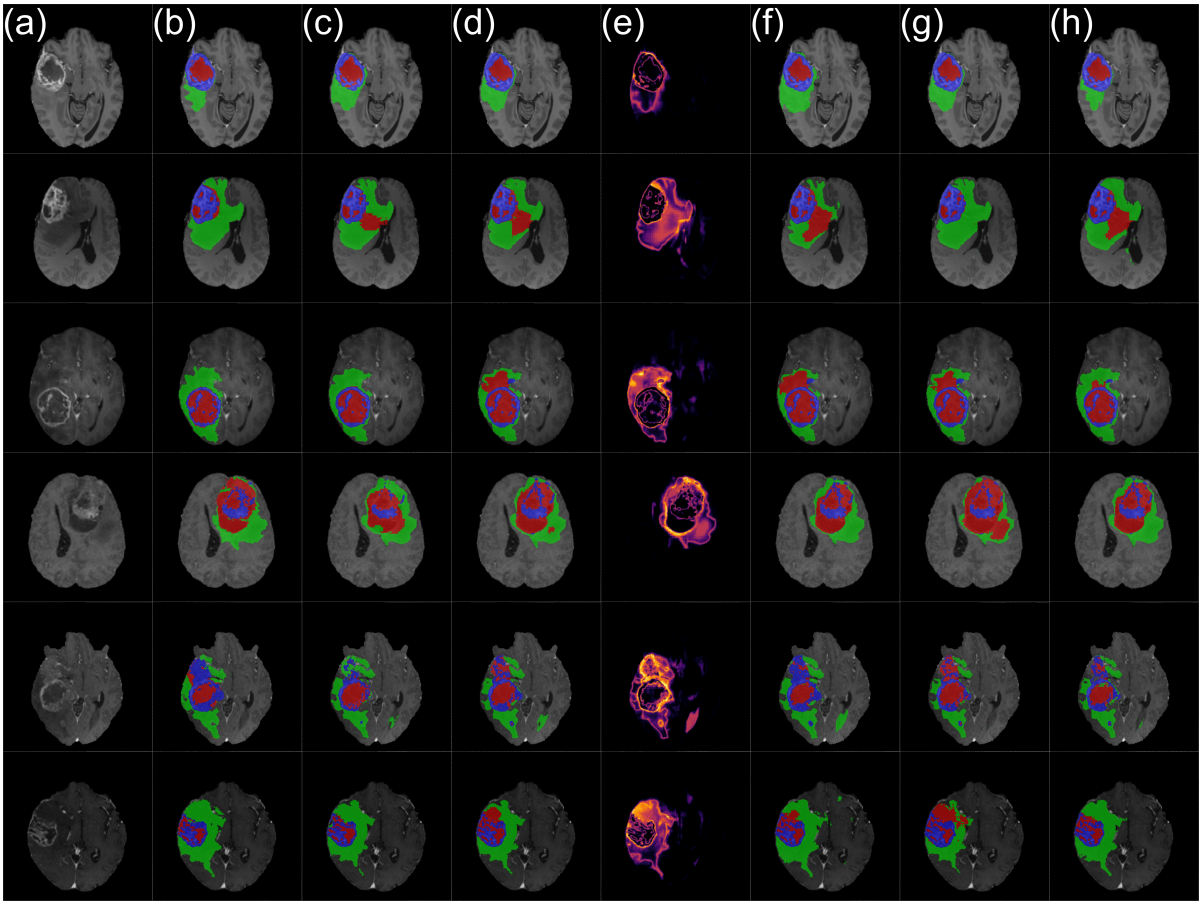


Figure 4.4: Qualitative results on the BraTS 2017 dataset for 30 mm^3 model: (a) T1ce slice; (b) ground-truth segmentation; (c) prediction of deterministic model; (d) prediction of proposed model; (e) marginal entropy; (f-h) samples. Samples were selected to show diversity.

1200 epochs. At each epoch, we randomly sampled 50 images and extracted 20 patches from each image. We randomly sampled the patches centred around a lesion or background voxel with equal probability. We used the RMSProp optimiser (Tieleman and Hinton 2012) with momentum 0.6 and a learning rate of 0.001 which we halved at the following epochs: 440, 640, 800, 900, 980, 1050. For augmentation, we used random elastic deformations, right-angle rotations, flips and linear intensity transformations. We used a batch size of 10, except for the 60 mm^3 model where we used a batch size of 4 due to GPU memory constraints. During inference, we stitched together the patches of the mean, covariance factor and diagonal to build a distribution over the entire image from which we can sample, this ensures no artefacts appear at patch borders. Due to the fully convolutional nature of the model, after it is trained, the patch size used for inference has no impact on the final result. We measured the DSC of the three lesion classes and the whole tumour (WT), consisting of all lesion combined. We measured sample diversity and D^2_{GED} using only 20 samples due to the quadratic dependency on the number of samples and the large image size. To calculate uncertainty maps, we used the marginal entropy of the categorical distributions

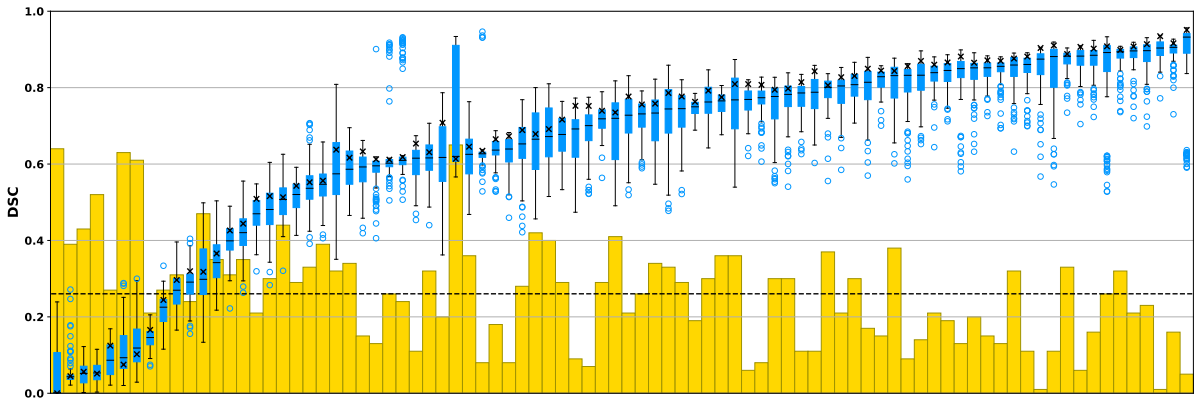


Figure 4.5: Distribution of sample average class DSC per case. The yellow bars denote the fraction of samples whose DSC is higher than the mean prediction, which is represented by a cross. The dashed line is the average fraction of samples better than the mean prediction (average height of the bars).

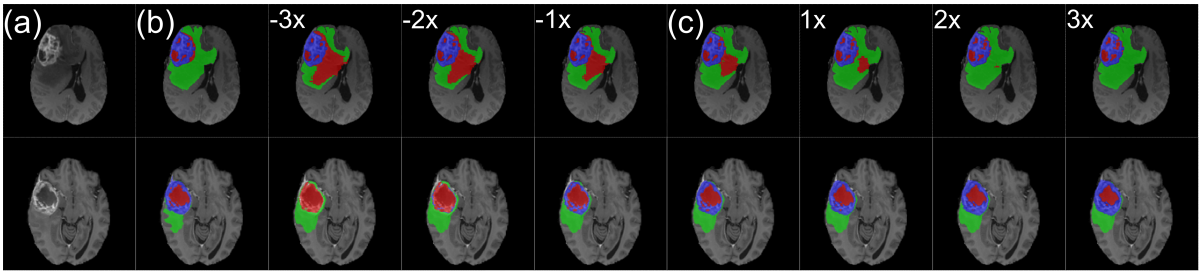


Figure 4.6: Sample manipulation after inference: (a) T1ce slice; (b) ground-truth segmentation; (c) sample surrounded by manipulated sample with scaling ranging from -3x to 3x.

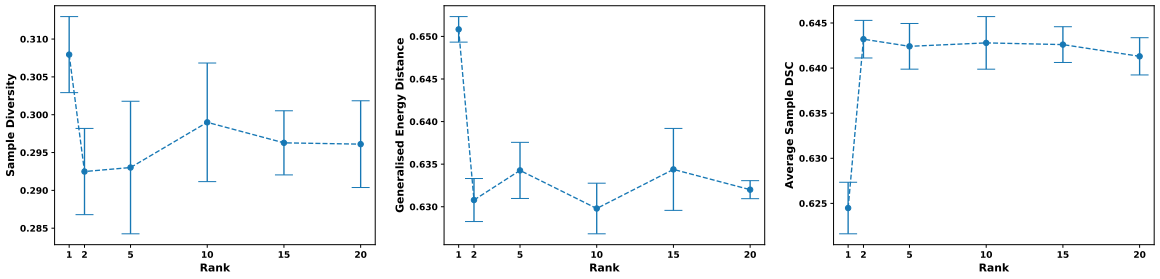
predicted for each voxel i:

$$H[y_i|\boldsymbol{x}] = \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[-\sum_{c=1}^{C} p(y_i=c|\boldsymbol{x})\log_C p(y_i=c|\boldsymbol{x})\right] \approx \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[-\sum_{c=1}^{C}\bar{p}_{ic}\log_C \bar{p}_{ic}\right], \tag{4.9}$$

where $\bar{p}_{ic} = \frac{1}{M}\sum_{m=1}^{M} p(y_i=c|\boldsymbol{\eta}_i^{(m)}) \approx \mathbb{E}_{p(\boldsymbol{\eta}|\boldsymbol{x})}[p(y_i=c|\boldsymbol{\eta}_i)] = p(y_i=c|\boldsymbol{x})$.

Table 4.2 shows the quantitative results for the deterministic and stochastic models. The stochastic models had no loss in performance when compared to the deterministic model. Comparing the two stochastic models, we observe that the added spatial context did not increase performance or yield a better-calibrated distribution. Regardless, the amount of needed spatial context is application dependent. Figure 4.4 shows qualitative results for six cases for the stochastic 30 mm^3 model. We observe entire structures in the segmentation appear and disappear between samples in regions of high uncertainty (e.g., row 4). Furthermore, mistakes made by the deterministic model or the stochastic model are corrected in at least one of the samples (e.g., row 2). Lastly, the high uncertainty in lesion borders makes them shrink and expand consistently between samples (e.g., row 1). For more samples see Figures B.1 – B.4.

Figure 4.5 shows per case sample distributions of the average lesion class DSC (100 samples).



(a) Sample diversity. **(b)** Generalised Energy Distance **(c)** Average Sample DSC

Figure 4.7: Impact of rank on different performance metrics for the BraTS dataset. Results are shown as mean and standard error over five random seeds.

As expected, for most cases, the majority of samples are worse than the mean prediction. However, on average, 26.0% samples are better than the mean prediction (dashed line). When looking at the best samples, the average (over the dataset) 95% quantile of the average class DSC was 70.3% when compared with the deterministic model average class DSC of 66.8%. This gain is not uniformly distributed as it tends to be higher for cases with low performance and decrease as the performance increases. In addition to being able to sample repeatedly after inference, another advantage of outputting a full distribution is the ability to manipulate samples post-inference. Since the covariance matrix has entries which are separable per class, by scaling only the part of the matrix relating to a given class, we are able to manipulate samples to increase or reduce the presence of that class. This can be used to correct possible mistakes or adjust borders, as shown in Figure 4.6. Similarly, we can trade sample diversity for quality by scaling the temperature of the entire distribution.

4.4.4 Rank ablation study

Intuitively, the rank of the multivariate normal distribution controls the number of independent clusters of pixels that are controlled together, thus, limiting the maximum possible sample complexity. In this section, we provide an ablation study of how the rank of the multivariate normal distribution impacts the performance metrics on the BraTS dataset using models trained on 110 mm image patches. Figure 4.7 shows the sample diversity, generalised energy distance and average class DSC for different six rank values: $rank \in [1, 2, 5, 10, 15, 20]$. The results are shown as the mean and standard error over five random seeds, that is $6 \times 5 = 30$ total training runs. We observe that as long as the rank is greater than one, there seems to be no clear relation between the rank and the performance metrics. From Figure 4.8, we see that increasing the rank increases the visual sample complexity, with more intricate structures appearing. Even though we haven't quantified sample complexity, we speculate that the increase in sample complexity does not improve performance because the structure of the aleatoric uncertainty in this dataset is very simple. This property is dataset-specific, which should be taken into account when choosing the rank for a new dataset.

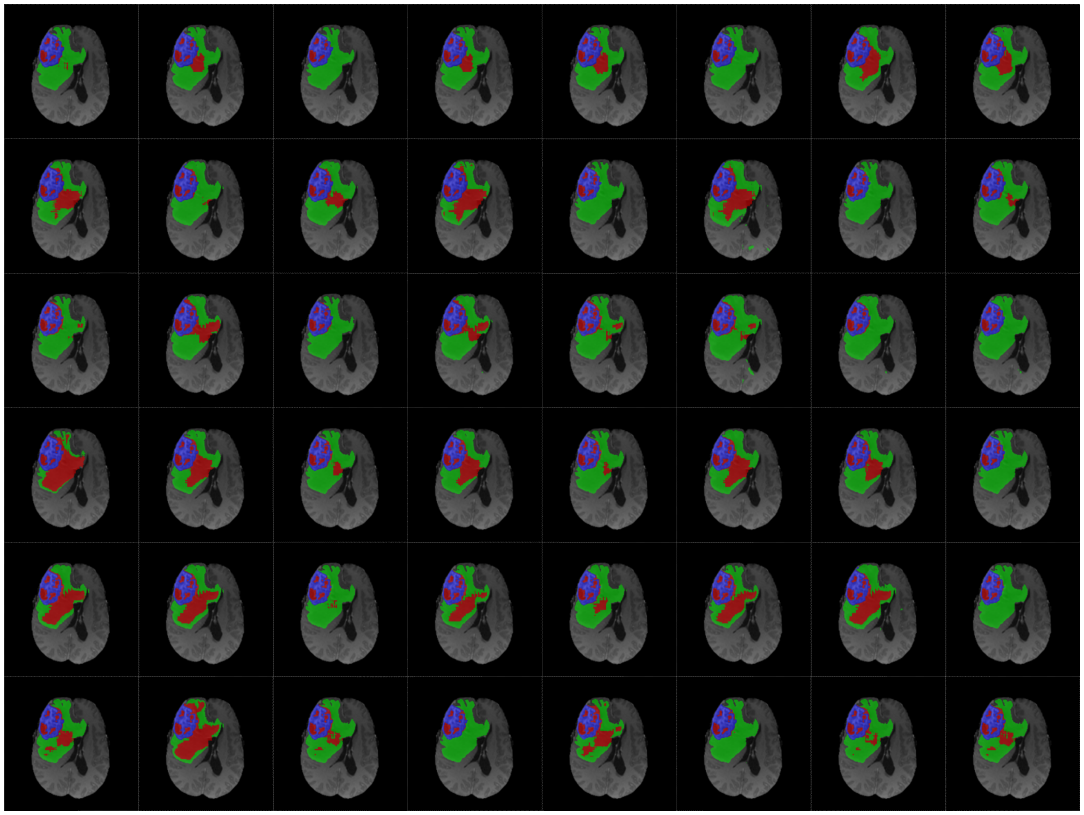


Figure 4.8: Visual impact of rank on samples for one case. Each row represents a model with different rank, and each column a different sample. Rank is increasing from top to bottom: $rank \in [1, 2, 5, 10, 15, 20]$.

4.4.5 Application to generative models

This section is based on the following publication:

- **Langley, J.**, Monteiro, M., Jones, C., Pawlowski, N., and Glocker, B. (2022). "Structured Uncertainty in the Observation Space of Variational Autoencoders". In: *Transactions on Machine Learning Research*

The publication is a result of JL master thesis project. JL applied the core idea of SSNs to generative modelling with VAEs. MM and BG came up with the idea. JL implemented the idea, developed the code and performed the experiments. CJ helped JL with the experiments. MM, NP and BG provided guidance throughout the project. JL, CJ, MM and BG co-wrote the publication's manuscript. BG secured the funding.

Code available at:

- https://github.com/biomedia-mira/sos-vae

The flexibility of the proposed approach means we can easily transfer it to other image tasks where modelling dependencies between pixels would be beneficial, such as image generation. One of

the most used and widespread classes of generative models is the Variational Autoencoder (VAE) (Kingma and Welling 2014). VAEs explicitly model the distribution of observations by assuming a latent variable model with low-dimensional latent space that is mapped to the observation space using a neural network decoder. This flexibility allows us to model arbitrarily complex observational distributions. However, one often-overlooked aspect is the choice of observational distribution. As an explicit likelihood model, the VAE assumes a distribution in observation space. The default choice for modelling natural images is a set of per-pixel independent normal distributions or, equivalently, a multivariate normal distribution with a diagonal covariance matrix. As we have demonstrated, sampling from this distribution would lead to grainy independent pixel noise, which results in poor image fidelity in the case of image generation.

Motivated by the above, we apply the same low-rank covariance matrix to the observational distribution of the VAE. The only modification necessary is a simplification. In the case of image generation, the target variable is continuous, as opposed to discrete in segmentation. As a result, we can compute the likelihood in closed form using the expression for the multivariate normal likelihood without using Monte Carlo estimation as we had to in segmentation.

The likelihood of a multivariate normal distribution with k dimensions distribution is given by:

$$p(x) = (2\pi)^{-\frac{k}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\intercal}\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})). \tag{4.10}$$

To calculate the likelihood of a multivariate normal distribution, we must invert the covariance matrix and compute its determinant. For a full-rank covariance matrix, the computational cost and memory footprint make this infeasible for all except very small images since these costs scale quadratically with image size. However, with the low-rank parametrisation, the full matrix never needs to be used directly. The inverse and determinant of the covariance matrix can be efficiently computed using the Woodbury matrix identity (Max 1950) and the matrix determinant lemma, respectively. Since both the covariance factor and the diagonal elements scale linearly with the size of the image, the computational costs of computing the log-likelihood scales linearly as well, in contrast to a full-rank covariance matrix where costs increase quadratically.

Figure 4.9 shows an example of images generated from a standard VAE when sampling from the pixel-wise independent observational distribution compared to samples from a VAE with a low-rank multivariate normal observational distribution. As we can see, regardless of other modelling choices, sampling from the observational distribution will result in spatially-incoherent samples with independent pixel noise unless we properly model the joint pixel distribution.

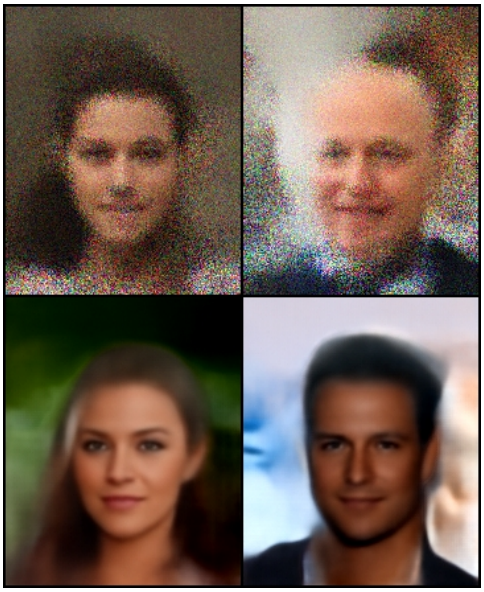


Figure 4.9: Top: Samples generated by sampling from the observational distribution of a standard VAE. **Bottom:** Samples generated by sampling from a VAE with a low-rank multivariate normal observational distribution. We can see the pixel-wise independent noise on the top and the spatially coherent samples on the bottom.

4.5 Discussion

We introduced an efficient approach for modelling spatially correlated aleatoric uncertainty in segmentation. We have shown that our method outperforms the baselines while being much simpler, improves predictive performance with added uncertainty, and the samples it generates can be better than those of a deterministic approach. The simplicity of the method enables it to be easily implemented over any existing neural network architecture, which enabled its use in a 3D application, something which had previously not been attempted. The ability to generate multiple plausible hypotheses post-inference is of value in human-in-the-loop scenarios, such as radiology, where a human could manipulate the segmentation semi-automatically according to the model's uncertainty. Furthermore, even in fully-autonomous systems such as autonomous vehicles being able to reason about spatially correlated uncertainty is essential. For example, uncertainty about whether a region is a pedestrian or not should be correlated over all pixels in the region.

Proper uncertainty quantification is crucial to increase trust and interpretability in deep learning systems, which is of particular importance in healthcare applications. Reliable uncertainty estimates could help inform clinical decision making, and importantly, provide clinicians with feedback on when to ignore automatically derived measurements. Moreover, uncertainty estimates could be propagated to downstream clinical tasks such as radiotherapy planning, e.g., the amount of radiation delivered to each anatomical region. In medicine, the notion of a second opinion is well established and an essential part of scrutinising the decision process. The ability to generate and manipulate multiple plausible hypotheses could be of great benefit in semi-automatic

settings, such as machine aided image segmentation, and help minimise the risk of missing important modes of the target distribution. A complementary prediction might be contradictory yet still very informative.

Chapter 5

Measuring axiomatic soundness of counterfactual image models

This chapter is based on the following publication:

- **Monteiro, M.**, Ribeiro, F. D. S., Pawlowski, N., Castro, D. C., and Glocker, B. (2023). "Measuring axiomatic soundness of counterfactual image models". In: *The Eleventh International Conference on Learning Representations*

MM developed the idea. MM did the implementation and experiments for the MNIST and 3d shapes datasets. FRdS did the implementation and experiments for the Celeb-A dataset. MM wrote the manuscript with the help of FRdS. FRdS, NP, DCC and BG provided useful discussion and feedback to MM on the method, experiments and manuscript. BG secured the funding.

5.1 Introduction

Faithfully answering counterfactual queries is a key challenge in representation learning and a cornerstone for aligning machine intelligence and human reasoning. While significant advances have been made in causal representation learning, enabling approximate counterfactual inference, there is surprisingly little methodology available to assess, measure, and quantify the quality of these models.

The structural causal model (SCM) is a mathematical tool that describes causal systems. It offers a convenient computational framework for operationalising causal and counterfactual inference (Pearl 2009). An SCM is a set of functional assignments (called *mechanisms*) that represent the relationship between a variable, its direct causes (called *parents*) and all other unaccounted sources of variation (called *exogenous noise*). In SCMs, we assume that the mechanisms are algorithmically independent of each other. Further, in Markovian SCMs, which we can represent by DAGs, we assume that the exogenous noise variables are statistically independent of each other (Peters et al. 2017). From here on out, by SCM, we mean Markovian SCM. When the functional form of a mechanism is unknown, learning it from data is a prerequisite for answering counterfactual queries (Bareinboim et al. 2022).

In the context of high-dimensional observations, such as images, the power and flexibility of deep generative models make them indispensable tools for learning the mechanisms of an SCM. However, these same benefits make model identifiability impossible in the general case (Khemakhem et al. 2020; Locatello, Bauer, et al. 2020; Peters et al. 2017), which can cause entanglement of causal effects (Pawlowski 2021) and lead to poor approximations of the causal quantities of interest. Regardless, even if the model or counterfactual query is unidentifiable, we can still measure the quality of the counterfactual approximation (Pearl 2010). However, evaluating image-counterfactual models is challenging without access to observed counterfactuals, which is unrealistic in real-world scenarios.

In this paper, we focus on what constraints a counterfactual inference model must satisfy and how we can use them to measure the model's *soundness* without having access to observed counterfactuals or the SCM that generated the data. We begin by framing mechanisms as functional assignments that directly translate an observation into a counterfactual, given its parents and counterfactual parents. Next, we use Galles and Pearl 1998's axiomatic definition of counterfactual to restrict the space of possible functions that can represent a mechanism. From these constraints, we derive a set of metrics which we can use to measure the soundness of any arbitrary black-box counterfactual inference engine. Lastly, we show how simulated interventions can mitigate model estimation issues due to confounding.

5.2 Related Work

Representation learning aims to capture semantically meaningful disentangled factors of variation in the data. Arguably, these representations can provide interpretability, reduced sample complexity, and improved generalisation (Bengio et al. 2013). From a causal perspective, these factors should represent the parents of a variable in the SCM (Schölkopf et al. 2021). Although there has been extensive research into unsupervised disentanglement (C. P. Burgess et al. 2018; R. T. Chen et al. 2018; Higgins et al. 2017; H. Kim and A. Mnih 2018; Kumar et al. 2018; Peebles et al. 2020), recent results (Locatello, Bauer, et al. 2020) reaffirm the impossibility of this task since the true causal generative model is not identifiable by observing a variable in isolation (Peters et al. 2017). In contrast, supervised disentanglement, where we observe the variable's parents, and weakly supervised disentanglement, where we observe "real" counterfactuals, can lead to causally identifiable generative models (Locatello, Bauer, et al. 2020).

The integration of causal considerations has led to the emerging field of causal representation learning (Schölkopf et al. 2021). In the supervised setting, extensive research has been conducted in adapting deep models for individualised treatment effect estimation (Jesson et al. 2020; Louizos et al. 2017; Madras et al. 2019; Shi et al. 2019; M. Yang et al. 2021; Yoon et al. 2018). Notably, Louizos et al. (2017) use deep latent variable models to estimate individualised and population-level treatment effects. M. Yang et al. (2021) use deep latent variable models for learning to trans-

form independent exogenous factors into endogenous causes that correspond to causally related concepts in the data. In the weakly-supervised setting, recent work has focused on using observations of "real"-counterfactuals instead of a variable's parents to obtain disentangled representations (Bouchacourt et al. 2018; Hosoya 2019; Locatello, Poole, et al. 2020; Shu et al. 2020). Besserve, Mehrjou, et al. (2020) and Besserve, Sun, et al. (2021) show how relaxing identifiability constraints can lead to some degree of identifiability in unsupervised settings.

In the context of image counterfactuals, Pawlowski et al. (2020) demonstrate how to jointly model all the functional assignments in an SCM using deep generative models. Despite presenting a general theory for any generative model, the authors implement only VAEs (Kingma and Welling 2014; Rezende et al. 2014) and normalising flows (Papamakarios et al. 2021), which Dash et al. (2022) complement by using GANs (Goodfellow, Pouget-Abadie, et al. 2020). Sanchez and Tsaftaris (2022) use diffusion models for the counterfactual estimation. Looveren and Klaise (2021) use class prototypes for finding interpretable counterfactual explanations. Sauer and Geiger (2021) use a deep network to disentangle object shape, object texture and background in natural images. Parascandolo et al. (2018) retrieve a set of independent mechanisms from a set of transformed data points in an unsupervised manner using multiple competing models. Additionally, many image-to-image translation models can be considered informal counterfactual inference engines (Choi et al. 2018; J. Hoffman et al. 2018; Isola et al. 2017; D. Li et al. 2021; M.-Y. Liu et al. 2017; Zhu, Park, et al. 2017).

The flexibility of deep models makes them susceptible to learning shortcuts (Geirhos et al. 2020). Consequently, when the data is biased, the effects of the parents can become entangled (Pawlowski 2021; Rissanen and Marttinen 2021). These issues create identifiability problems even when accounting for causality in representation learning. Simulating interventions through data augmentation or resampling can be used to debias the data (An et al. 2021; Ilse et al. 2021; Von Kügelgen et al. 2021). In a closely related field, research has focused on learning from biased data (Nam et al. 2020) or how to become invariant to a protected/spurious attribute (B. Kim et al. 2019).

From an algorithmic perspective of causality, Janzing and Schölkopf (2010) show how looking at the causal Markov condition in terms of conditional algorithmic independence allows causal inference with only one observation (a non-statistical scenario). Even though algorithmic independence is a more fundamental principle than statistical independence, in the general case, algorithmic mutual information (a.k.a. joint Kolmogorov complexity) is provably intractable due to the halting problem.

5.3 Methods

Generating counterfactuals is commonly performed in multiple steps. First, we abduct the exogenous noise from the observation and its parents. Second, we act on some parents. Finally, we use

a generative model to map the exogenous noise and the counterfactual parents back to the observation space. For deep models, true abduction is impossible in the general case, and identifiability issues are ubiquitous (Khemakhem et al. 2020; Locatello, Bauer, et al. 2020; Peters et al. 2017). There can exist multiple models capable of generating the data, and the true causal model cannot be identified from data alone. We argue that viewing counterfactual inference engines as black boxes, where we focus on what properties the model's output must obey rather than the model's theoretical shortcomings, leads us to a set of actionable and principled model constraints. While a full causal model of the data generation process is necessary to create new samples from a joint distribution, in many applications, we are only interested in generating counterfactuals for real observations. In this case, we can directly model the mapping between observation and counterfactual. We call this mapping a counterfactual function in which we consider the abduction to be an implicit process rather than an explicit step in the counterfactual inference.

5.3.1 Counterfactual Functions

Let x be a random variable (i.e., the observation) with parents $\mathbf{pa}$, x^* a counterfactual of x with parents $\mathbf{pa}^*$, and ϵ the exogenous noise variable pertaining to x. The functional assignment for x is given by $x \coloneqq \mathrm{g}(\epsilon, \mathbf{pa})$, and for its counterfactual by $x^* \coloneqq \mathrm{g}(\epsilon, \mathbf{pa}^*)$. Conceptually, counterfactual reasoning is a three-step process: 1) Abduction: infer the exogenous noise from the observation and its parents; 2) Action: intervene on the targeted parents; 3) Prediction: propagate the effect of the intervention through the modified model to generate the counterfactual (Pearl 2009). However, when the values for all the counterfactual parents are known, abduction, action and prediction do not need to be performed sequentially but can be formulated as a single functional assignment taking as arguments the observation, the parents and the counterfactual parents, as formalised in the following.

The abduction process involves inverting the mechanism with respect to the exogenous noise $\epsilon \coloneqq \mathrm{g}^{-1}(x, \mathbf{pa})$. In general, the mechanism is not invertible since there might be several possible counterfactuals corresponding to the same observation. In other words, the inverse mapping is not deterministic since an observation induces a distribution over the exogenous noise, which induces a distribution over possible counterfactuals: $\epsilon \sim P(\epsilon \,|\, x, \mathbf{pa}) \implies x^* \sim P(x^* \,|\, x, \mathbf{pa})$. Equivalently, we can formulate abduction as a deterministic functional assignment $\epsilon \coloneqq \mathrm{abduct}(x, \mathbf{pa})$ where the abduction function is drawn from a distribution over functions $\mathrm{abduct} \sim P(\mathrm{abduct})$. We can rewrite the functional assignment for the counterfactual as $x^* \coloneqq \mathrm{g}(\mathrm{abduct}(x, \mathbf{pa}), \mathbf{pa}^*)$ and obtain an equivalent expression by introducing a new function with the same arguments $x^* \coloneqq \mathrm{f}(x, \mathbf{pa}, \mathbf{pa}^*)$, where $\mathrm{f} \sim P(\mathrm{f})$. We call these functions counterfactual functions and denote the abduction as implicit since a value for the exogenous noise ϵ is never explicitly produced. Notice how abduct and f do not depend on the data due to the independence of cause and mechanism (Peters et al. 2017, Sec. 2.1).

5.3.1.1 Function properties from axiomatic definition of counterfactuals

To determine the properties of such functions, we review Pearl's axiomatic definition of counterfactuals (Pearl 2009, Sec. 7.3.1). The *soundness* theorem states that the properties of composition, effectiveness and reversibility hold true in all causal models (Galles and Pearl 1998). The *completeness* theorem states that these properties are complete (Halpern 2000). Together these theorems state that composition, effectiveness and reversibility are the necessary and sufficient properties of counterfactuals in any causal model. With this in mind. We aim to construct a functional system that obeys these axioms.

Effectiveness: Intervening on a variable to have a specific value will cause the variable to take on that value. Thus, suppose $\mathrm{Pa}(\cdot)$ is an oracle function that returns the parents of a variable, then we have the following equality: $\mathrm{Pa}(\mathrm{f}(x, \mathbf{pa}, \mathbf{pa}^*)) = \mathbf{pa}^*$.

Composition: Intervening on a variable to have the value it would otherwise have without the intervention will not affect other variables in the system. This implies the existence of a null transformation $\mathrm{f}(x, \mathbf{pa}, \mathbf{pa}) = x$ since if $\mathbf{pa}^* = \mathbf{pa}$, then x is not affected.

Reversibility: Reversibility prevents the existence of multiple solutions due to feedback loops. In recursive systems such as DAGs, it follows trivially from composition. However, in the general case, these properties are independent. If setting a variable A to a value a results in a value b for a variable B, and setting B to a value b results in a value a for A, then A and B will take the values a and b. If a mechanism is invertible, taking a twin network conception of the SCM (see Pearl 2009, Sec. 7.1.4) and replacing the previous variables with the observation x and its counterfactual x^*, it follows that if $x^* \coloneqq \mathrm{f}(x, \mathbf{pa}, \mathbf{pa}^*)$, then $x = \mathrm{f}(x^*, \mathbf{pa}^*, \mathbf{pa})$. In other words, the mapping between the observation and the counterfactual is deterministic for invertible mechanisms. Otherwise, there would be a feedback loop. See Appendix C.1 for proof.

5.3.1.2 Partial counterfactual functions

We can also consider partial counterfactual functions, which model the effect of a single parent on the observation *independently* of all other causes. These partial interventions allow us to decompose an intervention into steps whereby a single parent is changed while holding the remaining parents fixed. Each step acts on all parents, but only one parent changes value at a time: $x^* = \mathrm{f_k}(x, \mathrm{pa}_k, \mathrm{pa}_k^*)$, where $\mathbf{pa}^* = \mathbf{pa}_{\mathcal{K}\setminus k} \cup \{\mathrm{pa}_k^*\}$. Notice that this is still an intervention on all parents, in contrast with atomic interventions where only one variable is intervened upon and all others updated according to the SCM. These partial functions can be obtained by taking a full counterfactual function and fixing all parents to their initial values except one $\mathrm{f_k}(x, \mathrm{pa}_k, \mathrm{pa}_k^*) = \mathrm{f}\big(x, \mathbf{pa}_{\mathcal{K}\setminus k} \cup \{\mathrm{pa}_k\}, \mathbf{pa}_{\mathcal{K}\setminus k} \cup \{\mathrm{pa}_k^*\}\big)$, or they can directly implicitly infer the values of the fixed parents, as shown in Section 5.3.1.4.

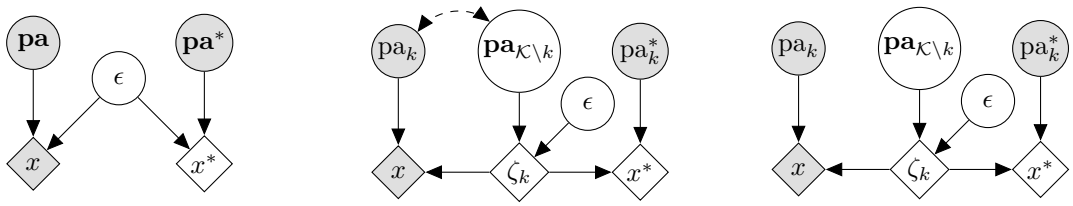
5.3.1.3 Partial counterfactual function decomposition

We show that a full intervention can be decomposed into a sequence of partial interventions and that the partial functions must obey the commutative property. Using the process described in Section 5.3.1, we can write a partial counterfactual function as: $x^* \coloneqq \mathrm{f_k}(x, \mathrm{pa}_k, \mathrm{pa}_k^*)$ where $\mathrm{f}_k \sim P(\mathrm{f}_k)$. Since the output of $\mathrm{f_k}(\cdot)$ is counterfactual of x, all the properties from Section 5.3.1.1 still apply. Repeating the process for each parent results in a set of independent partial functions allowing us to write the full counterfactual function as a composition of multiple independent partial functions:

$$x^* \coloneqq \left[\mathrm{f_K}(\,\cdot\,, \mathrm{pa}_K, \mathrm{pa}_K^*) \circ \ldots \circ \mathrm{f}_2(\,\cdot\,, \mathrm{pa}_2, \mathrm{pa}_2^*) \circ \mathrm{f}_1(\,\cdot\,, \mathrm{pa}_1, \mathrm{pa}_1^*)\right](x) \tag{5.1}$$

By construction, each partial function models the effect of a single parent on the observation independently of other parents. As a result, the decomposition in Equation 5.1 must be commutative.

5.3.1.4 Learning partial counterfactual functions from data



(a) Twin network SCM of a mechanism **(b)** Twin network SCM with target parent separated **(c)** Twin network SCM after intervention

Figure 5.1: Process of making a parent independent of the remainder using twin network SCMs and interventions.

Figure 5.1a shows a causal graph for a single mechanism, where there can exist unknown arbitrary causal links between the variable's parents. Suppose we take the endogenous causes of x and convert all but one into exogenous causes resulting in the causal graph in Figure 5.1b. The bidirectional arrow between the endogenous parent pa_k and the now exogenous parents $\mathbf{pa}_{\mathcal{K}\setminus k}$ denotes the unknown causal relationships between them. This manipulation results in a "partial" mechanism $x^* \coloneqq \mathrm{g_k}(\zeta_k, \mathrm{pa}_k)$ where only one parent is endogenous and thus susceptible to action. The remaining parents and the previous exogenous noise are grouped into a new noise variable $\zeta_k \coloneqq \mathrm{g_k^{-1}}(x, \mathrm{pa}_k)$. However, due to the dependence between the target parent pa_k and the unobserved exogenous parents $\mathbf{pa}_{\mathcal{K}\setminus k}$, abducting ζ_k is impossible. The path between ζ_k and pa_k breaks the assumption of independent exogenous noise variables, making it impossible to separate the effects of endogenous and exogenous causes. The mechanism induced by ζ_k now depends on the cause pa_k since $\zeta_k \not\perp\!\!\!\perp \mathrm{pa}_k$, violating the principle of independent causal mechanisms.

To preserve the principle of independent causal mechanisms, we must first make pa_k independent

of the remaining parents $\mathbf{pa}_{\mathcal{K}\setminus k}$. Since the causal direction between pa_k and $\mathbf{pa}_{\mathcal{K}\setminus k}$ is unknown, we intervene on both these variables, thus guaranteeing their link is severed. The modified SCM is shown in Figure 5.1c, where abduction is now possible since the mechanism no longer depends on the cause. Because there are no backdoor paths between pa_k and ζ_k, using *do*-calculus Pearl 2009, we have that $P(\zeta_k \,|\, x, \mathrm{do}(\mathrm{pa}_k)) = P(\zeta_k \,|\, x, \mathrm{pa}_k)$. Since x^* depends on x via ζ_k the distribution over potential counterfactuals $P(x^* \,|\, x, \mathrm{pa}_k)$ remains unchanged after the intervention. After the intervention, we can learn the partial counterfactual function without confounding being an issue. The partial function changes one parent while holding the remainder fixed since these are now part of the exogenous noise.

5.3.2 Measuring soundness of counterfactuals

We now distinguish between the ideal counterfactual function $\mathrm{f}(\cdot)$ and its approximation $\hat{\mathrm{f}}(\cdot)$. We consider a scenario where we want to evaluate how good our estimate of a counterfactual function is from observational data alone, without access to observed counterfactuals. Using the axiomatic properties described in Section 5.3.1.1, we can derive a set of soundness metrics that compare the approximate and ideal models.

Composition: Since the ideal model cannot change an observation under the null transformation, we can measure how much the approximate model deviates from the ideal by calculating the distance between the original observation and the mth time null-transformed observation. The repeated application of the function will highlight what types of corruptions the approximate model produces on the observation. Given a distance metric $\mathrm{d_X}(\cdot, \cdot)$, such as the l_1 distance, an observation x with parents $\mathbf{pa}$ and a functional power m, we can measure composition as:

$$\mathrm{composition}^{(m)}(x, \mathbf{pa}) \coloneqq \mathrm{d_X}\Big(x, \hat{\mathrm{f}}^{(m)}(x, \mathbf{pa}, \mathbf{pa})\Big). \tag{5.2}$$

See Section C.2 for a discussion on desirable metric properties.

Reversibility: When a mechanism is invertible, the ideal model must be cycle-consistent. Thus we can measure reversibility by calculating the distance between the original observation and the mth time cycled-back transformed observation. Setting $\hat{\mathrm{p}}(x, \mathbf{pa}, \mathbf{pa}^*) \coloneqq \hat{\mathrm{f}}\Big(\hat{\mathrm{f}}(x, \mathbf{pa}, \mathbf{pa}^*), \mathbf{pa}^*, \mathbf{pa}\Big)$. Given a distance metric $\mathrm{d_X}(\cdot, \cdot)$, an observation x with parents $\mathbf{pa}$ and a functional power m, we can measure reversibility as:

$$\mathrm{reversibility}^{(m)}(x, \mathbf{pa}, \mathbf{pa}^*) \coloneqq \mathrm{d_X}\big(x, \hat{\mathrm{p}}^{(m)}(x, \mathbf{pa}, \mathbf{pa}^*)\big). \tag{5.3}$$

Note that in most real-world scenarios, the inherent uncertainty regarding exogenous factors makes it hard to determine whether the true mechanism is invertible.

Effectiveness: Unlike composition and reversibility, which we can measure independently of the data distribution, effectiveness is difficult to measure objectively without relying on data-driven

methods or strong domain knowledge. We propose measuring effectiveness individually for each parent by creating a pseudo-oracle function $\widehat{\mathrm{Pa}}_{\mathrm{k}}(\cdot)$, which returns the value of the parent pa_k given the observation. These functions can be human-made programs or machine learning models learnt from data via classification/regression. The inevitable limitation of the data-driven approach is the approximation error. Moreover, we must be especially cautious in the presence of confounded parents to ensure that the pseudo-oracles do not exploit shortcuts and correctly retrieve the desired parent. To independently measure how well the effect of each parent is modelled, we measure effectiveness after applying partial counterfactual functions (see Section 5.3.1.2). Using an appropriate distance metric $\mathrm{d}_{\mathrm{k}}(\cdot,\cdot)$, such as accuracy for discrete variables or l_1 distance for continuous ones, we measure effectiveness for each parent as:

$$\text{effectiveness}_k(x, \mathbf{pa}, \mathbf{pa}^*) = \mathrm{d}_{\mathrm{k}}\left(\widehat{\mathrm{Pa}}_{\mathrm{k}}\left(\hat{\mathrm{f}}_{\mathrm{k}}(x, \mathrm{pa}_k, \mathrm{pa}_k^*)\right), \mathrm{pa}_k^*\right). \tag{5.4}$$

5.3.3 Simulated Interventions

When learning deep models from biased data, we must be careful not to allow the model to learn shortcuts which do not reflect the true causal relationships in the data. While some generative models have inductive priors that make them more robust to confounding (Higgins et al. 2017; D. Li et al. 2021), discriminative models are quite brittle (Geirhos et al. 2020). The causal approach to address confounding is to break the offending causal links in the SCM via an intervention, removing the possibility of learning shortcuts by de-biasing the data.

Consider the joint distribution of the data $P(x, \mathbf{pa}_{\mathcal{K}\setminus k}, \mathrm{pa}_k) = P(x\,|\,\mathrm{pa}_k, \mathbf{pa}_{\mathcal{K}\setminus k})\,P(\mathrm{pa}_k, \mathbf{pa}_{\mathcal{K}\setminus k})$, where $\mathcal{K}$ is the set of all parents of x. If we perform a soft intervention (see Peters et al. 2017, Sec. 3.2) where we set pa_k and $\mathbf{pa}_{\mathcal{K}\setminus k}$ to their respective marginal distributions, we obtain an interventional distribution where pa_k and $\mathbf{pa}_{\mathcal{K}\setminus k}$ are independent. The joint distribution now factorises as:

$$P^{\mathrm{do}(\mathrm{pa}_k\sim P(\mathrm{pa}_k);\mathbf{pa}_{\mathcal{K}\setminus k}\sim P(\mathbf{pa}_{\mathcal{K}\setminus k}))}(x, \mathbf{pa}_{\mathcal{K}\setminus k}, \mathrm{pa}_k) = P(x\,|\,\mathbf{pa}_{\mathcal{K}\setminus k}, \mathrm{pa}_k)\,P(\mathrm{pa}_k)\,P(\mathbf{pa}_{\mathcal{K}\setminus k}) \tag{5.5}$$

In the absence of interventional data, we use simulated interventions by resampling the data – note that this is only possible when the observed joint distribution has full support over the product of marginals. We sample pa_k and $\mathbf{pa}_{\mathcal{K}\setminus k}$ according to their respective marginal distributions and then randomly sample an observation conditioned on the sampled parent values. To make all parents independent of each other, following the same logic, we can sample each parent independently according to its marginal distribution.

5.3.4 Deep generative models as approximate counterfactual functions

In this section, we discuss two generative models commonly used as approximate counterfactual inference engines.

Conditional VAE: The evidence lower bound for a classic conditional VAE (Higgins et al. 2017; Kingma and Welling 2014) with a β penalty is given by:

$$\mathrm{ELBO}_\beta(\theta, \omega) = \mathbb{E}_{q_\theta(z|x,\mathbf{pa})}[\log p_\omega(x\,|\,z, \mathbf{pa})] - \beta\, D_{\mathrm{KL}}[q_\theta(z\,|\,x, \mathbf{pa})\,\|\,p(z)], \tag{5.6}$$

where $q_\theta(z\,|\,x, \mathbf{pa})$ is a normal distribution parameterised by a neural network encoder with parameters θ, $p_\omega(x\,|\,z, \mathbf{pa})$ is a set of pixel-wise independent Bernoulli or Normal distributions parameterised by a neural network decoder with parameters ω, and $p(z)$ is an isotropic normal prior distribution. The model is trained by maximising the ELBO with respect to parameters of the neural networks using the re-parametrisation trick to sample from the approximate latent posterior: $z = \mu_\theta(x, \mathbf{pa}) + \sigma_\theta(x, \mathbf{pa}) \odot \epsilon_z$ where $\epsilon_z \sim \mathcal{N}(0, I)$. The construction of the conditional VAE naturally leads to a composition constraint via the likelihood term. Even though the model can ignore the conditioning, using a bottleneck or a β penalty pushes the model towards using the conditioning, thus enforcing effectiveness, making the conditional VAE a natural choice for a counterfactual model.

We can produce counterfactuals by encoding an observation and its parents, sampling the latent posterior, and then decoding it along with the counterfactual parents: $x^* \sim p_\omega(x\,|\,z, \mathbf{pa}^*)$, where $z \sim q_\theta(z|x, \mathbf{pa})$. Rewriting the previous expression as: $x^* := \hat{\mathrm{f}}_{\theta,\omega}(x, \mathbf{pa}, \mathbf{pa}^*)$ where $\mathrm{f} \sim P(\hat{\mathrm{f}})$, we see the parallels to the formulation in Section 5.3.1. Even though it is possible to generate new samples from the model by sampling z, the VAE is not a full causal generative model. The latent variable z is not the same as the exogenous noise of the SCM ϵ. There are no guarantees that $z \perp\!\!\!\perp \mathrm{pa}|x$ or that the exogenous noise would be normally distributed. Furthermore, there are no guarantees that the forward model (decoder) can disentangle the effects of each parent on the observation.

Conditional GAN with a composition constraint: Given a joint distribution $x, \mathbf{pa} \sim P(x, \mathbf{pa})$ and the marginal distribution of each parent $\mathrm{pa}_k \sim P(\mathrm{pa}_k)$, if we independently sample each parent according to its marginal and perform an intervention, we obtain an interventional distribution $x, \mathbf{pa} \sim P^{do}(x, \mathbf{pa})$, where the parents are independent of each other $\mathbf{pa} \sim \prod_k P(\mathrm{pa}_k)$. We can obtain this distribution via a simulated intervention or by applying a counterfactual function to samples of a source distribution $P^{src}(x, \mathbf{pa})$, which can be the joint or the interventional distribution itself. Since these distributions must be equal, we can use GANs to minimise statistical

divergence between the two (Goodfellow, Pouget-Abadie, et al. 2020; Nowozin et al. 2016):

$$F(\theta,\omega) = \mathop{\mathbb{E}}_{x,\mathbf{pa}\sim P^{do}(x,\mathbf{pa})}\Big[\log\big(D_\omega(x,\mathbf{pa})\big)\Big] - \mathop{\mathbb{E}}_{\substack{x,\mathbf{pa}\sim P^{src}(x,\mathbf{pa})\\ \mathrm{pa}_k^*\sim P(\mathrm{pa}_k)}}\Big[\log\Big(1-D_\omega\big(\hat{\mathrm{f}}_\theta(x,\mathbf{pa},\mathbf{pa}^*),\mathbf{pa}\big)\Big)\Big], \tag{5.7}$$

where the conditional generator $\hat{\mathrm{f}}_\theta(x,\mathbf{pa},\mathbf{pa}^*)$ is a neural network parameterised by θ which approximates the counterfactual function, D_ω is the critic function parameterised by parameters ω, and $F(\theta,\omega)$ is minimised with respect to θ and maximised with respect to ω.

Unlike VAEs, GANs have no inherent mechanism to enforce composition; thus, we introduce a composition constraint which encourages the null transformed observation to be close to the original observation. Given distance metric $\mathrm{d}(\cdot,\cdot)$ such as the l_2 distance, we can write the following regulariser, which we add to the GAN objective:

$$R_{\text{composition}}(\theta) = \mathop{\mathbb{E}}_{x,\mathbf{pa}\sim P^{src}(x,\mathbf{pa})}\mathrm{d}\Big(x,\hat{\mathrm{f}}_\theta(x,\mathbf{pa},\mathbf{pa})\Big). \tag{5.8}$$

Similarly, for invertible mechanisms, we can add a reversibility constraint. However, early experiments proved it redundant while adding computational cost. The proposed GAN model has no distribution over functions, which is equivalent to assuming that the exogenous noise posterior $P(\epsilon\,|\,x,\mathbf{pa})$ is delta distributed. Although this is a strong assumption, assuming a normal posterior as is done in the VAE is also restrictive. Both are unlikely to be a good approximation of the true noise distribution in most real-world scenarios.

5.4 Results

We now demonstrate the utility of our evaluation framework by applying it to three datasets that pose different modelling challenges. For demonstration purposes, we assume invertible mechanisms so we can use the reversibility metric.

5.4.1 Colour MNIST

To illustrate the effect confounding has on counterfactual approximation, we construct a simple experiment using the MNIST dataset (LeCun, Bottou, et al. 1998) where we introduce a new parent: the digit's hue. We colour each image by triplicating the grey-scale channel, setting the saturation to 1 and setting the hue to a value between 0 and 1. The hue value is given by one of three possible SCMs:

- **Unconfounded:** where we draw the hue from a uniform distribution independently of the digit: $\mathrm{hue}\sim\mathrm{Uniform}(0,1)$;
- **Confounded without full support:** where the hue depends on the digit but the joint distribution does not have full support: $\mathrm{hue}\sim\mathcal{N}(\mathrm{digit}/10+0.05,\sigma)$, where $\sigma=0.05$;

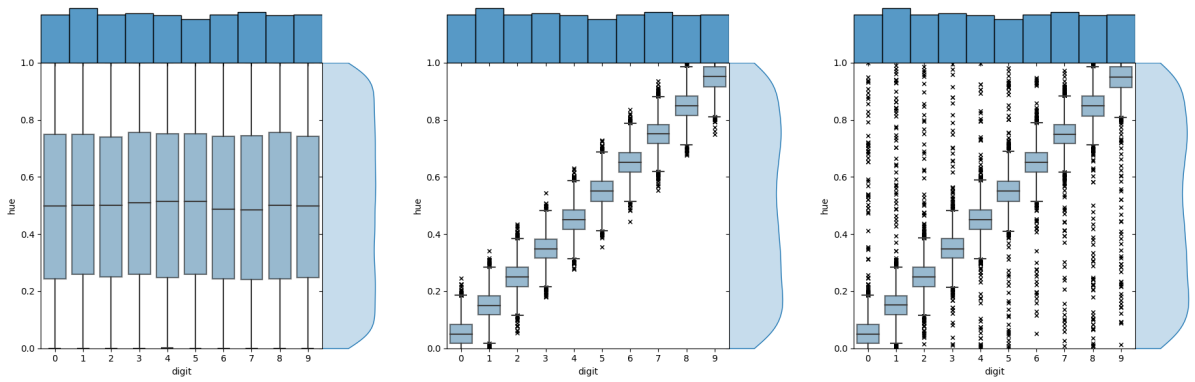
- **Confounded with full support:** like the previous scenario except we include a percentage of outliers where the hue is drawn independently of the digit in order to ensure the distribution has full support: $b \sim \text{Bernoulli}(p)$ and $\text{hue} \sim \mathcal{N}(\text{digit}/10 + 0.05, \sigma)$ if $b = 0$ else $\text{hue} \sim \text{Uniform}(0, 1)$, where $\sigma = 0.05$ and $p = 0.01$.

Figure 5.2 shows the joint distribution of digit and hue for the three SCMs, and Figure 5.3 shows samples from the confounded and unconfounded joint distributions.

We compare the counterfactual soundness of the following models: a VAE with Bernoulli log-likelihood and $\beta = \{1, 2\}$, a VAE with Normal log-likelihood with a fixed variance of 0.1 and $\beta = 5$, and a conditional GAN with a composition constraint. To obtain lower bounds for the soundness metrics, we include two models which, by design, cannot perform abduction: the identity function and a VAE without the encoder at inference time (Bernoulli VAE, $\beta = 1$).

Listings C.1 and C.2 show the architectures of the VAE and GAN respectively. We trained the models for 20000 steps with a batch size of 512. For the VAE, we used the AdamW (Loshchilov and Hutter 2019) optimiser with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\text{weight\_decay} = 0.0001$. For the GAN, we used the same optimiser with a learning rate of 0.0001, $\beta_1 = 0$, $\beta_2 = 0.9$ and $\text{weight\_decay} = 0.0001$. The learning rate is multiplied by 0.5 at 12000 steps and again multiplied by 0.2 at 16000 steps. Listing C.3 shows the architecture for pseudo-oracles. We trained the pseudo-oracles for 2000 steps with a batch size of 1024 using the AdamW (Loshchilov and Hutter 2019) optimiser with a learning rate of 0.0005, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\text{weight\_decay} = 0.0001$. For data augmentation we used random translations where the amount of horizontal and vertical translation is sampled uniformly from 0 to 10% of the width/height of the image.

For the confounded scenarios, we perform a simulated intervention to break the causal link between digit and hue. Since the hue is a continuous variable, to simulate the intervention, we calculate the histogram of its marginal distribution (5 bins) and resample it as if it were a discrete



(a) Unconfounded joint distribution.

(b) Confounded joint distribution without full support.

(c) Confounded joint distribution with full support.

Figure 5.2: Colour MNIST joint distribution of digit and hue for different SCMs.

(a) Samples from the unconfounded joint distribution.

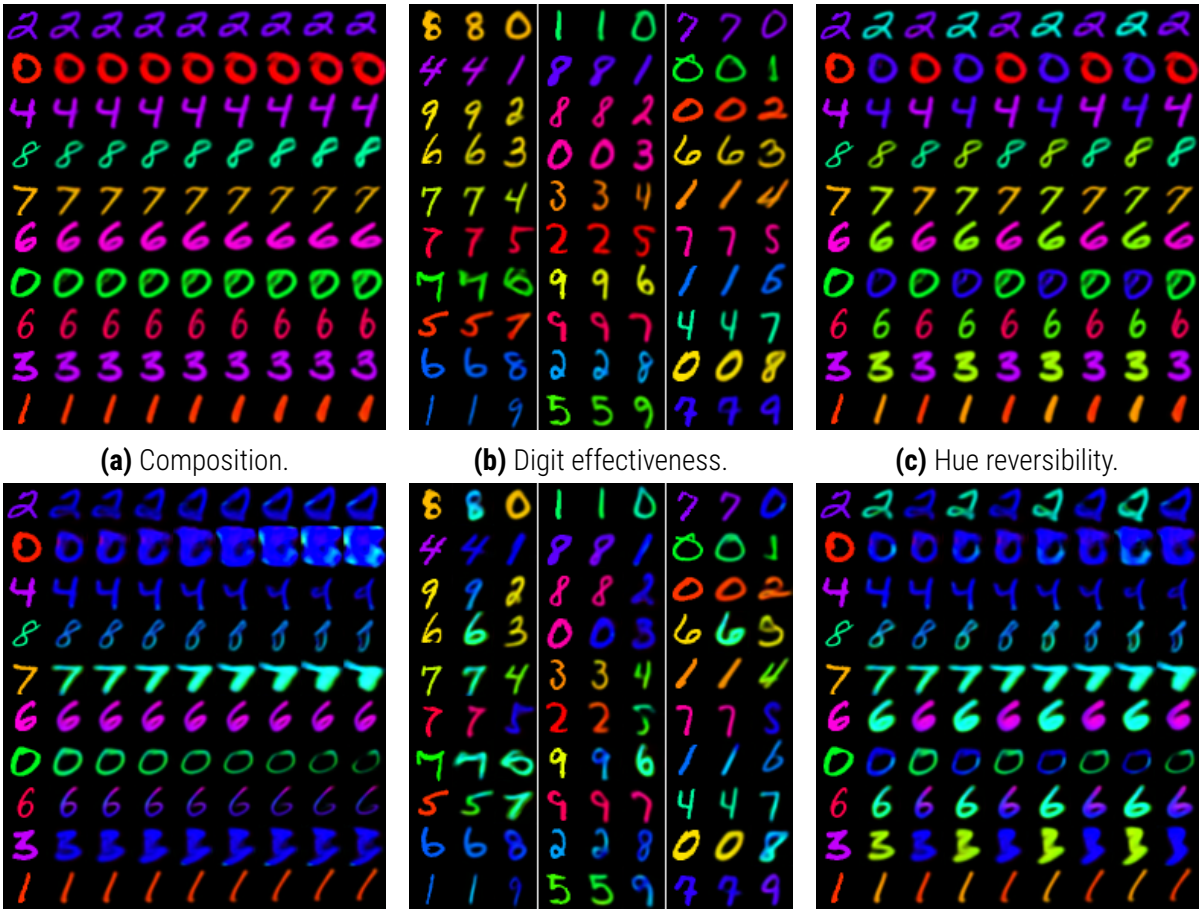
(b) Samples from the confounded joint distribution.

Figure 5.3: Colour MNIST samples.

Table 5.1: Soundness metrics on colour MNIST over 5 random seeds. We measure composition after the null intervention and reversibility after one intervention cycle. We measure effectiveness using digit accuracy and hue absolute error in percentage points since $\text{hue} \in [0, 1]$. ***** GAN always requires simulated intervention for target distribution and thus cannot be trained w/o full support.

dataset	inter-vention	model	null-intervention composition	digit intervention effectiveness		digit intervention reversibility	hue intervention effectiveness		hue intervention reversibility
			$l_1^{(1)} \downarrow$	$\text{acc}_{\text{digit}}(\%) \uparrow$	$\text{ae}_{\text{hue}}(\%) \downarrow$	$l_1^{(1)} \downarrow$	$\text{acc}_{\text{digit}}(\%) \uparrow$	$\text{ae}_{\text{hue}}(\%) \downarrow$	$l_1^{(1)} \downarrow$
	-	Identity	0.00	10.50	1.38	0.00	99.18	32.98	0.00
un-confounded	-	VAE w/o encoder	19.04 (0.09)	97.08 (0.25)	1.32 (0.05)	19.04 (0.09)	97.24 (0.26)	1.32 (0.06)	19.04 (0.09)
	-	Bernoulli VAE β=1	5.98 (0.06)	98.68 (0.13)	1.29 (0.04)	7.67 (0.06)	99.45 (0.09)	1.26 (0.05)	7.24 (0.05)
	-	Bernoulli VAE β=2	6.86 (0.07)	99.52 (0.07)	1.33 (0.15)	9.10 (0.12)	99.60 (0.04)	1.32 (0.15)	8.62 (0.11)
	-	Normal VAE β=5	6.26 (0.29)	97.24 (0.26)	1.52 (0.28)	8.07 (0.26)	99.38 (0.06)	1.47 (0.27)	7.51 (0.32)
	-	GAN	4.92 (0.05)	94.28 (1.01)	1.60 (0.22)	9.22 (0.27)	98.98 (0.05)	1.55 (0.23)	5.60 (0.03)
confounded w/o full support	no	Bernoulli VAE β=1	9.20 (1.31)	97.12 (1.05)	10.74 (4.77)	11.42 (1.49)	98.89 (0.16)	11.60 (6.14)	11.11 (1.61)
		Bernoulli VAE β=2	10.84 (0.45)	98.94 (0.17)	10.36 (1.39)	12.82 (0.45)	99.17 (0.05)	10.07 (1.39)	12.52 (0.41)
		Normal VAE β=5	11.21 (0.63)	94.74 (0.51)	14.17 (2.63)	13.32 (0.62)	98.81 (0.22)	14.27 (3.03)	12.69 (0.59)
	yes	Bernoulli VAE β=1	8.63 (0.50)	96.94 (0.26)	6.38 (1.58)	11.10 (0.75)	98.88 (0.25)	7.02 (1.96)	10.79 (0.75)
		Bernoulli VAE β=2	9.85 (0.33)	95.76 (1.63)	6.44 (1.24)	12.10 (0.39)	95.77 (1.56)	6.44 (1.37)	11.86 (0.29)
		Normal VAE β=5	9.32 (1.41)	95.35 (0.71)	7.54 (1.99)	11.29 (1.39)	98.79 (0.28)	7.30 (2.03)	10.85 (1.36)
confounded w/ full support	no	Bernoulli VAE β=1	6.68 (0.27)	96.62 (2.09)	8.52 (6.93)	8.89 (0.70)	99.20 (0.10)	12.15 (11.69)	8.45 (0.69)
		Bernoulli VAE β=2	7.56 (0.10)	99.36 (0.16)	2.70 (0.12)	9.67 (0.06)	99.47 (0.06)	2.54 (0.12)	9.32 (0.09)
		Normal VAE β=5	6.72 (0.30)	95.53 (0.28)	3.88 (1.12)	9.06 (0.68)	99.07 (0.04)	3.59 (1.20)	8.45 (0.67)
	yes*	GAN	6.05 (0.06)	95.17 (0.55)	1.95 (0.07)	11.18 (0.10)	99.18 (0.10)	1.73 (0.11)	7.79 (0.10)
	yes	Bernoulli VAE β=1	6.67 (0.10)	99.07 (0.15)	2.31 (0.24)	8.42 (0.16)	99.37 (0.13)	3.08 (1.08)	8.40 (0.48)
		Bernoulli VAE β=2	7.84 (0.09)	99.63 (0.03)	2.16 (0.06)	9.63 (0.08)	99.61 (0.06)	2.01 (0.10)	9.34 (0.09)
		Normal VAE β=5	6.51 (0.29)	97.75 (0.18)	3.05 (0.44)	8.35 (0.29)	99.31 (0.07)	2.73 (0.47)	7.83 (0.31)
		GAN	5.25 (0.06)	96.27 (0.26)	1.84 (0.11)	10.75 (0.34)	99.01 (0.06)	1.77 (0.14)	6.20 (0.04)

variable. We train the models on data generated from the three SCMs but always test on the unconfounded test set, which mimics a scenario where the correlations in the training data are spurious and ensures the results of the tests are not biased. Note that if the test set is biased, we can obtain biased estimates for the soundness metrics; this setting falls out of the scope of this study. We use accuracy for the digit and the absolute error for the hue. To measure composition and reversibility, we use the l_1 distance. Table 5.1 shows the results for all models and scenarios. For the VAEs, we sample a unique function (*i.e.*latent noise) for each observation but keep it fixed for repeated interventions.



(a) Composition. **(b)** Digit effectiveness. **(c)** Hue reversibility.

(d) Composition. **(e)** Digit effectiveness. **(f)** Hue reversibility.

Figure 5.4: Soundness tests on colour MNIST for a disentangled model **(a, b, c)** and a entangled model **(d, e, f)**. **Composition (a, d)**: one sample per row, the first column is the input, each column after is the result of applying the null intervention to the previous column. **Digit effectiveness (b, e)**: three sub-panels with one sample per row, the first column is the input, the second column is the result of the null intervention, the last column is the result of a partial intervention on the digit. **Hue reversibility (c, f)**; one sample per row, the first column is the input, the second column is the result of a partial intervention on the hue, the third column is the result of applying the partial intervention that cycles back to the original hue. The remaining columns repeat the cycle.

Comparing with poor counterfactuals models. We can see that the identity function achieves perfect composition and reversibility but, as expected, fails at effectiveness. In contrast, the VAE without encoder performs well in terms of effectiveness but fails at composition and reversibility. Interestingly, the same VAE with the encoder achieves a composition after ten null interventions of $l_1^{(10)} = 17.36\ (0.62)$, close to the value without the encoder $l_1^{(10)} = 19.04\ (0.09)$, indicating that the VAE progressively loses the identity of the image and converges to a random sample (Figure 5.4a).

Comparing models and scenarios. For each scenario, we can see how the proposed metrics allow us to compare models independently of the model class and directly compare the quality of counterfactual approximations. For the confounded scenario without full support, we see a significant

drop in performance which we cannot recover even when using a simulated intervention since the lack of support prevents us from sampling points from some areas of the joint distribution. Nevertheless, when the confounded distribution has full support, we can recover the performance using a simulated intervention or selecting an appropriate β penalty for the VAE. In addition to the numerical results, it is also helpful to visualise the results of the tests. To highlight the impact of confounding, in Figure 5.4, we show the soundness tests for a disentangled model (Normal VAE on the confounded scenario w/ full support and a simulated intervention) and an entangled model (Normal VAE on the confounded scenario w/o full support and no simulated intervention). We can see that the entangled model cannot preserve the image's identity, quickly changing its colour and distorting its shape (Figure 5.4d). In contrast, the disentangled model is far more capable of preserving the image's identity even if the shape gets distorted over repeated applications (Figure 5.4a). Regarding effectiveness, we can see that the entangled model consistently fails to change the digit without changing its hue (Figure 5.4e). Conversely, the disentangled model can change the digit independently of the hue, and its counterfactuals seem to preserve properties such as slant and thickness. Finally, regarding hue reversibility, we can see that only the disentangled model is able to correctly cycle back and forth between hue values even if there is some shape distortion (Figures 5.4f and 5.4c).

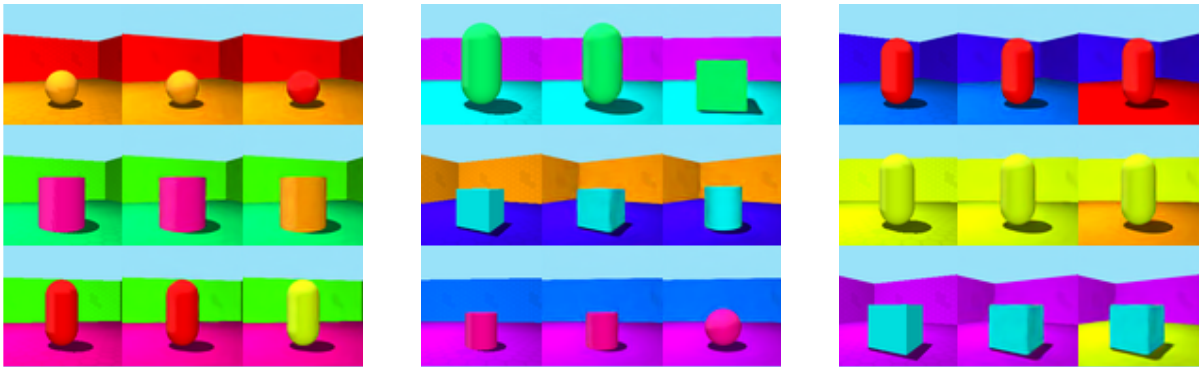
Measuring effectiveness using pseudo-oracles trained on biased data. In the previous experiment, we used pseudo-oracles trained on unconfounded data to compare models. However, in real-world scenarios where we do not control the data-generation process, we have to use pseudo-oracles trained on the available data. If we use pseudo-oracles trained on biased data, we can get biased estimates for effectiveness. We can use simulated interventions when training the oracles to address this problem. When training the pseudo-oracles on the confounded scenario w/ full support, we obtain a digit accuracy of 90.67% and hue absolute error of 6.34% – down from 99.18% and 1.38% respectively, for the oracles trained on unconfounded data. However, using a simulated intervention, we obtained 96.70% and 4.11% for the digit accuracy and hue absolute error, respectively. Regardless, we show in Tables C.1, C.2 that the relative performance ranking of the models is only slightly affected by the decrease in oracle quality due to confounding. Additionally, we show in Table C.3 that the ranking is also robust to using linear models (logistic/linear regression) as pseudo-oracles, giving us some margin for error when measuring the relative effectiveness.

5.4.2 3D Shapes

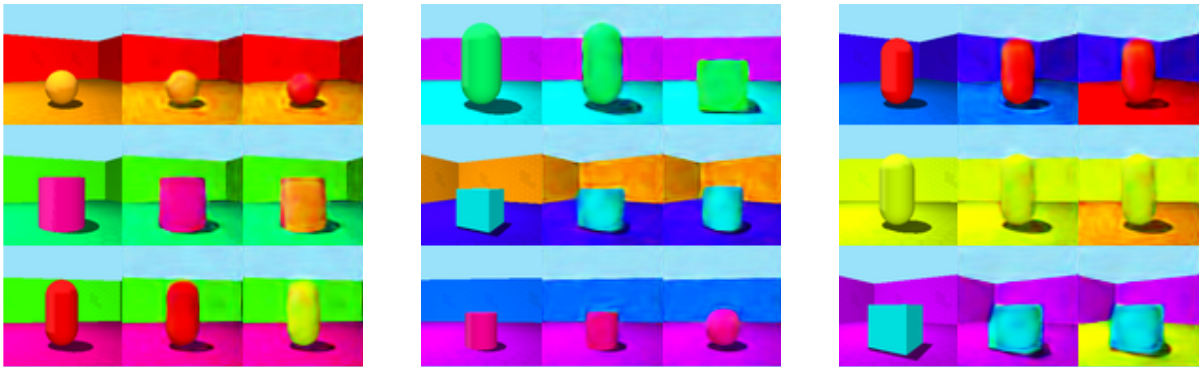
Next, we test our evaluation framework on the 3D shapes dataset (C. Burgess and H. Kim 2018), which comprises procedurally generated images where a 3D object is generated based on six independent parents: floor hue, wall hue, object hue, scale, shape and orientation. Each possible combination of attributes results in an image creating a dataset of 480000 images. In this scenario, there are no exogenous noise factors since each image is uniquely determined by its parents.

We compare the counterfactual properties of a Bernoulli VAE ($\beta = 1$) with a constrained GAN. We used the same training regimes as we used for the colour MNIST experiment with slight modifications the VAE and GAN architecture (Listings C.4 and C.5) and no data-augmentation. We keep 10% of images as a test set and train on the remaining 90%. We treat all variables as discrete since even the continuous ones, such as hue, only take a set of discrete values.

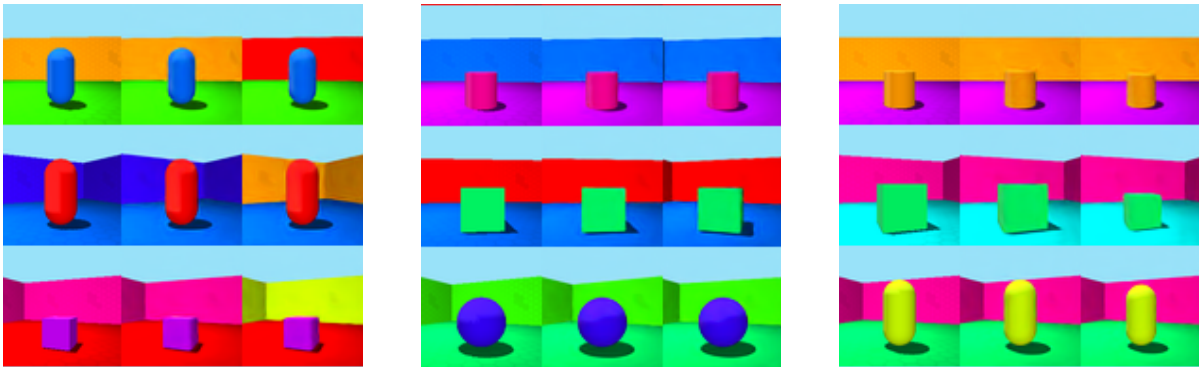
Figures 5.5 shows the a composition/effectiveness plots for both models. Visually we can see that the models can successfully change all parents without perturbing the remainder. In the numerical results shown in Table C.4 we see that both models achieve near-perfect effectiveness. However, we can see that the GAN introduces a slight distortion in the image, which the VAE does not. In fact, the VAE achieves a composition score of $l_1^{(1)} = 1.62 \pm (0.07)$ and $l_1^{(10)} = 1.62(0.07)$ while the GAN has a score of $l_1^{(1)} = 5.11(0.14)$ and $l_1^{(10)} = 5.16(0.15)$. The lack of exogenous factors explains the consistency in the composition scores after repeated null interventions. Abduction is trivial since there is nothing to abduct. Thus, the model can ignore the observation and rely solely on the parents to generate the counterfactual.



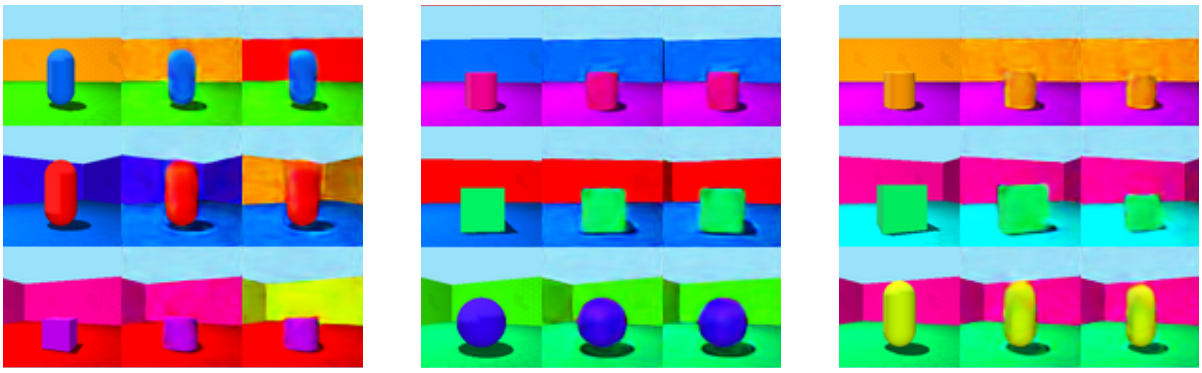
(a) Object hue (VAE). **(b)** Object shape (VAE). **(c)** Floor hue (VAE).



(d) Object hue (GAN). **(e)** Object shape (GAN). **(f)** Floor hue (GAN).



(g) Wall hue (VAE). **(h)** Orientation (VAE). **(i)** Object scale (VAE).



(j) Wall hue (GAN). **(k)** Orientation (GAN). **(l)** Object scale (GAN).

Figure 5.5: Effectiveness/Composition tests on 3D shapes for VAE and GAN. Each sub-figure has one sample per row: the first column is the input, the second column is the result of the null intervention, and the last column is the result of a partial intervention on the specified parent.

5.4.3 CelebA-HQ

Lastly, we demonstrate our evaluation framework on a natural image dataset which requires more complex generative models. We used the CelebA-HQ dataset (Karras et al. 2018) with a 64x64 resolution. We randomly split the 30,000 examples into 70% for training, 15% for validation and 15% for testing. We selected the 'smiling' and 'eyeglasses' binary parent attributes for conditioning our generative model. After careful consideration, we opted for these attributes for three main reasons: (i) they are gender neutral, which mitigates bias and allows us to use the entire dataset rather than a smaller subset; (ii) they are more objective than other attributes like 'attractive' and 'young'; (iii) we can train reasonably accurate pseudo-oracles for our counterfactual evaluation on these attributes, which is not the case for some others due to label noise/ambiguity and significant class imbalance.

For the model, we opted to extend the VDVAE (Child 2021) to a conditional model to enable counterfactuals. We follow the general very deep VAE (VDVAE) setup proposed by (Child 2021), and introduce some modifications to accommodate both parent conditioning and our compute constraints. The architecture is based on the ResNet VAE proposed by (Kingma, Salimans, et al. 2016) but is much deeper and uses bottleneck residual blocks. The VDVAE has several stochastic layers of latent variables which are conditionally dependent upon each other, and are organised into L groups $\mathbf{z} = \{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_L\}$. These latents are typically output as feature maps of varying resolutions, whereby $\mathbf{z}_0$ consists of fewer latents at low resolution up to many latents $\mathbf{z}_L$ at high resolution. The conditioning structure is organised following the ladder structure proposed by (Sønderby et al. 2016), where both the prior $p_\theta(\mathbf{z})$ and approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ generate latent latent variables in the same top-down order. As in previous work, the prior and posterior are diagonal Gaussian distributions and the model is trained end-to-end by optimizing the usual variational bound on the log-likelihood (ELBO) (Kingma and Welling 2014; Maaløe et al. 2019).

Conditional VDVAE. For our counterfactual generation purposes, we augmented the original prior and posterior top-down conditioning structure to include $\mathbf{x}$'s parents $\mathbf{pa}$ as follows:

$$p_\theta(\mathbf{z}|\mathbf{pa}) = p_\theta(\mathbf{z}_0)\prod_{i=1}^{L} p_\theta(\mathbf{z}_i|\mathbf{z}_{i-1}, \mathbf{pa}), \quad q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{pa}) = q_\phi(\mathbf{z}_0|\mathbf{x})\prod_{i=1}^{L} q_\phi(\mathbf{z}_i|\mathbf{z}_{i-1}, \mathbf{x}, \mathbf{pa}). \tag{5.9}$$

We call this the Conditional VDVAE. In practical terms, we simply expand and concatenate the parent conditioning attributes to the latent variables at each stochastic layer, then merge them into the downstream via a 1×1 convolution. Although there are many other conditioning structures one could consider, this one proved good enough for our experiments. In order to produce counterfactuals, we perform abduction by passing images and their parent attributes through the encoder and retrieving the posterior latent variables at each stochastic layer. We then fix these latents and

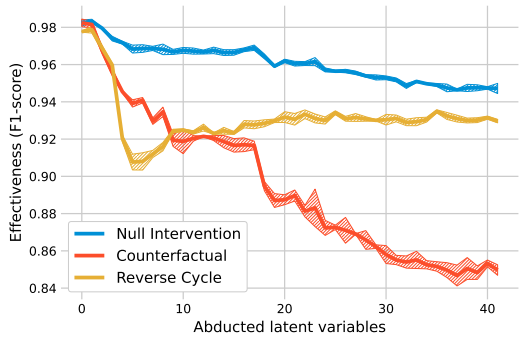
propagate them through the decoder along with the counterfactual parent conditioning. We also found it beneficial to replace the original diagonal discretized logistic mixture likelihood (Salimans et al. 2017) used in VDVAEs with a diagonal discretized Gaussian likelihood (Ho et al. 2020), as it produced visually sharper counterfactuals in our experiments. Importantly, our model was trained with a β penalty (Higgins et al. 2017) of 5 which discouraged it from focusing mostly on maximising the likelihood term in the ELBO and ignoring counterfactual conditioning at inference time. We found that this introduces a trade-off between reconstruction quality and obeying counterfactual conditioning.

We found that a naive application of a VDVAE to produce high resolution counterfactuals leads to ignored counterfactual parent conditioning. Although the setup we describe next worked well enough in our experiments, understanding and overcoming this issue in the general case likely warrants further investigation.

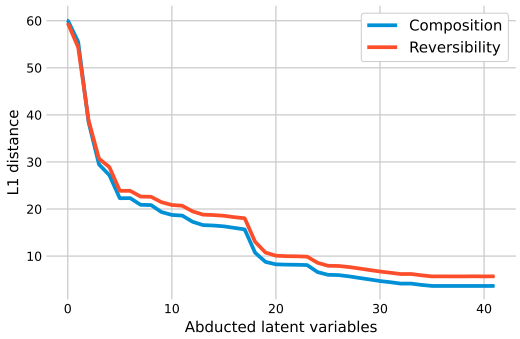
For our experiments, we used a VDVAE with stochastic latent variables spanning 6 resolution scales up to the 64×64 input resolution: $\{1^2, 4^2, 8^2, 16^2, 32^2, 64^2\}$, where each latent variable has 16 channels. We used the following number of residual blocks per resolution scale: $\{4, 4, 8, 12, 12, 4\}$, resulting in a total of 42 stochastic latent variable layers and 19M trainable parameters. Additionally, we modified the original architecture from a fixed channel width (e.g. 384) across all resolutions to the following custom channel widths per resolution: $\{32, 64, 128, 256, 512, 1024\}$. We found that reducing the number of channels at higher resolutions and increasing them for lower resolutions performed well enough in our experiments, whilst reducing both memory and runtime requirements significantly.

We trained the VDVAE for 1.7M steps with a batch size of 32 using the AdamW (Loshchilov and Hutter 2019) optimiser with an initial learning rate of 1e-3, $\beta_1 = 0.9$, $\beta_2 = 0.9$ and a weight decay of 0.01. The learning rate was linearly warmed-up from 0 to 1e-3 over the first 100 steps then reduced to 1.5e-4 at 175K steps and again to 1.5e-5 at 900K steps. We set gradient clipping to 220 and gradient L2 norm skipping threshold to 380, and observed no significant instabilities during training. The final artefact is an exponential moving average of the model parameters with a rate of 0.999 which we use at inference time. For data-augmentation, since the initial dataset was 128×128 resolution, during training we applied random cropping to 120×120 and resized to 64×64 before applying random horizontal flipping with probability 0.5 and finally scaling pixel intensities to $[-1, 1]$. At test time we simply resize to 64×64.

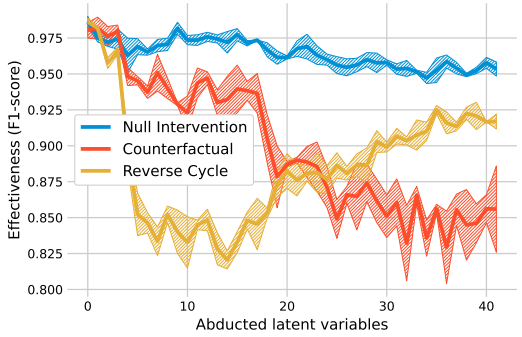
Our attribute classifiers were trained for 300 epochs with a batch size of 32 using the AdamW (Loshchilov and Hutter 2019) optimizer with an initial learning rate of 1e-3, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and weight decay of 0.01. The learning rate followed a plateau schedule on the validation set F1-score, where it was halved after 50 epochs of no improvement. The final artefact is an exponential moving average of the model parameters with a rate of 0.99 which we use at inference time. To find the best binary classification threshold between 0 and 1, we evaluated the validation set at threshold intervals of



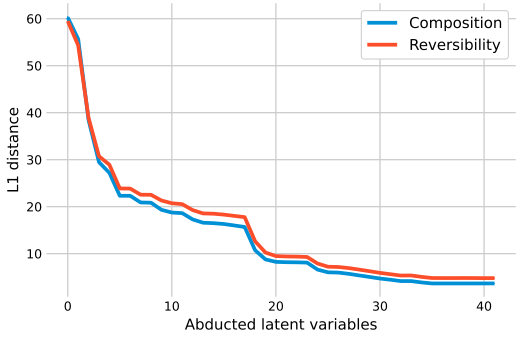
(a) Effectiveness ('smiling').



(b) Composition & Reversibility ('smiling').



(c) Effectiveness ('eyeglasses').



(d) Composition & Reversibility ('eyeglasses').

Figure 5.6: Measuring CelebA-HQ test set effectiveness, composition and reversibility as a function of the number of latent variables abducted.

0.001 and took the best performing threshold in terms of F1-score. We ran training from 3 random seeds and the final test set F1-score for the 'smiling' attribute is 0.9342 ± 0.0018, and for the 'eyeglasses' attribute it is 0.9758 ± 0.0037. For data augmentation, we followed the VDVAE setup but also included the $\mathrm{torchvision.transforms.AutoAugment}()$ augmentation module from PyTorch (Paszke et al. 2019) during training to strengthen regularisation. Since the CelebA-HQ binary attributes can be highly imbalanced, we also used a weighted random data sampler (with replacement) during training, which ensures minibatches have balanced representation of each class at each step.

Since the latent code is hierarchical, the abduction process can be decomposed into abducting subsets of latent variables at different resolutions. This added complexity enables higher fidelity images but makes abduction harder in practice. When producing counterfactuals, abducting all latent variables can result in the model ignoring the counterfactual parent conditioning. Conversely, abducting only a subset trends the model towards obeying the conditioning at the cost of faithfulness to the input observation. In other words, there is a trade-off between effectiveness and composition/reversibility.

Figure 5.6 show the evolution of effectiveness, composition, and reversibility with the number of fixed posterior latent variables. We can see the effectiveness of the counterfactual intervention

rapidly dropping as we abduct more (posterior) latent variables. The same is not true for the null-intervention and cycled-back counterfactuals, indicating that the model can maintain the original parent value at the cost of the ability to change it. Furthermore, we can see the opposite trend for composition and reversibility as they rapidly increase (decrease in l1) as we abduct more and more variables. We reveal a trade-off between effectiveness and composition mediated by the number of latent variables we abduct. Interestingly, as we abduct more and more variables, the effectiveness of the null-intervention counterfactual drops slightly, and the effectiveness of the cycled-back counterfactual dips before coming back up, this behaviour is somewhat unexpected since we should be closer to the input when we abduct more variables. We believe this can be explained by the interaction between the data-driven pseudo-oracles and the generative model, where examples where the conditioning is more obvious are easier to classify correctly.

Figure 5.7 visually shows the trade-off between effectiveness and composition. Figures 5.7a and 5.7c show 'smiling' and 'eyeglasses' counterfactuals for a model with 42 latent variables when abducting all variables. Figures 5.7b and 5.7d show the same counterfactuals when abducting only a subset of 8 variables. The respective numerical results are available in Table C.5. We can see a greater loss of subject identity when abducting only a subset of variables, but the conditioning is more evident than when abducting all variables. We see that, under full abduction, it is harder for the counterfactuals to obey the conditioning, but they are more faithful to the input (e.g. higher 'smiling' composition $l_1^{(1)} = 3.657 \pm 0.0006$ but lower effectiveness F1-score $= 0.848 \pm 0.0006$. In contrast, when using partial abduction, we obtain higher effectiveness F1-score $= 0.933 \pm 0.002$ at the cost of much lower composition $l_1^{(1)} = 20.890 \pm 0.018$. Figures 5.8 and 5.9 show granular evolution of the same counterfactuals when abducting fewer and fewer latent variables. We can see that the identity is progressively lost until we essentially end up with a random sample.

(a) Smiling (all latents) **(b)** Smiling (8 latents) **(c)** Eyeglasses (all latents) **(d)** Eyeglasses (8 latents)

Figure 5.7: Effectiveness/Composition plots for CelebA-HQ when abducting all 42 latent variables and only a subset of 8 variables. Each sub-figure has one sample per row: the first column is the input, the second column is the result of the null intervention, and the last column is the result of a partial intervention that flips the binary value of the initial parent.

Figure 5.8: Evolution of 'smiling' CelebA-HQ counterfactuals from full abduction (42 latents) to partial abduction (32-1 latents) and finally random samples (0 latents fixed).

Figure 5.9: Evolution of 'eyeglasses' CelebA-HQ counterfactuals from full abduction (42 latents) to partial abduction (32-1 latents) and finally random samples (0 latents fixed).

5.5 Discussion

We have presented a theoretically grounded framework for evaluating counterfactual inference models without observed counterfactuals or knowledge of the underlying SCM. While guarantees of truly identifiable models using deep learning are not possible in the general case, we show that we can measure the *soundness* of such models by observing which constraints they should obey ideally. Further, we show the impact of confounding on training and evaluation of deep counterfactual inference models and how its effect can be mitigated by using simulated interventions. We hope the ideas presented here can help inform the development and evaluation of future counterfactual inference models that use deep models as their base.

The ability to generate plausible image counterfactuals can have both productive and nefarious applications. On the positive side, counterfactual explanations have the potential to improve the interpretability of deep learning models and help bridge the gap between human and machine intelligence. Counterfactual queries may help to identify disparities in model performance, and counterfactual data augmentation can mitigate dataset bias against underrepresented groups in downstream tasks such as classification. With that said, if incorrectly used, counterfactual image models may also further exacerbate such biases. Moreover, visually plausible artificially generated counterfactual images could be misused by ill-intended parties to deceive, mislead or spread misinformation. We argue that the opportunities and risks of counterfactual image generation must be carefully considered throughout development and a comprehensive evaluation framework is integral to this process.

Chapter 6

Conclusion

6.1 Summary of contributions

In this section, we summarise the main contributions of the thesis and discuss their impact.

In Chapter 3, we developed an algorithm for the multiclass semantic segmentation of traumatic brain injuries (TBI) lesions in head CT. We demonstrated the ability of our algorithm to separately segment, quantify, and detect multiclass haemorrhagic lesions and perilesional oedema. This work illustrates the importance of working with clinicians to determine tasks we should tackle and which metrics are actually clinically relevant. In addition to the future promise of alleviating radiologists' workload, our work on automatic segmentation of TBI has already enabled large clinical studies which otherwise would be unfeasible. Automatically segmenting thousands of scans has enabled clinical researchers to better understand the disease and the impact of specific treatments on a population level.

- Mathieu, F., Güting, H., Gravesteijn, B., **Monteiro, M.**, Glocker, B., Kornaropoulos, E. N., Kamnistas, K., Robertson, C. S., Levin, H., Whitehouse, D. P., et al. (2020). "Impact of antithrombotic agents on radiological lesion progression in acute traumatic brain injury: a CENTER-TBI propensity-matched cohort analysis". In: *Journal of neurotrauma* 37.19, pp. 2069–2080
- Mathieu, F., Zeiler, F. A., Ercole, A., **Monteiro, M.**, Kamnitsas, K., Glocker, B., Whitehouse, D. P., Das, T., Smielewski, P., Czosnyka, M., et al. (2020). "Relationship between measures of cerebrovascular reactivity and intracranial lesion progression in acute traumatic brain injury patients: a CENTER-TBI study". In: *Journal of Neurotrauma* 37.13, pp. 1556–1565
- Zeiler, F. A., Mathieu, F., **Monteiro, M.**, Glocker, B., Ercole, A., Beqiri, E., Cabeleira, M., Stocchetti, N., Smielewski, P., Czosnyka, M., et al. (2020). "Diffuse intracranial injury patterns are associated with impaired cerebrovascular reactivity in adult traumatic brain injury: a CENTER-TBI Validation Study". In: *Journal of Neurotrauma* 37.14, pp. 1597–1608
- Zeiler, F. A., Mathieu, F., **Monteiro, M.**, Glocker, B., Ercole, A., Cabeleira, M., Stocchetti, N., Smielewski, P., Czosnyka, M., Newcombe, V., et al. (2021). "Systemic Markers of Injury and Injury Response Are Not Associated with Impaired Cerebrovascular Reactivity in Adult Traumatic Brain Injury: A Collaborative European Neurotrauma Effectiveness Research in Traumatic Brain Injury (CENTER-TBI) Study". In: *Journal of Neurotrauma* 38.7, pp. 870–878

- Whitehouse, D. P., **Monteiro, M.**, Czeiter, E., Vyvere, T. V., Valerio, F., Ye, Z., Amrein, K., Kamnitsas, K., Xu, H., Yang, Z., et al. (2022). "Relationship of admission blood proteomic biomarkers levels to lesion type and lesion burden in traumatic brain injury: A CENTER-TBI study". In: *EBioMedicine* 75, p. 103777

In Chapter 4, we presented an algorithm that models aleatoric uncertainty in semantic segmentation by considering dependencies between pixels. Unlike previous approaches, our model can produce spatially coherent label-map samples and more accurate uncertainty estimates. In short, it is a better model for the observed data distribution. Proper uncertainty estimation is crucial when applying deep learning to critical decision-making. For example, uncertainty estimation and the ability to produce multiple plausible outputs could be instrumental in applications such as radiotherapy planning. The simplicity of our approach means it is not limited to segmentation, and thus it has since been extended for image generation.

- Langley, J., **Monteiro, M.**, Jones, C., Pawlowski, N., and Glocker, B. (2022). "Structured Uncertainty in the Observation Space of Variational Autoencoders". In: *Transactions on Machine Learning Research*

In Chapter 5, we proposed a framework for evaluating counterfactual inference engines. Starting from the axiomatic definition of counterfactual, we developed a set of metrics that allows us to compare our approximate model to the ideal model without access to the latter. Our framework can be used to compare different approximate models and choose the best available. Additionally, it can be used to diagnose which aspects of identifiability our models are struggling with and point to solutions to these issues.

6.2 Limitations and Future Research

This section discusses the limitations of the work presented in this thesis and suggests future research directions to address these limitations.

6.2.1 Granularity and accuracy of automatic traumatic brain injury segmentation

Due to the chaotic nature of its mechanism of injury, traumatic brain injury is an extremely heterogeneous condition. It can display itself in a variety of different forms, which change over time after the initial injury. For this reason, its accurate segmentation and quantification are exceptionally difficult compared to other medical imaging tasks. While the segmentation performance of our methods was satisfactory, it is still far from those usually observed in easier tasks. We addressed this issue by showing that the clinically relevant metrics are not segmentation performance but lesion volume quantification and lesion detection. However, despite the performance achieved for those metrics being good enough for clinical research, there is a need for improvement before a

method like this can be used in clinical practice. Specifically, we need to increase the performance for extra-axial haemorrhage and perilesional oedema. For this, larger and more diverse datasets will be crucial.

Another significant limitation of our method is the lack of granularity in our class labels. In order to set ourselves an attainable goal, and given the limited amount of training data, we had to merge some classes. We merged petechial haemorrhages and intraparenchymal haemorrhages and traumatic subarachnoid haemorrhage, extradural haematoma and subdural haematoma. From a clinical perspective, the presence and volume of these lesion types imply different prognoses and lead to different treatment decisions. As a result, an important next step would be to develop an algorithm which can accurately separate these lesions. Alas, the heterogeneity of TBI and the noisiness of CT images make separating these classes challenging even for human experts. Imbuing an algorithm with this capability would require significantly more data than we currently have, data whose labelling cost would be extremely high.

Lastly, localising lesions within the brain can add relevant information for prognosis and treatment decisions. We have shown a proof-of-concept for this capability, but developing a fully-fledged tool capable of lesion localisation could enable more in-depth clinical research studies.

6.2.2 Local and global spatially correlated uncertainty

The low-parameterisation of the logit map covariance matrix proposed in Chapter 4 is, by design, a global parameterisation, meaning it does not consider the spatial proximity between pixels when measuring the degree of their dependency. Dependencies between pixels are calculated using a rank-dimensional vector, allowing the network to structure the covariance as it pleases within the rank-constrained space. While this gives us an easy and flexible parameterisation, it ignores a strong inductive prior, which is that pixels close together in space are more likely to have strong dependencies. Therefore, including a local component in the covariance matrix where we allow spatially neighbouring pixels to be modelled separately from the global component could improve our probabilistic model and give better uncertainty estimates. Our model would be capable of capturing long and short-range dependencies separately without a significant increase in computational cost, which is closer to the intuition we have for the true model of the data-generation process.

Another limitation of our work and of measuring the correctness of aleatoric uncertainty estimates in the broader sense is the lack of access to ground-truth uncertainty. The only way of truly measuring the correctness of our aleatoric uncertainty estimates is to compare the predicted and true data distributions. Given that we are modelling the conditional distribution of the label map given the input image, the only way of achieving this is to have multiple annotations for the same data point. While very few datasets like the LIDC-IDRI dataset (Armato III et al. 2011) already have this property, this is certainly not common for the vast majority of publicly available datasets. Collecting

multiple annotations per image is linearly more expensive and time-consuming. However, to make substantial advances in uncertainty modelling, we will need to invest in its proper measurement and objective quantification. In the future, we should strive to include uncertainty considerations from the start and collect multiple annotations for each data-point. We should also develop methods to understand how many annotations are necessary to provide confident uncertainty measurements.

Lastly, even correct uncertainty estimates are difficult to interpret and use productively. More investigation is necessary into how these uncertainty estimates are communicated to humans and how they can be used to influence decision-making positively.

6.2.3 Effectiveness metric depends on data and lack of full support for the observational distribution

The main limitation of the method presented in Chapter 5 is the necessity for pseudo-oracle classifiers/regressors trained from data to measure the effectiveness of the counterfactual inference model. Training the pseudo-oracles from data inevitably results in some empirical error that limits the accuracy of our evaluation framework. If we are not careful, it can also result in incorrect measurements due to spurious correlations in the data. Even though we showed that the relative ranking of models does not change much when using less accurate pseudo-oracles, an ideal measurement should be independent of the data. An objective measurement for effectiveness, like the ones we obtained for composition and reversibility, would increase the utility and confidence of our framework. However, it is hard to conceive a data-independent and accurate measurement for effectiveness for high-dimensional data with complex parents. In reality, we should aim to create a data-driven metric that, although imperfect, gives us some theoretical guarantees about the relative performance of models and its gap to the ideal unattainable metric.

Another related limitation of our method is that simulated interventions are restricted by the available data. We cannot perform a simulated intervention if the observational distribution does not have full support over the parent marginals. The lack of support means that we cannot train pseudo-oracles to measure the independent effect of a specific parent on the counterfactual in a disentangled manner. Additionally, in instances where there is full support but it is provided by a limited amount of data points, the performance of the pseudo-oracle can be hindered by the repetition of the same data points too often.

Bibliography

Adamson, A. S. and Smith, A. (2018). "Machine learning and health care disparities in dermatology". In: *JAMA dermatology* 154.11, pp. 1247–1248.

Amyot, F., Arciniegas, D. B., Brazaitis, M. P., Curley, K. C., Diaz-Arrastia, R., Gandjbakhche, A., Herscovitch, P., Hinds, S. R., Manley, G. T., Pacifico, A., et al. (2015). "A review of the effectiveness of neuroimaging modalities for the detection of traumatic brain injury". In: *Journal of Neurotrauma* 32.22, pp. 1693–1721.

An, J., Ying, L., and Zhu, Y. (2021). "Why resampling outperforms reweighting for correcting sampling bias with stochastic gradients". In: *International Conference on Learning Representations*.

Armato III, S. G., McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, A. P., Zhao, B., Aberle, D. R., Henschke, C. I., Hoffman, E. A., et al. (2011). "The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans". In: *Medical Physics* 38.2, pp. 915–931.

Arnab, A., Zheng, S., Jayasumana, S., Romera-Paredes, B., Larsson, M., Kirillov, A., Savchynskyy, B., Rother, C., Kahl, F., and Torr, P. H. (2018). "Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction". In: *IEEE Signal Processing Magazine* 35.1, pp. 37–52.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473*.

Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J. S., Freymann, J. B., Farahani, K., and Davatzikos, C. (2017). "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features". In: *Scientific data* 4.1, pp. 1–13.

Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R. T., Berger, C., Ha, S. M., Rozycki, M., et al. (2018). "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge". In: *arXiv preprint arXiv:1811.02629*.

Bardera, A., Boada, I., Feixas, M., Remollo, S., Blasco, G., Silva, Y., and Pedraza, S. (2009). "Semi-automated method for brain hematoma and edema quantification using computed tomography". In: *Computerized medical imaging and graphics* 33.4, pp. 304–311.

Bareinboim, E., Correa, J. D., Ibeling, D., and Icard, T. (2022). "On Pearl's hierarchy and the foundations of causal inference". In: *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 507–556.

Batra, D., Yadollahpour, P., Guzman-Rivera, A., and Shakhnarovich, G. (2012). "Diverse M-best solutions in Markov random fields". In: *European Conference on Computer Vision*, pp. 1–16.

Baumgartner, C. F., Tezcan, K. C., Chaitanya, K., Hötker, A. M., Muehlematter, U. J., Schawkat, K., Becker, A. S., Donati, O., and Konukoglu, E. (2019). "PHiSeg: capturing uncertainty in medical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention – MICCAI 2019*, pp. 119–127.

Bengio, Y., Courville, A., and Vincent, P. (2013). "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1798–1828.

Bernhardt, M., Castro, D. C., Tanno, R., Schwaighofer, A., Tezcan, K. C., Monteiro, M., Bannur, S., Lungren, M. P., Nori, A., Glocker, B., et al. (2022). "Active label cleaning for improved dataset quality under resource constraints". In: *Nature communications* 13.1, pp. 1–11.

Besserve, M., Mehrjou, A., Sun, R., and Schölkopf, B. (2020). "Counterfactuals uncover the modular structure of deep generative models". In: *International Conference on Learning Representations*.

Besserve, M., Sun, R., Janzing, D., and Schölkopf, B. (2021). "A theory of independent mechanisms for extrapolation in generative models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 8, pp. 6741–6749.

Bhadauria, H. S., Singh, A., and Dewal, M. (2013). "An integrated method for hemorrhage segmentation from brain CT imaging". In: *Computers & Electrical Engineering* 39.5, pp. 1527–1536.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. Vol. 4. 4. Springer.

Blake, A., P. Kohli, and C. Rother, eds. (2011). *Markov random fields for vision and image processing*. MIT Press.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). "Weight uncertainty in neural network". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research, pp. 1613–1622.

Bouchacourt, D., Tomioka, R., and Nowozin, S. (2018). "Multi-level variational autoencoder: Learning disentangled representations from grouped observations". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Buolamwini, J. and Gebru, T. (2018). "Gender shades: Intersectional accuracy disparities in commercial gender classification". In: *Conference on fairness, accountability and transparency*, pp. 77–91.

Burgess, C. and Kim, H. (2018). *3D Shapes Dataset*. https://github.com/deepmind/3dshapes-dataset/.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). "Understanding disentangling in beta-VAE". In: *arXiv preprint arXiv:1804.03599*.

Carney, N., Totten, A. M., O'Reilly, C., Ullman, J. S., Hawryluk, G. W., Bell, M. J., Bratton, S. L., Chesnut, R., Harris, O. A., Kissoon, N., et al. (2017). "Guidelines for the management of severe traumatic brain injury". In: *Neurosurgery* 80.1, pp. 6–15.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018). "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4, pp. 834–848.

Chen, R. T., Li, X., Grosse, R. B., and Duvenaud, D. K. (2018). "Isolating sources of disentanglement in variational autoencoders". In: *Advances in neural information processing systems* 31.

Chilamkurthy, S., Ghosh, R., Tanamala, S., Biviji, M., Campeau, N. G., Venugopal, V. K., Mahajan, V., Rao, P., and Warier, P. (2018). "Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study". In: *The Lancet* 392.10162, pp. 2388–2396.

Child, R. (2021). "Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images". In: *International Conference on Learning Representations*.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. (2018). "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8789–8797.

Clopper, C. J. and Pearson, E. S. (1934). "The use of confidence or fiducial limits illustrated in the case of the binomial". In: *Biometrika* 26.4, pp. 404–413.

Collaborators, C. T. et al. (2004). "Effect of intravenous corticosteroids on death within 14 days in 10 008 adults with clinically significant head injury (MRC CRASH trial): randomised placebo-controlled trial". In: *The Lancet* 364.9442, pp. 1321–1328.

Collaborators, C. T. et al. (2005). "Final results of MRC CRASH, a randomised placebo-controlled trial of intravenous corticosteroid in adults with head injury—outcomes at 6 months". In: *The Lancet* 365.9475, pp. 1957–1959.

Criminisi, A., Shotton, J., Konukoglu, E., et al. (2012). "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning". In: *Foundations and trends® in computer graphics and vision* 7.2–3, pp. 81–227.

Dash, S., Balasubramanian, V. N., and Sharma, A. (2022). "Evaluating and mitigating bias in image classifiers: A causal perspective using counterfactuals". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 915–924.

Dewan, M. C., Rattani, A., Gupta, S., Baticulon, R. E., Hung, Y.-C., Punchak, M., Agrawal, A., Adeleye, A. O., Shrime, M. G., Rubiano, A. M., et al. (2018). "Estimating the global incidence of traumatic brain injury". In: *Journal of neurosurgery* 130.4, pp. 1080–1097.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). "Dermatologist-level classification of skin cancer with deep neural networks". In: *nature* 542.7639, pp. 115–118.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., and Dean, J. (2019). "A guide to deep learning in healthcare". In: *Nature medicine* 25.1, pp. 24–29.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). "The Pascal visual object classes challenge: A retrospective". In: *International Journal of Computer Vision* 111.1, pp. 98–136.

Figueira Rodrigues Vieira, G. and Guedes Correa, J. F. (2020). "Early computed tomography for acute post-traumatic diffuse axonal injury: a systematic review". In: *Neuroradiology* 62.6, pp. 653–660.

Gal, Y. and Ghahramani, Z. (2016). "Dropout as a Bayesian approximation: representing model uncertainty in deep learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research, pp. 1050–1059.

Galles, D. and Pearl, J. (1998). "An axiomatic characterization of causal counterfactuals". In: *Foundations of Science* 3.1, pp. 151–182.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). "Shortcut learning in deep neural networks". In: *Nature Machine Intelligence* 2.11, pp. 665–673.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). "Generative adversarial networks". In: *Communications of the ACM* 63.11, pp. 139–144.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., et al. (2016). "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs". In: *Jama* 316.22, pp. 2402–2410.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). "On Calibration of Modern Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research, pp. 1321–1330.

Haacke, E. M., Duhaime, A. C., Gean, A. D., Riedy, G., Wintermark, M., Mukherjee, P., Brody, D. L., DeGraba, T., Duncan, T. D., Elovic, E., et al. (2010). "Common data elements in radiologic imaging of traumatic brain injury". In: *Journal of Magnetic Resonance Imaging* 32.3, pp. 516–543.

Halpern, J. Y. (2000). "Axiomatizing causal reasoning". In: *Journal of Artificial Intelligence Research* 12, pp. 317–337.

Hanley, J. A. and McNeil, B. J. (1982). "The meaning and use of the area under a receiver operating characteristic (ROC) curve." In: *Radiology* 143.1, pp. 29–36.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

– (2016b). "Identity mappings in deep residual networks". In: *European Conference on Computer Vision*, pp. 630–645.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *International Conference on Learning Representations*.

Ho, J., Jain, A., and Abbeel, P. (2020). "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33, pp. 6840–6851.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). "CyCADA: Cycle-Consistent Adversarial Domain Adaptation". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research, pp. 1989–1998.

Hosoya, H. (2019). "Group-based Learning of Disentangled Representations with Generalizability for Novel Contents". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 2506–2513.

Hu, S., Worrall, D., Knegt, S., Veeling, B., Huisman, H., and Welling, M. (2019). "Supervised uncertainty quantification for segmentation with multiple annotations". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 137–145.

Ilse, M., Tomczak, J. M., and Forré, P. (2021). "Selecting Data Augmentation for Simulating Interventions". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research, pp. 4555–4562.

Ioffe, S. and Szegedy, C. (2015). "Batch normalization: accelerating deep network training by reducing internal covariate shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research, pp. 448–456.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). "Image-To-Image Translation With Conditional Adversarial Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1125–1134.

Jain, S., Vyvere, T. V., Terzopoulos, V., Sima, D. M., Roura, E., Maas, A., Wilms, G., and Verheyden, J. (2019). "Automatic quantification of computed tomography features in acute traumatic brain injury". In: *Journal of Neurotrauma* 36.11, pp. 1794–1803.

Janzing, D. and Schölkopf, B. (2010). "Causal inference using the algorithmic Markov condition". In: *IEEE Transactions on Information Theory* 56.10, pp. 5168–5194.

Jesson, A., Mindermann, S., Shalit, U., and Gal, Y. (2020). "Identifying causal-effect inference failure with uncertainty-aware models". In: *Advances in Neural Information Processing Systems* 33, pp. 11637–11649.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873, pp. 583–589.

Jungo, A., Balsiger, F., and Reyes, M. (2020). "Analyzing the Quality and Challenges of Uncertainty Estimations for Brain Tumor Segmentation". In: *Frontiers in Neuroscience* 14, p. 282.

Kamnitsas, K., Ferrante, E., Parisot, S., Ledig, C., Nori, A. V., Criminisi, A., Rueckert, D., and Glocker, B. (2016). "DeepMedic for brain tumor segmentation". In: *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 138–149.

Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., and Glocker, B. (2017). "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation". In: *Medical image analysis* 36, pp. 61–78.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *International Conference on Learning Representations*.

Kendall, A. and Gal, Y. (2017). "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems*. Vol. 30.

Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. (2020). "Variational Autoencoders and Nonlinear ICA: A Unifying Framework". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Vol. 108. Proceedings of Machine Learning Research, pp. 2207–2217.

Kim, B., Kim, H., Kim, K., Kim, S., and Kim, J. (2019). "Learning Not to Learn: Training Deep Neural Networks With Biased Data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9012–9020.

Kim, H. and Mnih, A. (2018). "Disentangling by Factorising". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research, pp. 2649–2658.

Kingma, D. P. and Ba, J. (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR*.

Kingma, D. P. and Welling, M. (2014). "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014*.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). "Improved variational inference with inverse autoregressive flow". In: *Advances in neural information processing systems* 29.

Kirillov, A., Savchynskyy, B., Schlesinger, D., Vetrov, D., and Rother, C. (2015). "Inferring M-Best diverse labelings in a single one". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1814–1822.

Kohl, S., Romera-Paredes, B., Maier-Hein, K. H., Rezende, D. J., Eslami, S. M. A., Kohli, P., Zisserman, A., and Ronneberger, O. (2019). "A hierarchical probabilistic U-Net for modeling multi-scale ambiguities". In: *arXiv preprint arXiv:1905.13077*.

Kohl, S., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J. R., Maier-Hein, K., Eslami, S., Jimenez Rezende, D., and Ronneberger, O. (2018). "A probabilistic U-Net for segmentation of mmbiguous images". In: 31.

Krähenbühl, P. and Koltun, V. (2011). "Efficient inference in fully connected CRFs with Gaussian edge potentials". In: *Advances in Neural Information Processing Systems* 24.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6, pp. 84–90.

Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. (2019). "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration". In: *Advances in neural information processing systems* 32.

Kumar, A., Sattigeri, P., and Balakrishnan, A. (2018). "Variational Inference of Disentangled Latent Concepts from Unlabeled Observations". In: *International Conference on Learning Representations*.

Kuo, W., Häne, C., Mukherjee, P., Malik, J., and Yuh, E. L. (2019). "Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning". In: *Proceedings of the National Academy of Sciences* 116.45, pp. 22737–22745.

Kurland, D., Hong, C., Aarabi, B., Gerzanich, V., and Simard, J. M. (2012). "Hemorrhagic progression of a contusion after traumatic brain injury: a review". In: *Journal of Neurotrauma* 29.1, pp. 19–31.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems* 30.

Langley, J., Monteiro, M., Jones, C., Pawlowski, N., and Glocker, B. (2022). "Structured Uncertainty in the Observation Space of Variational Autoencoders". In: *Transactions on Machine Learning Research*.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4, pp. 541–551.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., and Batra, D. (2015). "Why M heads are better than one: Training a diverse ensemble of deep networks". In: *arXiv preprint arXiv:1511.06314*.

Lee, S., Purushwalkam Shiva Prakash, S., Cogswell, M., Ranjan, V., Crandall, D., and Batra, D. (2016). "Stochastic multiple choice learning for training diverse deep ensembles". In: *Advances in Neural Information Processing Systems* 29.

Li, D., Yang, J., Kreis, K., Torralba, A., and Fidler, S. (2021). "Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8300–8311.

Liu, M.-Y., Breuel, T., and Kautz, J. (2017). "Unsupervised image-to-image translation networks". In: *Advances in neural information processing systems* 30.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2020). "A Sober Look at the Unsupervised Learning of Disentangled Representations and their Evaluation". In: *Journal of Machine Learning Research* 21.209, pp. 1–62.

Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. (2020). "Weakly-supervised disentanglement without compromises". In: *International Conference on Machine Learning*, pp. 6348–6359.

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Looveren, A. V. and Klaise, J. (2021). "Interpretable counterfactual explanations guided by prototypes". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 650–665.

Loshchilov, I. and Hutter, F. (2019). "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. (2017). "Causal effect inference with deep latent-variable models". In: *Advances in neural information processing systems* 30.

Ma, Y.-A., Chen, T., and Fox, E. (2015). "A complete recipe for stochastic gradient MCMC". In: *Advances in neural information processing systems* 28.

Maaløe, L., Fraccaro, M., Liévin, V., and Winther, O. (2019). "Biva: A very deep hierarchy of latent variables for generative modeling". In: *Advances in neural information processing systems* 32.

Maas, A. I., Menon, D. K., Adelson, P. D., Andelic, N., Bell, M. J., Belli, A., Bragge, P., Brazinova, A., Büki, A., Chesnut, R. M., et al. (2017). "Traumatic brain injury: integrated approaches to improve prevention, clinical care, and research". In: *The Lancet Neurology* 16.12, pp. 987–1048.

Maas, A. I., Menon, D. K., Steyerberg, E. W., Citerio, G., Lecky, F., Manley, G. T., Hill, S., Legrand, V., and Sorgner, A. (2015). "Collaborative European NeuroTrauma effectiveness research in traumatic

brain injury (CENTER-TBI) a prospective longitudinal observational study". In: *Neurosurgery* 76.1, pp. 67–80.

MacKay, D. J. (1992). "Bayesian interpolation". In: *Neural Computation* 4.3, pp. 415–447.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. (2019). "Fairness through causal awareness: Learning causal latent-variable models for biased data". In: *Proceedings of the conference on fairness, accountability, and transparency*, pp. 349–358.

Malinin, A. and Gales, M. (2018). "Predictive uncertainty estimation via prior networks". In: *Advances in neural information processing systems* 31.

– (2019). "Reverse KL-divergence training of prior networks: Improved uncertainty and adversarial robustness". In: *Advances in Neural Information Processing Systems* 32.

Marshall, L. F., Marshall, S. B., Klauber, M. R., Berkum Clark, M. van, Eisenberg, H. M., Jane, J. A., Luerssen, T. G., Marmarou, A., and Foulkes, M. A. (1991). "A new classification of head injury based on computerized tomography". In: *Journal of Neurosurgery* 75.Supplement, S14–S20.

Mathieu, F., Güting, H., Gravesteijn, B., Monteiro, M., Glocker, B., Kornaropoulos, E. N., Kamnistas, K., Robertson, C. S., Levin, H., Whitehouse, D. P., et al. (2020). "Impact of antithrombotic agents on radiological lesion progression in acute traumatic brain injury: a CENTER-TBI propensity-matched cohort analysis". In: *Journal of neurotrauma* 37.19, pp. 2069–2080.

Mathieu, F., Zeiler, F. A., Ercole, A., Monteiro, M., Kamnitsas, K., Glocker, B., Whitehouse, D. P., Das, T., Smielewski, P., Czosnyka, M., et al. (2020). "Relationship between measures of cerebrovascular reactivity and intracranial lesion progression in acute traumatic brain injury patients: a CENTER-TBI study". In: *Journal of Neurotrauma* 37.13, pp. 1556–1565.

Max, A. W. (1950). "Inverting modified matrices". In: *Memorandum Rept. 42, Statistical Research Group*. Princeton University, p. 4.

Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al. (2014). "The multimodal brain tumor image segmentation benchmark (BRATS)". In: *IEEE transactions on medical imaging* 34.10, pp. 1993–2024.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). "Human-level control through deep reinforcement learning". In: *nature* 518.7540, pp. 529–533.

Monteiro, M., Kamnitsas, K., Ferrante, E., Mathieu, F., McDonagh, S., Cook, S., Stevenson, S., Das, T., Khetani, A., Newman, T., et al. (2019). "Tbi lesion segmentation in head ct: Impact of preprocessing and data augmentation". In: *International MICCAI Brainlesion Workshop*, pp. 13–22.

Monteiro, M., Le Folgoc, L., Coelho de Castro, D., Pawlowski, N., Marques, B., Kamnitsas, K., Wilk, M. van der, and Glocker, B. (2020). "Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty". In: *Advances in Neural Information Processing Systems* 33, pp. 12756–12767.

Monteiro, M., Newcombe, V. F., Mathieu, F., Adatia, K., Kamnitsas, K., Ferrante, E., Das, T., Whitehouse, D., Rueckert, D., Menon, D. K., et al. (2020). "Multiclass semantic segmentation and quantification of traumatic brain injury lesions on head CT using deep learning: an algorithm development and multicentre validation study". In: *The Lancet Digital Health* 2.6, e314–e322.

Monteiro, M., Ribeiro, F. D. S., Pawlowski, N., Castro, D. C., and Glocker, B. (2023). "Measuring axiomatic soundness of counterfactual image models". In: *The Eleventh International Conference on Learning Representations*.

Murray, G. D., Butcher, I., McHugh, G. S., Lu, J., Mushkudiani, N. A., Maas, A. I., Marmarou, A., and Steyerberg, E. W. (2007). "Multivariable prognostic analysis in traumatic brain injury: results from the IMPACT study". In: *Journal of Neurotrauma* 24.2, pp. 329–337.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. (2020). "Learning from failure: De-biasing classifier from biased classifier". In: *Advances in Neural Information Processing Systems* 33, pp. 20673–20684.

Narayan, R. K., Maas, A. I., Servadei, F., Skolnick, B. E., Tillinger, M. N., Marshall, L. F., and Group, T. I. H. S. (2008). "Progression of traumatic intracerebral hemorrhage: a prospective observational study". In: *Journal of Neurotrauma* 25.6, pp. 629–639.

Neal, R. M. (1993). *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, ON, Canada.

Nowozin, S., Cseke, B., and Tomioka, R. (2016). "f-gan: Training generative neural samplers using variational divergence minimization". In: *Advances in neural information processing systems* 29.

Obermeyer, Z., Powers, B., Vogeli, C., and Mullainathan, S. (2019). "Dissecting racial bias in an algorithm used to manage the health of populations". In: *Science* 366.6464, pp. 447–453.

Oertel, M., Kelly, D. F., McArthur, D., Boscardin, W. J., Glenn, T. C., Lee, J. H., Gravori, T., Obukhov, D., McBride, D. Q., and Martin, N. A. (2002). "Progressive hemorrhage after head trauma: predictors and consequences of the evolving injury". In: *Journal of neurosurgery* 96.1, pp. 109–116.

Paisley, J., Blei, D., and Jordan, M. (2012). "Variational Bayesian inference with stochastic search". In: *arXiv preprint arXiv:1206.6430*.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). "Normalizing Flows for Probabilistic Modeling and Inference". In: *Journal of Machine Learning Research* 22.57, pp. 1–64.

Parascandolo, G., Kilbertus, N., Rojas-Carulla, M., and Schölkopf, B. (2018). "Learning Independent Causal Mechanisms". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research, pp. 4036–4044.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32.

Pawlowski, N. (2021). "Probabilistic and Causal Reasoning in Deep Learning for Imaging". PhD thesis. Imperial College London.

Pawlowski, N., Coelho de Castro, D., and Glocker, B. (2020). "Deep structural causal models for tractable counterfactual inference". In: *Advances in Neural Information Processing Systems* 33, pp. 857–869.

Pearl, J. (2009). *Causality*. Cambridge University Press.

– (2010). "An introduction to causal inference". In: *The International Journal of Biostatistics* 6.2, pp. 1–62.

Peebles, W., Peebles, J., Zhu, J.-Y., Efros, A., and Torralba, A. (2020). "The Hessian penalty: A weak prior for unsupervised disentanglement". In: *European Conference on Computer Vision*, pp. 581–597.

Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

Phillips, J. M. (2021). *Mathematical Foundations for Data Analysis*. Springer.

Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2008). *Dataset shift in machine learning*. MIT Press.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. Proceedings of Machine Learning Research 2, pp. 1278–1286.

Rissanen, S. and Marttinen, P. (2021). "A critical look at the consistency of causal estimation with deep latent variable models". In: *Advances in Neural Information Processing Systems* 34, pp. 4207–4217.

Roberts, M., Driggs, D., Thorpe, M., Gilbey, J., Yeung, M., Ursprung, S., Aviles-Rivero, A. I., Etmann, C., McCague, C., Beer, L., et al. (2021). "Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans". In: *Nature Machine Intelligence* 3.3, pp. 199–217.

Robinson, R. D. (2020). "Reliable Machine Learning for Medical Imaging Data through Automated Quality Control and Data Harmonization". In.

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241.

Rosnati, M., Soreq, E., Monteiro, M., Li, L., Graham, N. S., Zimmerman, K., Rossi, C., Carrara, G., Bertolini, G., Sharp, D. J., et al. (2022). "Automatic lesion analysis for increased efficiency in outcome prediction of traumatic brain injury". In: *International Workshop on Machine Learning in Clinical Neuroimaging*, pp. 135–146.

Roy, S., Wilkes, S., Diaz-Arrastia, R., Butman, J. A., and Pham, D. L. (2015). "Intraparenchymal hemorrhage segmentation from clinical head CT of patients with traumatic brain injury". In: *Medical Imaging 2015: Image Processing*. Vol. 9413, pp. 125–130.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning representations by back-propagating errors". In: *Nature* 323.6088, pp. 533–536.

Rupprecht, C., Laina, I., DiPietro, R., Baust, M., Tombari, F., Navab, N., and Hager, G. D. (2017). "Learning in an Uncertain World: Representing Ambiguity Through Multiple Hypotheses". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3591–3600.

Sainath, T. N., Kingsbury, B., Saon, G., Soltau, H., Mohamed, A.-r., Dahl, G., and Ramabhadran, B. (2015). "Deep convolutional neural networks for large-scale speech tasks". In: *Neural networks* 64, pp. 39–48.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. (2017). "PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications". In: *International Conference on Learning Representations*.

Sanchez, P. and Tsaftaris, S. A. (2022). "Diffusion Causal Models for Counterfactual Estimation". In: *Proceedings of the First Conference on Causal Learning and Reasoning*. Vol. 177, pp. 647–668.

Saon, G., Soltau, H., Nahamoo, D., and Picheny, M. (2013). "Speaker adaptation of neural network acoustic models using i-vectors". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 55–59.

Sauer, A. and Geiger, A. (2021). "Counterfactual Generative Networks". In: *International Conference on Learning Representations*.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. (2021). "Toward causal representation learning". In: *Proceedings of the IEEE* 109.5, pp. 612–634.

Sensoy, M., Kaplan, L., and Kandemir, M. (2018). "Evidential deep learning to quantify classification uncertainty". In: *Advances in neural information processing systems* 31.

Shi, C., Blei, D., and Veitch, V. (2019). "Adapting neural networks for the estimation of treatment effects". In: *Advances in neural information processing systems* 32.

Shu, R., Chen, Y., Kumar, A., Ermon, S., and Poole, B. (2020). "Weakly Supervised Disentanglement with Guarantees". In: *International Conference on Learning Representations*.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play". In: *Science* 362.6419, pp. 1140–1144.

Simonyan, K. and Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Sohn, K., Lee, H., and Yan, X. (2015). "Learning structured output representation using deep conditional generative models". In: *Advances in neural information processing systems* 28, pp. 3483–3491.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. (2016). "Ladder variational autoencoders". In: *Advances in neural information processing systems* 29.

Steyerberg, E. W., Wiegers, E., Sewalt, C., Buki, A., Citerio, G., De Keyser, V., Ercole, A., Kunzmann, K., Lanyon, L., Lecky, F., et al. (2019). "Case-mix, care pathways, and outcomes in patients with traumatic brain injury in CENTER-TBI: a European prospective, multicentre, longitudinal, cohort study". In: *The Lancet Neurology* 18.10, pp. 923–934.

Székely, G. J. and Rizzo, M. L. (2013). "Energy statistics: A class of statistics based on distances". In: *Journal of statistical planning and inference* 143.8, pp. 1249–1272.

Tanno, R., Worrall, D. E., Ghosh, A., Kaden, E., Sotiropoulos, S. N., Criminisi, A., and Alexander, D. C. (2017). "Bayesian image quality transfer with CNNs: exploring uncertainty in dMRI super-resolution". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 611–619.

Tieleman, T. and Hinton, G. (2012). "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude". In: *COURSERA: Neural networks for machine learning* 4.2, pp. 26–31.

Vande Vyvere, T., Wilms, G., Claes, L., Martin Leon, F., Nieboer, D., Verheyden, J., Van Den Hauwe, L., Pullens, P., Maas, A. I., Parizel, P. M., et al. (2019). "Central versus local radiological reading of acute computed tomography characteristics in multi-center traumatic brain injury research". In: *Journal of Neurotrauma* 36.7, pp. 1080–1092.

Von Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., and Locatello, F. (2021). "Self-supervised learning with data augmentations provably isolates content from style". In: *Advances in neural information processing systems* 34, pp. 16451–16467.

Wang, G., Li, W., Aertsen, M., Deprest, J., Ourselin, S., and Vercauteren, T. (2019). "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks". In: *Neurocomputing* 338, pp. 34–45.

Welling, M. and Teh, Y. W. (2011). "Bayesian learning via stochastic gradient Langevin dynamics". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. Proceedings of Machine Learning Research, pp. 681–688.

Whitehouse, D. P., Monteiro, M., Czeiter, E., Vyvere, T. V., Valerio, F., Ye, Z., Amrein, K., Kamnitsas, K., Xu, H., Yang, Z., et al. (2022). "Relationship of admission blood proteomic biomarkers levels to lesion type and lesion burden in traumatic brain injury: A CENTER-TBI study". In: *EBioMedicine* 75, p. 103777.

Wiens, J., Saria, S., Sendak, M., Ghassemi, M., Liu, V. X., Doshi-Velez, F., Jung, K., Heller, K., Kale, D., Saeed, M., et al. (2019). "Do no harm: a roadmap for responsible machine learning for health care". In: *Nature medicine* 25.9, pp. 1337–1340.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144*.

Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., and Wang, J. (2021). "CausalVAE: Disentangled representation learning via neural structural causal models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9593–9602.

Yoon, J., Jordon, J., and Van Der Schaar, M. (2018). "GANITE: Estimation of Individualized Treatment Effects using Generative Adversarial Nets". In: *International Conference on Learning Representations*.

Zaki, W. M. D. W., Fauzi, M. F. A., Besar, R., and Ahmad, W. M. W. (2011). "Qualitative and quantitative comparisons of haemorrhage intracranial segmentation in CT brain images". In: *TENCON 2011-2011 IEEE Region 10 Conference*, pp. 369–373.

Zeiler, F. A., Mathieu, F., Monteiro, M., Glocker, B., Ercole, A., Beqiri, E., Cabeleira, M., Stocchetti, N., Smielewski, P., Czosnyka, M., et al. (2020). "Diffuse intracranial injury patterns are associated with impaired cerebrovascular reactivity in adult traumatic brain injury: a CENTER-TBI Validation Study". In: *Journal of Neurotrauma* 37.14, pp. 1597–1608.

Zeiler, F. A., Mathieu, F., Monteiro, M., Glocker, B., Ercole, A., Cabeleira, M., Stocchetti, N., Smielewski, P., Czosnyka, M., Newcombe, V., et al. (2021). "Systemic Markers of Injury and Injury Response Are Not Associated with Impaired Cerebrovascular Reactivity in Adult Traumatic Brain Injury: A Collaborative European Neurotrauma Effectiveness Research in Traumatic Brain Injury (CENTER-TBI) Study". In: *Journal of Neurotrauma* 38.7, pp. 870–878.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. (2015). "Conditional random fields as recurrent neural networks". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1529–1537.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., and Shechtman, E. (2017). "Toward multimodal image-to-image translation". In: *Advances in neural information processing systems* 30.

Appendix A

Supplementary material for segmentation of traumatic brain injury

A.1 Absolute volume error

Figure A.4 shows that the absolute error tends to increase with lesion volume. The absolute error in volume is driven by the surface area of lesions since voxel misclassification typically occurs at the boundary zone with the surrounding tissue. It follows that large lesions have larger absolute error because they have a larger surface. However, as can be seen in Figure 3.3, the predicted volume is still well-calibrated. For small lesions, although there are many cases where volumes are predicted even though no lesion is present and vice versa, the absolute error remains low (Figure A.4). Hence, it follows that the predicted segmentation can still be useful for quantifying the lesion volume even in cases where location estimates are less accurate.

A.2 Lesion localisation

This section illustrates how predicted segmentation maps can be used to localise and quantify lesions within the brain. By registering the image and the model's prediction to a labelled atlas space, we can intersect the prediction with the atlas' labelled regions to obtain how much volume of each lesion class is present in each of the regions. For demonstrative purposes, we built a CT atlas by using 20 normal CT scans that show no disease using an iterative unbiased atlas construction.8 This atlas was subsequently aligned to an MNI MRI atlas where 19 regions were labelled: background; brain stem; left cerebellum; right cerebellum; left basal forebrain; left basal ganglia; left frontal lobe; left insula; left occipital lobe; left parietal lobe; left temporal; right basal forebrain; right basal ganglia; right frontal lobe; right insula; right occipital lobe; right parietal lobe; right temporal; ventricles. In Figure A.6, each column represents one case: from top to bottom (matching slices), we can see the atlas, atlas with labelled regions, image and image overlaid with lesion segmentation. The image is registered to the atlas. Once the image is in the same space as the atlas, since the atlas is labelled, we can intersect the lesion segmentation with the labelled regions and hence localise the lesions in atlas space. For example, for the second column, after registration, we now can say that there is an IPH lesion with a volume of 2mL in the left frontal lobe.

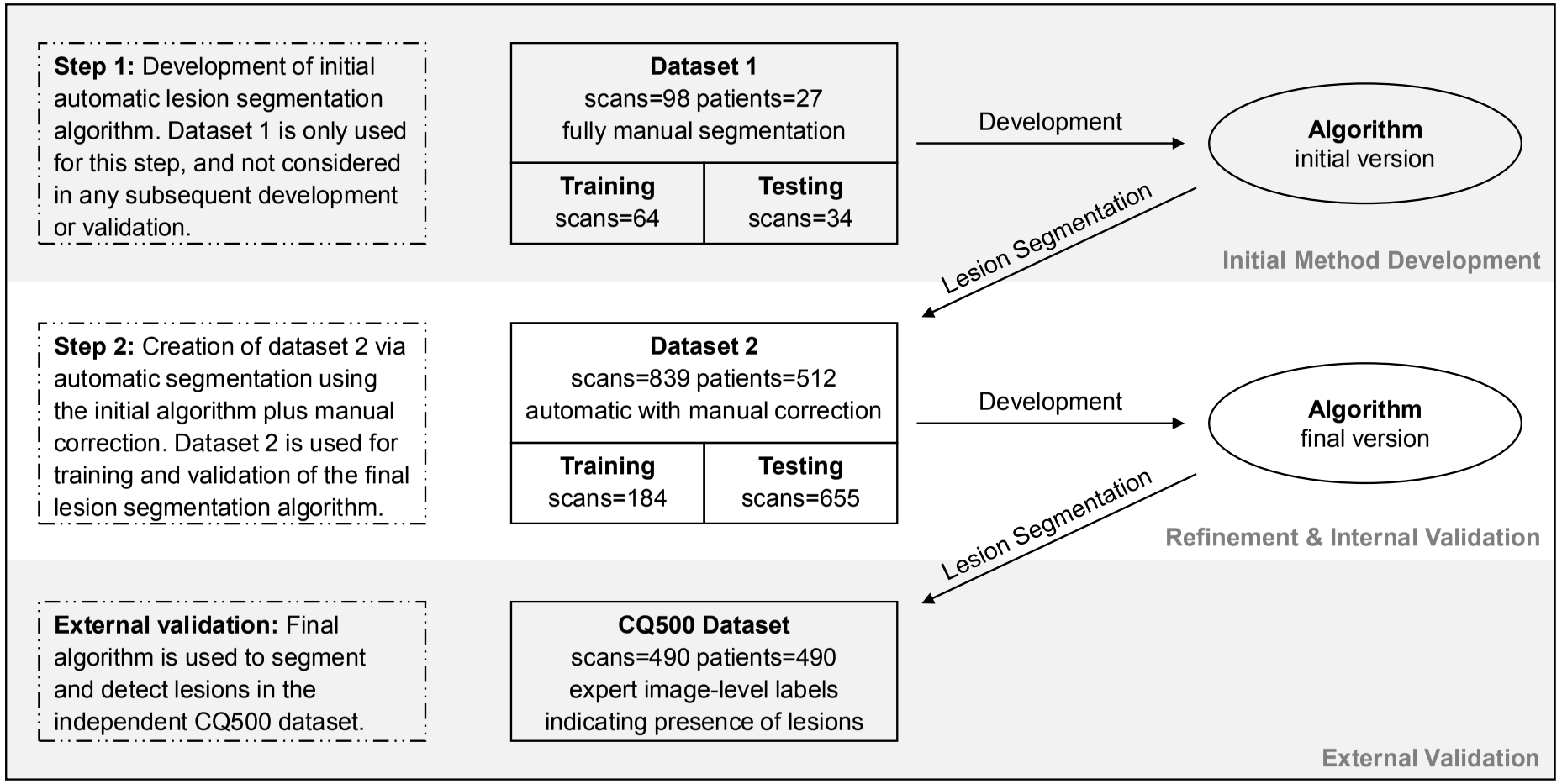


Figure A.1: Data flow diagram for model development and validation.

Table A.1: Distribution of lesion presence and volume for the train and test set. n denotes the fraction of scans with lesions over the total number of scans in the set; lesion volume is presented as mean (standard deviation).

set	**lesion (any)**		**IPH**		**EAH**		**oedema**		**IVH**	
	n	volume (mL)	n	volume (mL)	n	volume (mL)	n	volume (mL)	n	volume (mL)
train	184/184 (100.0%)	34.49 (40.08)	137/184 (74.5%)	5.51 (11.67)	178/184 (96.7%)	17.79 (29.79)	145/184 (78.8%)	10.64 (16.99)	47/184 (25.5%)	0.55 (4.30)
test (>=1mL)	328/328 (100.0%)	31.72 (38.77)	246/328 (75.0%)	5.57 (10.53)	309/328 (94.2%)	16.19 (27.26)	254/328 (77.4%)	9.70 (16.69)	74/328 (22.6%)	0.25 (1.05)
test (<1mL)	141/327 (43.1%)	0.17 (0.27)	60/327 (18.3%)	0.04 (0.12)	93/327 (28.4%)	0.09 (0.19)	33/327 (10.1%)	0.03 (0.10)	22/327 (6.7%)	0.01 (0.07)

Table A.2: Intra-class correlation coefficient (ICC) for intra-rater reproducibility and inter-rater variability. ICCs are presented as mean (95% CI).

	n	lesion (any)	IPH	EAH	oedema	IVH
intra-rater	20	0.997 (0.992-0.999)	0.999 (0.998-1.000)	0.998 (0.995-0.999)	0.958 (0.894-0.983)	0.989 (0.971-0.996)
inter-rater	25	0.915 (0.813-0.962)	0.978 (0.949-0.990)	0.923 (0.831-0.966)	0.895 (0.773-0.953)	0.908 (0.799-0.959)

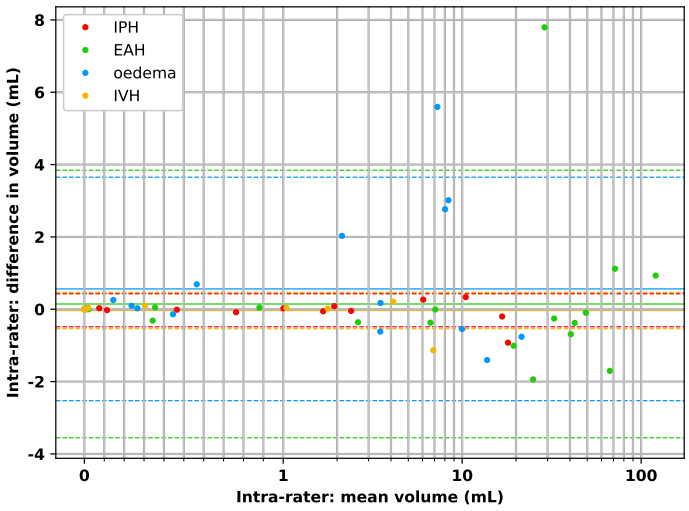


Figure A.2: Per-class Bland-Altman plots for intra-rater reproducibility. Means are represented as solid lines and 95% CIs as dashed lines. The mean difference is -0.03mL (95% CI - 0.48 to 0.43) for IPH, 0.14mL (95% CI -3.55 to 3.84) for EAH, 0.56mL (95% CI -2.53 to 3.65) for oedema and -0.04mL (95% CI -0.54 to 0.47) for IVH.

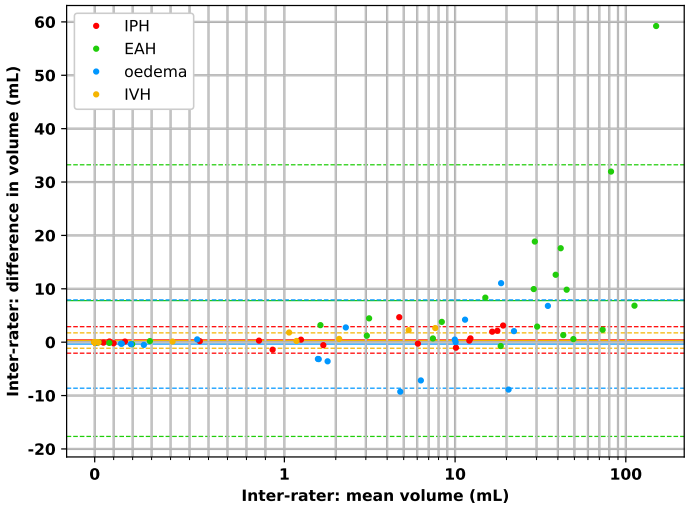


Figure A.3: Per-class Bland-Altman plot inter-rater variability. Means are represented as solid lines and 95% CIs as dashed lines. The mean difference is 0.42mL (95% CI - 2.06 to 2.90) for IPH, 7.79mL (95% CI -17.66 to 33.23) for EAH, -0.35mL (95% CI -8.62 to 7.92) for oedema and 0.30mL (95% CI -1.14 to 1.74) for IVH.

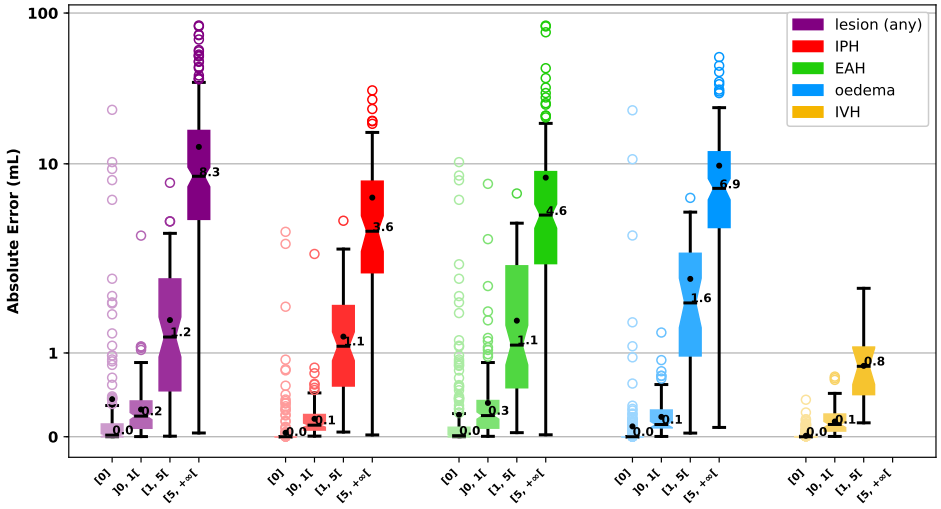


Figure A.4: Per-class boxplots of the absolute error for different volume groups. In increasing shade: [0];]0, 1[; [1, 5[; [5, $+\infty$[mL. The y-axis in on logarithmic scale for visualization. The corresponding table is available in Table A.3.

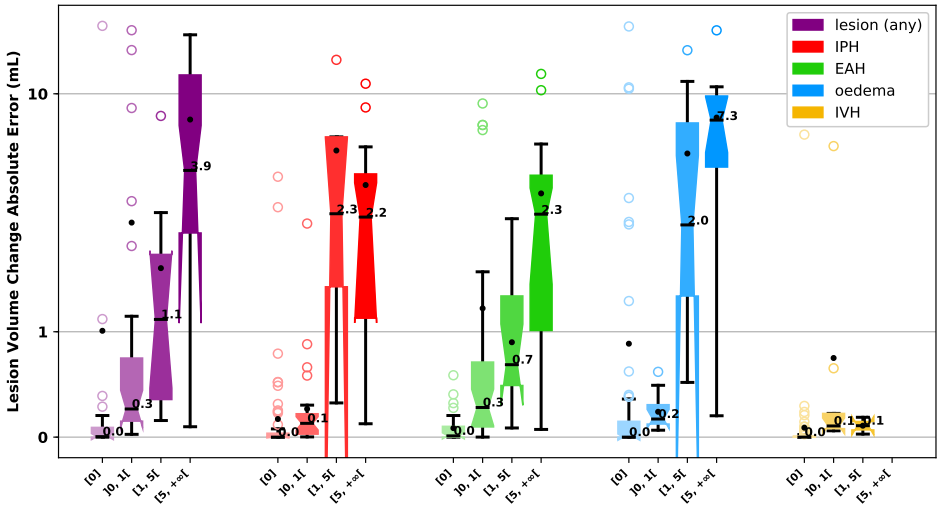


Figure A.5: Per-class boxplots of the absolute error between the true lesion volume change and the predicted lesion volume change for different volume groups. In increasing shade: [0];]0, 1[; [1, 5[; [5, $+\infty$[mL. The y-axis in on logarithmic scale for visualization. The corresponding table is available in Table A.3.

Table A.3: Evaluation metrics stratified by volume group. Numbers presented as number of scans (n), mean (standard deviation), median (range). For lesion change the volume group is defined by the initial lesion volume.

class	group	DSC (%)			Absolute Error (mL)			Lesion Change Abs Error (mL)		
		count	mean	median	count	mean	median	count	mean	median
lesion (any)	[0]	130	0.0 (0.0)	0.0 (0.0)	186	0.45 (2.06)	0.02 (0.15)	25	1.01 (4.46)	0.00 (0.09)
	]0, 1[	141	18.5 (20.5)	12.0 (32.5)	141	0.33 (0.36)	0.25 (0.33)	28	2.09 (5.07)	0.27 (0.60)
	[1, 5[	77	37.8 (20.8)	39.5 (25.3)	77	1.39 (1.18)	1.19 (1.34)	11	1.60 (2.04)	1.12 (1.38)
	[5, 10[	251	62.2 (17.7)	66.9 (22.8)	251	12.95 (14.21)	8.27 (12.36)	34	7.31 (6.69)	3.94 (10.63)
IPH	[0]	55	0.0 (0.0)	0.0 (0.0)	349	0.05 (0.28)	0.00 (0.00)	58	0.17 (0.58)	0.00 (0.04)
	]0, 1[	139	23.7 (28.9)	2.9 (47.4)	139	0.21 (0.26)	0.14 (0.19)	23	0.27 (0.44)	0.13 (0.19)
	[1, 5[	77	54.0 (22.1)	58.2 (26.8)	77	1.20 (0.77)	1.08 (0.96)	4	5.02 (5.89)	2.33 (4.50)
	[5, 10[	90	68.2 (17.2)	72.6 (23.5)	90	5.97 (6.14)	3.57 (5.71)	13	3.30 (3.17)	2.24 (2.65)
EAH	[0]	157	0.0 (0.0)	0.0 (0.0)	253	0.26 (1.05)	0.01 (0.11)	31	0.09 (0.14)	0.01 (0.10)
	]0, 1[	140	13.6 (17.5)	5.5 (22.7)	140	0.40 (0.71)	0.25 (0.33)	31	1.22 (2.37)	0.28 (0.62)
	[1, 5[	102	39.1 (18.0)	40.7 (21.0)	102	1.39 (1.07)	1.09 (1.52)	12	0.90 (0.59)	0.69 (0.85)
	[5, 10[	160	63.5 (18.6)	67.5 (25.7)	160	8.10 (12.15)	4.57 (6.70)	24	2.98 (3.00)	2.32 (2.71)
oedema	[0]	68	0.0 (0.0)	0.0 (0.0)	368	0.12 (1.32)	0.00 (0.00)	63	0.89 (3.36)	0.00 (0.15)
	]0, 1[	79	14.9 (20.5)	0.0 (27.6)	79	0.24 (0.23)	0.15 (0.21)	10	0.24 (0.18)	0.17 (0.18)
	[1, 5[	71	22.8 (24.7)	14.8 (40.8)	71	1.88 (1.25)	1.60 (1.58)	11	4.84 (5.17)	2.04 (5.68)
	[5, 10[	137	51.3 (24.1)	54.6 (36.1)	137	9.72 (9.09)	6.87 (8.23)	14	7.49 (4.99)	7.26 (5.63)
IVH	[0]	141	0.0 (0.0)	0.0 (0.0)	559	0.01 (0.04)	0.00 (0.00)	87	0.09 (0.65)	0.00 (0.01)
	]0, 1[	75	26.9 (25.4)	24.5 (51.2)	75	0.18 (0.16)	0.15 (0.20)	9	0.75 (1.62)	0.11 (0.14)
	[1, 5[	17	51.9 (14.0)	50.5 (26.1)	17	0.85 (0.43)	0.84 (0.57)	2	0.11 (0.08)	0.11 (0.08)
	[5, 10[	4	28.2 (18.6)	32.6 (26.8)	4	6.50 (1.14)	5.93 (0.94)	0	0.00 (0.00)	0.00 (0.00)

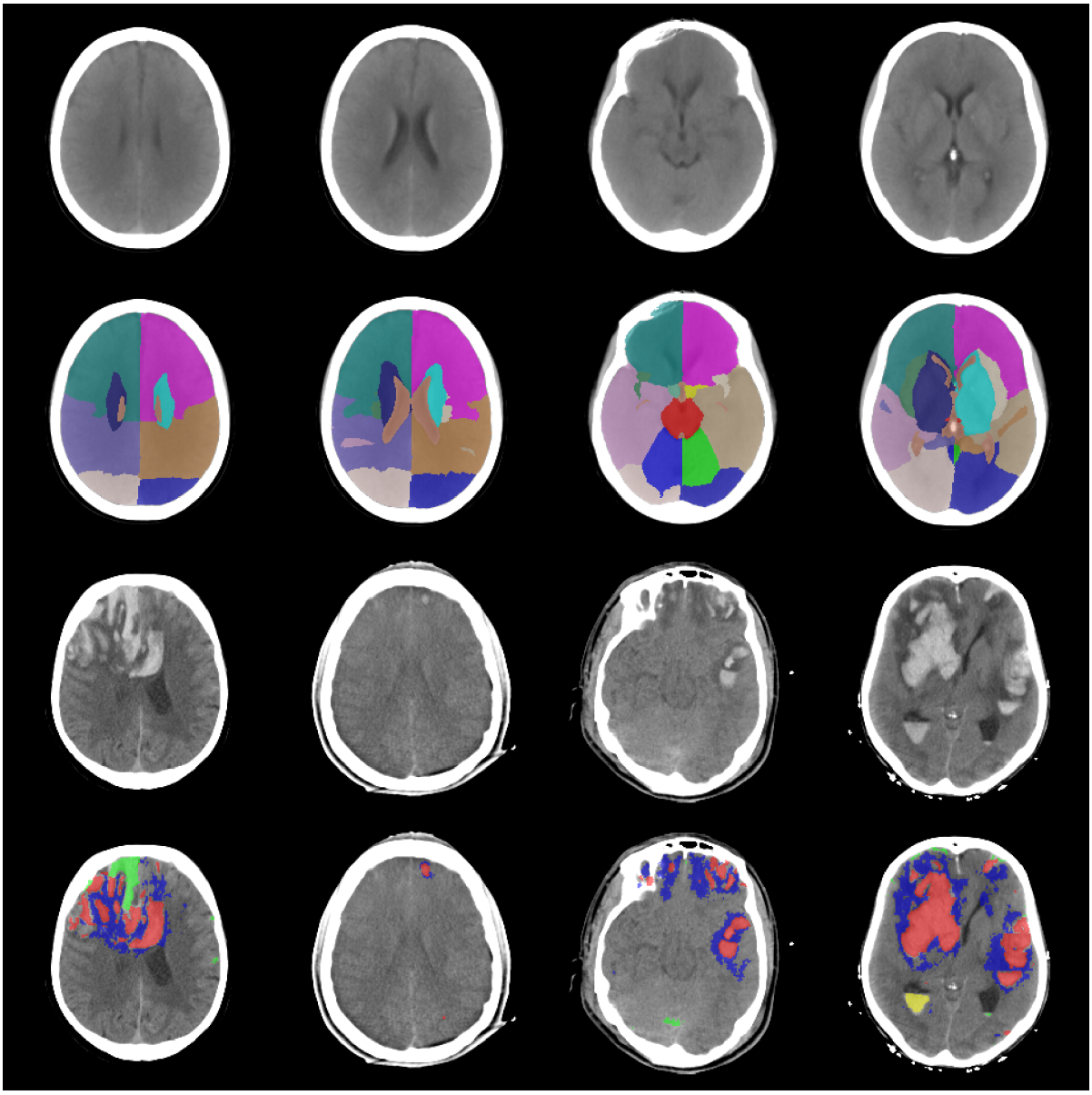


Figure A.6: Lesion localisation example. From top to bottom: atlas, labelled atlas, image registered to atlas, image overlaid with lesion segmentation. Lesion location is obtained by intersecting the lesion segmentation with the labelled atlas. For the lesion segmentation, IPH is segmented in red, EAH in green, oedema in blue and IVH in yellow.

Appendix B

Supplementary material for Stochastic Segmentation Networks

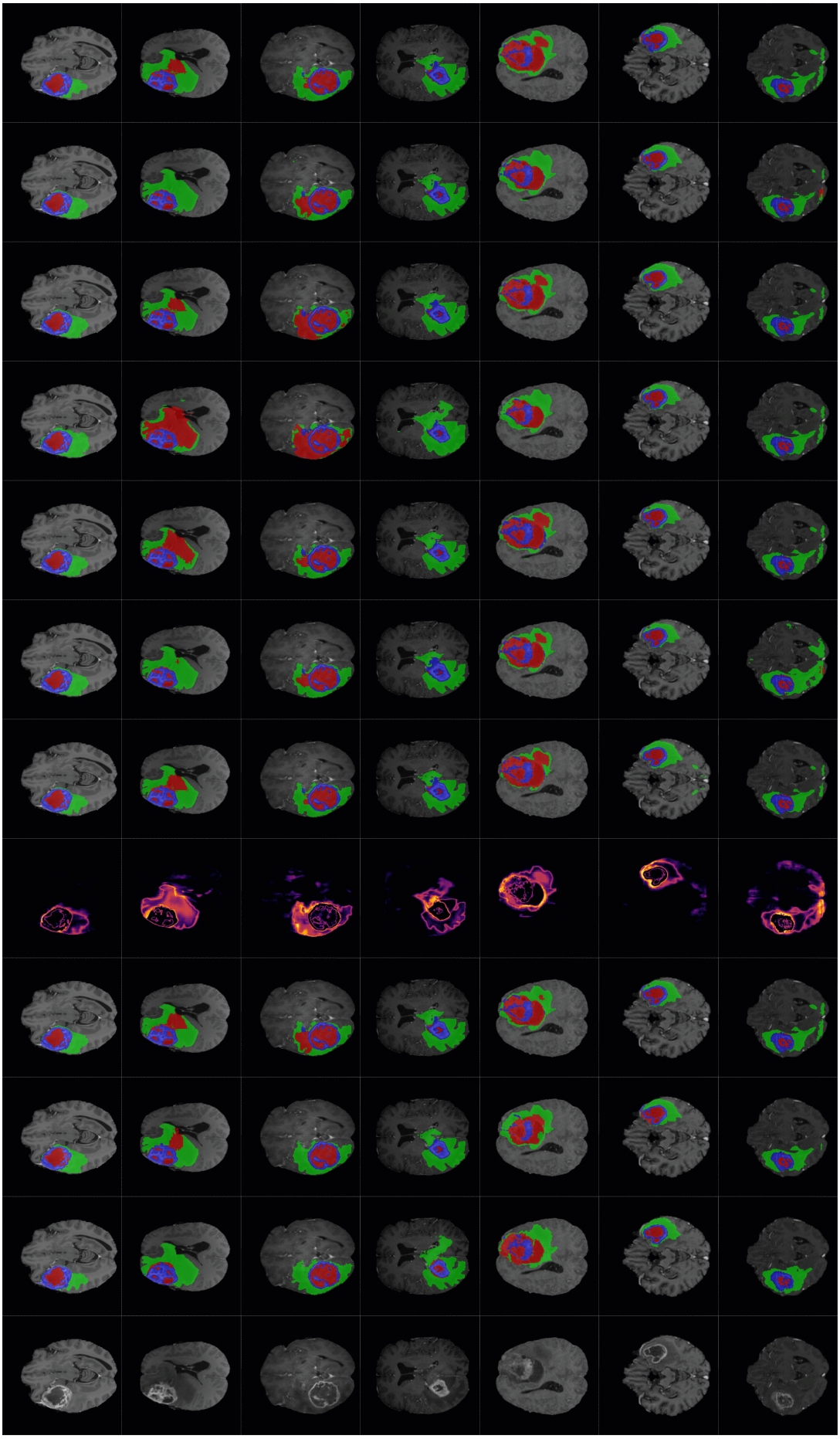


Figure B.1: Results of sampling from the proposed stochastic model. From left to right: T1ce slice; ground-truth; prediction of deterministic model; prediction of stochastic model; marginal entropy; seven random samples (1/4).

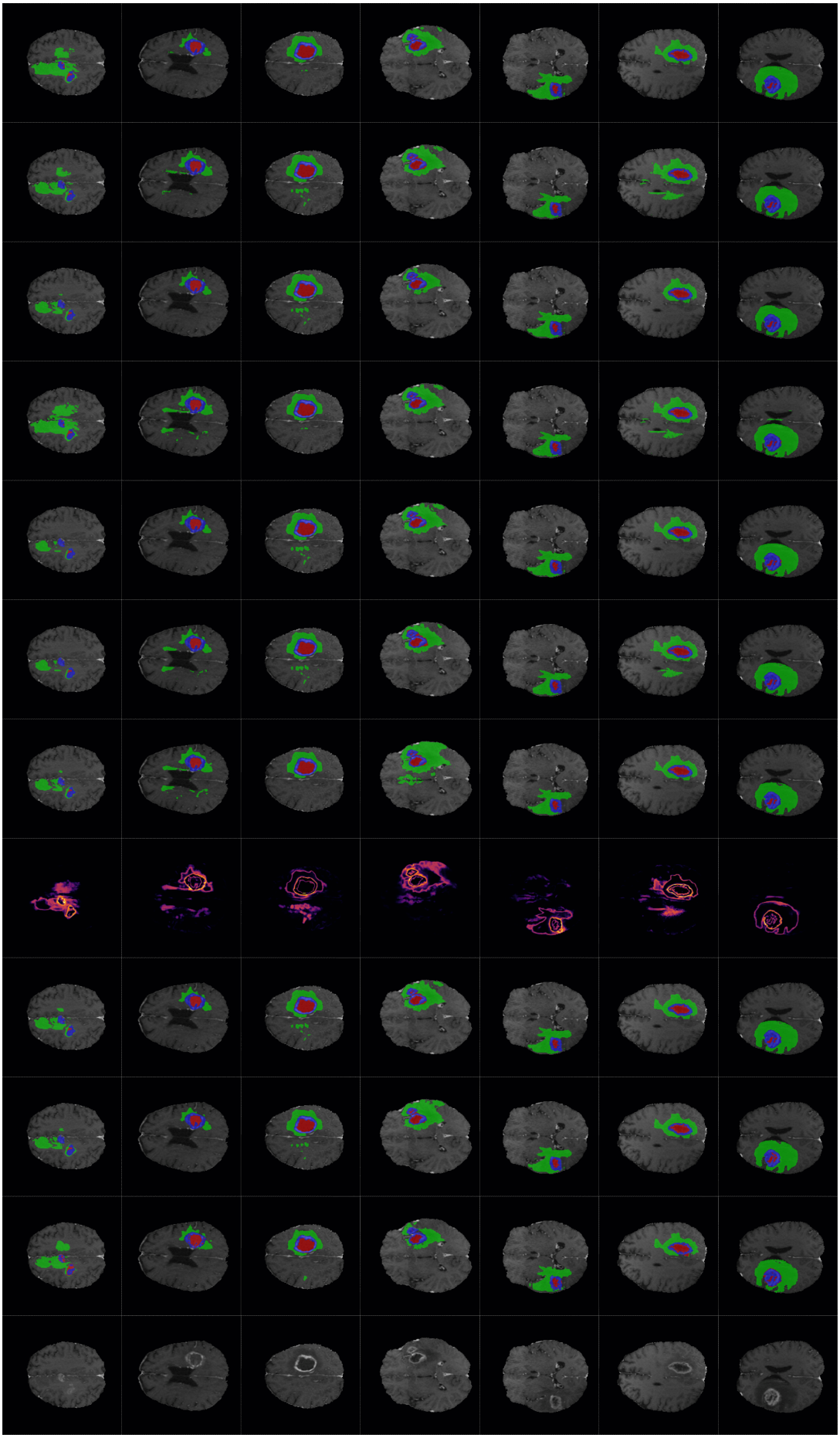


Figure B.2: Results of sampling from the proposed stochastic model. From left to right: T1ce slice; ground-truth; prediction of deterministic model; prediction of stochastic model; marginal entropy; seven random samples (2/4).

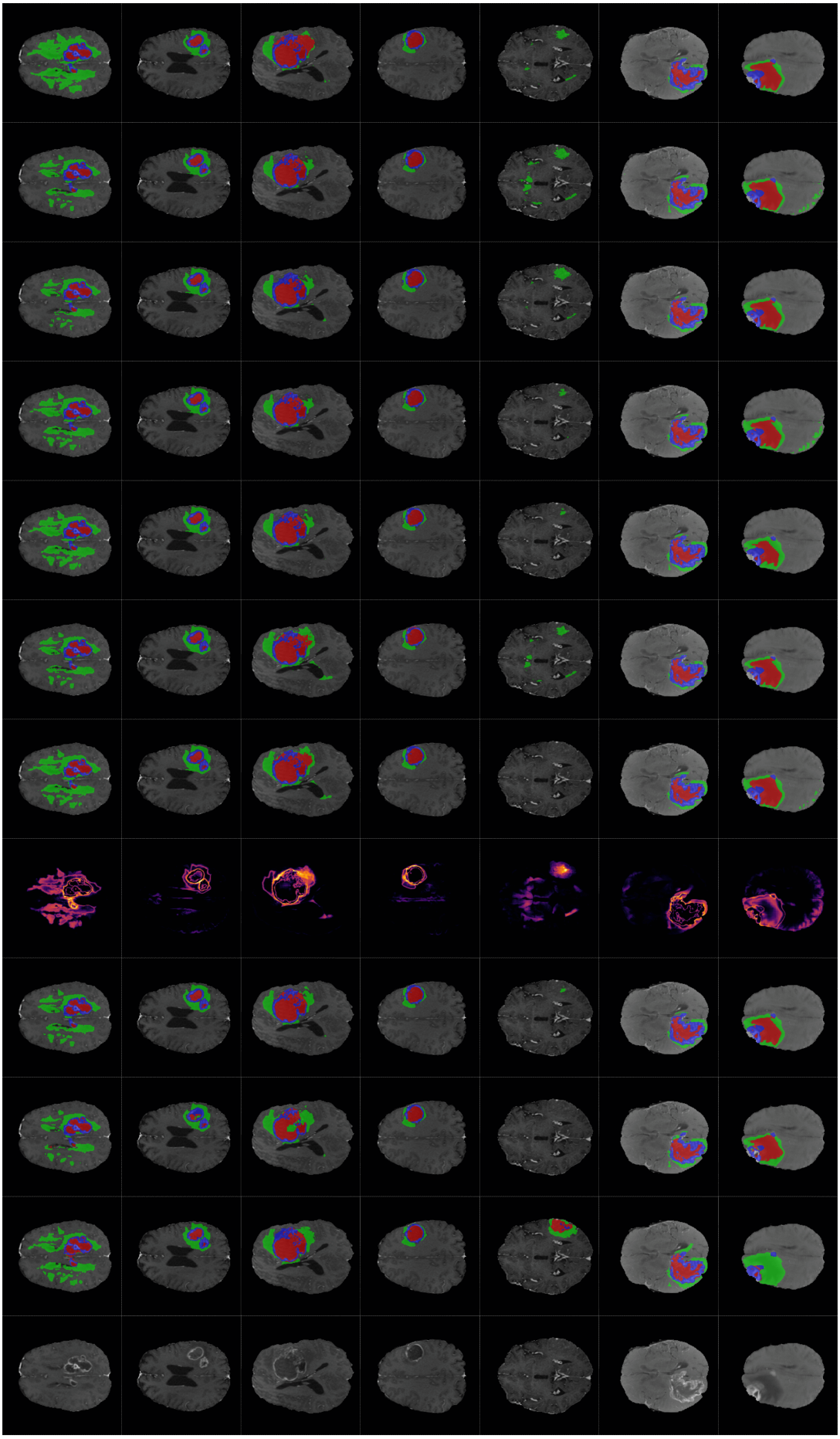


Figure B.3: Results of sampling from the proposed stochastic model. From left to right: T1ce slice; ground-truth; prediction of deterministic model; prediction of stochastic model; marginal entropy; seven random samples (3/4).

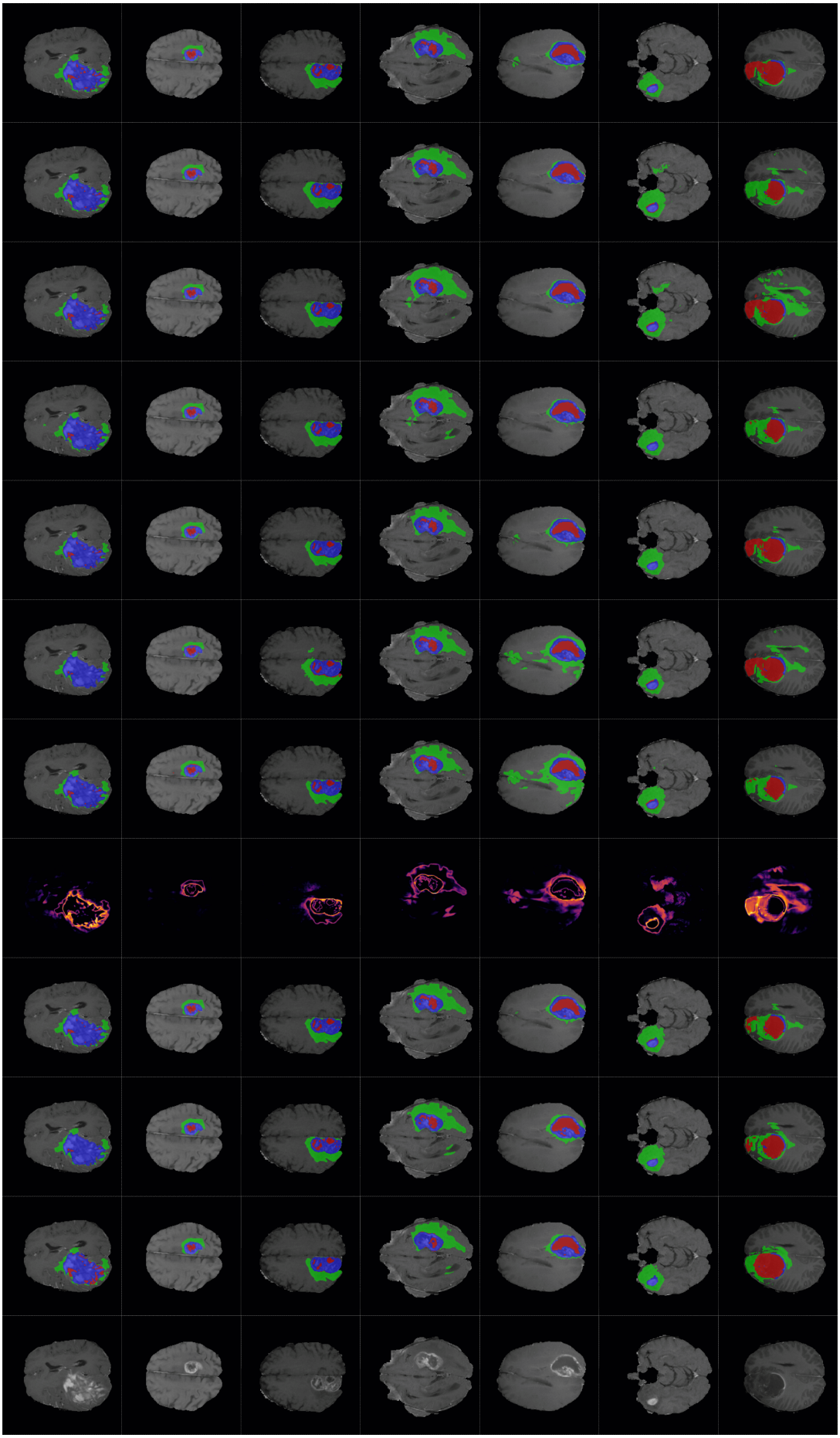


Figure B.4: Results of sampling from the proposed stochastic model. From left to right: T1ce slice; ground-truth; prediction of deterministic model; prediction of stochastic model; marginal entropy; seven random samples (4/4).

Appendix C

Supplementary material for axiomatic quality of counterfactuals

C.1 Reversibility proof

Lemma C.1.1. *Let $x = \mathrm{g}(\epsilon, \mathbf{pa})$ be a mechanism where x is the observation with parents $\mathbf{pa}$ and exogenous noise variable ϵ. Further, let x^* be a counterfactual of x with parents $\mathbf{pa}^*$. If the mechanism is invertible, then the exogenous noise is deterministically given by: $\epsilon = \mathrm{g}^{-1}(x, \mathbf{pa})$ or $\epsilon = \mathrm{g}^{-1}(x^*, \mathbf{pa}^*)$. Under these conditions, in the counterfactual function form we have that if $x^* \coloneqq \mathrm{f}(x, \mathbf{pa}, \mathbf{pa}^*)$, then $x = \mathrm{f}(x^*, \mathbf{pa}^*, \mathbf{pa})$.*

Proof.

$$\begin{aligned} x^* &= \mathrm{g}(\epsilon, \mathbf{pa}^*) \\ &= \mathrm{g}\big(\mathrm{g}^{-1}(x, \mathbf{pa}), \mathbf{pa}^*\big) \\ &= \mathrm{f}(x, \mathbf{pa}, \mathbf{pa}^*) \end{aligned} \tag{C.1}$$

And:

$$\begin{aligned} x &= \mathrm{g}(\epsilon, \mathbf{pa}) \\ &= \mathrm{g}\big(\mathrm{g}^{-1}(x^*, \mathbf{pa}^*), \mathbf{pa}\big) \\ &= \mathrm{f}(x, \mathbf{pa}^*, \mathbf{pa}) \end{aligned} \tag{C.2}$$

□

C.2 Distance metrics

Given two points $a \in \mathbb{R}^N$ and $b \in \mathbb{R}^N$ and a distance $\mathrm{d_X}(\cdot, \cdot) : (\mathbb{R}^N, \mathbb{R}^N) \to \mathbb{R}^+$, for the distance to be considered a metric, it must obey the following properties (see Phillips 2021, Sec. 6.1):

- Non-negativity: $\mathrm{d_X}(a, b) \geq 0$;
- Identity: $\mathrm{d_X}(a, b) = 0$ if $a = b$;
- Symmetry: $\mathrm{d_X}(a, b) = \mathrm{d_X}(b, a)$;
- Triangular inequality: $\mathrm{d_X}(a, b) \leq \mathrm{d_X}(a, c) + \mathrm{d_X}(c, b)$.

In this paper, we opted to use the l_1 distance, which is a metric, because we can directly interpret

its value as the average pixel intensity by which two images differ. While there is an argument to be made for the use of perceptual distances for images, they are not metrics and have the same data-driven weaknesses, which pose a problem for our pseudo-oracles. As a result, we opted for a data-independent distance/metric.

Listing C.1: VAE architecture for the colour MNIST dataset.

```
        image_encoder = serial(
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Flatten,
            Dense(out_dim=128),
            LeakyRelu
        )

        image_decoder = serial(
            Dense(7 * 7 * 64),
            LeakyRelu,
            Reshape((-1, 7, 7, 64)),
            Resize((-1, 14, 14, 64), method='linear'),
            Conv(out_chan=64, filter_shape=(4, 4), strides=(1, 1), padding='SAME'),
            LeakyRelu,
            Resize((-1, 28, 28, 64), method='linear'),
            Conv(out_chan=64, filter_shape=(4, 4), strides=(1, 1), padding='SAME'),
            LeakyRelu,
            Conv(out_chan=3, filter_shape=(3, 3), strides=(1, 1), padding='SAME')
        )
        # encoder
        encoded_image = image_encoder(image)
        tmp = Dense(out_dim=128)(concat(encoded_image, parents))
        mu = Dense(out_dim=16)(tmp)
        sigma = softplus(Dense(out_dim=16)(tmp))
        # decoder
        z = sample_from_standard_normal(mu, sigma)
        counterfactual = image_decoder(concat(z, counterfactual_parents))
```

Listing C.2: GAN architecture for the colour MNIST dataset.

```
        image_encoder = serial(
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Flatten,
            Dense(out_dim=128),
            LeakyRelu
        )

        image_decoder = serial(
            Dense(7 * 7 * 64),
            LeakyRelu,
            Reshape((-1, 7, 7, 64)),
            Resize((-1, 14, 14, 64), method='linear'),
            Conv(out_chan=64, filter_shape=(4, 4), strides=(1, 1), padding='SAME'),
            LeakyRelu,
            Resize((-1, 28, 28, 64), method='linear'),
            Conv(out_chan=64, filter_shape=(4, 4), strides=(1, 1), padding='SAME'),
            LeakyRelu,
            Conv(out_chan=3, filter_shape=(3, 3), strides=(1, 1), padding='SAME')
        )
        #generator
        gan_decoder = serial(
                        Dense(out_dim=128),
                        LeakyRelu,
                        image_decoder,
                        Tanh
                    )
        encoded_image = image_encoder(image)
        tmp = concat(encoded_image, parents, counterfactual_parents)
        counterfactual = gan_decoder(tmp)

        # critic
        critic = serial(
                Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2), padding='SAME'),
                LeakyRelu,
                Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2), padding='SAME'),
                LeakyRelu,
                Flatten,
                Dense(out_dim=128),
                LeakyRelu,
                Dense(out_dim=128),
                LeakyRelu
                Dense(1)
            )
        critic_input = concat(image, broadcast_to_shape(parents, image.shape))
        logits = critic(critic_input)
```

Listing C.3: Pseudo-oracle architecture for the colour MNIST and 3D shapes datasets.

```
        pseudo_oracle = serial(
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Flatten,
            Dense(out_dim=128),
            LeakyRelu,
            Dense(out_dim=num_classes if classification else 1)
        )
        y_hat = pseudo_oracle(image)
```

Listing C.4: VAE architecture for the 3D shapes dataset.

```
        image_encoder = serial(
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Flatten,
            Dense(out_dim=128),
            LeakyRelu
        )

        image_decoder = serial(
            Dense(8 * 8 * 64),
            LeakyRelu,
            Reshape((-1, 8, 8, 64)),
            Resize((-1, 16, 16, 64), method='linear'),
            Conv(out_chan=64, filter_shape=(4, 4), strides=(1, 1), padding='SAME'),
            LeakyRelu,
            Resize((-1, 32, 32, 64), method='linear'),
            Conv(out_chan=64, filter_shape=(4, 4), strides=(1, 1), padding='SAME'),
            LeakyRelu,
            Resize((-1, 64, 64, 64), method='linear'),
            Conv(out_chan=64, filter_shape=(4, 4), strides=(1, 1), padding='SAME'),
            LeakyRelu,
            Conv(out_chan=3, filter_shape=(3, 3), strides=(1, 1), padding='SAME')
        )
        # encoder
        encoded_image = image_encoder(image)
        tmp = Dense(out_dim=128)(concat(encoded_image, parents))
        mu = Dense(out_dim=16)(tmp)
        sigma = softplus(Dense(out_dim=16)(tmp))
        # decoder
        z = sample_from_standard_normal(mu, sigma)
        counterfactual = image_decoder(concat(z, counterfactual_parents))
```

Listing C.5: GAN architecture for the 3D shapes dataset.

```
        image_encoder = serial(
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2)),
            LeakyRelu,
            Flatten,
            Dense(out_dim=128),
            LeakyRelu
        )

        image_decoder = serial(
            Dense(8 * 8 * 64),
            LeakyRelu,
            Reshape((-1, 8, 8, 64)),
            Resize((-1, 16, 16, 64), method='linear'),
            Conv(out_chan=64, filter_shape=(4, 4), strides=(1, 1), padding='SAME'),
            LeakyRelu,
            Resize((-1, 32, 32, 64), method='linear'),
            Conv(out_chan=64, filter_shape=(4, 4), strides=(1, 1), padding='SAME'),
            LeakyRelu,
            Resize((-1, 64, 64, 64), method='linear'),
            Conv(out_chan=64, filter_shape=(4, 4), strides=(1, 1), padding='SAME'),
            LeakyRelu,
            Conv(out_chan=3, filter_shape=(3, 3), strides=(1, 1), padding='SAME')
        )

        #generator
        gan_decoder = serial(
                        Dense(out_dim=128),
                        LeakyRelu,
                        image_decoder,
                        Tanh
                    )
        encoded_image = image_encoder(image)
        tmp = concat(encoded_image, parents, counterfactual_parents)
        counterfactual = gan_decoder(tmp)

        # critic
        critic = serial(
                Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2), padding='SAME'),
                LeakyRelu,
                Conv(out_chan=64, filter_shape=(4, 4), strides=(2, 2), padding='SAME'),
                LeakyRelu,
                Flatten,
                Dense(out_dim=128),
                LeakyRelu,
                Dense(out_dim=128),
                LeakyRelu
                Dense(1)
            )
        critic_input = concat(image, broadcast_to_shape(parents, image.shape))
        logits = critic(critic_input)
```

Listing C.6: Pseudo-oracle architecture for the CelebA-HQ dataset.

```
        class ConvNet(nn.Module):
            def __init__(self):
                super().__init__()
                self.cnn = nn.Sequential(
                    nn.Conv2d(3, 16, 3, 1, 1),
                    nn.BatchNorm2d(16),
                    nn.ReLU(),
                    nn.Conv2d(16, 32, 3, 2, 1),
                    nn.BatchNorm2d(32),
                    nn.ReLU(),
                    nn.Conv2d(32, 32, 3, 1, 1),
                    nn.BatchNorm2d(32),
                    nn.ReLU(),
                    nn.Conv2d(32, 64, 3, 2, 1),
                    nn.BatchNorm2d(64),
                    nn.ReLU(),
                    nn.Conv2d(64, 64, 3, 1, 1),
                    nn.BatchNorm2d(64),
                    nn.ReLU(),
                    nn.Conv2d(64, 128, 3, 2, 1),
                    nn.BatchNorm2d(128),
                    nn.ReLU(),
                    nn.AdaptiveAvgPool2d(1),
                )
                self.fc = nn.Sequential(
                    nn.Linear(128, 128),
                    nn.BatchNorm1d(128),
                    nn.ReLU(),
                    nn.Linear(128, 1)
                )

            def forward(self, x):
                return self.fc(self.cnn(x).squeeze())
```

Table C.1: Effectiveness results on colour MNIST when using pseudo oracles trained from biased data without using a simulated intervention. We measure effectiveness using digit accuracy and hue absolute error in percentage points since $hue \in [0, 1]$. The average model ranking change compared to Table 5.1 (oracles trained from unbiased data) was 0.8(3) for the dataset without full support and 0.75 for the dataset with full support.

dataset	inter-ven-tion	model	digit intervention effectiveness $\mathrm{acc}_{\mathrm{digit}}(\%)\uparrow$	digit intervention effectiveness $\mathrm{ae}_{\mathrm{hue}}(\%)\downarrow$	hue intervention effectiveness $\mathrm{acc}_{\mathrm{digit}}(\%)\uparrow$	hue intervention effectiveness $\mathrm{ae}_{\mathrm{hue}}(\%)\downarrow$	avg. rank	original rank	abs. diff.
con-found-ed w/o full support	no	Bernoulli VAE β=1	53.77 (9.88)	11.43 (4.56)	53.68 (6.08)	11.80 (5.85)	3.5	3.5	0
		Bernoulli VAE β=2	48.34 (2.27)	10.42 (1.64)	49.32 (2.01)	10.15 (1.62)	4	2.5	1.5
		Normal VAE β=5	57.65 (8.41)	13.92 (3.38)	60.69 (6.96)	13.63 (3.51)	3.5	5.5	2
	yes	Bernoulli VAE β=1	49.83 (2.52)	9.15 (0.87)	50.92 (2.07)	9.51 (1.35)	3	2.25	0.75
		Bernoulli VAE β=2	46.48 (1.22)	8.51 (0.87)	47.30 (1.34)	8.53 (0.87)	3.5	3.25	0.25
		Normal VAE β=5	43.86 (1.28)	8.28 (1.38)	47.19 (1.10)	7.97 (1.34)	3.5	4	0.5
con-found-ed w/ full support	no	Bernoulli VAE β=1	87.71 (0.92)	10.90 (6.53)	90.15 (0.48)	14.02 (10.59)	7	6.5	0.5
		Bernoulli VAE β=2	90.05 (0.44)	5.94 (0.20)	90.64 (0.52)	5.92 (0.23)	5.25	3.25	2
		Normal VAE β=5	82.98 (0.91)	6.19 (0.73)	89.23 (0.19)	5.77 (0.65)	6.5	7	0.5
	yes*	GAN	87.03 (0.62)	4.71 (0.16)	92.51 (0.36)	5.28 (0.34)	3	4.25	1.25
	yes	Bernoulli VAE β=1	92.07 (0.25)	5.64 (0.19)	92.19 (0.35)	6.15 (0.59)	3.75	4	0.25
		Bernoulli VAE β=2	93.98 (0.26)	5.72 (0.25)	93.76 (0.15)	5.82 (0.23)	2.5	2	0.5
		Normal VAE β=5	88.68 (0.33)	5.92 (0.46)	91.85 (0.12)	5.87 (0.47)	4.5	4.75	0.25
		GAN	87.40 (0.14)	4.78 (0.18)	90.90 (0.21)	4.90 (0.16)	3.5	4.25	0.75

Table C.2: Effectiveness results on colour MNIST when using pseudo oracles trained from biased data using a simulated intervention. We measure effectiveness using digit accuracy and hue absolute error in percentage points since $hue \in [0, 1]$. The average model ranking change compared to Table 5.1 (oracles trained from unbiased data) was 0.8(3) for the dataset without full support and 0.625 for the dataset with full support.

dataset	inter-ven-tion	model	digit intervention effectiveness $\mathrm{acc}_{\mathrm{digit}}(\%)\uparrow$	digit intervention effectiveness $\mathrm{ae}_{\mathrm{hue}}(\%)\downarrow$	hue intervention effectiveness $\mathrm{acc}_{\mathrm{digit}}(\%)\uparrow$	hue intervention effectiveness $\mathrm{ae}_{\mathrm{hue}}(\%)\downarrow$	avg. rank	original rank	abs. diff.
con-found-ed w/o full support	no	Bernoulli VAE β=1	57.09 (10.01)	11.17 (4.88)	57.14 (6.63)	11.42 (6.06)	3.5	3.5	0
		Bernoulli VAE β=2	53.43 (2.44)	10.08 (1.57)	54.03 (2.57)	9.73 (1.55)	4	2.5	1.5
		Normal VAE β=5	62.93 (8.26)	13.81 (3.27)	67.23 (7.21)	13.55 (3.38)	3.5	5.5	2
	yes	Bernoulli VAE β=1	53.72 (2.61)	8.57 (1.00)	55.36 (2.37)	8.88 (1.45)	3	2.25	0.75
		Bernoulli VAE β=2	50.96 (0.76)	7.93 (0.96)	51.79 (0.72)	7.89 (0.90)	3.5	3.25	0.25
		Normal VAE β=5	47.89 (1.75)	7.83 (1.58)	51.23 (2.01)	7.42 (1.59)	3.5	4	0.5
con-found-ed w/ full support	no	Bernoulli VAE β=1	93.25 (1.48)	8.93 (7.08)	96.57 (0.42)	12.36 (11.61)	7.5	6.5	1
		Bernoulli VAE β=2	96.90 (0.40)	3.17 (0.11)	96.94 (0.20)	2.98 (0.14)	4.25	3.25	1
		Normal VAE β=5	90.37 (0.45)	4.16 (1.07)	95.92 (0.25)	3.85 (1.11)	7.5	7	0.5
	yes*	GAN	93.30 (0.69)	2.36 (0.19)	97.47 (0.23)	2.52 (0.09)	3.25	4.25	1
	yes	Bernoulli VAE β=1	97.36 (0.18)	2.80 (0.14)	97.70 (0.25)	3.67 (0.83)	3.5	4	0.5
		Bernoulli VAE β=2	98.74 (0.11)	2.68 (0.17)	98.84 (0.16)	2.61 (0.16)	2	2	0
		Normal VAE β=5	94.68 (0.39)	3.26 (0.24)	97.50 (0.21)	3.20 (0.25)	4.5	4.75	0.25
		GAN	93.90 (0.18)	2.38 (0.16)	96.58 (0.15)	2.44 (0.17)	3.5	4.25	0.75

Table C.3: Effectiveness on colour MNIST when using linear/logistic regression as pseudo oracles trained from unbiased data. We measure effectiveness using digit accuracy and hue absolute error in percentage points since $hue \in [0, 1]$. The average model ranking change compared to Table 5.1 (oracles trained from unbiased data) was 0.91(6) for the dataset without full support and 0.6875 for the dataset with full support.

dataset	inter-ven-tion	model	digit intervention effectiveness $\mathrm{acc}_{\mathrm{digit}}(\%)\uparrow$	digit intervention effectiveness $\mathrm{ae}_{\mathrm{hue}}(\%)\downarrow$	hue intervention effectiveness $\mathrm{acc}_{\mathrm{digit}}(\%)\uparrow$	hue intervention effectiveness $\mathrm{ae}_{\mathrm{hue}}(\%)\downarrow$	avg. rank	original rank	abs. diff.
con-found-ed w/o full support	no	Bernoulli VAE β=1	80.83 (1.91)	17.77 (4.12)	81.79 (1.42)	17.28 (4.14)	4.75	3.5	1.25
		Bernoulli VAE β=2	86.18 (0.63)	16.17 (1.32)	86.03 (0.52)	15.91 (1.27)	2.5	2.5	0
		Normal VAE β=5	78.67 (4.32)	18.97 (1.56)	82.43 (3.80)	18.37 (1.22)	5.25	5.5	0.25
	yes	Bernoulli VAE β=1	82.23 (0.81)	15.22 (0.63)	82.31 (1.01)	14.24 (0.74)	2.75	2.25	0.5
		Bernoulli VAE β=2	80.52 (2.11)	15.93 (1.34)	79.37 (2.49)	15.74 (1.41)	4.25	3.25	1
		Normal VAE β=5	82.50 (1.61)	14.17 (1.01)	85.20 (1.13)	13.82 (1.08)	1.5	4	2.5
con-found-ed w/ full support	no	Bernoulli VAE β=1	82.26 (2.97)	18.05 (4.15)	84.23 (1.16)	18.73 (5.29)	7.25	6.5	0.75
		Bernoulli VAE β=2	87.55 (0.47)	14.75 (0.19)	87.00 (0.28)	14.80 (0.20)	3.5	3.25	0.25
		Normal VAE β=5	81.78 (0.93)	15.14 (0.39)	84.90 (0.59)	14.36 (0.44)	5.25	7	1.75
	yes*	GAN	80.01 (1.28)	14.71 (0.08)	87.09 (0.84)	14.87 (0.17)	5	4.25	0.75
	yes	Bernoulli VAE β=1	87.17 (0.64)	14.88 (0.11)	86.09 (1.73)	14.37 (0.64)	3.75	4	0.25
		Bernoulli VAE β=2	89.87 (0.51)	14.70 (0.20)	89.32 (0.34)	14.56 (0.16)	1.5	2	0.5
		Normal VAE β=5	85.43 (0.59)	15.00 (0.17)	87.23 (0.53)	14.84 (0.25)	4.5	4.75	0.25
		GAN	80.39 (0.56)	14.77 (0.15)	85.51 (0.51)	14.65 (0.12)	5.25	4.25	1

Table C.4: Results of quality tests on 3D shapes: we measure composition after the null intervention and reversibility after one intervention cycle. Effectiveness is measured using accuracy.

intervention	model	effectiveness (%) $\mathrm{acc}_{\mathrm{floorhue}}$	$\mathrm{acc}_{\mathrm{objecthue}}$	$\mathrm{acc}_{\mathrm{objectorientation}}$	$\mathrm{acc}_{\mathrm{objectscale}}$	$\mathrm{acc}_{\mathrm{objectshape}}$	$\mathrm{acc}_{\mathrm{wallhue}}$	reversibility $l_1^{(1)}$	$l_1^{(10)}$
floor hue	VAE	100.00	100.00	100.00	100.00	100.00	100.00	1.62 (0.07)	1.62 (0.07)
	GAN	100.00	100.00	99.90 (0.16)	99.37 (0.13)	99.30 (0.16)	100.00	5.23 (0.15)	5.25 (0.15)
object hue	VAE	100.00	100.00	100.00	100.00	100.00	100.00	1.62 (0.07)	1.62 (0.07)
	GAN	100.00	99.99 (0.01)	99.91 (0.15)	99.29 (0.16)	99.16 (0.25)	100.00	5.33 (0.16)	5.35 (0.16)
object orientation	VAE	100.00	100.00	100.00	100.00	100.00	100.00	1.62 (0.07)	1.62 (0.07)
	GAN	100.00	100.00	99.88 (0.16)	99.07 (0.15)	99.00 (0.22)	100.00	5.32 (0.14)	5.35 (0.14)
object scale	VAE	100.00	100.00	100.00	100.00	100.00	100.00	1.62 (0.07)	1.62 (0.07)
	GAN	100.00	100.00	99.91 (0.15)	98.83 (0.27)	99.19 (0.27)	100.00	5.30 (0.15)	5.31 (0.15)
object shape	VAE	100.00	100.00	100.00	100.00	100.00	100.00	1.62 (0.07)	1.62 (0.07)
	GAN	100.00	100.00	99.92 (0.12)	98.76 (0.27)	99.10 (0.16)	100.00	5.37 (0.15)	5.39 (0.15)
wall hue	VAE	100.00	100.00	100.00	100.00	100.00	100.00	1.62 (0.07)	1.62 (0.07)
	GAN	100.00	100.00	99.86 (0.20)	99.30 (0.19)	99.32 (0.19)	100.00	5.27 (0.16)	5.29 (0.16)

Table C.5: Quality metrics on the CelebA-HQ test set over 3 random seeds, with different subsets of abducted latent variables from our conditional VDVAE model. Composition is measured via the null intervention and reversibility after one intervention cycle. Effectiveness of test set counterfactuals is measured using F1-score given by our 'smiling'/'eyeglasses' attribute classifiers.

	smiling intervention			eyeglasses intervention		
latents abducted	composition $l_1^{(1)} \downarrow$	reversibility $l_1^{(1)} \downarrow$	effectiveness F1-score ↑	composition $l_1^{(1)} \downarrow$	reversibility $l_1^{(1)} \downarrow$	effectiveness F1-score ↑
1	60.100 (0.127)	59.521 (0.164)	0.984 (0.0005)	60.273 (0.144)	59.263 (0.013)	0.979 (0.002)
2	55.521 (0.068)	54.360 (0.092)	0.982 (0.0008)	55.443 (0.036)	54.314 (0.055)	0.982 (0.005)
4	29.489 (0.018)	30.705 (0.030)	0.957 (0.001)	29.491 (0.048)	30.758 (0.025)	0.976 (0.005)
8	20.890 (0.018)	22.604 (0.025)	0.933 (0.002)	20.896 (0.011)	22.526 (0.005)	0.932 (0.006)
16	15.997 (0.003)	18.269 (0.021)	0.921 (0.003)	16.312 (0.009)	18.323 (0.009)	0.932 (0.019)
24	6.606 (0.001)	9.880 (0.017)	0.886 (0.006)	8.092 (0.002)	9.290 (0.006)	0.872 (0.003)
32	4.160 (0.002)	6.189 (0.015)	0.858 (0.001)	4.451 (0.0006)	5.606 (0.010)	0.865 (0.023)
42 (all)	3.657 (0.0006)	5.701 (0.017)	0.848 (0.0006)	3.656 (0.001)	4.808 (0.004)	0.854 (0.006)