Opportunities and Risks of Stochastic Deep Learning

Author:
Panagiotis Eustratiadis

Date:
November 10th, 2023

Supervisor:
Prof. Timothy M. Hospedales

A thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy in Artificial Intelligence

Institute of Perception, Action, and Behaviour
School of Informatics
University of Edinburgh

To Christina.

For when I was sad, you were sad
and we were sad together.

For when I was dumb, you were smart
and with our hearts we'd giggle.

For when it was dark, you were light
and I could see the way.

Acknowledgements

I am deeply grateful to my academic advisor, prof. Timothy Hospedales, who gave me the opportunity to be part of his research group, and provided me with the guidance and support I needed to complete one of the biggest challenges of my life. Timothy is a great and uncompromising scientist who plays the game of life in the hard difficulty setting, and I hope to put everything he taught me – willingly, or unintentionally – to good use.

Diogenis Laertius, in his work "Lives of Eminent Philosophers" attributes the following quote to Socrates: "There is only one good, that is, knowledge, and only one evil, that is, ignorance; wealth and good birth bring their possessor no dignity, but on the contrary evil.". This quote means more to me than it probably should. In my life, I have the ambition to wake up every day, more knowledgeable than the last. The people that I thank in this paragraph, my PhD colleagues and friends, with their brilliant minds and warm personalities, had a role in helping me realise that ambition. I would like to thank Henry, Linus, Boyan, Ondrej, Raman, and Ruchika for their support. I would especially like to thank Arushi and Kiyoon for their unconditional friendship, and promise that I will forever remember the evenings we spent together when life got too overwhelming. Finally, I thank my friends, Andreas and Marina, for doing their best to help me.

But most of all, I thank Christina for her love and patience, without which none of this would have been written.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Panagiotis Eustratiadis)

Abstract

This thesis studies opportunities and risks associated with stochasticity in deep learning that specifically manifest in the context of adversarial robustness and neural architecture search (NAS). On the one hand, opportunities arise because stochastic methods have a strong impact on robustness and generalisation, both from a theoretical and an empirical standpoint. In addition, they provide a framework for navigating non-differentiable search spaces, and for expressing data and model uncertainty. On the other hand, trade-offs (i.e., risks) that are coupled with these benefits need to be carefully considered. The three novel contributions that comprise the main body of this thesis are, by these standards, instances of opportunities and risks.

In the context of adversarial robustness, our first contribution proves that the impact of an adversarial input perturbation on the output of a stochastic neural network (SNN) is theoretically bounded. Specifically, we demonstrate that SNNs are maximally robust when they achieve weight-covariance alignment, i.e., when the vectors of their classifier layer are aligned with the eigenvectors of that layer's covariance matrix. Based on our theoretical insights, we develop a novel SNN architecture with excellent empirical adversarial robustness and show that our theoretical guarantees also hold experimentally.

Furthermore, we discover that SNNs partially owe their robustness to having a noisy loss landscape. Gradient-based adversaries find this landscape difficult to ascend during adversarial perturbation search, and therefore fail to create strong adversarial examples. We show that inducing a noisy loss landscape is not an effective defence mechanism, as it is easy to circumvent. To demonstrate that point, we develop a stochastic loss-smoothing extension to state-of-the-art gradient-based adversaries that allows them to attack successfully. Interestingly, our loss-smoothing extension can also (i) be successful against non-stochastic neural networks that defend by altering their loss landscape in different ways, and (ii) strengthen gradient-free adversaries.

Our third and final contribution lies in the field of few-shot learning, where we develop a stochastic NAS method for adapting pre-trained neural networks to previously unseen classes, by observing only a few training examples of each new class. We determine that the adaptation of a pre-trained backbone is not as simple as adapting all of its parameters. In fact, adapting or fine-tuning the entire architecture is sub-optimal, as a lot of layers already encode knowledge optimally. Our NAS algorithm searches for the optimal subset of pre-trained parameters to be adapted or fine-tuned, which yields a significant improvement over the existing paradigm for few-shot adaptation.

Contents

Chapter 1

Introduction

In the era of modern machine learning, we[1] seek to exploit the recent gigantic leaps in hardware technology to solve important and difficult problems that were out of reach in the past. Nowadays, we can process large datasets of anything between thousands and millions of data instances (e.g., images, text, sensory data, measurements, etc.), and attempt to model problems using Deep Neural Networks (DNNs) parameterised by anything between millions and billions of parameters. The high-level objective of this deep modelling is to generalise, i.e., to make predictions about previously unseen data instances as accurately as possible.

The depth and, therefore, number of learnable parameters in DNNs is both a blessing and a curse. It is a blessing, because for the first time in the history of artificial intelligence (AI) we are able to tackle challenging problems such as natural language modelling (Brown et al., 2020), text-guided image generation (Radford et al., 2021), and protein structure prediction (Jumper et al., 2021) with staggering success. And it is also a curse, because there is a considerable aspect of this kind of deep modelling that we do not fully understand, and are not yet able to interpret. One of the main reasons for this flaw is that while in the traditional machine learning paradigm we would pre-process raw data before giving it as input to a pattern recognition algorithm (Bishop, 2007, Ch. 1), DNNs are able to directly "ingest" raw data and form their own latent representations of the world (i.e., features) without need for human intervention.

1.1 Stochasticity in Deep Learning

Stochasticity, or randomness, is an essential component of modern machine learning that has been exploited in solution to some of its most fundamental challenges:

- **Hardware limitations**: The copious amount of data required to train DNNs

[1] Note that the term "we" in this context refers to "our research community" or "machine learning scientists". Starting from Section 1.2, and for the rest of the thesis, "we" refers to "my co-contributors and I".

does not fit in the physical memory of even our most technologically-advanced processing units. For that reason, DNNs can only be trained by observing small segments of the available training data at a time.

- **Generalisation** to unseen data: During optimisation, DNNs run the risk of getting stuck in local minima or overfitting to the training data, thus failing to generalise to the entire possible data distribution.

- **Robustness**: AI systems should develop an understanding of the world that is invariant to both naturally- and adversarially-induced data corruptions – similarly to how a human would recognise a picture of a cat, regardless of whether the image were blurred, rotated, cropped, or noisy.

- **Discontinuous** or **non-differentiable** search spaces: Learning algorithms are often required to optimise an objective that is not differentiable and, therefore, gradient-based search and optimisation methods are not directly applicable.

- **Uncertainty** estimation: Datasets that are collected to train DNNs are, in general, clean and carefully-curated. However, this is not an accurate reflection of the real-world environments in which they are deployed, which have inherent variability and unpredictability. Despite this reality gap, AI systems need to be able to perform reliably in the face of uncertainty.

The rest of this section provides an overview of stochastic methods that have been proposed in the past to tackle each of these problems. We can categorise stochasticity in deep learning into two major categories: (i) Stochasticity during learning, and (ii) stochasticity during inference.

1.1.1 Stochastic Learning

To address **hardware limitations** and **generalisation** to unseen data, several learning algorithms based on Stochastic Gradient Descent (SGD) (Kiefer and Wolfowitz, 1952) have been proposed, a comprehensive review of which has been written by Bottou et al. (2018). Extensions to SGD include momentum (Qian, 1999), parameter-wise adaptive learning rates (Duchi et al., 2011), and others. This family of stochastic optimisation methods is an improvement upon the traditional deterministic gradient descent (GD), as GD is less computationally efficient and more prone to getting stuck in local minima (Bottou et al., 2018).

Further in the context of **generalisation**, dropout (Srivastava et al., 2014) and variational dropout (Kingma et al., 2015) have been proposed as stochastic regularisation techniques that randomly deactivate subsets of neurons at each training iteration. The main intuition behind dropout is to soften the implicit dependencies between neurons, and allow different parts of a neural network to specialise on different aspects of the data. This results in stronger features and richer latent representations.

Another common family of approaches that can boost model **generalisation** and can be used in combination with the aforementioned, involve artificially increasing the volume of the available training data by inducing small random corruptions to each data sample. Methods belonging to this family are known as data augmentation methods (Shorten and Khoshgoftaar, 2019). It should be mentioned that data augmentation is not only beneficial because it increases the size of the training dataset. More importantly, it encourages the trained model to be invariant to artifacts and corruptions that can manifest naturally in the data (Ericsson et al., 2022; Chavhan et al., 2023).

Besides random and natural corruptions, training data can be augmented with adversarial (i.e., loss-maximising) perturbations. Training models with this type of data augmentation is known as adversarial training (AT) (Goodfellow et al., 2015; Carlini and Wagner, 2017), and it is the most common way to achieve **robustness** against adversarial attacks (Szegedy et al., 2014). Unlike the standard data augmentation-based training regimes, AT is defined as a min-max problem in which a model observes training samples and attempts to minimise its prediction error, while an adversary attempts to generate training samples that maximise it. The literature is rich with various proposed AT methods, a systematic review of which has been written by Bai et al. (2021).

Finally, it is often the case when the learning objective is a **discontinuous**, or otherwise **non-differentiable** function, making gradient-based optimisation methods like SGD inapplicable. These types of objectives are typically optimised with stochastic learning or stochastic search. Prime examples of such objectives include:

- The reward function in reinforcement learning (RL) problems that require function approximation for policy learning. In such settings, RL algorithms randomly sample episodes from a data distribution (i.e., “environment”) and stochastically collect rewards from these episodes to use as a training signal (e.g., policy gradient methods, Sutton et al., 1999).

- The success rate of black-box adversarial attacks that can only observe the attacked model’s discrete predictions. Popular methods in this line of work use stochastic gradient approximation (e.g., Liu et al., 2020), random search (e.g., Andriushchenko et al., 2020), and evolutionary search (e.g., Su et al., 2019) to train strong black-box adversaries.

- Performing neural architecture search (NAS) to select the best-performing DNN architecture for a given task from a discrete, but intractably large search space of possible layer and parameter configurations. Exhaustive search is computationally unfeasible in this setting – so it is common to perform evolutionary search instead (e.g., Miikkulainen et al., 2019).

1.1.2 Stochastic Inference

In recent years, variational Bayesian inference (Paisley et al., 2012) has been an important direction of research in stochastic deep learning, as it provides us with effective and (mostly) theoretically-grounded frameworks for developing Stochastic Neural Networks (SNNs). The aforementioned variational dropout (Kingma et al., 2015), if used during inference, is the simplest example of such a framework. In addition, a highly influential example of variational inference is the variational information bottleneck (Alemi et al., 2017), that enables SNNs to learn maximally expressive latent representations of the world.

Expressivity is an important property of SNNs that improves their capacity for **generalisation** considerably, as shown by the highly influential work of Kingma and Welling (2014) on variational autoencoders with enormous success. The intuitive explanation for their expressivity is that they model the world flexibly as distributions of features rather than point estimates. Consider a face detection algorithm. Faces consist of a number of visual features, e.g., eyes, a nose, a mouth, to name a few. But within these features exists variation, e.g., eyes can be of several different colors, noses can be of several different shapes, mouths can be smiling or frowning, etc. An SNN trained for face detection can capture such variations, and develop richer latent representations for identifying facial characteristics.

Furthermore, it is believed that since SNNs sample their parameters from a distribution, they exhibit **robustness** against imperceptibly small adversarial input perturbations (Alemi et al., 2017). Even though there exists a significant amount of research that confirms this hypothesis empirically (e.g., He et al., 2019, Yu et al., 2021), it has also been shown that SNNs have strong theoretical ties to adversarial robustness (Alemi et al., 2017; Eustratiadis et al., 2021).

Stochastic inference is also a valuable tool when modelling and estimating **uncertainty**. Consider the following two scenarios, which correspond to two different types of uncertainty, epistemic and aleatoric, respectively:

- The smart voice assistant of a newly-bought smartphone prompts the user to repeat a number of small phrases during its initial setup.

- A self-driving vehicle is planning its course, and makes a left turn to enter a previously unseen road.

The first scenario describes an instance of epistemic uncertainty: the uncertainty in the way an AI system perceives the world. This uncertainty can be dampened by observing more data, i.e., the smart voice assistant is uncertain about its user's voice, so it collects a few data samples of it to better recognise it in the future. In contrast, the second scenario describes an instance of aleatoric uncertainty: the uncertainty that exists inherently in the data. This uncertainty is irreducible, i.e., even a perfectly-trained autonomous vehicle cannot predict what lies around the corner with absolute confidence.

Bayesian Neural Networks (BNNs) (Blundell et al., 2015), are a subclass of SNNs that enjoy all of the aforementioned benefits associated with SNNs (e.g., expressivity, generalisation), but are additionally capable of modelling and estimating uncertainty (Kendall and Gal, 2017). In BNNs, every weight is assigned a probability distribution – unlike the classical definition of DNNs, where every weight is assigned a point value. This allows BNNs to be sampled multiple times during inference to effectively approximate an ensemble of infinite models. In this formulation, the weight uncertainty of BNNs is a direct reflection of their predictive uncertainty. Interestingly, it has been shown that the total predictive uncertainty of BNNs can be further decomposed into its epistemic and aleatoric components (Depeweg et al., 2018), which adds a layer of interpretability on top of their other capabilities.

1.2 Contributions

The focus of this thesis is on adversarial robustness and non-differentiability. It is a compilation of three novel contributions on how stochasticity can benefit or harm an SNN's robustness against adversarial perturbations, and how it can act as an effective means of searching a non-differentiable architecture search space in the context of few-shot adaptation.

In Chapter 3, we prove that the impact of an adversarial input perturbation on the output of an SNN is theoretically bounded. Specifically, we demonstrate that SNNs are maximally robust when they achieve weight-covariance alignment, i.e., when the vectors of their classifier layer are aligned with the eigenvectors of that layer's covariance matrix. Based on our theoretical insights, we develop a novel SNN architecture with excellent empirical adversarial robustness, and show that our theoretical guarantees also hold experimentally.

Furthermore, in Chapter 4 we present our discovery that SNNs partially owe their robustness to having a noisy loss landscape. Gradient-based adversaries find this landscape difficult to ascend during adversarial perturbation search, and therefore fail to create strong adversarial examples. Our analysis shows that manipulating the loss landscape in such a way is not an effective defence, but a vulnerability. To demonstrate that point, we develop a stochastic loss-smoothing extension to existing state-of-the-art gradient-based adversaries that allows them to attack successfully. Interestingly, our loss-smoothing extension can also (i) be successful against non-stochastic neural networks that defend by altering their loss landscape in different ways, and (ii) strengthen gradient-free adversaries.

Chapter 6 details our third and final contribution that lies in the context of few-shot learning, where we develop a stochastic NAS method for adapting pre-trained DNNs to previously unseen classes, by observing only a few training examples of each new class. We determine that the adaptation of a pre-trained backbone is not as simple as adapting all of its parameters. In fact, adapting or fine-tuning the entire architecture is sub-optimal, as a lot of layers already encode knowledge optimally. Our NAS algorithm searches for the optimal subset of pre-trained parameters to be adapted or fine-tuned, which yields a significant improvement over the existing paradigm for few-shot adaptation.

1.2.1 Publications and Contributors

All contributions presented in this thesis were made under the supervision of Prof. Timothy Hospedales, who contributed with project ideas and guidance, along with my co-authors, Dr. Henry Gouk, Dr. Da Li, and Łukasz Dudziak. All other work, including the core body of research, materialisation of ideas and experimentation was carried out by myself.

The core contributions of this thesis correspond to the following peer-reviewed publications:

- **Chapter 3:** P. Eustratiadis, H. Gouk, D. Li, and T. M. Hospedales. Weight-covariance alignment for adversarially-robust neural networks. In *International Conference on Machine Learning (ICML)*, 2021.

- **Chapter 4:** P. Eustratiadis, H. Gouk, D. Li, and T. M. Hospedales. Attacking adversarial defences by smoothing the loss landscape. In *ICML Workshop on Adversarial Machine Learning*, 2022.

- **Chapter 6:** P. Eustratiadis, Ł. Dudziak, D. Li, and T. M. Hospedales. Neural fine-tuning search for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2024. **(under review)**

1.2.2 Thesis Structure

Chapter 1 introduces stochasticity in deep learning, motivates its importance, and highlights the primary pieces of work that comprise this thesis. Then, the thesis is divided into two parts, each detailing the research conducted on two different aspects of stochastic deep learning: Part I discusses stochastic learning and inference in the context of adversarial robustness, while Part II discusses stochastic learning and search in the context of few-shot adaptation.

Each part is further divided into its background, and main contribution(s). The purpose of the background chapters (chapters 2 and 5 for Parts I and II, respectively) is to ease the reader into the terminology and notation that is used in the contribution chapters, and provide a brief summary of the related work upon which these contributions were built. Chapters 3, 4, and 6 are the core contribution chapters; they present three novel pieces of work that are instances of opportunities and risks of stochastic deep learning.

The contribution chapters are accompanied by their own discussion sections, individually presenting interesting insights based on the results of each piece of work. The main body of the thesis is followed by an epilogue that highlights the scientific impact of my research contributions at the time of writing, as well as their future potential. Finally, Appendices A, B, and C provide more information and technical details, complementary to the work presented in the contribution chapters.

Part I

Adversarial Robustness of Stochastic Neural Networks

Chapter 2

Background

It has been shown that, even though DNNs can perform exceptionally well in computer vision tasks such as image recognition (e.g., He et al., 2016), they can also be easily misdirected by carefully-crafted but humanly-imperceptible pixel-level perturbations to the input (Szegedy et al., 2014). The search for such perturbations is commonly referred to as an *adversarial attack*, and images perturbed in such a way are called *adversarial examples*.

The discovery of adversarial examples poses a serious existential threat to modern machine learning, as it impacts the security and reliability of models that operate in mission-critical settings where reliable outcomes are crucial (e.g., medicine, justice). This has led to adversarial defence emerging as an important field of machine learning research, with the purpose of creating models that are robust against adversarial perturbations.

2.1 Adversarial Attacks

In the context of image classification, an adversarial attack is defined as the search for a perturbation, δ, that when applied to a clean input image, x, creates an adversarial example $\tilde{x} = x + \delta$ that is misclassified. Formally:

$$f_\theta(x) \neq f_\theta(\tilde{x}), \tag{2.1}$$

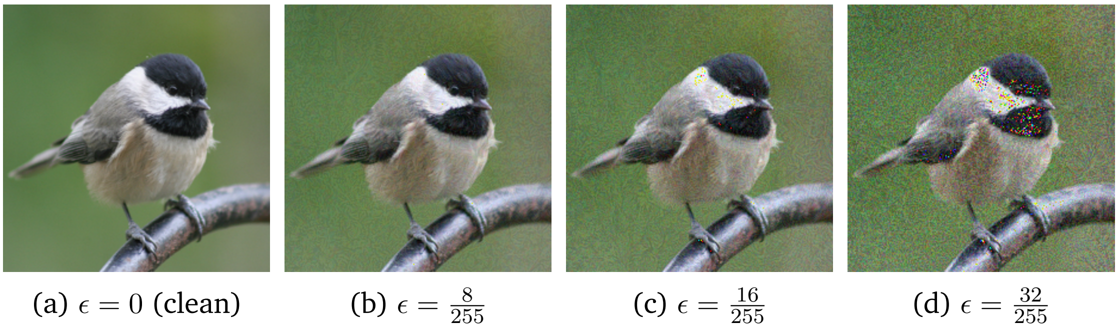
given a deep neural classifier f_θ parameterised by θ that classifies x correctly. In this thesis, we consider norm-bounded adversarial attacks under the ℓ_p threat model that imposes a norm constraint on δ during perturbation search:

$$||x - \tilde{x}||_p \leq \epsilon, \tag{2.2}$$

where ϵ is a small value indicating the attack strength, and p is typically in $\{1, 2, \infty\}$. The set of all adversarial examples built from x w.r.t. f_θ can then be defined as:

$$\Delta(x, f_\theta) = \{\tilde{x} \mid f_\theta(x) \neq f_\theta(\tilde{x}), ||x - \tilde{x}||_p \leq \epsilon\}. \tag{2.3}$$

The role of the norm constraint is twofold:



(a) $\epsilon = 0$ (clean) (b) $\epsilon = \frac{8}{255}$ (c) $\epsilon = \frac{16}{255}$ (d) $\epsilon = \frac{32}{255}$

Figure 2.1: Example of an adversarially-perturbed image subject to the ℓ_∞ norm, with various perturbation strengths. In this specific example, only (a) is classified as the correct class (chickadee, ImageNet index: 19). (b) is classified as a wall clock (ImageNet index: 892), (c) is classified as a pillow (ImageNet index: 721), and (d) is classified as chainmail armour (ImageNet index: 490).

- It defines the imperceptibility of the attack, i.e., the lower the value of ϵ, the more imperceptible to the human eye the attack is.
- The literature uses it as a fair means of comparison between proposed attacks and defences.

Figure 2.1 illustrates an image from ImageNet (Deng et al., 2009), adversarially-perturbed subject to the ℓ_∞ norm, with different values of ϵ.

Adversarial attacks can be further categorised according to their misclassification objective, or the perturbation search method:

- **Misclassification objective:** *Targeted* attacks search for a perturbation, δ, such that the model misclassifies an image belonging to one class, as an image of a specific target class. *Untargeted* attacks do not define a target class, and are considered successful as long as any misclassification occurs.
- **Perturbation search method:** *Gradient-based* search allows the adversary to guide the perturbation vector towards the direction that maximises the classification loss. This type of attack is typically stronger and faster than its gradient-free counterpart, but assumes white-box access to the model's parameters. In *gradient-free* search, on the other hand, the adversary is constrained to query-level access, i.e., it treats the model as a black box, and only observes its predictions to guide perturbation search.

An extensive survey on the taxonomy of adversarial attacks has been written by Chakraborty et al. (2021). In this thesis, we mainly consider untargeted attacks, but explore both gradient-based and gradient-free approaches.

2.1.1 Gradient-Based Adversaries

Consider an image classification problem with C classes. Let f_θ be a DNN with parameters θ, and x an input image belonging to class $c \in C$. The first and simplest gradient-based adversary outlined in prior work is the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015); a single-step attack that adds a small perturbation to x in the direction indicated by the sign of the gradient of an objective function $\mathcal{L}(f_\theta(x), c)$ w.r.t. x. Formally:

$$\tilde{x} = x + \epsilon \cdot \mathrm{sgn}(\nabla_x \mathcal{L}(f_\theta(x), c)), \tag{2.4}$$

where ϵ denotes the attack strength. The Basic Iterative Method (BIM) (Kurakin et al., 2017) was introduced shortly thereafter as an iterative variant of FGSM,

$$\tilde{x}_{t+1} = \tilde{x}_t + \eta \cdot \mathrm{sgn}(\nabla_{\tilde{x}_t} \mathcal{L}(f_\theta(\tilde{x}_t), c)) \quad \text{s.t. } ||\tilde{x}_t - \tilde{x}_{t+1}||_p \leq \epsilon, \tag{2.5}$$

where η denotes the step size, or learning rate, and $\tilde{x}_0 = x$. Projected Gradient Descent (PGD) (Madry et al., 2018) was later proposed as an improved version of BIM, where the initial perturbation is a randomly selected point in the ϵ-ball of x. Recent contributions have improved upon this scheme, e.g., through Nesterov's acceleration (Lin et al., 2020) and variance tuning (Wang and He, 2021).

2.1.2 Gradient-Free Adversaries

Unlike gradient-based adversaries, gradient-free adversaries assume that the details of the target model are unknown, and can only access the model's predictions through queries. In this setting, it is common to employ transfer attacks (Papernot et al., 2017a); where someone can typically train a substitute of the target model, search for successful adversarial examples using gradient-based search on the substitute model, and use them to attack the target model. Alternatively, Chen et al. (2017) show that the gradients of the target model can be approximated using Zeroth-Order Optimization (ZOO). Another line of research on gradient-free adversaries make use of evolutionary search (Su et al., 2019), or random search (Andriushchenko et al., 2020), to find a pixel or small image segment that, if perturbed, has maximal capacity to fool the target model.

2.2 Overview of Stochastic Defences

In this section, we review recent related work that has shown that SNNs can yield promising performance in adversarial robustness, by injecting either fixed or learnable noise into the models.

The idea behind Random Self Ensemble (RSE) (Liu et al., 2018) is that one can simulate an ensemble of virtually infinite models while only training one. This can be achieved by injecting additive, spherical, Gaussian noise into various layers of a DNN and performing multiple forward passes during inference. Simple as this approach may be, it effectively improves the model's robustness in comparison to a conventional deterministic model. RSE treats the variance of the injected noise as a hyperparameter that is heuristically tuned, rather than learned in conjunction with the other network parameters. In contrast, He et al.

(2019) propose Parametric Noise Injection (PNI), in which a fixed spherical noise distribution is controlled by a learnable "intensity" parameter, further improving adversarial robustness. The authors show that the noise can be incorporated into different locations of a neural network, i.e., to both feature activations and model weights. The injected noise is trained together with the model parameters via adversarial training. Learn2Perturb (L2P) (Jeddi et al., 2020) is an extension of PNI, except instead of learning a single spherical noise parameter, L2P learns a set of parameters defining an isotropic noise perturbation injection module. The parameters of both the perturbation injection module and the model are updated using adversarial training in an alternating manner. Finally, the Simple and Effective SNN (SESNN) (Yu et al., 2021) features fully-trainable stochastic layers, which are trained for adversarial robustness by adding a regularisation term to the objective function that maximizes the entropy of the learned noise distribution. Unlike the other SNNs, SESNN only requires clean images during training.

Another class of stochastic defences apply noise to the input images, rather than injecting noise to intermediate activations (Pinot et al., 2019; Li et al., 2019; Cohen et al., 2019). From a theoretical point of view, this can be seen as "smoothing" the function implemented by the neural network in order to reduce the amount the output of the network can change when the input is changed only slightly. This type of defence can be considered a black-box defence, in the sense that it does not actually involve regularizing the weights of the DNN, but only modifies the input. While interesting, it has primarily been applied in scenarios where one is using a model-as-a-service framework, and an adversary cannot be certain about whether or not the attacked model was trained with an adversarial defence method (Cohen et al., 2019).

2.2.1 Pseudo-Robustness via Gradient Obfuscation

In their paper, Athalye et al. (2018a) demonstrate that many existing defences create a false impression of robustness to gradient-based adversaries by masking the gradient of the loss function from the attacker. They identify three types of gradient obfuscation: shattered, stochastic, and vanishing gradients; and show that gradient-obfuscating defences are not reliable.

In this thesis we are primarily concerned with stochastic gradients, which manifest in stochastic defences. Stochastic defences use SNNs that sample their weights or activations from a distribution and, as a result, their computed gradients are also sampled from a distribution. Consider a Bayesian linear estimator, f, with weights $W \sim \mathcal{N}(\mu_W, \Sigma_W)$ and a bias term $b \sim \mathcal{N}(\mu_b, \Sigma_b)$ that predicts a continuous value, y, for a given input, x, as $f(x) = Wx^T + b$. Because W and b are independent random variables, we can say that $f(x) \sim \mathcal{N}(\mu_W x^T + \mu_b, \Sigma_W + \Sigma_b)$. If we were to perform gradient-based search (e.g., Equations 2.4, 2.5) to attack this Bayesian estimator, we would compute the gradient of its regression loss function, $\mathcal{L}$, w.r.t. the input, e.g., $g = \nabla_x (f(x) - y)^2$ in the case where $\mathcal{L}$ is the squared error. Therefore, g is also a random variable and its distribution depends on the derivative of $\mathcal{L}$. In this simplified example, the distribution of g is Chi-squared, because $\frac{d}{dx}(f(x) - y)^2 = 2W(f(x) - y)$.

To deal with stochastic gradients, Athalye et al. (2018a) developed Expectation over Transformation (EoT) (Athalye et al., 2018b); a method that repeatedly samples the target model's gradient w.r.t. the input, and computes the average of these samples to obtain a more accurate estimation of the "true" gradient. Further, Gaussian sampling has been previously used in order to circumvent non-stochastic, but otherwise obfuscating defences (e.g., shattered gradients) (Tramèr et al., 2020; Pintor et al., 2021). Following this line of work, it has recently become a requirement for stochastic defence research to incorporate a series of checks that ensure new stochastic defence methods do not owe their success to gradient obfuscation. Our proposed stochastic defence detailed in Chapter 3 includes these checks in its evaluation (Section 3.3.5).

Expectation over Transformation We now highlight a few technical details about EoT. Let f_θ be a SNN with parameters θ, and x an input image belonging to class $c \in C$. The stochastic weights or activations of f_θ cause $f_\theta(x)$ to be randomised; as a result, $\nabla_x \mathcal{L}(f_\theta(x), c)$ is a distribution of gradients. EoT is, in essence, a Monte-Carlo method that estimates the true gradient of the loss function by averaging n gradient samples as

$$\omega = \frac{1}{n} \sum_{i=0}^{n} \nabla_x \mathcal{L}(f_\theta(x)_i, c) \ , \tag{2.6}$$

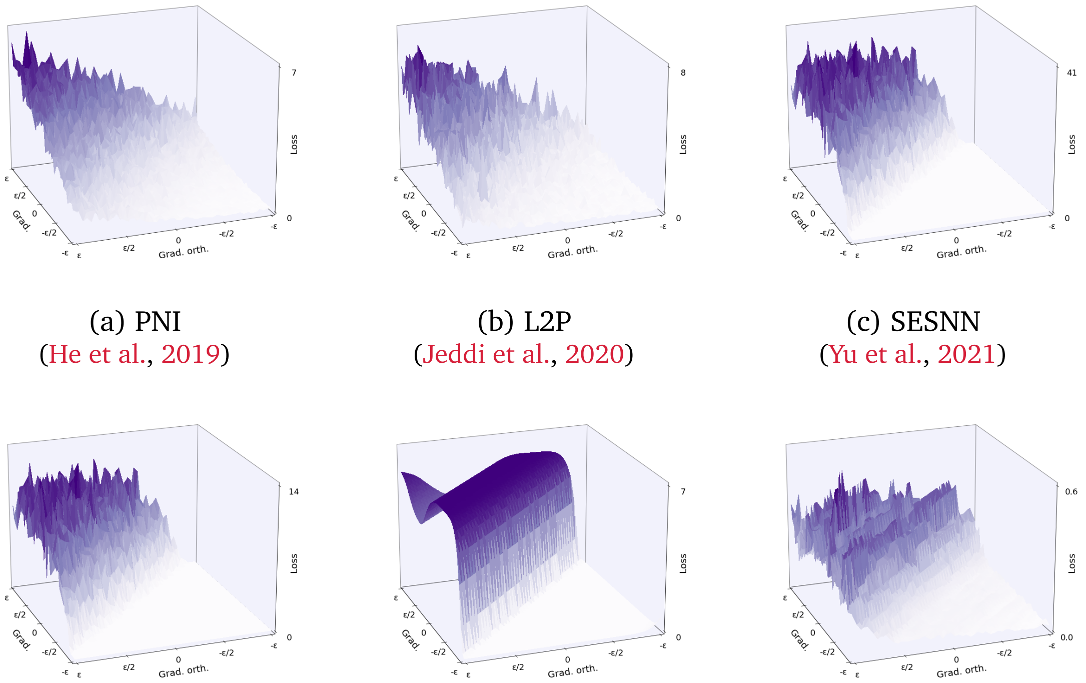
where ω denotes the approximation of the true gradient, g.

2.3 Defences with an Obfuscating Loss Landscape

In our work we consider both stochastic and non-stochastic defences that we found to create a rough, discontinuous, or otherwise rugged loss surface that is difficult for gradient-based adversaries to navigate. In the case of stochastic defences, we only consider related work that have applied EoT in their model evaluation. These defences are highlighted in detail in Section 2.2.

However, an obfuscating loss landscape is not an exclusive characteristic of SNNs. k-Winner Takes All (k-WTA) (Xiao et al., 2020) is a defence that replaces the ReLU activation with a discontinuous function. Further, Anti-Adversaries (AA) (Alfarra et al., 2021) is a recent training-free adversarial defence that could be categorised as a "black-box" defence. It improves adversarial robustness by prepending a layer that induces discontinuity to the loss landscape.

Our observation is that all of these methods can defend successfully against white-box adversarial attacks, largely through inducing rough loss landscapes that gradient-based adversaries struggle to ascend. 3-dimensional slices through the loss landscapes of the aforementioned defences are shown in Fig. 2.2. For comparison, we provide examples of smooth loss landscapes in Appendix B, as well as more details about how these plots are computed.



(a) PNI (He et al., 2019)

(b) L2P (Jeddi et al., 2020)

(c) SESNN (Yu et al., 2021)

(d) WCA (Eustratiadis et al., 2021)

(e) AA (Alfarra et al., 2021)

(f) k-WTA (Xiao et al., 2020)

Figure 2.2: Loss landscapes of each of the adversarial defences considered in this paper. All defences use a ResNet-18 backbone and the loss surfaces are constructed on a correctly-classified test image from CIFAR-10. The X axis is the gradient w.r.t. the clean input image, and the Y axis is chosen to be orthogonal to X. The Z axis is the value of the loss function for each perturbation within the ϵ-ball of the input image, where $\epsilon = \frac{8}{255}$. Refer to Section 4.2.5 for more details about how these loss surfaces are computed.

Chapter 3

Weight-Covariance Alignment for Adversarially-Robust Neural Networks

This Chapter corresponds to the paper: P. Eustratiadis, H. Gouk, D. Li, and T. M. Hospedales. Weight-covariance alignment for adversarially-robust neural networks. In *International Conference on Machine Learning (ICML)*, 2021.

Stochastic neural networks (SNNs) that either inject noise into their hidden layers or sample their parameters from a probability distribution have recently been shown to be robust against adversarial attacks (Alemi et al., 2017). Despite that, the vast majority of existing SNN-based methods for adversarial robustness are heuristically-motivated. In addition, these methods often rely on adversarial training, which is computationally costly. In this Chapter, we propose a novel SNN architecture that achieves state-of-the-art robust performance without the need for adversarial training, and is coupled with solid theoretical justification. Specifically, while existing SNNs inject learned or hand-tuned isotropic noise, our SNN learns an anisotropic noise distribution to optimise a learning-theoretic bound on adversarial robustness. We evaluate our method on various popular image classification benchmarks and show that it can be applied to different backbone architectures. Overall, it provides adversarial robustness against a variety of white-box and black-box attacks, while being simple and fast to train compared to existing alternatives.

3.1 Introduction

In this section, we identify and discuss three limitations of current state-of-the-art stochastic defences. First, most contemporary adversarial defence methods use a mixture of clean and adversarial samples during training; this approach is formally known as adversarial training (AT) (Goodfellow et al., 2015). The process of generating good adversarial examples during training is complicated (e.g., Cai et al., 2018), and it leads to significantly higher computational cost and longer training time. Second, a lot of existing adversarial defences (e.g., Mustafa et al., 2019), and especially stochastic defences (e.g., He et al., 2019, Jeddi et al., 2020), are heuristically-motivated – even though there is empirical evidence of their effectiveness against adversarial attacks, they lack theoretical support. Third, the noise incorporated by existing stochastic models is *isotropic* i.e., generated from a multivariate Gaussian distribution with a diagonal covariance matrix, meaning that it perturbs learned features of different dimensions independently. Our theoretical analysis shows that this is a strong assumption, and that *anisotropic* noise provides a higher degree of flexibility, leading to better performance.

Our research addresses the aforementioned limitations, as we propose an SNN architecture and training regime that makes use of learnable anisotropic noise. Specifically, we theoretically analyse the margin between the clean and adversarial performance of a stochastic model and derive an upper bound on the difference between these two quantities. This novel theoretical insight suggests that the anisotropic noise covariance in an SNN should be optimised to align with the classifier weights, which has the effect of tightening the bound between clean and adversarial performance. This alignment can be achieved with an easy-to-implement regulariser, which can be efficiently optimised on clean samples alone, without need for adversarial training. Our work belongs in the family of certified defences (Raghunathan et al., 2018; Lécuyer et al., 2019; Pinot et al., 2019; Li et al., 2019), although the robustness guarantees are theoretically provable only for simple linear models.

Finally, we show that our method, Weight-Covariance Alignment (WCA), can be applied to architectures of varied depth and complexity – namely, LeNet++ (Wen et al., 2016) and ResNet-18 (He et al., 2016) – and achieves state-of-the-art robustness across widely-used benchmarks, including CIFAR-10/100 (Krizhevsky and Hinton, 2009), SVHN (Netzer et al., 2011) and F-MNIST (Xiao et al., 2017). Moreover, our method can handle high-resolution images, as we show by additionally including Imagenette (Howard, 2019) and mini-ImageNet (Vinyals et al., 2016) in our experimental setup. This high level of robustness is demonstrated against both white-box and black-box adversaries. We name our proposed model WCA-Net.

The scientific contributions described in this Chapter can be summarised as follows:

- Unlike the majority of existing stochastic defences that are heuristically-motivated, our proposed stochastic defence is trained to optimise a derived learning-theoretic bound; resulting in a solid theoretical justification for its robust performance.

- To the best of our knowledge, this is the first stochastic defence with learned anisotropic noise to be proposed in the literature.

- WCA only requires clean samples for training, and does not depend on costly adversarial training.

- We demonstrate the state-of-the-art performance of our stochastic defence on various benchmarks, as well as its resilience against both white- and black-box attacks.

3.2 Method

Based on theoretical analysis of how the injected noise can impact generalisation performance, further expanded in Section 3.2.1, we propose a weight-covariance alignment loss term that encourages the weight vectors associated with the final linear classification layer to be aligned with the covariance matrix of the injected noise. Consequently, our theory leads us to use anisotropic noise, rather than the isotropic noise typically employed by previous approaches.

Our method fits into the category of SNNs that apply additive noise to the penultimate activations of the network. Consider the function, $f_\theta(x)$, as the *backbone*, or feature extractor portion of the network i.e., everything except the final classification layer. Our WCA-Net architecture is defined as

$$h(x) = W(f(x) + z) + b\,, \quad z \sim \mathcal{N}(0, \Sigma), \tag{3.1}$$

where W and b are the parameters of the final linear layer, and z is the vector of additive noise. We choose this stochastic architecture because it offers the following benefits:

- It is flexible w.r.t. the selection of a backbone architecture, and as feature extractors become more powerful over time, this method remains relevant.

- There is only one source of noise, and the noise itself is additive. This makes the theoretical derivation of our method clear and straightforward, as opposed to when the noise is a parametric function of the data, or is injected in multiple places (e.g., He et al., 2019, Jeddi et al., 2020).

- In our method, we are not interested in uncertainty estimation or any type of variational inference; therefore we do not have to use a fully-fledged BNN.

The objective function used to train our model is

$$\mathcal{L} = \mathcal{L}_{\text{C}} - \mathcal{L}_{\text{WCA}}, \tag{3.2}$$

where $\mathcal{L}_{\text{C}}$ and $\mathcal{L}_{\text{WCA}}$ represent the classification loss (e.g. softmax composed with cross-entropy) and weight-covariance alignment term respectively. We describe each of our technical contributions in the remainder of this section.

3.2.1 Weight-Covariance Alignment

Non-stochastic methods for defending against adversarial examples typically try to guarantee that the prediction for a given input image cannot be changed. In contrast, a defence that is stochastic should aim to minimise the probability that the prediction can be changed. In this section, we present a theoretical analysis of the probability that the prediction of an SNN will be changed by an adversarial attack. For simplicity, we restrict our analysis to the case of binary classification.

Denoting a feature extractor as f, we define an SNN h, trained for binary classification as

$$h(x) = w^T(f(x) + z) + b \ , \quad z \sim \mathcal{N}(0, \Sigma), \tag{3.3}$$

where w is the weight vector of the classification layer and b is the bias. We denote the non-stochastic version of h, where the value of z is always a vector of zeros, as $\tilde{h}$. The margin of a prediction is given by

$$m_h(x, y) = yh(x), \tag{3.4}$$

for $y \in \{-1, 1\}$. It is positive if the prediction is correct, and negative otherwise.

The quantity in which we are interested is the difference in probabilities of misclassification when the model is and is not under adversarial attack δ, which is given by

$$G^h_{p,\epsilon}(x, y) = \max_{\delta : \|\delta\|_p \leq \epsilon} P(m_h(x + \delta, y) \leq 0) - P(m_h(x, y) \leq 0). \tag{3.5}$$

Our main theoretical result shows how one can take an adversarial robustness bound, $\Delta^{\tilde{h}}_p(x, \epsilon)$, for the deterministic version of a network, and transform it to a bound on G for the stochastic version of the network.

Theorem 1. *The quantity $G^h_{p,\epsilon}(x, y)$, as defined in Equation 3.5, is bounded as*

$$G^h_{p,\epsilon}(x, y) \leq \frac{\Delta^{\tilde{h}}_p(x, \epsilon)}{\sqrt{2\pi w^T \Sigma w}}, \tag{3.6}$$

where the robustness of the deterministic version of h is known to be bounded as $|\tilde{h}(x) - \tilde{h}(x + \delta)| \leq \Delta^{\tilde{h}}_p(x, \epsilon)$ for any $\|\delta\|_p \leq \epsilon$.

The proof is provided in Appendix A. From Theorem 1 we can observe that increasing the bi-linear form, $w^T \Sigma w$, of the noise distribution covariance and the classifier reduces the gap between clean and robust performance. As such, we define the loss term,

$$\mathcal{L}_{\text{WCA}} = \sum_{i=1}^{C} \ln(w_i^T \Sigma w_i), \tag{3.7}$$

where C is the number of classes in the classification problem, and w_i is the weight vector of the final layer that is associated with the i^th^ class. We found that including the logarithm results in balanced growth rates between the $\mathcal{L}_C$

and $\mathcal{L}_{\text{WCA}}$ terms in Equation 3.2 as training progresses, hence improving the reliability of training loss convergence.

The key insight of Theorem 1, operationalised by Equation 3.7, is that the noise and weights should co-adapt to align the noise and weight directions. We call this loss Weight-Covariance Alignment (WCA) because it is maximised when each w_i is well-aligned with the eigenvectors of the covariance matrix.

This WCA loss term runs into the risk of maximizing the magnitude of w, rather than encouraging alignment or increasing the scale of the noise. To avoid the uncontrollable scaling of network parameters, it is common to penalise large weights by means of ℓ^2 regularization:

$$\mathcal{L} = \mathcal{L}_{\mathcal{C}} - \mathcal{L}_{WCA} + \lambda w^T w, \tag{3.8}$$

where λ controls the strength of the penalty. In our case, we apply the ℓ^2 penalty on the parameters of the classification layer and the covariance matrix. Another approach to limiting parameter magnitude would be to enforce norm constraints on w and Σ, e.g., using a projected subgradient method at each update. More details about this alternative approach are provided in Appendix A. Empirically, we found that the penalty-based approach outperformed the constraint-based approach, so we focus on the former by default.

3.2.2 Injecting Anisotropic Noise

In contrast to previous work that only considers injecting isotropic Gaussian noise (Liu et al., 2019b; He et al., 2019; Jeddi et al., 2020; Yu et al., 2021), we make use of anisotropic noise, providing a richer noise distribution than previous approaches. Crucially, it also means that the principal directions in which the noise is generated no longer have to be axis-aligned. I.e., prior work suffers from the inability to simultaneously optimise the alignment between the noise and the weight vectors (required to minimise the adversarial gap, as per Theorem 1), while maintaining the freedom to place the weight vectors off the axis (required for good clean performance). Our use of anisotropic noise combined with WCA encourages alignment between the weight vectors of the classifier and the eigenvectors of the covariance matrix, while allowing non-axis aligned weights, thus providing more freedom about where to place the classification decision boundaries.

Previous approaches are able to train the variance of each dimension of the isotropic noise via the use of the “reparameterization trick” (Kingma and Welling, 2014), where one samples noise from a distribution with zero mean and unit variance, then rescales the samples to get the desired variance. Because the rescaling process is differentiable, this allows one to learn variance jointly with the other network parameters with backpropagation. In order to sample anisotropic noise, one can instead sample a vector of zero mean unit variance and multiply this vector by a lower triangular matrix, L. This lower triangular matrix is related to the covariance matrix as

$$\Sigma = L \cdot L^T. \tag{3.9}$$

This guarantees that the covariance matrix remains positive semi-definite after each gradient update.

3.3 Experiments

In this section, we present the experiments that demonstrate the efficacy of our model and verify our theoretical analysis.

3.3.1 Experimental Setup

Datasets For comparison against the state-of-the-art and for our ablation study we use four benchmarks: CIFAR-10/100 (Krizhevsky and Hinton, 2009), SVHN (Netzer et al., 2011) and Fashion-MNIST (Xiao et al., 2017). CIFAR-10 and CIFAR-100 contain 60K 32x32 color images, 50K for training and 10K for testing, evenly spread across 10 and 100 classes respectively. SVHN can be considered a more challenging version of MNIST (LeCun et al., 2010); it contains almost 100K 32x32 color images of digits (0-9) collected from Google's Street View imagery, with roughly 73K for training and 26K for testing. Fashion-MNIST is a collection of 70K 28x28 grayscale images of clothing, 60K for training and 10K for testing, also spread across 10 classes.

Models For all benchmarks except F-MNIST we use a ResNet-18 (He et al., 2016) backbone, while for F-MNIST, being a relatively simpler dataset, we use LeNet++ (Wen et al., 2016). After the backbone we add a penultimate layer for dimensionality reduction; this enables us to always train a reasonably-sized covariance matrix, regardless of the original dimensionality of the backbone[1]. The only restriction for the dimensionality of the penultimate layer is that it needs to be a number greater than or equal to the number of classes in the task, so as to allow the covariance matrix to align with at least one classifier vector. The two hyperparameters of note across all of our experiments are the learning rate and ℓ^2 penalty (i.e., weight decay), the exact values of which are provided in the supplementary material.

Attacks We evaluate our method using three white-box adversaries: FGSM (Goodfellow et al., 2015), PGD (Madry et al., 2018) and C&W (Carlini and Wagner, 2017), and one black-box attack: the One-Pixel attack (Su et al., 2019).

We parameterise the attacks following the literature (He et al., 2019; Jeddi et al., 2020). More specifically, FGSM and PGD are set with an attack strength of $\epsilon = 8/255$ for CIFAR-10, CIFAR-100 and SVHN, and $\epsilon = 0.3$ for F-MNIST. PGD has a step size of $\alpha = \epsilon/10$ and number of steps $k = 10$ for all benchmarks. C&W has a learning rate of $\alpha = 5 \cdot 10^{-4}$, number of iterations $k = 1000$, initial constant $c = 10^{-3}$ and maximum binary steps $b_{\text{max}} = 9$.

[1] 32x32 for the benchmarks with 10 classes, 256x256 for the benchmarks with 100 classes.

To set the hyperparameters of the One-Pixel attack, we tried to replicate the exact experimental setup described in the supplementary material of Jeddi et al. (2020) for attack strengths of 1, 2 and 3 pixels. We followed their setup with population size $\mathrm{N} = 400$ and maximum number of iterations $\mathrm{k}_{\mathrm{max}} = 75$. However, we noticed that the more pixels we added to our attack the weaker it became, which is counter-intuitive. We attribute that to the small number of iterations; every added pixel substantially increases the search space of the differential evolution algorithm, and 75 iterations are no longer enough to converge when the number of pixels is 2 and 3. Therefore we maintain a population size of $\mathrm{N} = 400$, but increase the number of iterations to $\mathrm{k}_{\mathrm{max}} = 1000$. We further clarify that for the differential evolution algorithm we use a crossover probability of $\mathrm{r} = 0.7$, a mutation constant of $\mathrm{m} = 0.5$, and the following criterion for convergence:

$$\sqrt{\text{Var}(\mathcal{E}(X))} \leq \left| \frac{1}{100N} \sum_{x \in X} \mathcal{E}(x) \right|, \tag{3.10}$$

where X denotes the population, $\mathcal{E}(X)$ the energy of the population and $\mathcal{E}(x)$ the energy of a single sample.

Expectation over Transformation As a consequence of the noise injected by SNNs, the gradients used by white-box adversaries are stochastic (Athalye et al., 2018a); hence the true gradients cannot be correctly estimated for attacks that use only one sample to compute the perturbation. To avoid this issue, we apply Expectation over Transformation (EoT) following Athalye et al. (2018a). When generating an attack, we compute gradients of multiple forward passes using Monte-Carlo sampling and perturb the inputs using the averaged gradient at each update. We empirically found that a reliable number of MC samples is 50 (as we observed performance begins to saturate from around 35 and converges at 40); thus, we use 50 across all experiments.

3.3.2 Comparison to Prior Stochastic defences

Competitors

We compare the performance of WCA-Net to recent state-of-the-art stochastic defences to verify its efficacy. **AdvBNN** (Liu et al., 2019b): adversarially trains a BNN for defence. **PNI** (He et al., 2019): learns an “intensity” parameter to control the variance of the SNN. **Learn2Perturb (L2P)** (Jeddi et al., 2020): an improvement upon PNI that features a learnable isotropic perturbation injection module. Furthermore, we have included partial comparisons against **SESNN** (Yu et al., 2021) and **IAAT** (Xie et al., 2019). All experiments use a ResNet-18 backbone and are conducted on CIFAR-10 for fair comparison.

White-box Attacks

We first compare our proposed WCA-Net to the existing state-of-the-art stochastic adversarial defences in the white-box attack setting. From the results in Table

Table 3.1: Comparison of state-of-the-art SNNs for FGSM and PGD attacks on CIFAR-10 and CIFAR-100 with a ResNet-18 backbone. Performance of competitor methods was taken from the original published papers.

	CIFAR-10			CIFAR-100		
Method	Clean	FGSM	PGD	Clean	FGSM	PGD
AdvBNN (Liu et al., 2019b)	82.2	60.0	53.6	$\sim$ 58.0	$\sim$ 30.0	$\sim$ 27.0
PNI (He et al., 2019)	87.2	58.1	49.4	$\sim$ 61.0	$\sim$ 27.0	$\sim$ 22.0
L2P (Jeddi et al., 2020)	85.3	62.4	56.1	$\sim$ 50.0	$\sim$ 30.0	$\sim$ 26.0
SESNN (Yu et al., 2021)	92.3	74.3	-	-	-	-
IAAT (Xie et al., 2019)	-	-	-	63.9	-	18.5
WCA-Net	**93.2**	**77.6**	**71.4**	**70.1**	**51.5**	**42.7**

Table 3.2: Comparison of state-of-the-art SNNs for white box C&W attack and black box n-Pixel attack on CIFAR-10 with a ResNet-18 backbone. Performance of competitor methods was taken from the original published papers.

	Attack Strength	AdvBNN	PNI	L2P	WCA-Net
	Clean	82.2	87.2	85.3	**93.2**
C&W	$\kappa = 0.1$	78.1	66.1	84.0	**89.4**
	$\kappa = 1$	65.1	34.0	76.4	**78.4**
	$\kappa = 2$	49.1	16.0	66.5	**71.9**
	$\kappa = 5$	16.0	0.08	34.8	**55.0**
n-Pixel	1 pixel	68.6	50.9	64.5	**90.8**
	2 pixels	64.6	39.0	60.1	**85.5**
	3 pixels	59.7	35.4	53.9	**81.2**
	5 pixels	-	-	-	**64.3**

3.1, we can see that our WCA-Net shows noticeable improvement of $\sim 15\%$ over the strongest competitor, L2P. Moreover, we find that our method does not sacrifice its performance on clean data to afford such strong robustness.

An important aspect of WCA that needs to be assessed is its potential to scale with the number of classes. For this reason we conduct experiments on CIFAR-100, comparing against our previously mentioned competitors, plus IAAT (Xie et al., 2019), all of which use a ResNet-18 backbone in their architectures. From Table 3.1 we can see that the adversarial robustness of WCA-Net outperforms the other methods.

We also present the evaluation of our method against the C&W attack in Table 3.2. Here, the confidence level κ indicates the attack strength. Our WCA-Net achieves the best performance, with the accuracy degrading gracefully as the confidence increases.

Table 3.3: Comparison of WCA-Net to recent state-of-the-art, both stochastic and non-stochastic, on CIFAR-10. All competitors evaluate their models on the untargeted PGD attack, with attack strength $\epsilon = 8/255$, and number of iterations $k \in \{7, 10, 20\}$. Some results are extracted from He et al. (2019). Performance of competitor methods was taken from the original published papers. AT: Use of adversarial training.

Defence	Architecture	AT	Clean	PGD
RSE (Liu et al., 2018)	ResNext	✗	87.5	40.0
DP (Lécuyer et al., 2019)	28-10 Wide ResNet	✗	87.0	25.0
TRADES (Zhang et al., 2019a)	ResNet-18	✓	84.9	56.6
PCL (Mustafa et al., 2019)	ResNet-110	✓	91.9	46.7
PNI (He et al., 2019)	ResNet-20 (4x)	✓	87.7	49.1
AdvBNN (Liu et al., 2019b)	VGG-16	✓	77.2	54.6
L2P (Jeddi et al., 2020)	ResNet-18	✓	85.3	56.3
MART (Wang et al., 2020a)	ResNet-18	✓	83.0	55.5
BPFC (Addepalli et al., 2020)	ResNet-18	✗	82.4	41.7
RLFLAT (Song et al., 2020)	32-10 Wide ResNet	✓	82.7	58.7
MI (Pang et al., 2020)	ResNet-50	✗	84.2	64.5
SADS (S. and Babu, 2020)	28-10 Wide ResNet	✓	82.0	45.6
WCA-Net	ResNet-18	✗	**93.2**	**71.4**

Black-box Attacks

To further verify the robustness of our WCA-Net, we conduct experiments on a black-box attack, the One-Pixel attack (Su et al., 2019). This attack is gradient-free and relies on evolutionary optimization. Its attack strength is controlled by the number of pixels it compromises. We follow Jeddi et al. (2020) and consider pixel numbers in $\{1, 2, 3\}$. Additionally, we report results for a stronger 5-pixel attack. From Table 3.2, we can see that our method demonstrates the strongest robustness in all cases, showing $\sim 13\%$ to $\sim 22\%$ improvement over the best competitor AdvBNN. Importantly, these results show that the robustness of our method does not rely on stochastic gradients.

Stronger Attacks

In addition, we evaluate WCA-Net against two stronger attacks that are common among recent adversarial robustness literature, but are not mentioned in the stochastic defences we outline as direct competitors. These are: (i) PGD$_{100}$; a stronger variant of PGD with 100 random restarts, and (ii) the Square Attack (Andriushchenko et al., 2020); a black-box attack that compromises the attacked image in small localised square-shaped updates. We present the results of our evaluation in Table 3.4.

Table 3.4: Evaluation of WCA-Net with a ResNet-18 backbone on CIFAR-10, against the white-box PGD$_{100}$ and black-box Square Attack, for different values of attack strength ϵ.

	$\epsilon/255$	Clean	1	2	4	8	16	32	64	128
PGD$_{100}$	No Defence	93.3	45.3	14.6	0	0	0	0	0	0
	WCA-Net	93.2	73.2	72.2	72.1	71.2	69.7	56.4	28.2	10.5
Square	No Defence	93.3	32.9	31.7	12.4	6.0	1.2	0	0	0
	WCA-Net	93.2	51.7	51.7	50.4	49.0	48.8	44.3	36.9	28.6

Table 3.5: Ablation study for FGSM and PGD attacks on CIFAR-10, CIFAR-100, SVHN and F-MNIST. For CIFAR-10, CIFAR-100 and SVHN we use a ResNet-18, and for F-MNIST a LeNet++ backbone. (I): Isotropic, (A): Anisotropic.

	CIFAR-10			CIFAR-100		
Model	Clean	FGSM	PGD	Clean	FGSM	PGD
No Defence	93.3	14.9	3.9	72.2	12.3	1.2
WCA-Net (I)	93.1	60.7	55.9	70.1	27.5	21.8
WCA-Net (A)	93.2	77.6	71.4	70.1	51.5	42.7
	SVHN			F-MNIST		
Model	Clean	FGSM	PGD	Clean	FGSM	PGD
No Defence	93.4	55.6	23.5	90.8	26.4	12.0
WCA-Net (I)	93.4	45.0	40.1	90.1	63.5	37.2
WCA-Net (A)	93.4	87.6	85.7	90.1	65.2	48.5

3.3.3 Comparison to State of the Art

Direct comparison to a wider range of competitors is difficult due to the variety of backbones and settings used. However, such a comparison is useful, because it places our work within the wider context of adversarial robustness literature. Table 3.3 provides comparison to recent state of the art stochastic and non-stochastic defences. We can see that WCA-Net achieves excellent performance including comparing to methods that use bigger backbones and make the stronger assumption of adversarial training. Upon inspection of the results, we make an interesting observation: Regularising a neural network's parameters for robustness via adversarial training, or otherwise, takes a toll on its clean performance. WCA is a strong method in that regard, because the clean performance of our WCA-trained architecture matches the clean performance of the original, non-robust backbone.

Table 3.6: Control experiments on CIFAR-10 for further analysis. See Sec. 3.3.4. AT: Training purely with adversarial examples. CT+AT: Training with a mix of clean and adversarial examples.

Experiment	Clean	FGSM	PGD
No defence	93.3	14.9	3.9
WCA-Net (Penalty regulariser)	93.2	77.6	71.4
WCA-Net (Constraint regulariser)	92.2	62.9	53.2
E1: Test without EoT	93.2	82.9	75.1
E2: Average multiple noise samples	93.2	70.3	68.8
E3: Noise trained independently	93.1	45.0	41.6
WCA-Net: AT	88.1	75.4	70.4
WCA-Net: CT+AT	90.0	75.6	70.7

Table 3.7: Comparison between the undefended ResNet-18 baseline and WCA-Net with a ResNet-18 backbone for Imagenette (high-res, 10 categories) and mini-ImageNet (large-scale, 100 categories) under PGD attack.

	Imagenette			mini-ImageNet		
Model	Clean	FGSM	PGD	Clean	FGSM	PGD
No defence	75.5	8.4	0	51.9	5.0	0
WCA-Net	74.2	59.3	48.7	51.3	41.6	30.4

3.3.4 Further Analysis

Ablation Study

We perform an ablation study on four benchmarks, CIFAR-10, CIFAR-100, SVHN and F-MNIST, to investigate the contribution of anisotropic noise, as shown in Table 3.5. For each benchmark, we evaluate a "clean" baseline architecture, consisting only of the backbone and the classification layer. We then evaluate a variant of WCA-Net with isotropic, and one with anisotropic noise. We observe that our anisotropic noise provides consistent benefit to adversarial robustness.

Another important observation is that there is no trade-off between the robust and clean performance of our models; both the isotropic and anisotropic variants of WCA-Net maintain the clean performance of the baseline defenceless model.

All the FGSM and PGD attacks in Table 3.5 use attack strength $\epsilon = 8/255$. For completeness, we report the performance of all the variants above against FGSM and PGD with various attack strengths $\epsilon = 2^n$, $n \in \{0...7\}$ on CIFAR-10 shown in Figure 3.1. From these results, we can see the overall trend here is consistent with the observations in Table 3.5. Also, we can see that the performance of our variants degrades more gracefully than the defenceless baseline.

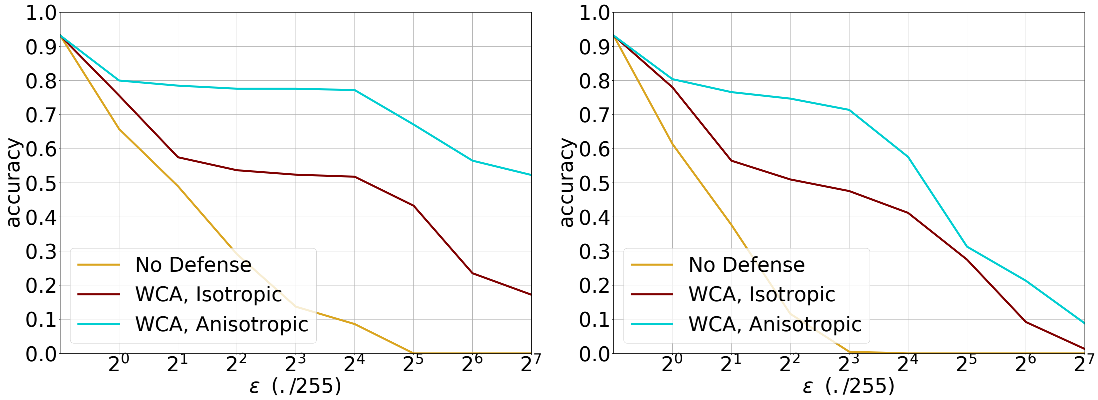


Figure 3.1: Evaluation of our model variants (see Table 3.5) for different attack strengths $\epsilon = 2^n, \; n \in \{0...7\}$, specifically for the FGSM (left) and PGD (right) attacks on CIFAR-10.

Large-scale, high-resolution We are further interested to show that our WCA method can handle high-resolution images and more challenging datasets. For that purpose, we extend our evaluation to two additional benchmarks that are not considered by related work: (i) Imagenette (Howard, 2019), a subset of ImageNet with 10 classes and full-resolution images, and (ii) mini-ImageNet (Vinyals et al., 2016), a large subset of ImageNet with 100 classes and 84x84 images, designed to be more challenging than CIFAR-100. The results presented in Table 3.7 demonstrate that our method generalises quite well to both high-resolution images as well as more challenging datasets.

Norm-constrained architecture As explained in Section 3.2.1, we control the magnitude of the weights in our architecture by means of ℓ^2 regularization. An alternative option to achieve the same effect is to apply norm constraints to the classification vectors w_i and covariance matrix Σ. A detailed explanation of how we apply these norm constraints is given in Appendix A. In Table 3.6, we report results of a WCA-Net variant with a norm-constrained regulariser. Constraint-based regularization still provides good robustness, but is weaker than the ℓ^2 penalty-based variant.

E1: Importance of EoT To show the impact of EoT, we also evaluate the test performance without it. Table 3.6 shows that the test performance increases without using EoT. This makes sense as critiqued in Athalye et al. (2018a); one gradient sample is not enough to construct an effective attack.

E2: Average multiple noise samples at test time Our model's forward pass performs the following: (i) Extract features from the penultimate layer of the backbone, (ii) inject additive noise, and (iii) compute the logits. By default we draw a single noise sample as suggested by our theory. In this experiment, we sample from the distribution multiple times and average the final logits. The more noise samples we average, the more we expect the additive noise to lose its regularization effect. The experimental results in Table 3.6 confirm that using more $(n = 10)$ samples degrades performance.

E3: Train noise and model independently In this experiment, we first train the model without injecting any noise. Then, keeping the model parameters frozen we train the noise independently. In Table 3.6 we can see that this variant achieves an elementary level of robustness that is better than the defenceless baseline shown in Table 3.5, however, not as strong as the isotropic baseline. As mentioned in Section 3.2.1, a key insight of Theorem 1 is that the noise and weights should co-adapt. As expected, keeping the weight vectors w_i frozen, overall limits the ways the WCA term (see Equation 3.7) can inflate, thus never realizing its full potential.

Adversarial training Our proposed method achieves adversarial robustness by only requiring clean data for training. To show this, we adversarially train our anisotropic WCA-Net in two settings: (i) purely with adversarial examples and (ii) with a mix of clean and adversarial examples. We train with a PGD attack with $\epsilon = 8/255$ and $k = 10$. Our results in Table 3.6 show that incorporating AT harms our performance on clean data as expected (Goodfellow et al., 2015); while providing no consistent benefit for adversarial defence.

3.3.5 Inspection of Gradient Obfuscation

Athalye et al. (2018a) proposed a set of criteria to inspect whether a stochastic defence method relies on obfuscated gradients. Following He et al. (2019), we summarise these criteria as a checklist. If any item in this checklist holds true, the stochastic defence is deemed unreliable. The following analysis verifies that our model's strong robustness is not caused by gradient obfuscation.

Criterion 1: One-step attacks perform better than iterative attacks. Given that PGD is an iterative variant of FGSM, we use our existing evaluation to refute this criterion. From the results in Tables 3.1, 3.5 and 3.6, we can see that our WCA-Net performs consistently better against FGSM than against PGD.

Criterion 2: Black-box attacks perform better than white-box attacks. From Tables 3.1 and 3.2 we observe that FGSM and PGD outperform the 1-pixel attack. In Figure 3.1 we see the effect of increasing the attack strength on both white-box attacks, and they still outperform the stronger 2-, 3- and 5-pixel attacks.

Criterion 3: Unbounded attacks do not reach 100% success. To compare against previous work in fair terms, FGSM and PGD are parameterised following He et al. (2019). However, for this check we deliberately increase the attack

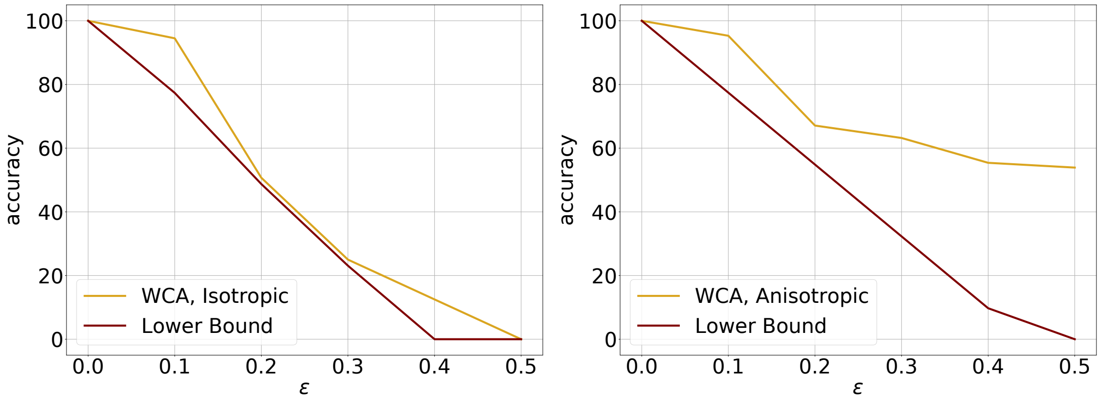


Figure 3.2: Evaluating our bound. Plots of the test set accuracy of SVMs trained on the zero and one digits found in MNIST. We report the performance of models trained with isotropic (left) and anisotropic (right) noise, and the worst-case performance according to Theorem 1. The anisotropic model provides a more robust bound than the isotropic model as well as better empirical performance. Best viewed in color.

strength of PGD to $\epsilon = 255/255$ and number of iterations to $k = 20$. We evaluate all of our models against this attack, and they achieve an accuracy of 0%.

Criterion 4: Random sampling finds adversarial examples. To assess this, we hand-pick 100 CIFAR-10 test images that our model successfully classifies during standard testing (100% accuracy), but misclassifies under FGSM with $\epsilon = 8/255$ (0% accuracy). For each of these test images, we randomly sample 1,000 perturbed images within the same ϵ-ball, and replace the original image if any of the samples result in misclassification. We then evaluate our model on these 100 images to get a performance of 98%.

Criterion 5: Increasing the distortion bound doesn't increase success. Our ϵ-ablation in Figure 3.1 shows that increasing the distortion bound increases the attack's success.

3.3.6 Empirical Evaluation of Theorem 1

To evaluate the tightness of our bound presented in Theorem 1, we train linear Support Vector Machines (SVM) on the zero and one digits found in the MNIST dataset. Using a linear model allows us to compute the numerator using the technique of Gouk and Hospedales (2020),

$$\Delta_{\infty}^{\tilde{h}}(x, \epsilon) = \epsilon \|w\|_1, \tag{3.11}$$

where w is the weight vector of the SVM. We reduce the images to 32 dimensions using principal components analysis, and apply learned isotropic and anisotropic

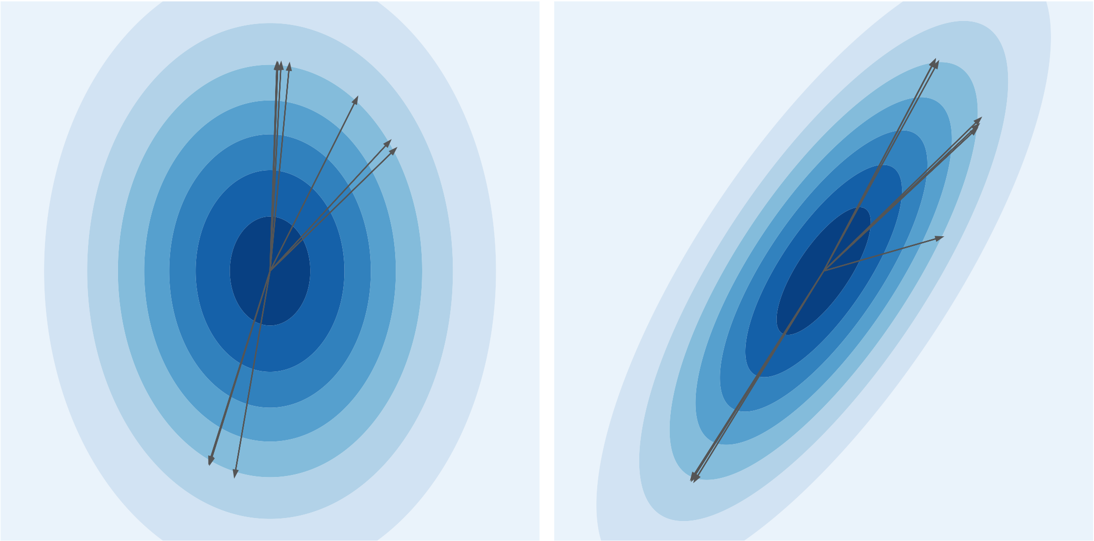


Figure 3.3: Visualisation of our WCA models on F-MNIST with a 2-dimensional bottleneck. Contours and arrows indicate noise covariance Σ and weights w_i. Left: WCA-Net with isotropic noise. Right: WCA-Net with anisotropic noise. Evidently, our WCA-Net with anisotropic noise allows covariance to be aligned with off-axis weights.

noise to these reduced features before classification with the SVM. The covariance matrix and SVM weights are found by minimizing the hinge loss plus the WCA loss term using gradient descent. Results of attacking these models with PGD, and the lower bound on performance as computed by Theorem 1, are given in Figure 3.2. From these plots we can see: (i) the bound is not violated at any point, corroborating our analysis; (ii) as the strength of the adversarial attack is increased, the bound remains non-vacuous for reasonable (i.e., imperceptible) values of the attack strength; and (iii) the model with the anisotropic noise is more adversarially robust than the model with the isotropic noise. This finding is particularly interesting because in the linear model regime PGD attacks are able to find globally optimal adversarial examples.

3.3.7 Empirical Observations about WCA

In Figure 3.3 we use a biviriate Gaussian to show the effect of our regularization methods, by plotting the contours of the distribution against the weight vectors of the classification layer. These visualizations are obtained by training our WCA-Net isotropic (left) and anisotropic (right) variants with a LeNet++ backbone on F-MNIST, with a 2-dimensional bottleneck and 2x2 covariance matrix. The X and Y axes in this plot construct a 2-dimensional vector space with real values, on which we project 10 2-dimensional weight vectors, with origin at (0, 0), each corresponding to a class from F-MNIST. In the background, we plot the contours of the noise distribution. This type of figure can serve as a qualitative diagnostic that answers the question: "Is the performance boost of our model attributed to weight-covariance alignment?”. The resulting figure shows the following:

- First, in the left of Figure 3.3, we can see that the learned noise is axis-aligned since the injected noise is isotropic. Further, we can see that the weight vectors are near-axis-aligned, as WCA pushes them to align with the learned noise.
- Then, in the right Figure, due to the combination of anisotropic noise and WCA, our model has weight-aligned noise, and the weights are free to be non-axis-aligned. Overall, we observe better alignment between the learned weight vectors and the eigenvectors of the covariance matrix in our proposed anisotropic WCA-Net.

3.4 Discussion

This Chapter describes our contribution of the first stochastic adversarial defence that features fully-trained anisotropic Gaussian noise, and its robustness does not rely on adversarial training. Our training algorithm is considered to be “hyperparameter-free”, as it involves the same hyperparameters as a common training algorithm, i.e., learning rate and weight decay. We provide both theoretical support for the core ideas behind it, and experimental evidence of its excelling performance. We extensively evaluate WCA-Net on a variety of white-box and black-box attacks, and further show that its high performance is not a result of stochastic (obfuscated) gradients. Thus, we consider the proposed model to push the boundary of adversarial robustness.

3.4.1 Insights About Weight-Covariance Alignment

As is the theme of this thesis, there are opportunities and risks associated with the use of WCA. In this section, we present some key insights that can guide the scientific community into extracting the maximum benefit out of our research.

Maximisation of the WCA term should be approached with care

It is important to stress that a quick look at Equation 3.6 might give the reader a false impression about the simplicity of the WCA objective. While it can be easily translated into a regularisation term to accompany the classification loss, how to maximise it correctly with gradient descent is far from obvious.

The main caveat in this line of work is the tendency of gradient descent-based optimisers to maximise $w^T \Sigma w$ by blindly inflating the values of w and Σ. Not only can this lead to overfitting, but even worse, it can lead to numeric overflow – especially after softmax is applied to the activations of the classification layer. It is important to force gradient descent to prioritise weight-covariance alignment over inflation. In this work we propose two methods for achieving this: applying ℓ^2 regularisation (see Section 3.2.1) and constraining the norm of the weight vectors (see Appendix A).

The use of anisotropic noise is worth the performance/complexity trade-off

We hypothesise that one of the reasons why we are the first in the literature to propose an SNN architecture trained with anisotropic noise, is the substantial increase in training-time complexity for little benefit. In our work, this is not true; the benefit is substantial because we explicitly care about alignment. By simultaneously giving the classification weight vectors the freedom to align with the eigenvectors of the covariance matrix, and the noise covariance the freedom to align with the classification weight vectors, we can reliably achieve both WCA and model convergence between training executions.

Furthermore, the use of anisotropic noise breaks the strong assumption that learned features are independent. Intuitively, this assumption is unlikely to hold true when learning directly from pixels, but in practical terms, the simplicity of incorporating isotropic noise may outweigh this concern. However, when it comes to adversarial robustness this concern is more severe, as pixel perturbations of most gradient-based and gradient-free adversaries do not treat pixels independently. Specifically, the reason why deep learning models are vulnerable against adversarial attacks in the first place, as discovered by Szegedy et al. (2014), is that a lot of small pixel-level perturbations can synergise into causing large feature-level changes in the output.

In our work, we further provide a framework for training anisotropic noise in a more straightforward way than learning the entire Σ matrix. We recommend learning a lower-triangular matrix, L, whereby it is easier to impose constraints related to gradient updates, and guarantees that Σ remains positive and semi-definite, as per Equation 3.9.

3.4.2 Limitations

The stochastic defences outlined in this paper, including WCA, owe part of their robustness to creating a noisy loss landscape that gradient-based adversaries find difficult to ascend. Manipulating the loss landscape in such a way is a type of gradient obfuscation that Chapter 4 explains in detail, but was not known at the time when this research was conducted. Therefore, any adversarial attacks that include loss-smoothing were not included in the experimental setting of WCA-Net, as was true for the competing stochastic defences AdvBNN, PNI, L2P, and SESNN. It should be noted that this does not mean that the tightening of our learning-theoretic bound no longer yields robustness. In fact, the experimental results in Section 4.3.2 show that our method displays higher robustness against competing methods, even when under attack by loss-smoothing adversaries. This robustness can still be entirely attributed to WCA.

Chapter 4

Attacking Adversarial Defences by Smoothing the Loss Landscape

This chapter corresponds to the paper: P. Eustratiadis, H. Gouk, D. Li, and T. M. Hospedales. Attacking adversarial defences by smoothing the loss landscape. In *ICML Workshop on Adversarial Machine Learning*, 2022.

This chapter investigates a family of adversarial defence methods that owe part of their success to creating a noisy, discontinuous, or otherwise rugged loss landscape that adversaries find difficult to navigate. One common (but not universal) way to achieve this effect is via the use of stochastic neural networks (SNNs). We show that this way of defending is a form of gradient obfuscation, and propose a general extension to gradient-based adversaries that smooths the surface of the loss function and provides more reliable gradient estimates. We further show that the same principle can strengthen gradient-free adversaries. Our loss-smoothing method proves to be effective against both stochastic and non-stochastic adversarial defences that exhibit robustness due to this type of obfuscation, as demonstrated in our experimental analysis. Furthermore, we provide analysis of how our smoothing method interacts with Expectation over Transformation; a popular gradient-sampling method currently used to attack stochastic defences.

4.1 Introduction

In this chapter, we reveal a form of gradient obfuscation that, to the best of our knowledge, is not yet known[1]. So far, it is understood that SNNs can defend effectively against adversarial attacks because having stochastic weights reduces overfitting, with similar effect to training a non-stochastic neural network with Lipschitz regularisation (Liu et al., 2018), a property with strong theoretical links to adversarial robustness (Hein and Andriushchenko, 2017). We show that there

[1] At the time of writing, 2021-2022.

is an additional reason for their robust performance. Stochastic defences, even when averaging multiple gradient samples with EoT, tend to create a rough loss landscape that gradient-based adversaries find difficult to ascend. A second, and perhaps more interesting finding, is that this property is not exclusive to stochastic adversarial defences; there exist non-stochastic defences that have the same effect (e.g., Alfarra et al., 2021).

We show that the aforementioned property is a weakness that can be attacked by an adversary. Specifically, we propose a stochastic extension to gradient-based attacks that approximates performing the Weierstrass Transform (WT) (Zayed, 1996, Ch. 18) on the loss function in order to smooth it before computing its gradient. Interestingly, we find that the same method can be applied in a gradient-free setting to effectively circumvent the same type of obfuscation.

We support our insights experimentally by applying this novel extension to Projected Gradient Descent (PGD) (Madry et al., 2018) and other recent iterative FGSM variants (Lin et al., 2020; Wang and He, 2021) as well as Zeroth Order Optimization (ZOO) (Chen et al., 2017), in the gradient-based and gradient-free settings respectively. We demonstrate the efficacy of our loss-smoothing method against both stochastic (He et al., 2019; Jeddi et al., 2020; Yu et al., 2021; Eustratiadis et al., 2021) and non-stochastic defences (Xiao et al., 2020; Alfarra et al., 2021) that create a noisy or discontinuous loss surface, and damage their robust performance by as much as 20%. Finally, we analyse how the WT interacts with EoT when attacking stochastic defences. We show that these two methods serve different purposes and are complementary. However, unlike an attack that applies EoT, a WT-based attack is effective against both stochastic and non-stochastic defences.

4.2 Method

4.2.1 The Weierstrass Transform

The Weierstrass Transform (Zayed, 1996, Ch. 18) of a function f is defined as the convolution of f with a Gaussian kernel function k in order to obtain g, a smoothed version of f. Formally,

$$g(x) = \int_{-\infty}^{+\infty} k(x-y)\, f(y) \cdot dy, \quad k(x) = \frac{1}{\sqrt{4\pi}}\, e^{\frac{-x^2}{4}}. \tag{4.1}$$

The conventional WT is defined for functions of scalar variables and utilises a Gaussian kernel with a variance of $\sqrt{2}$. In our work we are applying it to neural networks that are functions of many variables which may need to be smoothed to different extents; therefore we relax these two conditions by using a multivariate Gaussian with a tuneable isotropic covariance matrix.

4.2.2 Using the Weierstrass Transform to Attack

Let $\mathcal{L}(h_\theta(x), c)$ be the classification loss function, x an input image belonging to class $c \in C$, and h_θ a function approximator with parameters θ. We can use Equation 4.1 to define the smoothed loss function $\tilde{\mathcal{L}}$ as

$$\tilde{\mathcal{L}}(h_\theta(x), c) = \int_{\mathbb{R}^d} k(x-y)\ \mathcal{L}(h_\theta(y), c) \cdot dy, \tag{4.2}$$

where d is the dimensionality of x. This can also be interpreted as an expectation

$$\tilde{\mathcal{L}}(h_\theta(x), c) = \mathbb{E}_\eta[\mathcal{L}(h_\theta(x+\eta), c)], \quad \eta \sim \mathcal{N}(0, \sigma^2 I). \tag{4.3}$$

The dimensionality of the integral in Equation 4.2 corresponds to the number of input pixels; so computing it directly is computationally unfeasible. However, it is possible to compute a stochastic unbiased estimate of $\tilde{\mathcal{L}}$ by using Monte-Carlo sampling,

$$\hat{\mathcal{L}}(h_\theta(x), c) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(h_\theta(X_i), c), \tag{4.4}$$

where m is the number of perturbations sampled around x and

$$X_i = x + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \sigma^2 I). \tag{4.5}$$

The error introduced by this approximation of the WT is bounded (with high confidence), as shown in the following Theorem. It can be seen that the quality of the approximation improves as the number of samples, m, is increased.

Theorem 2. *For a k-Lipschitz network, h_θ, applied to a fixed instance (x, c), and a loss function, $\mathcal{L}$, that is L-Lipschitz on the co-domain of h_θ, we have with probability at least $1-\delta$ that*

$$|\hat{\mathcal{L}}(h_\theta(x), c) - \tilde{\mathcal{L}}(h_\theta(x), c)| \leq kL\sigma\sqrt{\frac{4d\ln(1/\delta)}{m}} + \frac{2kL\ln(1/\delta)}{3m}, \tag{4.6}$$

where we assume that x is contained within the unit ball in d-dimensional Euclidean space.

The proof of Theorem 2 is provided in Appendix B.

4.2.3 A Stochastic WT Extension of Gradient-Based Attacks

Conceptually, any gradient-based adversary can be extended with the WT to smooth rugged loss landscapes and estimate gradients more reliably. Algorithm 1 describes PGD$_{\text{WT}}$, our proposed method that is an extension of PGD. In addition to the standard hyperparameters of PGD, i.e., the number of iterations k, step size α, and attack strength ϵ, we add m as the number of images sampled around x, and the standard deviation σ of the zero-mean normal distribution from which the images are sampled.

Algorithm 1: PGD$_{\text{WT}}$

```
Data: $x$, $c$
Model: $h_\theta$
Input: $k$, $m$, $n$, $\alpha$, $\epsilon$, $\sigma$
Output: $\tilde{x}$
$\tilde{x} \longleftarrow x + z, \quad z \sim \mathcal{U}(-\epsilon, \epsilon)$
for $k$ iterations do
    $\tilde{X} \longleftarrow$ sample $m$ points around $\tilde{x}$  [Eq. 4.5]
    if defence is stochastic then
        $\omega \leftarrow \frac{1}{mn} \sum_{i=0}^{m} \sum_{j=0}^{n} \nabla_x \mathcal{L}(h_\theta^j(\tilde{X}_i), c)$  [Eq. 4.8]
    else
        $\omega \leftarrow \frac{1}{m} \sum_{i=0}^{m} \nabla_x \mathcal{L}(h_\theta(\tilde{X}_i), c)$  [Eq. 4.7]
    end
    $\tilde{x} \longleftarrow \tilde{x} + \alpha \, \mathrm{sgn}(\omega)$
    project $\tilde{x}$ to $\ell_p$-ball of $\epsilon$
end
```

The main idea is that, given enough samples in close proximity to x, we can compute the true slope of the loss function as the average slope of the surface where these samples lie. Therefore, within the context of PGD$_{\text{WT}}$, we define the true gradient ω as

$$\omega = \frac{1}{m} \sum_{i=0}^{m} \nabla_x \mathcal{L}(h_\theta(\tilde{X}_i), c), \tag{4.7}$$

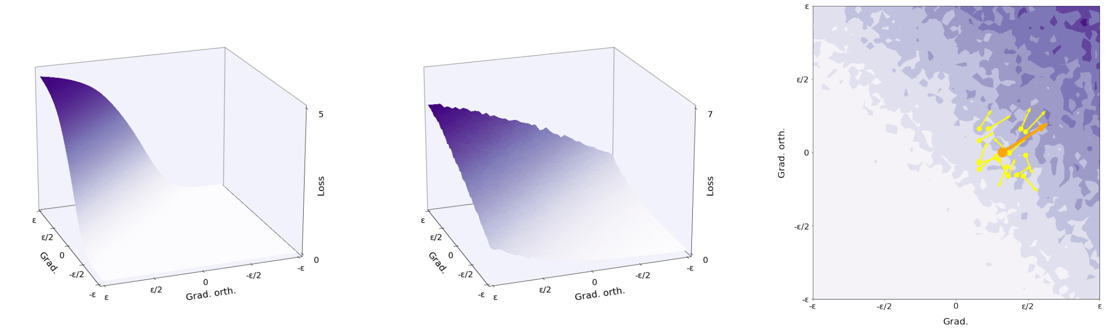
where $\tilde{X}$ denotes the set of images sampled around the perturbed image $\tilde{x}$, as per Equation 4.5.

Figure 4.1c illustrates the concept of this attack. While the gradient at a particular image x and samples nearby are individually noisy (random small yellow arrows), their aggregate direction (large orange arrow) ascends the loss surface.

Generalisation Properties Note that the WT only affects part of a gradient-based attack that performs the gradient computation. In this chapter we choose to illustrate the WT extension on PGD as a proof of concept, due to its convenient mathematical formulation as well as its efficacy as an attack. However, Equation 4.7 can effectively replace the gradient computation step in any gradient-based adversary (e.g., Goodfellow et al., 2015; Lin et al., 2020; Wang and He, 2021).

Integration with EoT

When we use Equation 4.4 and 4.5 to smooth the loss landscape of a stochastic defence, the gradient w.r.t. the input x, $\nabla_x \mathcal{L}(h_\theta(\tilde{X}), c)$, remains stochastic (Athalye et al., 2018a). It is therefore sensible to apply EoT (Athalye et al., 2018b)



(a) RN-18 (no defence). (b) PNI + PGD$_{\text{WT}}$ (c) PNI (top-down)

Figure 4.1: Illustration of the intuition behind our WT attack. **Left:** The smooth surface of an undefended ResNet-18. **Middle:** When under attack by PGD$_{\text{WT}}$, PNI's original noisy loss landscape (see Figure 2.2a) is smoothed to better approximate one of an undefended network e.g., left figure. Refer to Figure B.3 in Appendix B for the smoothed surfaces of all other defences. **Right:** Top-down view of Figure 2.2a. The loss landscape around x (dark orange point) is noisy, and the adversary cannot find a reliable direction to follow. To overcome this, it samples m images around x (yellow points) and follows the average gradient obtained at each of those points.

on the sampled $\tilde{X}$, and average over the output distribution of h_θ. Incorporating Equation 2.6 into Equation 4.7 we get

$$\omega = \frac{1}{mn}\sum_{i=0}^{m}\sum_{j=0}^{n}\nabla_x\mathcal{L}(h_\theta^j(\tilde{X}_i), c). \tag{4.8}$$

A thorough empirical analysis of how the WT interacts with EoT is presented in Section 4.3.3, along with an ablation study for each individual component.

4.2.4 A Stochastic WT Extension of Gradient-Free Attacks

Although we primarily consider the WT to be an extension of gradient-based attacks, its potential impact when applied to gradient-free attacks cannot be ignored. In this section, we demonstrate WT's generality by integrating it with ZOO (Chen et al., 2017), a black-box adversary that uses gradient approximation instead of surrogate models (Papernot et al., 2016; Chen et al., 2017; Papernot et al., 2017b), assuming access only to the per-class posterior $p\big(h(x)\big)$.

Given an input image x and a pixel coordinate ρ, ZOO iteratively constructs a perturbation δ on x_ρ as

$$\delta(x, c) = \begin{cases} -\alpha\hat{g}_\rho(x, c) & \hat{h}_\rho \leq 0 \\ -\alpha\frac{\hat{g}_\rho(x,c)}{\hat{h}_\rho(x,c)} & \text{otherwise} \end{cases}, \tag{4.9}$$

where α denotes the learning rate. $\hat{g}_i$ and $\hat{h}_i$ are the first- and second-order

Algorithm 2: ZOO$_{\text{WT}}$ (Newton's Coordinate Descent)

```
Data: $x^d$, $c$
Model: $h$
Input: $k$, $m$, $n$, $\alpha$, $\epsilon$, $\sigma$
Output: $\tilde{x}$
for $k$ iterations do
    Randomly pick coordinates $\vec{\rho} \in \{1, \dots, d\}$
    $\tilde{X} \longleftarrow$ sample $m$ points around $\tilde{x}$  [Eq. 4.5]
    if defence is stochastic then
        $\delta^* \leftarrow \frac{1}{mn} \sum_{i=0}^{m} \sum_{j=0}^{n} \delta_j(X_i, c)$ [Eq. 4.12]
    else
        $\delta^* \leftarrow \frac{1}{m} \sum_{i=0}^{m} \delta(X_i, c)$ [Eq. 4.11]
    end
    $\tilde{x} \longleftarrow \tilde{x} + \delta^*$
    project $\tilde{x}$ to $\ell_p$-ball of $\epsilon$
end
```

approximate gradients of a hinge-like loss function

$$f(x, c_0) = \max\{\log h(x)_{c_0} - \max_{c \neq c_0} \log h(x)_c, -\kappa\} , \tag{4.10}$$

where $\kappa \geq 0$. Algorithm 2 details ZOO$_{\text{WT}}$. Note that the principle behind the WT extension remains the same as in the white-box setting. Adapting Equation 4.7 and 4.8 with ZOO's gradient approximation (Equation 4.9) we respectively get

$$\delta^* = \frac{1}{m} \sum_{i=0}^{m} \delta(X_i, c) , \tag{4.11}$$

and for stochastic defences

$$\delta^* = \frac{1}{mn} \sum_{i=0}^{m} \sum_{j=0}^{n} \delta_j(X_i, c) . \tag{4.12}$$

As ZOO estimates gradients with finite difference it is susceptible to being mislead by a rough loss surface (Fig. 2.2). Smoothing the loss estimates at each point improves the quality of approximate gradient estimation for the ZOO attacker.

4.2.5 Visualising the Loss Landscapes

In this section, we describe a diagnostic method that we use to visually identify whether an adversarial defence produces a noisy loss landscape, and to generate the visualisations in Fig. 2.2 and B.3.

Given an unperturbed input image, x, that the target model h_θ classifies correctly as class c, we compute the gradient of the loss w.r.t. x as $g_1 = \nabla_x \mathcal{L}(h_\theta(x, c))$.

Table 4.1: Robust accuracy % of PGD and PGD$_{\text{WT}}$ attacks on CIFAR. All defences use a RN-18 backbone.

	CIFAR-10				CIFAR-100			
Method	PGD$_{10}$	PGD$_{\text{WT10}}$	PGD$_{100}$	PGD$_{\text{WT100}}$	PGD$_{10}$	PGD$_{\text{WT10}}$	PGD$_{100}$	PGD$_{\text{WT100}}$
PNI	49.4	34.8 (-14.6)	31.4	13.7 (-17.7)	22.2	17.9 (-4.3)	10.1	9.4 (-0.7)
L2P	56.1	47.2 (-8.9)	20.5	18.2 (-2.3)	26.1	11.5 (-14.6)	18.4	10.3 (-8.1)
SE-SNN	39.8	21.3 (-18.5)	13.9	12.5 (-1.4)	18.6	8.0 (-10.6)	15.9	5.9 (-10.0)
WCA-Net	61.7	53.3 (-8.4)	58.6	37.6 (-21.0)	41.7	27.4 (-14.3)	39.0	10.8 (-28.2)
AA	63.2	43.9 (-19.3)	43.6	25.9 (-17.7)	47.9	29.6 (-18.3)	43.6	21.2 (-22.4)
k-WTA	58.0	33.1 (-24.9)	48.2	30.7 (-17.5)	44.3	24.1 (-20.2)	37.5	15.3 (-22.2)

Table 4.2: Robust accuracy % of PGD and PGD$_{\text{WT}}$ attacks on CIFAR-100 and Imagenette (full-resolution). All defences use a WRN-34-10 backbone.

	CIFAR-100				Imagenette			
Method	PGD$_{10}$	PGD$_{\text{WT10}}$	PGD$_{100}$	PGD$_{\text{WT100}}$	PGD$_{10}$	PGD$_{\text{WT10}}$	PGD$_{100}$	PGD$_{\text{WT100}}$
PNI	51.6	32.5 (-19.1)	48.4	31.3 (-17.1)	51.8	39.6 (-12.2)	42.3	24.3 (-18.0)
L2P	45.3	32.4 (-12.9)	40.0	29.5 (-10.5)	63.4	46.9 (-16.5)	42.4	23.2 (-19.2)
SE-SNN	44.6	34.9 (-9.7)	46.0	31.0 (-15.0)	47.2	22.9 (-24.3)	41.1	21.7 (-19.4)
WCA-Net	63.6	54.5 (-9.1)	56.7	44.5 (-12.2)	67.5	51.0 (-16.5)	50.3	35.6 (-14.7)
AA	76.1	59.2 (-16.9)	62.4	54.0 (-8.4)	69.3	44.8 (-24.5)	57.1	39.4 (-17.7)
k-WTA	60.2	46.1 (-14.1)	51.3	34.4 (-16.9)	55.7	33.6 (-22.1)	52.0	28.3 (-23.7)

We then arbitrarily choose a dimension g_2, such that $g_1 \perp g_2$. Finally, we create evenly-spaced query images (and potential adversarial examples) $\tilde{x}_i$ in the ϵ-ball of x as

$$\tilde{x}_i = x + \epsilon_1 \text{sgn}(g_1) + \epsilon_2 \text{sgn}(g_2) \ , \tag{4.13}$$

where $\epsilon_1, \epsilon_2 \in [-\frac{8}{255}, \frac{8}{255}]$, and project their calculated loss values $\mathcal{L}(h_\theta(\tilde{x}_i, c))$ to the g_1 and g_2 axes. The intuitive explanation of the resulting 3d plot is as follows: The X and Y axes have an origin of (0, 0), that corresponds to the unperturbed input, x. Every point on the 2-dimensional plane these axes form, represents an additive input perturbation on x, to the direction opposite to the gradient of $\mathcal{L}$ w.r.t. x, indicated by $\text{sgn}(g_1)$ and $\text{sgn}(g_2)$ respectively, in Equation 4.13. The magnitude of the perturbation is a maximum of $\frac{8}{255}$ in either direction. Finally, the Z axis corresponds to the loss value of the target model, when the input image has been corrupted by the perturbation at each location of (X,Y). During gradient-based perturbation search, this is the type of landscape that a gradient-based adversary is trying to ascend.

Fig. 2.2 shows the above 2D slice through the loss landscapes of PNI, L2P, SE-SNN, WCA, AA and k-WTA defences. In Fig. B.3 we show the corresponding smoothed loss landscapes, when under attack by PGD$_{\text{WT}}$, side-by-side for easier means of visual comparison. Further, Appendix B.4 includes the loss surfaces of the highest scoring non-stochastic adversarial defences listed in RobustBench (Croce et al., 2021), to give the reader a frame of reference of how non-rugged

Table 4.3: Robust accuracy % of SI-NI-FGSM (M1, Lin et al., 2020) and VMI-FGSM (M2, Wang and He, 2021) attacks and their respective WT extensions on CIFAR (RN-18 backbone) and Imagenette (WRN-34-10 backbone). Names are shortened for better readability.

	CIFAR-10				CIFAR-100			
Method	(M1)	WT-(M1)	M2	WT-(M2)	(M1)	WT-(M1)	M2	WT-(M2)
PNI	48.2	35.5 (-12.7)	38.3	27.4 (-10.9)	24.9	13.0 (-11.9)	25.7	18.6 (-7.1)
L2P	56.1	44.9 (-11.2)	31.7	19.2 (-12.5)	27.2	18.5 (-8.7)	30.1	21.0 (-9.1)
SE-SNN	40.5	31.6 (-8.9)	38.1	22.8 (-15.3)	25.3	12.2 (-13.1)	28.9	15.0 (-13.9)
WCA-Net	58.5	54.0 (-4.5)	55.7	34.8 (-20.9)	45.8	30.4 (-15.4)	44.0	33.2 (-10.8)
AA	61.8	53.6 (-8.2)	58.0	41.4 (-16.6)	46.7	31.8 (-14.9)	41.1	23.3 (-17.8)
k-WTA	55.3	43.0 (-12.3)	46.9	38.9 (-8.0)	49.4	38.0 (-11.4)	37.2	27.6 (-9.6)

	Imagenette			
Method	(M1)	WT-(M1)	M2	WT-(M2)
PNI	47.4	37.2 (-10.2)	42.5	33.2 (-9.3)
L2P	59.6	46.1 (-13.5)	42.4	30.5 (-11.9)
SE-SNN	44.8	33.9 (-10.9)	40.7	38.4 (-2.3)
WCA-Net	64.0	59.0 (-5.0)	51.6	42.3 (-9.3)
AA	66.5	49.3 (-17.2)	56.9	43.0 (-13.9)
k-WTA	57.9	46.5 (-11.4)	46.6	38.7 (-7.9)

loss landscapes should look like in state-of-the-art defences.

4.3 Experiments

4.3.1 Experimental Setup

For our experiments we consider four stochastic defences (PNI (He et al., 2019), L2P (Jeddi et al., 2020), SE-SNN (Yu et al., 2021) and WCA (Eustratiadis et al., 2021)) and two non-stochastic (k-WTA (Xiao et al., 2020) and AA (Alfarra et al., 2021)). For fair comparison these defences use the same backbone architecture, ResNet-18 (RN-18) and Wide ResNet-34-10 (WRN-34-10) (He et al., 2016; Zagoruyko and Komodakis, 2016) in the corresponding experiments. The robustness scores of all competitor methods were reproduced, and not taken from the original papers, to ensure that any change in performance comes from the WT extension. We evaluate their performance against the gradient-based PGD_{WT10} and $\text{PGD}_{\text{WT100}}$, and the gradient-free ZOO_{WT}. In terms of datasets, we consider CIFAR-10, CIFAR-100 (Krizhevsky and Hinton, 2009) and Imagenette (Howard, 2019) with high-resolution images. Our hyperparameter selection is outlined in Appendix B.

Table 4.4: Robust accuracy scores % of gradient-free attacks ZOO and ZOO$_{\mathrm{WT}}$ on CIFAR (RN-18 backbone) and Imagenette (WRN-34-10 backbone).

	CIFAR-10		CIFAR-100		Imagenette	
Method	ZOO	ZOO$_{\mathrm{WT}}$	ZOO	ZOO$_{\mathrm{WT}}$	ZOO	ZOO$_{\mathrm{WT}}$
PNI	62.1	54.3 (-7.8)	38.1	25.7 (-12.4)	59.2	41.0 (-18.2)
L2P	63.7	56.1 (-7.6)	37.5	29.7 (-7.8)	65.8	54.3 (-11.5)
SE-SNN	59.4	44.3 (-15.1)	28.3	21.5 (-6.8)	49.8	37.6 (-12.2)
WCA-Net	70.9	64.8 (-6.1)	48.8	42.8 (-6.0)	72.3	61.9 (-10.4)
AA	74.1	66.5 (-7.6)	52.7	42.3 (-10.4)	77.9	60.6 (-17.3)
k-WTA	70.2	64.5 (-5.7)	55.2	43.2 (-12.0)	70.1	53.7 (-16.4)

4.3.2 Quantitative Evaluation

In Tables 4.1 and 4.2 we report the accuracy of our selection of adversarial defences when under our PGD$_{\mathrm{WT}}$ attack against the baselines. It is evident that PGD$_{\mathrm{WT}}$ outperforms base PGD consistently across different benchmarks, defences, attack strengths, and network depths. In particular, we can observe that: (i) Every defence considered suffers substantially; in some cases even with more than -20% in robust accuracy. (ii) Weaker defences are almost completely defeated, with L2P, SE-SNN, and k-WTA failing on CIFAR-10; and PNI, L2P, SE-SNN and k-WTA failing on CIFAR-100. (iii) The stronger WCA and AA defences tend to suffer large hits, especially under PGD$_{\mathrm{WT100}}$. (iv) Our loss-smoothing attack is particularly effective with high-resolution images, with most defences suffering a performance reduction of over 15%.

To show the generality of our method, we apply the WT extension to the more sophisticated and recently proposed gradient-based adversaries NI-FGSM (Lin et al., 2020) and VMI-FGSM (Wang and He, 2021) that use acceleration and variance tuning to improve attack strength and transferability. Table 4.3 shows results consistent with our previous evaluation, and proves that our loss-smoothing method can effectively strengthen recently proposed attacks of higher complexity than PGD. Finally, in Table 4.4 we present our evaluation of ZOO$_{\mathrm{WT}}$. It is evident that even though (i) the performance reduction is on average slightly lower than the gradient-based setting and (ii) ZOO$_{\mathrm{WT}}$ imposes an additional query-efficiency cost, ZOO$_{\mathrm{WT}}$ is still successful in attacking these obfuscating defences.

These experimental results support that rugged loss surfaces can be exploited, and loss-smoothing adversaries are significantly stronger against this type of gradient obfuscation.

4.3.3 Interaction between WT and EoT

Let us now explicitly highlight the difference between the WT and EoT.

- **WT:** Approximates the *true loss landscape* of the neural network around

Table 4.5: Ablation study: effect of the WT and EoT individually against stochastic defences. The scores are the robust accuracy % on CIFAR-10.

(Attack: PGD$_{\text{WT10}}$)	No WT + No EoT	No WT + EoT$_{16}$	WT$_{16}$ + No EoT	WT$_{16}$ + EoT$_{16}$
PNI	50.6	49.1	48.7	34.8
L2P	58.9	54.4	55.0	47.2
SE-SNN	46.6	39.5	39.7	21.3
WCA-Net	72.0	58.4	61.1	53.3

input x, by averaging the gradients of the loss function w.r.t. m data points sampled in the neighborhood of x. Forward inference is performed using the *same neural network parameters*.

- **EoT:** Approximates *the true gradient* of the loss function w.r.t. the input, by averaging the gradients obtained after sampling neural network parameters n times. Forward inference is performed using the *same batch of data*.
- **WT + EoT:** Approximates the *true loss landscape* of the neural network around input x, by averaging the *true gradients* (obtained via EoT) of the loss function w.r.t. m data points sampled in the neighborhood of x.

In this section, we analyse how the WT and EoT interact with each other when attacking stochastic defences.

An ablation study is presented in Table 4.5, where we evaluate the two methods individually and in combination when attacking PNI, L2P, SE-SNN and WCA. We start by setting the baseline to regular PGD, and then vary each of the two components by setting the number of WT samples and EoT iterations to 16 (Appendix B explains why 16), to keep consistent with our evaluation in Section 4.3.2. Our ablation study shows that, while each method increases attack strength, neither is significantly better than the other in terms of individual performance. We conclude the WT and EoT are most effective when used in combination, to deal with the noisy loss landscape and the stochastic gradients respectively. Further analysis on this is provided in Appendix B.

4.4 Discussion

In this chapter, we reveal a new form of gradient obfuscation against adversarial attacks that can be a property of stochastic, as well as non-stochastic defences. It occurs when a neural network creates a noisy or discontinuous loss landscape to mislead gradient-based adversaries, and it does not constitute an adequate defence, as it can be circumvented by smoothing the surface of the loss function before following the gradient w.r.t. the input. We propose a smoothing method with which both gradient-based and gradient-free adversaries can be extended, utilising a Monte-Carlo variant of the Weierstrass transform. As demonstrated by applying the WT on PGD, ZOO and [SI-NI/VMI]-FGSM, this extension enables strong, successful attacks.

We further illustrate the smoothing capabilities of our adversary beyond the quantitative evaluation presented in Section 4.3.2, by plotting the loss surfaces of the defences before and after WT smoothing (Figure 2.2 main paper and Figure B.3 in Appendix B). We hope that highlighting this novel type of attack against this class of adversarial defences will inspire future research to avoid relying on this weak defence strategy for robustness.

4.4.1 Insights About the Robustness of SNNs

Upon reading this chapter and observing the generality of our method, i.e., how easily gradient-based attacks may be extended with loss-smoothing capabilities, a question arises: Should we avoid using SNNs, as they are not robust? To complicate matters further, it has been shown that SNNs trained with the VIB (Alemi et al., 2017) or WCA (Eustratiadis et al., 2021) objectives are provably robust. What happens when provably robust SNNs are under attack by loss-smoothing adversaries? Our experimental results outlined in Section 4.3 can help answer these questions.

As mentioned in Chapter 3, most stochastic defences, including PNI, L2P, and SE-SNN, are heuristically-motivated. Conversely, VIB and WCA have theoretical justification for their robust performance: VIB forms latent representations that are minimally expressive of the input and maximally expressive of the output, and WCA tightens the learning-theoretic bound that determines how invariant SNNs are to input perturbations. In our experiments, we see that WCA-Net consistently outperforms the heuristically-motivated competing defences, which means that even after smoothing its loss landscape, its theoretical robustness still holds.

In conclusion, we recommend that the research community uses SNNs that are theoretically-grounded and are coupled with clear and solid justification for their performance. This way the benefits that are associated with the use of SNNs, e.g., uncertainty estimation (Kendall and Gal, 2017), can be enjoyed while minimising the corresponding risks.

Part II

Stochastic Learning and Search for Few-Shot Adaptation

Chapter 5

Background

Learning from a few examples using prior knowledge is a key characteristic not only of human intelligence, but also of numerous variational/Bayesian meta-learning approaches to few-shot recognition (e.g., Finn et al., 2018; Gordon et al., 2019; Zhang et al., 2019b, 2021b). In this archetype of few-shot learning (FSL), prior knowledge is encoded by placing a prior distribution (usually Gaussian) over the learned parameters. During inference, the prior knowledge and the evidence (i.e., data) are considered jointly to make predictions.

In modern machine learning, however, prior knowledge is often interpreted in a non-distributional sense. Nowadays, there exist large, pre-trained foundation models (e.g., Brown et al., 2020) that can be used as few-shot learners, exploiting the fact that they have been trained on datasets orders of magnitude broader than the smaller datasets that represent target domains. This paradigm encourages few-shot adaptation, rather than few-shot learning. In Chapter 6, we showcase a novel stochastic approach in this line of work, where we train a vast search space of adaptation architectures, and then perform stochastic search within it to find the highest-performing ones w.r.t. a target domain.

5.1 Overview of Few-Shot Learning

In the past decade, there has been a substantial volume of work in the topic of few-shot (meta-)learning, a survey of which has been written by Wang et al. (2021b). In this section, however, we only focus on the subset of this work that is relevant to the scope of this thesis: variational approaches to few-shot learning, and few-shot adaptation of foundation models.

5.1.1 Variational Approaches

Making predictions on previously unseen tasks involves a high degree of ambiguity, even when a strong prior has been meta-learned on related tasks. To deal with this challenge, Finn et al. (2018) introduce a Bayesian version of the model-agnostic meta-learning (MAML) algorithm (Finn et al., 2017). Bayesian MAML samples models from a trained parameter distribution, allowing for stochastic adaptation to new tasks during meta-testing by injecting noise into gradient descent. In a related line of work, VERSA (Gordon et al., 2019) is an instance of a meta-learning framework for approximate probabilistic inference that takes few-shot learning datasets as inputs, and outputs distributions over task-specific parameters that are sampled during inference. These distributions can be learned without the use of second-order derivatives during meta-training, and can be sampled repeatedly and efficiently during meta-testing, without needing to take gradient steps.

While Bayesian MAML and VERSA belong to the family of methods that meta-learn model distributions, there exists another that meta-learn feature distributions. Zhang et al. (2019b) propose a Bayesian variant of nearest-centroid classification (Snell et al., 2017), where every prototype is mapped to a distribution instead of a point. Finally, MetaQDA (Zhang et al., 2021b) is a Bayesian meta-learning generalization of classifiers that are based on quadratic discriminant analysis (Hastie et al., 2009, Ch. 4). It operates solely on the classification layer, and it is agnostic of the latent representations produced by the backbone DNN.

5.1.2 Few-Shot Adaptation

Gradient-Based Adaptation Parameter-efficient adaptation modules have been previously applied for multi-domain learning, and transfer learning. A seminal example of this are Residual Adapters (Rebuffi et al., 2017), which are lightweight 1x1 convolutional filters added to ResNet blocks. They were initially proposed for multi-domain learning, but are also useful for FSL, by providing the ability to update the feature extractor while being lightweight enough to avoid severe overfitting in the few-shot regime. Task-Specific Adapters (TSA) (Li et al., 2022) use such adapters together with a URL (Li et al., 2021) pre-trained backbone to achieve state of the art results for CNNs on the Meta-Dataset benchmark (Triantafillou et al., 2020). Meanwhile, prompt (Jia et al., 2022) and prefix (Li and Liang, 2021) tuning are established examples of parameter-efficient adaptation for transformer architectures for similar reasons. In FSL, Efficient Transformer Tuning (ETT) (Xu et al., 2022) apply a similar strategy to few-shot ViT adaptation using a DINO (Caron et al., 2021) pre-trained backbone.

FT (Dhillon et al., 2020), FLUTE (Triantafillou et al., 2021), and PMF (Hu et al., 2022b) focus on adaptation of existing parameters without inserting new ones. To manage the adaptation/overfitting trade-off in the few-shot regime, PMF fine-tunes the whole ResNet or ViT backbone, but with carefully-managed learning rates. Meanwhile, FLUTE hand-picks a set of FILM parameters with a modified ResNet backbone for few-shot fine-tuning, while keeping the majority of the feature extractor frozen.

All of the methods above make heuristic choices about where to place adapters within the backbone, or for which parameters to allow/disallow fine-tuning. However, as different input layers represent different features (Zeiler and Fergus, 2014; Chen et al., 2021), there is scope for making better decisions about which features to update. Furthermore, in the multi-domain setting different target datasets may benefit from different choices about which modules to update. This paper takes an Auto-ML NAS-based approach to systematically address this issue.

Feed-Forward Adaptation The aforementioned methods all use stochastic gradient descent to update the features during adaptation. We briefly mention CNAPS (Requeima et al., 2019) and derivatives (Bateni et al., 2020) as a competing line of work that use feed-forward networks to modulate the feature extraction process. However, these dynamic feature extractors are less able to generalise to completely novel domains than gradient-based methods (Finn and Levine, 2018), as the adaptation module itself suffers from an out of distribution problem.

5.2 Overview of Neural Architecture Search

Neural Architecture Search (NAS) is a large and well-studied topic (Elsken et al., 2019) which we do not attempt to review in detail here. Mainstream NAS aims to discover new architectures that achieve high performance when training on a single dataset from scratch in a many-shot regime. To this end, research aims to develop faster search algorithms (e.g., Liu et al., 2019a; Guo et al., 2020; Abdelfattah et al., 2021; Xiang et al., 2023), and more effective search spaces (e.g., Radosavovic et al., 2019; Fang et al., 2020; Ci et al., 2021; Zhou et al., 2021).

5.2.1 Stochastic Single-Path One-Shot NAS

Our work builds upon the popular family of search strategies based on Single Path One-Shot (SPOS) (Guo et al., 2020). SPOS adopts a weight-sharing strategy: it encapsulates the entire search space inside a supernet that is trained by sampling paths randomly, and then a search algorithm determines the optimal path. It requires less memory and is more efficient than traditional NAS methods because only a portion of the candidates are activated and optimised.

While there exist some recent NAS works that try to address a similar “train once, search many times” problem efficiently (e.g., Cai et al., 2020; Li et al., 2020a; Moons et al., 2021; Molchanov et al., 2022), naively using these approaches has two serious shortcomings:

- They assume that after the initial supernet training, subsequent searches do not involve any training (e.g., a search is only performed to consider a different FLOPs constraint while accuracy of different configurations is assumed to stay the same) and thus can be done efficiently. This assumption does not hold true in the few-shot learning setting.

- Even if naively searching for each dataset at test time were computationally feasible, the few-shot nature of our setting poses a significant risk of overfitting the architecture to the small support set considered in each episode.

Chapter 6

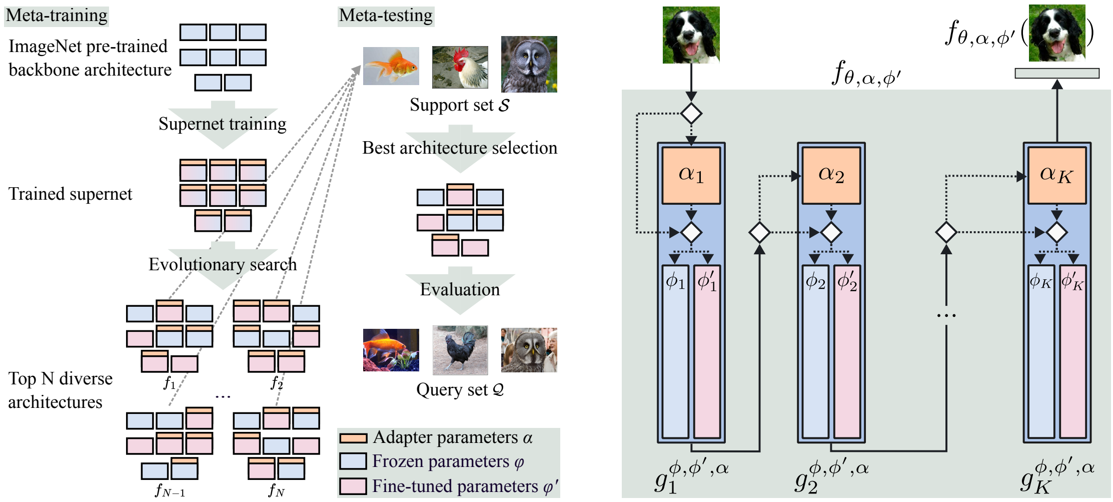
Neural Fine-Tuning Search for Few-Shot Learning

This chapter corresponds to the paper: P. Eustratiadis, Ł. Dudziak, D. Li, and T. M. Hospedales. Neural fine-tuning search for few-shot learning. In *International Conference on Learning Representations*, 2024. **(under review)**

In few-shot recognition, a classifier that has been trained on one set of classes is required to rapidly adapt and generalize to a disjoint, novel set of classes. To that end, recent studies have shown the efficacy of fine-tuning with carefully crafted adaptation architectures. However this raises the question of: How can one design the optimal adaptation strategy? In this chapter, we examine this question through the lens of neural architecture search (NAS). Given a pre-trained neural network, our algorithm discovers the optimal arrangement of adapters, which layers to keep frozen and which to fine-tune. We demonstrate the generality of our NAS method by applying it to both residual networks and vision transformers and report state-of-the-art performance on Meta-Dataset and Meta-Album.

6.1 Introduction

Few-shot recognition (Lake et al., 2011; Miller et al., 2000; Wang et al., 2021b) aims to learn novel concepts from few examples, often by rapid adaptation of a model trained on a disjoint set of labels. Many solutions adopt a meta-learning perspective (Finn et al., 2017; Snell et al., 2017; Ravi and Larochelle, 2017; Lee et al., 2019; Rusu et al., 2019), or train a powerful feature extractor on the source classes (Wang et al., 2019; Tian et al., 2020) – both of which assume that the training and testing classes are drawn from the same underlying distribution e.g., handwritten characters (Lake et al., 2015), or ImageNet categories (Vinyals et al., 2016). Later work considers a more realistic and challenging vari-


Meta-training
ImageNet pre-trained backbone architecture
Supernet training
Trained supernet
Evolutionary search
Top N diverse architectures
f_1
f_2
...
f_{N-1}
f_N
Meta-testing
Support set $\mathcal{S}$
Best architecture selection
Evaluation
Query set $\mathcal{Q}$
Adapter parameters α
Frozen parameters φ
Fine-tuned parameters φ'
$f_{\theta,\alpha,\phi'}$
$f_{\theta,\alpha,\phi'}($ $)$
α_1
α_2
α_K
ϕ_1
ϕ'_1
ϕ_2
ϕ'_2
ϕ_K
ϕ'_K
...
$g_1^{\phi,\phi',\alpha}$
$g_2^{\phi,\phi',\alpha}$
$g_K^{\phi,\phi',\alpha}$

(a) After a supernet is trained, evolutionary search finds the top-performing candidates (validation set). During a new test episode, the shortlisted candidates are evaluated on the support set (Eq. 6.12), and the best architecture for that test episode is selected.

(b) The dotted lines represent possible paths that can be sampled during SPOS training. Every adaptable layer in the architecture (g_i) has its own pre-trained ($\phi_i \subset \theta$), fine-tuned (ϕ'_i), and adapter (α_i) parameters.

Figure 6.1: Our proposed NAS paradigm for few-shot adaptation. (a) Overall meta-train/meta-test workflow. (b) The supernet architecture. The supernet contains all combinations of pre-trained, fine-tuned and adapter parameters. f denotes the feature extractor, which is composed of many layers, g, which are the minimal unit for adaptation in our search space.

ant of this problem, whereby a classifier should perform few-shot adaptation not only across visual categories, but also across diverse visual domains (Triantafillou et al., 2020; Ullah et al., 2022). In this cross-domain problem formulation, customising the feature extractor to the novel domains is important, and several studies address this through dynamic feature extractors (Requeima et al., 2019; Bateni et al., 2020) or ensembles of features (Dvornik et al., 2020; Li et al., 2021; Liu et al., 2021). Another group of studies employ simple, yet effective, fine-tuning strategies for adaptation (Dhillon et al., 2020; Hu et al., 2022b; Li et al., 2022; Xu et al., 2022) that are predominantly heuristically motivated. Thus, an important question that arises from previous work is: How can one design the *optimal* adaptation strategy? In this chapter, we take a step towards answering this question.

Fine-tuning approaches to few-shot adaptation must manage a trade-off between adapting a large or small number of parameters. The former allows for better adaptation, but risks overfitting on a few-shot training set. The latter reduces the risk of overfitting, but limits the capacity for adaptation to novel categories and domains. The recent PMF (Hu et al., 2022b) manages this trade-off through careful tuning of learning rates while fine-tuning the entire feature extractor. TSA (Li et al., 2022) and ETT (Xu et al., 2022) manage it by freezing the feature extractor weights, and inserting some parameter-efficient adaptation modules, lightweight enough to be trained in a few-shot manner. FLUTE (Triantafillou et al., 2021) manages it through selective fine-tuning of a tiny set of FILM (Perez et al., 2018) parameters, while keeping most of them fixed. Despite this progress, the best way to manage the adaptation/generalisation trade-off in fine-tuning approaches to few-shot learning (FSL) is still an open question. For example, which layers should be fine-tuned? What kind of adapters should be inserted, and where? While PMF, TSA, ETT, FLUTE, and others provide some intuitive recommendations, we propose a more systematic approach to answer these questions.

We advance the adaptation-based paradigm for FSL by developing a neural architecture search (NAS) algorithm that searches for and finds the optimal adaptation architecture. Given an initial pre-trained feature extractor, our NAS determines the subset of the architecture that should be fine-tuned, as well as the subset of layers where adaptation modules should be inserted. We draw inspiration from recent work in NAS (Guo et al., 2020; Cai et al., 2020; Chen et al., 2021; Chu et al., 2021; Zhang et al., 2022) that proposes revised versions of the stochastic Single-Path One-Shot (SPOS) (Guo et al., 2020) weight-sharing strategy. Specifically, given a strong pre-trained backbone such as a ResNet (He et al., 2016) or a Vision Transformer (ViT) (Dosovitskiy et al., 2021), we form a search space defined by the inclusion or non-inclusion of task-specific adapters per layer, and the freezing or fine-tuning of learnable parameters per layer. Based on this search space, we construct a supernet (Brock et al., 2018) that we train by sampling a random path in each forward pass (Guo et al., 2020). Our supernet architecture is illustrated schematically in Figure 6.1, where the aforementioned decisions are drawn as decision nodes (◇), and possible paths are marked in dotted lines.

While the supernet training remains somewhat similar to the standard NAS approaches, the subsequent search poses new challenges due to the inherent characteristics of the FSL setting. Specifically, as cross-domain FSL considers a number of datasets including novel domains at test time, it becomes questionable whether searching for a single model – which is the prevalent paradigm in NAS (Liu et al., 2019a; Cai et al., 2019; Li et al., 2020b; Wang et al., 2021a) – is the best choice. On the other hand, per-episode architecture selection is too slow and might overfit to the small support set.

Motivated by the aforementioned challenges, we propose a novel NAS algorithm that shortlists a small number of architecturally-diverse configurations at training time, but defers the final selection until the dataset and episode is known at test time. We empirically show that this is not only computationally efficient, but also improves results noticeably, especially when only a limited amount of domains is available at training time. We term our method Neural Fine-Tuning Search (NFTS).

NFTS defines a generic search space that is relevant to both major backbone architecture families (i.e., convolutional networks and transformers), and the choice of which specific adapter modules to consider is a hyperparameter, rather than a hard constraint. In this chapter, we consider using adapter modules that are currently state-of-the-art for ResNets and ViTs (TSA and ETT, respectively), but more adaptation architectures can be added to the search space.

Our contributions are summarised as follows:

- We provide the first systematic Auto-ML approach to finding the optimal adaptation strategy that trades-off adaptation flexibility and overfitting risk in multi-domain FSL.
- Our novel NFTS algorithm automatically determines which layers should be frozen or adapted, and where new adaptation parameters should be inserted for best few-shot adaptation.
- We advance the state-of-the-art in the well-established and challenging Meta-Dataset (Triantafillou et al., 2020), and the more recent and diverse Meta-Album (Ullah et al., 2022) benchmarks.

6.2 Neural Fine-Tuning Search

In this chapter, we develop an instantiation of the SPOS strategy for the multi-domain FSL problem. We construct a search space suited for parameter-efficient adaptation of a prior architecture to a new set of categories, and extend SPOS to learn on a suite of datasets, and efficiently generalise to novel datasets. This is different than the traditional SPOS paradigm of training and evaluating on the same dataset and same set of categories.

6.2.1 Few-Shot Learning Background

Let $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{D}$ be the set of D classification domains, and $\bar{\mathcal{D}} = \{X, Y\} \in \mathcal{D}$ a task containing n samples along with their designated true labels $\{\bar{X}, \bar{Y}\} = \{x_j, y_j\}_{j=1}^{n}$. Few-shot classification is the problem of learning to correctly classify a query set $\mathcal{Q} = \{X_\mathcal{Q}, Y_\mathcal{Q}\} \sim \bar{\mathcal{D}}$ by training on a support set $\mathcal{S} = \{X_\mathcal{S}, Y_\mathcal{S}\} \sim \bar{\mathcal{D}}$ that contains very few examples of each class in $\mathcal{Q}$. This can be achieved by finding the parameters θ of a classifier f_θ with the objective

$$\arg\max_{\theta} \prod_{\mathcal{D}} p(Y_\mathcal{Q} | f_\theta(\mathcal{S}, X_\mathcal{Q})). \tag{6.1}$$

In practice, if θ is randomly initialised and trained using stochastic gradient descent on a small support set $\mathcal{S}$, it will overfit and fail to generalise to $\mathcal{Q}$. To address this issue, one can exploit knowledge transfer from some seen classes to the novel classes. Formally, each domain $\bar{\mathcal{D}}$ is partitioned into two disjoint sets $\bar{\mathcal{D}}_{\text{train}}$ and $\bar{\mathcal{D}}_{\text{test}}$, which are commonly referred to as “meta-train” and “meta-test”, respectively. The labels in these sets are also disjoint, i.e., $Y_{\text{train}} \cap Y_{\text{test}} = \emptyset$. In that case, θ is trained by maximising the objective in Equation 6.1 using the meta-train set, but the overall objective is to perform adequately when transferring knowledge to meta-test.

The knowledge transferred from meta-train to meta-test can take various forms (Hospedales et al., 2022). As discussed earlier, we aim to generalise a family of few-shot methods (Hu et al., 2022b; Li et al., 2022; Xu et al., 2022) where parameters θ are transferred before a subset of them $\phi \subset \theta$ are fine-tuned; and possibly extended by attaching additional "adapter" parameters α that are trained for the target task. For meta-test, Equation 6.1 can therefore be rewritten as

$$\arg\max_{\alpha,\phi} \prod_{\mathcal{D}_{\text{test}}} p(Y_{\mathcal{Q}} | f_{\alpha,\phi}(\mathcal{S}, X_{\mathcal{Q}})), \tag{6.2}$$

In this chapter, we focus on finding the optimal adaptation strategy in terms of (i) the optimal subset of parameters $\phi \subset \theta$ that need to be fine-tuned, and (ii) the optimal task-specific parameters α to add.

6.2.2 Defining the Search Space

Let g_{ϕ_k} be the minimal unit for adaptation in an architecture. We consider these to be the repeated units in contemporary deep architectures, e.g., a convolutional layer in a ResNet, or a self-attention block in a ViT. If the feature extractor f_θ comprises of K such units with learnable parameters ϕ_k, then we denote $\theta = \bigcup_{k=1}^{K} \phi_k$, assuming all other parameters are kept fixed. For brevity in notation we will now omit the indices and refer to every such layer as g_ϕ. Following the state-of-the-art (Triantafillou et al., 2021; Hu et al., 2022b; Li et al., 2022; Xu et al., 2022), let us also assume that task-specific adaptation can be performed either by inserting additional adapter parameters α into g_ϕ, or by fine-tuning the layer parameters ϕ.

This allows us to define the search space as two independent binary decisions per layer: (i) The inclusion or non-inclusion of an adapter module attached to g_ϕ, and (ii) the decision of whether to use the pre-trained parameters ϕ, or replace them with their fine-tuned counterparts ϕ'. The size of the search space is, therefore, $(2^2)^K = 4^K$. For ResNets, we use the proposed adaptation architecture of TSA (Li et al., 2022), where a residual adapter h_α, parameterised by α, is

Table 6.1: The search space, as described in Section 6.2.2. When sampling a layer $g_{\phi,\phi',\alpha}$, it can be sampled in one of the following variants: (i) ϕ: fixed pre-trained parameters, no adaptation, (ii) α: fixed pre-trained parameters, with adaptation, (iii) ϕ': fine-tuned parameters, no adaptation, (iv) ϕ', α fine-tuned parameters, with adaptation.

	$g_{\phi,\phi',\alpha}(x)$ (ResNet)	$g_{\phi,\phi',\alpha}(x)$ (ViT)
$\phi\,, -$	$g_\phi(x)$	$z(A_{qkv}[q\ ;\ g_\phi(x)])$
$\phi\,, \alpha$	$g_\phi(x) + h_\alpha(x)$	$z(A_{qkv}[q\ ;\ g_\phi(x)] + h_{\alpha 1}) + h_{\alpha 2}$
$\phi', -$	$g_{\phi'}(x)$	$z(A_{qkv}[q\ ;\ g_{\phi'}(x)])$
ϕ', α	$g_{\phi'}(x) + h_\alpha(x)$	$z(A_{qkv}[q\ ;\ g_{\phi'}(x)] + h_{\alpha 1}) + h_{\alpha 2}$

Algorithm 3: Supernet training.

```
Input: Supernet $f_{\theta,\alpha,\phi'}$. Datasets $\mathcal{D}$. Step sizes $\eta_1$, $\eta_2$. Path pool $P$.
       Prototypical loss $\mathcal{L}$ (Eq. 6.5).
Output: Trained supernet $f_{\theta,\alpha,\phi'}$.
repeat
    Sample dataset $\bar{\mathcal{D}} \sim \mathcal{D}$
    Sample episode $\mathcal{S}$, $\mathcal{Q} \sim \bar{\mathcal{D}}$
    Sample path $p \sim P$ with learnable parameters $\alpha_p$, $\phi'_p$ and frozen
      parameters $\phi_p \subset \theta$
    $\alpha_p \longleftarrow \alpha_p - \eta_1 \nabla_{\alpha_p} \mathcal{L}(f^p_{\theta,\alpha,\phi'}, \mathcal{S}, \mathcal{Q})$
    $\phi'_p \longleftarrow \phi'_p - \eta_2 \nabla_{\phi'_p} \mathcal{L}(f^p_{\theta,\alpha,\phi'}, \mathcal{S}, \mathcal{Q})$
until prototypical loss converges
```

connected to g_ϕ

$$g_{\phi,\phi',\alpha}(x) = g_{\phi,\phi'}(x) + h_\alpha(x), \tag{6.3}$$

where $x \in \mathbb{R}^{W,H,C}$. For ViTs, we use the proposed adaptation architecture of ETT (Xu et al., 2022), where a tuneable prefix is prepended to the multi-head self-attention module A_{qkv}, and a residual adapter is appended to both A_{qkv} and the feed-forward module z in each decoder block

$$g_{\phi,\phi',\alpha}(x) = z(A_{qkv}[q \; ; \; g_{\phi,\phi'}(x)] + h_{\alpha 1}) + h_{\alpha 2}, \tag{6.4}$$

where $x \in \mathbb{R}^D$ and $[\cdot \; ; \; \cdot]$ denotes the concatenation operation. Note that in the case of ViTs the adapter is not a function of the input features, but simply an added offset.

Irrespective of the architecture, every layer $g_{\phi,\phi',\alpha}$ is parameterised by three sets of parameters, ϕ, ϕ', and α, denoting the initial parameters, fine-tuned parameters and adapter parameters respectively. Consequently, when sampling a configuration (i.e., path) from that search space, every such layer can be sampled as one of the variants listed in Table 6.1.

6.2.3 Training the Supernet

Following SPOS (Guo et al., 2020), our search space is actualised in the form of a supernet $f_{\theta,\alpha,\phi'}$; a “super” architecture that contains all possible architectures derived from the decisions detailed in Section 6.2.2. It is parameterised by: (i) θ, the frozen parameters from the backbone architecture f_θ, (ii) α, from the adapters h_α, and (iii) ϕ', from the fine-tuned parameters per layer $g_{\phi,\phi',\alpha}$.

We use a prototypical loss $\mathcal{L}(f, S, Q)$ as the core objective during supernet training and the subsequent search and fine-tuning.

$$\mathcal{L}(f, \mathcal{S}, \mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum_{i=1}^{|\mathcal{Q}|} \log \frac{e^{-d_{cos}(C_{\mathcal{Q}_i}, f(\mathcal{Q}_i))}}{\sum_{j=1}^{|C|} e^{-d_{cos}(C_j, f(\mathcal{Q}_i))}}, \tag{6.5}$$

Algorithm 4: Training time evolutionary search.

```
Input: Supernet $f_{\theta,\alpha,\phi'}$. Datasets $\mathcal{D}$. Step sizes $\eta_1$, $\eta_2$. Prototypical loss $\mathcal{L}$
       (Eq. 6.5). NCC accuracy $A$ (Eq. 6.11).
Output: Optimal path $p^*$.
Initialise population $P$ randomly
Initialise fitness of $P$ as $\Psi_P \longleftarrow 0$
repeat
    Sample episodes from all datasets $\mathcal{S}$, $\mathcal{Q} \sim \mathcal{D}$
    for each candidate $p \in P$ do
        for a small number of epochs do
            $\alpha_p \longleftarrow \alpha_p - \eta_1 \nabla_{\alpha_p} \mathcal{L}(f^p_{\theta,\alpha,\phi'}, \mathcal{S}, \mathcal{S})$
            $\phi'_p \longleftarrow \phi'_p - \eta_2 \nabla_{\phi'_p} \mathcal{L}(f^p_{\theta,\alpha,\phi'}, \mathcal{S}, \mathcal{S})$
        end
        $\Psi_p \longleftarrow A(f^p_{\theta,\alpha,\phi'}, \mathcal{S}, \mathcal{Q})$
    end
    offspring $\longleftarrow$ recombine the $M$ best candidates of $P$ w.r.t. $\Psi_P$
    $P \longleftarrow P$ + offspring
    eliminate the $M$ worst candidates of $P$ w.r.t. $\Psi_P$
until population fitness converges or max. iterations
```

where $C_{\mathcal{Q}_i}$ denotes the embedding of the class centroid that corresponds to the true class of $\mathcal{Q}_i$, and d_{cos} denotes the cosine distance. The set of class centroids C is computed as the mean embeddings of support examples that belong to the same class:

$$C = \left\{ \frac{1}{|\mathcal{S}^{y=l}|} \sum_{i=1}^{|\mathcal{S}|} f(\mathcal{S}_i^{y=l}) \right\}_{l=1}^{L}, \tag{6.6}$$

where L denotes the number of unique labels in $\mathcal{S}$.

For supernet training specifically, let P be a set of size 4^K, enumerating all possible sequences of K layers that can be sampled from the search space. Denoting a path sampled from the supernet as $f^p_{\theta,\alpha,\phi'}$, we minimise an expectation of the loss in Equation 6.5 over multiple episodes and paths, so the final objective becomes:

$$\arg\min_{\alpha,\phi'} \; \mathbb{E}_{p\sim P} \mathbb{E}_{\mathcal{S},\mathcal{Q}} \; \mathcal{L}(f^p_{\theta,\alpha,\phi'}, \mathcal{S}, \mathcal{Q}). \tag{6.7}$$

Algorithm 3 summarises the supernet training algorithm in pseudocode.

6.2.4 Searching for an Optimal Path

A supernet $f_{\theta,\alpha,\phi'}$ trained with the method described in Section 6.2.3 contains 4^K models, intertwined via weight sharing. As explained in Section 6.1, our goal is to search for the best-performing one, but the main challenge is related to the fact that we do not know what data is going to be used for adaptation at test time. One extreme approach, would be to search for a single solution at training time and simply use it throughout the entire test, regardless of the potential domain shift. Another, would be to defer the search and perform it from scratch each time a new support set is given to us at test time. However, both have their shortcomings. As such, we propose a generalization of this process where searching is split into two phases – one during training, and a subsequent one during testing.

Meta-training time. The search algorithm is responsible for pre-selecting a set of N models from the entire search space. Its main purpose is to mitigate potential overfitting that can happen at test time, when only a small amount of data is available, while providing enough diversity to successfully adjust the architecture to the diverse set of test domains. Formally, we search for a sequence of paths $(p_1, p_2, ..., p_N)$ where:

$$p_k = \arg\max_{p \in P} \mathbb{E}_{\mathcal{S},\mathcal{Q}} A(f^p_{\theta,\alpha^*,\phi'^*}, \mathcal{S}, \mathcal{Q}), \quad \text{s.t.} \tag{6.8}$$

$$\alpha^*, \phi'^* = \arg\min_{\alpha,\phi'} \mathcal{L}(f^p_{\theta,\alpha,\phi'}, \mathcal{S}, \mathcal{S}) \tag{6.9}$$

$$\forall_{j=1,\dots,k-1} \; d_{cos}(p_k, p_j) \geq T, \tag{6.10}$$

where T denotes a scalar threshold for the cosine distance between paths p_k and p_j, and A is the classification accuracy of a nearest centroid classifier (NCC) (Snell et al., 2017),

$$A(f, \mathcal{S}, \mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum_{i=1}^{|\mathcal{Q}|} [\arg\min_j d_{cos}(C_{\mathcal{Q}_j}, f(\mathcal{Q}_i)) = Y_{\mathcal{Q}_i}]. \tag{6.11}$$

Noticeably, we measure accuracy of a solution using a query set, after fine-tuning on a separate support set (Equation 6.9), then average across multiple episodes to avoid overfitting to a particular support set (Equation 6.8). We also employ a diversity constraint, in the form of cosine distance between binary encodings of selected paths (Equation 6.10), to allow for sufficient flexibility in the following test time search.

To efficiently obtain sequence $\{p_1, ..., p_N\}$, we use evolutionary search to find points that maximise Equation 6.8, and afterwards select the N best performers from the evolutionary search history that satisfy the constraint in Equation 6.10. Algorithm 4 summarises training-time search.

Meta-testing time. For a given meta-test episode, we decide which one of the pre-selected N models is best suited for adaptation on the given support set data. It acts as a failsafe to counteract the bias of the initial selection made at training time in cases when the support set might be particularly out-of-domain.

Table 6.2: Comparison to the state-of-the art methods on Meta-Dataset. Single domain setting – only ImageNet is seen during training and search. Performance of competitor methods was taken from Li et al. (2022) and Hu et al. (2022b) when possible, otherwise from the original published papers. Reporting mean accuracy over 600 episodes. *Additional data used for training.

	Method	Aircrafts	Birds	Textures	Fungi	ImageNet	Omniglot	QuickDraw	Flowers	CIFAR-10	CIFAR-100	MNIST	MS COCO	Traffic Signs	Average
ResNet-18	Liu et al. (2021)	48.5	47.9	63.8	31.8	46.9	61.6	57.5	80.1	65.4	52.7	80.8	41.4	46.5	52.6
	Snell et al. (2017)	53.1	68.8	66.6	39.7	50.5	60.0	49.0	85.3	-	-	-	41.0	47.1	56.1
	Saikia et al. (2020)	54.1	70.7	68.3	41.4	51.9	67.6	50.3	87.3	-	-	-	48.0	51.8	59.2
	Triantafillou et al. (2020)	63.4	69.8	70.8	41.5	52.8	61.9	59.2	86.0	-	-	-	48.1	60.8	61.4
	Li et al. (2022)	72.2	74.9	77.3	44.7	59.5	78.2	**67.6**	90.9	82.1	70.7	93.9	59.0	**82.5**	73.3
	Ours	**74.9**	**76.5**	**81.6**	**50.5**	**62.7**	**80.2**	67.2	**94.5**	**83.0**	**71.5**	**94.0**	**59.7**	81.9	**75.2**
ViT-S	*Hu et al. (2022b)	76.8	85.0	86.6	54.8	**74.7**	80.7	71.3	94.6	-	-	-	**62.6**	**88.3**	77.5
	Xu et al. (2022)	79.9	**85.9**	**87.6**	61.8	67.4	78.1	71.3	**96.6**	-	-	-	62.3	85.1	77.6
	Ours	**83.0**	85.5	**87.6**	**62.2**	71.0	**81.9**	**74.5**	96.0	79.4	72.6	95.2	**62.6**	87.9	**79.2**

Formally, the final path p^* to be used in a particular episode is defined as:

$$p^* = \underset{p \in \{p_1, \dots, p_N\}}{\arg\min} \; \mathcal{L}(f^p_{\theta,\alpha^*,\phi'^*}, \mathcal{S}, \mathcal{S}), \quad \text{s.t.} \tag{6.12}$$

$$\alpha^*, \phi'^* = \underset{\alpha,\phi'}{\arg\min} \, \mathcal{L}(f^p_{\theta,\alpha,\phi'}, \mathcal{S}, \mathcal{S}) \tag{6.13}$$

Noticeably, we test each of the N models by fine-tuning it on the support set (Equation 6.13) and testing its performance on the same support set (Equation 6.12). This is because the support set is the only source of data we have at test time and we cannot extract a disjoint validation set from it without risking the quality of the fine-tuning process. It is important to note that, while this step risks overfitting, the pre-selection of models at training time, as described previously, should already limit the subsequent search to only models that are unlikely to overfit. Since N is kept small in our experiments, we use a naive grid search to find p^*.

This approach is a generalization of existing NAS approaches, as it recovers both when $N = 1$ or $N = 4^K$. Our claim is that intermediate values of N are more likely to give us better results than any of the extremes, due to the reasons mentioned earlier. In particular, we would expect pre-selecting $1 < N \ll 4^K$ models to introduce reasonable overhead at test time while improving results, especially in cases when exposure to different domains might be limited at training time. In our evaluation we compare $N = 3$ and $N = 1$ to test this hypothesis. We do not include comparison to $N = 4^K$ as it is computationally unfeasible in our setting (performing equivalent of training time search for each test episode would require us to fine-tune $\approx 14 * 10^6$ models in total).

Table 6.3: Comparison to the state-of-the art methods on Meta-Dataset. Multi-domain setting – the first 8 datasets are seen during training and search. Performance of competitor methods was taken from Li et al. (2022) and Hu et al. (2022b) when possible, otherwise from the original published papers. Reporting mean accuracy over 600 episodes. *Additional data used for training.

	Method	Aircrafts	Birds	Textures	Fungi	ImageNet	Omniglot	QuickDraw	Flowers	CIFAR-10	CIFAR-100	MNIST	MS COCO	Traffic Signs	Average
ResNet-18	Requeima et al. (2019)	83.7	73.6	59.5	50.2	50.8	91.7	74.7	88.9	-	-	-	39.4	56.5	66.9
	Wang et al. (2019)	82.0	74.8	68.8	46.6	58.4	91.6	76.5	90.5	74.9	61.3	94.6	48.9	57.2	69.5
	Dvornik et al. (2020)	85.5	71.0	71.0	64.3	56.2	94.1	81.8	82.9	66.5	56.9	94.3	52.0	51.0	71.4
	Liu et al. (2021)	85.8	76.2	71.6	64.0	56.8	94.2	82.4	87.9	67.0	57.3	90.6	51.5	48.2	71.8
	Triantafillou et al. (2021)	82.8	75.3	71.2	48.5	58.6	92.0	77.3	90.5	75.4	62.0	96.2	52.8	63.0	72.7
	Li et al. (2021)	89.4	80.7	77.2	68.1	58.8	94.5	82.5	92.0	74.2	63.5	94.7	57.3	63.3	76.6
	Li et al. (2022)	89.9	81.1	77.5	66.3	59.5	**94.9**	81.7	**92.2**	82.9	70.4	**96.7**	57.6	**82.8**	78.4
	Ours	**90.1**	**83.8**	**82.3**	**68.4**	**61.4**	94.3	**82.6**	**92.2**	**83.0**	**75.1**	95.4	**58.8**	81.9	**80.7**
ViT-S	*Hu et al. (2022b)	88.3	91.0	**86.6**	74.2	**74.6**	91.8	79.2	**94.1**	-	-	-	62.6	**88.9**	83.1
	Ours	**89.1**	**92.5**	86.3	**75.1**	**74.6**	**92.0**	**80.6**	93.5	75.9	70.8	91.3	**62.8**	87.2	**83.4**

6.3 Experiments

6.3.1 Experimental Setup

Evaluation on Meta-Dataset We evaluate NFTS on the extended Meta-Dataset (Requeima et al., 2019; Triantafillou et al., 2020), currently the most commonly used benchmark for few-shot classification, that consists of 13 datasets: FGVC Aircraft, CU Birds, Describable Textures (DTD), FGVCx Fungi, ImageNet (ILSVRC 2012), Omniglot, QuickDraw, VGG Flowers, CIFAR-10/100, MNIST, MSCOCO, and Traffic Signs. There are 2 evaluation protocols: single domain learning and multi-domain learning. In the single domain setting, only ImageNet is seen during training and meta-training, while in the multi-domain setting the first eight datasets are seen (FGVC Aircraft to VGG Flower). For meta-testing at least 600 episodes are sampled for each domain.

Evaluation on Meta-Album Further, we evaluate NFTS on the more recently introduced Meta-Album (Ullah et al., 2022). Meta-Album is more diverse than Meta-Dataset. We use the currently available Sets 0-2, which contain over 1000 unique labels across 30 datasets spanning 10 domains including microscopy, remote sensing, manufacturing, plant disease, character recognition and human action recognition tasks, etc. Unlike Meta-Dataset, where their default evaluation protocol is variable-way variable-shot, Meta-Album evaluation follows a 5-way variable-shot setting, where the number of shots is typically 1, 5, 10 and 20. For meta-testing, results are averaged over 1800 episodes.

Architectures We employ two different backbone architectures, a ResNet-18 (He et al., 2016) and a ViT-small (Dosovitskiy et al., 2021). Following TSA (Li et al., 2022), the ResNet-18 backbone is pre-trained on the seen domains with the knowledge-distillation method URL (Li et al., 2021) and, following ETT (Xu

Table 6.4: Comparison of our method against Meta-Album baselines, as reported in Fig. 2b of their paper (Ullah et al., 2022). The setting is cross-domain 5-way [1, 5, 10, 20]-shot, and accuracy scores are averaged over 1800 tasks drawn from Set0, Set1 and Set2.

	From Scratch	Fine Tuning	Matching Net	ProtoNet	FO-MAML	NFTS
1-shot	30.42	40.43	34.49	38.07	33.94	**43.76**
5-shot	38.31	50.87	44.32	51.17	44.50	**57.59**
10-shot	39.58	53.42	49.23	55.18	48.62	**60.10**
20-shot	39.83	55.12	52.99	59.67	51.35	**60.97**

et al., 2022), the ViT-small backbone is pre-trained on the seen portion of ImageNet with the self-supervised method DINO (Caron et al., 2021). We consider TSA residual adapters (Rebuffi et al., 2017; Li et al., 2022) for ResNet and Prefix Tuning (Li and Liang, 2021; Xu et al., 2022) adapters for ViT. This is mainly to enable direct comparison with prior work on the same base architectures that use exactly these same adapter families, without introducing new confounders in terms of mixing adapter types (Li et al., 2022; Xu et al., 2022). However our framework is flexible, meaning it can accept any adapter type, or even multiple types in its search space.

6.3.2 Comparison to State of the Art

Meta-Dataset The results on Meta-Dataset are shown in Table 6.2 and Table 6.3 for single-domain and multi-domain training setting respectively. We can see that NFTS obtains the best average performance across all the competitor methods for both ResNet and ViT architectures. The margins over prior state-of-the-art are often substantial for this benchmark with +1.9% over TSA in ResNet-18 single domain, +2.3% in multi-domain and +1.6% over ETT in VIT-small single domain. The increased margin in the multi-domain case is intuitive, as our framework has more data with which to learn the optimal path(s).

We re-iterate that PMF, ETT, and TSA are special cases of our search space corresponding respectively to: (i) Fine-tune all and include no adapters, (ii) Include ETT adapters at every layer while freezing all backbone weights and (iii) Include TSA adapters at every layer while freezing all backbone weights. We also share initial pre-trained backbones with ETT and TSA (but not PMF, as it uses a stronger pre-trained model with additional data). Thus the margins achieved over these competitors are attributable to our systematic approach to finding suitable architectures in terms of where to fine-tune and where to insert new adapter parameters.

Meta-Album The results on Meta-Album are shown in Figure 6.4 as a function of number of shots within the 5-way setting, following Ullah et al. (2022). We can see that across the whole range of support set sizes, our NFTS outperforms

Table 6.5: Ablation study on Meta-Dataset comparing four special cases of the search space in terms of average accuracy: (i) $\phi, -$: No adaptation, no fine-tuning, (ii) ϕ, α: Adapt all, (iii) $\phi', -$: Fine-tune all, (iv) ϕ', α: Adapt and fine-tune all. NFTS-{1,N} refer to conventional and deferred episode-wise NAS respectively.

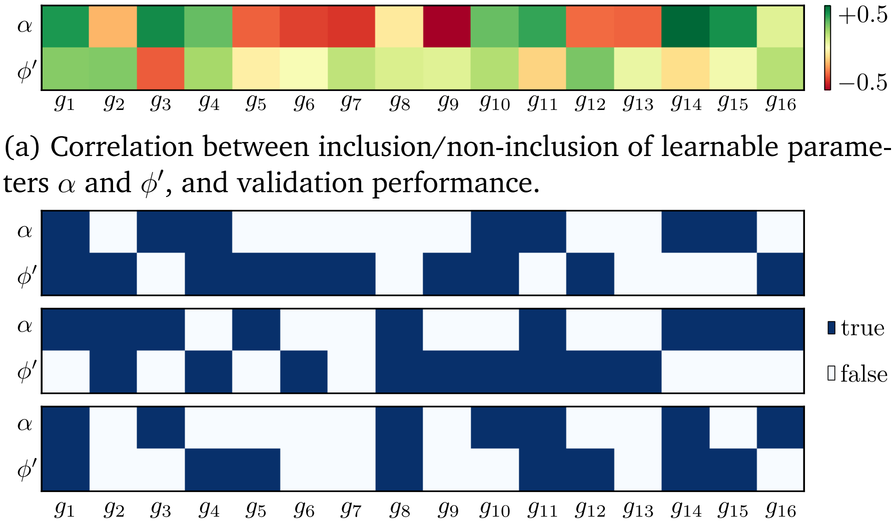
	Method	Single Domain	Multi-Domain
ResNet-18	$\phi\,, -$	67.8	67.8
	$\phi\,, \alpha$	70.4	76.5
	$\phi', -$	70.2	76.3
	ϕ', α	70.8	76.9
	NFTS-1	73.6	80.1
	NFTS-N	75.2	80.7
ViT-S	$\phi\,, -$	71.8	71.8
	$\phi\,, \alpha$	73.8	77.3
	$\phi', -$	74.0	77.5
	ϕ', α	74.4	78.9
	NFTS-1	78.7	83.1
	NFTS-N	79.2	83.4

the well-tuned baselines from Ullah et al. (2022). The margins are substantial, e.g., greater than 5% at 5-way/5-shot operating point. This result confirms that our framework scales to even more diverse datasets and domains than those considered previously in Meta-Dataset.

6.3.3 Ablation study

To analyse more precisely the role that our architecture search plays in few-shot performance, we also conduct an ablation study of our final model against four corners of our search space: (i) Initial model only, using a pre-trained feature extractor and simple NCC classifier, which loosely corresponds to SimpleShot (Wang et al., 2019), (ii) Full adaptation only, using a fixed feature extractor, which loosely corresponds to TSA (Li et al., 2022), ETT (Xu et al., 2022), FLUTE (Triantafillou et al., 2021), and others – depending on base architecture and choice of adapter, (iii) Fully fine-tuned model, which loosely corresponds to PMF (Hu et al., 2022b), and (iv) Combination of full fine-tuning and adaptation. From the results in Table 6.5 we can see that both fine-tuning (ii), adapters (iii), and their combination (iv) give improvement on the linear readout baseline (i). However, all of them are worse than the systematically optimised adaptation architecture of NFTS.

Furthermore, the ablation compares the results using the top-1 adaptation architecture found by SPOS architecture search against our novel progressive approach that defers the final architecture selection to an episode-wise decision. Our deferred architecture selection improves on fixing the top-1 architecture



(a) Correlation between inclusion/non-inclusion of learnable parameters α and ϕ', and validation performance.

(b) Top 3 performing paths subject to diversity constraint.

Figure 6.2: Qualitative analysis of our architecture search. Figure 6.2a summarises the whole search space by answering the question: *How important is to adapt (α) or fine-tune (ϕ') each block?* The color of each square indicates the point-biserial correlation (over all searched architectures) between adapting/fine-tuning layer g_i and validation performance. Figure 6.2b shows the top 3 performing candidates subject to a diversity constraint, after 15 generations of evolutionary search. Dark blue indicates that the layer is adapted/fine-tuned and light blue that it is not.

from meta-train, demonstrating the value of per-dataset/episode architecture selection (see also Sec 6.3.4).

6.3.4 Further Analysis

The ablation study quantitatively demonstrates the benefit of architecture search over common fixed adaptation strategies. In this section, we aim to analyse: What kind of adaptation architecture is discovered by our NAS strategy, and how it is discovered?

Discovered Architectures We first summarise results of the entire search space in terms of which layers are preferential to fine-tune or not, and which layers are preferential to insert adapters or not in Figure 6.2a. The blocks indicate layers (columns) and adapters/fine-tuning (rows), with the color indicating whether that architectural decision was positively (green) or negatively (red) correlated with validation performance. We can see that the result is complex, without a simple pattern, as assumed by existing work (Hu et al., 2022b; Li et al., 2022; Xu et al., 2022). That said, our NAS does discover some interpretable trends. For example, adapters should be included at early/late ResNet-18 layers and not at layers 5-9.

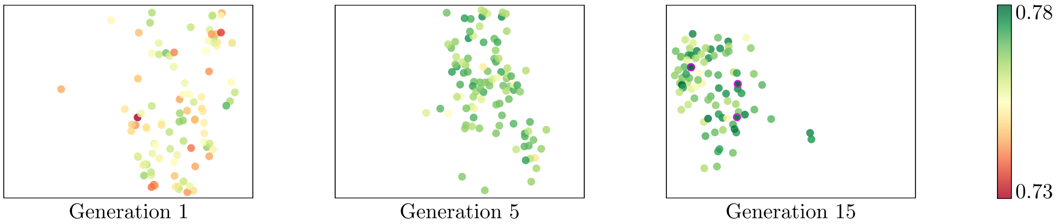
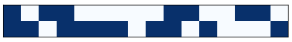
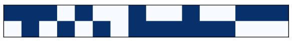
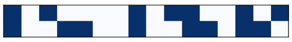


Figure 6.3: Population of paths(candidate architectures) in the search space after 1, 5, and 15 generations of evolutionary search. Each dot is a 2-d TSNE projection of the binary vector representing an architecture, and its color shows the validation performance for that architecture. The supernet contains a wide variety of models in terms of validation performance, and the search algorithm converges to a well-performing population. The top 3 performing paths that are given in 6.2b are highlighted in the far right figure (Generation 15) in purple outline.

Table 6.6: How the diverse selection of architectures from Figure 6.2b perform per unseen downstream domain in Meta-Dataset. Shading indicates episode-wise architecture selection frequency, numbers indicate accuracy using the corresponding architecture. The best dataset-wise architecture (bold) is most often selected (shading).

			
CIFAR-10	82.0	81.2	**83.3**
CIFAR-100	**75.9**	75.0	75.1
MNIST	**95.5**	94.4	95.1
MSCOCO	**58.1**	57.8	56.4
Tr. Signs	81.7	**82.2**	81.8

We next show the top three performing paths subject to diversity constraint in Figure 6.2b. We see that these follow the strong trends in the search space from Figure 6.2a. For example, they always adapt (α) block 14 and never adapt block 9. However, otherwise they do include diverse decisions (such as whether to fine-tune (ϕ') block 15) which was not strongly indicated in Figure 6.2a.

Finally, we analyse how our small set of $N = 3$ candidate architectures in Figure 6.2b is used during meta-test. This small set allows us to perform an efficient minimal episode-wise NAS, including for novel datasets unseen during training. The results in Table 6.6 show how often each architecture is selected by held out datasets during meta-test (shading), and what is the per-dataset performance using only that architecture. It is evidence of how our approach successfully learns to select the most suitable architecture on a per-dataset basis, even for unseen datasets. This unique capability goes beyond prior work (Hu et al., 2022b; Li et al., 2022; Xu et al., 2022) where all domains must rely on the same adaptation strategy despite their diverse adaptation needs.

Path Search Process In addition, Figure 6.3 illustrates the path search process. In this figure, we illustrate our $2K$-dimensional architecture search space as a 2D t-SNE projection, where the dots are candidate architectures of the evolutionary search process at different iterations. The dots are colored red (low) to green (high), according to their validation accuracy. From the results we can see that: The initial set of candidates is broadly dispersed and generally low performing (left), and gradually converge toward a tighter cluster of high performing candidates (right). The top 3 performing paths subject to a diversity constraint (also illustrated in Figure 6.2b) are annotated in purple outline.

6.4 Discussion

This chapter presents NFTS, a novel neural architecture-search based approach that discovers the optimal adaptation architecture for gradient-based few-shot learning. NFTS contains several recent strong heuristic adaptation architectures as special cases within its search space, which are all outperformed by our proposed systematic architecture search, leading to a new state-of-the-art on Meta-Dataset and Meta-Album. While we use a simple and coarse search space for easy and direct comparison to prior work's hand-designed adaptation strategies, future work can extend this framework to include a richer range of adaptation strategies, and a finer-granularity of search.

6.4.1 Insights About Neural Fine-Tuning Search

Our proposed framework displays high performance when deployed in a few-shot learning setting. However, there are some trade-offs to be considered when training a large foundation model.

Search space complexity/search time trade-off

In this section we discuss the subtleties of designing the search space for NFTS. In Section 6.2, we formulate the search space as the inclusion/non-inclusion of a **single** adaptation architecture per layer, and the decision of whether to fine-tune that layer or keep it frozen. With these two simple binary decisions, the search space is 4^K possible models, where K is the number of layers to be adapted or fine-tuned.

Of course, the search space could be expanded with many more adaptation architectures, e.g., BitFit (Zaken et al., 2022), LoRA (Hu et al., 2022a), bias tuning, and others. However, it is important to keep in mind that while enriching the search space can lead to better models, it can also substantially increase the convergence times of both supernet training and evolutionary search. For example, consider that adding one more binary decision can increase the search space from 4^K to 8^K, and $4^K << 8^K$ for deep architectures, e.g., if $K = 32$.

Increased cost when searching N architectures

As analysed in Section 6.3.3, our approach can be used in either

- **Top-1** architecture mode: Where each episode is a pure fine-tuning operation given the chosen architecture.
- **Top-N** architecture mode: Where each episode can perform its own small architecture selection routine based on the shortlisted architectures produced during evolutionary search, as well as fine-tuning

We remark that while the latter imposes a slightly increased cost during testing ($N = 3\times$ in practice), this is similar or less than competitors who repeat adaptation with different learning rates during testing (Hu et al., 2022b) ($4\times$ cost), or exploit a backbone ensemble ($8\times$ cost) (Dvornik et al., 2020; Liu et al., 2021).

Epilogue

Chapter 7

Scientific Impact of Thesis

The contributions presented in this thesis have advanced the world's knowledge of stochastic deep learning in the context of two major fields of modern machine learning research: (i) adversarial robustness, and (ii) neural architecture search. This section details how these contributions either *can* affect, or *have already* affected these fields of research.

7.1 Contributions

7.1.1 Weight-Covariance Alignment

The main finding of WCA is that one is theoretically guaranteed to dampen the effect of an adversarial perturbation in the input, by using a specific type of stochastic classifier. The weight vectors of that classifier need to be aligned with the eigenvectors of its covariance matrix in order for our theoretical guarantees to hold. We further show that it is more effective to use an anisotropic covariance matrix to achieve this kind of alignment. Since we are the first to propose the use of an anisotropic covariance matrix in the field of stochastic adversarial defence, our method of optimising it with stochastic gradient descent is also an important contribution.

Impact WCA has been studied by a number of follow-up research efforts as related work (Liang and Chan, 2022; Däubener and Fischer, 2022; Lee et al., 2023). It was most impactful in the work of Däubener and Fischer (2022), which discusses the effect of stochasticity during inference, as well as the importance of sample size during attack. Their work can also be seen as a generalisation of our analysis of stochastic loss landscapes in Chapter 4.

Limitations and Future Work Our study of WCA is limited to stochastic classifiers. This immediately leaves two open questions for future work and further analysis:

- How could we extend WCA to improve robustness against adversarial attacks when the target variable is continuous, rather than a class label?

- Is there a positive effect when WCA is achieved in layers other than the classification layer, e.g., if the backbone is also stochastic?

Furthermore, our theoretical analysis of WCA that is associated with guarantees of robustness, is conducted for the case of binary classification. There is a strong assumption in our work, that our theoretical analysis generalises gracefully to a multi-class classification setting. We present experimental evidence of this, of course, but it is not explicitly proven.

Finally, our proposed WCA regularisation term is optimised along with the rest of the network parameters using stochastic gradient descent. As mentioned in Chapter 3, this is a challenge, because stochastic gradient descent tends to maximise the dot product of the classifier weight vectors and the covariance matrix by inflating their values rather than achieving alignment. We remedy this issue by means of constraints, and ℓ^2 regularisation. Future work may consider to optimise the value of the WCA term by means other than gradient-based search that directly, rather than implicitly, focus on alignment; options include bi-level optimisation, evolutionary search, and others.

7.1.2 Smoothing the Loss Landscape

This work is the first to contribute a thorough analysis of the loss landscapes of several stochastic and non-stochastic adversarial defences. We demonstrate that a stochastic attack that repeatedly samples adversarial input images in the neighborhood of an initially-generated adversarial example, effectively smooths the rugged loss surface of several adversarial defences.

Impact Our research has been cited as related work in a recent paper preprint of Niroomand et al. (2023) that analyses the loss landscapes of Gaussian processes.

Limitations and Future Work In our work, we have analysed the loss landscape of stochastic and other defences as a function of the input, to illustrate how SNNs hinder the process of perturbation search. Future work can consider analysing the loss landscape as a function of the network parameters, rather than the input. This analysis can help us answer questions such as: We know that the TRADES (Zhang et al., 2019a) adversarial training objective outperforms the PGD (Madry et al., 2018) objective when it comes to downstream robustness. How does TRADES navigate the loss landscape w.r.t. the parameters compared to PGD, and why does it lead to a better solution?

7.1.3 Neural Fine-Tuning Search

Our latest work combines ideas of neural architecture search and parameter-efficient adaptation to fine-tune large pre-trained models that achieve state-of-the-art performance on previously unseen downstream tasks. This work contains two main findings:

- Fine-tuning/adapting an entire pre-trained architecture to a downstream task is sub-optimal. There exist fine-tuning/adaptation configurations that are non-trivial to find, but achieve a significant performance boost during downstream evaluation.
- Different architectures in the defined search space are able to specialise on different aspects of the data, as a consequence of having been trained with a path dropout strategy. Using a number of different architectures as an ensemble, is therefore beneficial.

We consider these findings to be of great importance, especially in the current era of machine learning, where using large pre-trained models as backbones has become more common than ever.

Impact This work is currently being applied on large language models (LLMs), in participation to the NeurIPS LLM Efficiency Challenge[1], where the goal is to adapt 7-billion parameter pre-trained LLMs to downstream language tasks, using 1 GPU for 1 day.

Limitations and Future Work We have limited the scope of our work to apply NFTS to a few-shot learning setting. Future work may consider applying NFTS to other common supervised tasks such as domain adaptation, and self-supervised tasks such as contrastive learning. There are two significant differences between each family of downstream tasks from the perspective of NFTS: (i) the data separation, e.g., (meta-) train, validation, and test, and (ii) the search objective.

The core idea of this work – that it is effective to place adapters and fine-tune only certain parts of the architecture that are not easy to find – unveils a problematic property in the modern machine learning paradigm that uses pre-trained foundation models, that relates to model explainability. Why do some layers encode knowledge optimally while others do not? Can these layers be identified without the need for complicated and expensive retraining and search? It is important for future work to focus on questions like these, especially due to this paradigm's sudden increase in popularity.

[1] https://llm-efficiency-challenge.github.io/

Bibliography

M. S. Abdelfattah, A. Mehrotra, Ł. Dudziak, and N. D. Lane. Zero-cost proxies for lightweight NAS. In *International Conference on Learning Representations (ICLR)*, 2021.

S. Addepalli, V. B. S., A. Baburaj, G. Sriramanan, and R. V. Babu. Towards achieving adversarial robustness by enforcing feature consistency across bit planes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2017.

M. Alfarra, J. C. Perez, A. Thabet, A. Bibi, P. Torr, and B. Ghanem. Combating adversaries with anti-adversaries. In *ICML Workshop on Adversarial Machine Learning*, 2021.

M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision (ECCV)*, 2020.

A. Athalye, N. Carlini, and D. A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018a.

A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018b.

T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang. Recent advances in adversarial training for adversarial robustness. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

P. Bateni, R. Goyal, V. Masrani, F. Wood, and L. Sigal. Improved few-shot visual classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

C. M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Information Science and Statistics. Springer, 2007.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. In *International Conference on Machine Learning (ICML)*, 2015.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2), 2018.

A. Brock, T. Lim, J. M. Ritchie, and N. Weston. SMASH: one-shot model architecture search through hypernetworks. In *International Conference on Learning Representations (ICLR)*, 2018.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

H. Cai, L. Zhu, and S. Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations (ICLR)*, 2019.

H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations (ICLR)*, 2020.

Q. Cai, C. Liu, and D. Song. Curriculum adversarial training. In *International Joint Conference on Artificial Intelligence, IJCAI*, 2018.

N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *Symposium on Security and Privacy*, 2017.

Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. Liang. Unlabeled data improves adversarial robustness. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *International Conference on Computer Vision (ICCV)*, 2021.

A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay. A survey on adversarial attacks and defences. *Transactions on Intelligent Technology*, 6, 2021.

R. Chavhan, J. Stuehmer, C. Heggan, M. Yaghoobi, and T. M. Hospedales. Amortised invariance learning for contrastive self-supervision. In *International Conference on Learning Representations (ICLR)*, 2023.

M. Chen, H. Peng, J. Fu, and H. Ling. Autoformer: Searching transformers for visual recognition. In *International Conference on Computer Vision (ICCV)*, 2021.

P. Chen, H. Zhang, Y. Sharma, J. Yi, and C. Hsieh. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Workshop on Artificial Intelligence and Security*, 2017.

X. Chu, B. Zhang, and R. Xu. Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. In *International Conference on Computer Vision (ICCV)*, 2021.

Y. Ci, C. Lin, M. Sun, B. Chen, H. Zhang, and W. Ouyang. Evolving search space for neural architecture search. In *International Conference on Computer Vision, (ICCV)*, 2021.

J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019.

F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein. RobustBench: a standardized adversarial robustness benchmark. In *Conference on Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2021.

S. Däubener and A. Fischer. How sampling impacts the robustness of stochastic neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning (ICML)*, 2018.

G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations (ICLR)*, 2020.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

J. C. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 12, 2011.

N. Dvornik, C. Schmid, and J. Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *European Conference on Computer Vision (ECCV)*, 2020.

T. Elsken, J. H. Metzen, and F. Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research (JMLR)*, 20, 2019.

L. Ericsson, H. Gouk, and T. M. Hospedales. Why do self-supervised models transfer? on the impact of invariance on downstream tasks. In *British Machine Vision Conference (BMVC)*, 2022.

P. Eustratiadis, H. Gouk, D. Li, and T. M. Hospedales. Weight-covariance alignment for adversarially-robust neural networks. In *International Conference on Machine Learning (ICML)*, 2021.

J. Fang, Y. Sun, Q. Zhang, Y. Li, W. Liu, and X. Wang. Densely connected search space for more flexible neural architecture search. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

C. Finn and S. Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In *International Conference on Learning Representations (ICLR)*, 2018.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

C. Finn, K. Xu, and S. Levine. Probabilistic model-agnostic meta-learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. E. Turner. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations (ICLR)*, 2019.

H. Gouk and T. M. Hospedales. Optimising network architectures for provable adversarial robustness. In *Sensor Signal Processing for Defence Conference (SSPD)*, 2020.

H. Gouk, T. M. Hospedales, and M. Pontil. Distance-based regularisation of deep networks for fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2021.

S. Gowal, C. Qin, J. Uesato, T. A. Mann, and P. Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *CoRR*, abs/2010.03593, 2020.

Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun. Single path one-shot neural architecture search with uniform sampling. In *European Conference on Computer Vision (ECCV)*, 2020.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer, 2009.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Z. He, A. S. Rakin, and D. Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

M. Hein and M. Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

D. Hendrycks, K. Lee, and M. Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning (ICML)*, 2019.

T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey. Meta-learning in neural networks: A survey. *Transactions on Pattern Analysis and Machine Intelligence*, 2022.

J. Howard. Imagenette. *[Online]. Available: https://github.com/fastai/imagenette/*, 2019.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022a.

S. X. Hu, D. Li, J. Stühmer, M. Kim, and T. M. Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

A. Jeddi, M. J. Shafiee, M. Karg, C. Scharfenberger, and A. Wong. Learn2perturb: An end-to-end feature perturbation learning to improve adversarial robustness. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, 2022.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596, 2021.

A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23, 1952.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations (ICLR)*, 2017.

J. Lafferty, H. Liu, and L. Wasserman. Concentration of measure. *[Online]. Available: http://www.stat.cmu.edu/ larry/=sml/Concentration.pdf*, 2008.

B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum. One shot learning of simple visual concepts. In *Annual Meeting of the Cognitive Science Society*, 2011.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350, 2015.

Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2010.

M. Lécuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *Symposium on Security and Privacy*, 2019.

K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

S. Lee, H. Kim, and J. Lee. Graddiv: Adversarial robustness of randomized neural networks via gradient diversity regularization. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45, 2023.

S. Lefkimmiatis, J. P. Ward, and M. Unser. Hessian schatten-norm regularization for linear inverse problems. *Transactions on Image Processing*, 22, 2013.

B. Li, C. Chen, W. Wang, and L. Carin. Certified adversarial robustness with additive noise. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

C. Li, J. Peng, L. Yuan, G. Wang, X. Liang, L. Lin, and X. Chang. Block-wisely supervised neural architecture search with knowledge distillation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020a.

G. Li, G. Qian, I. C. Delgadillo, M. Müller, A. K. Thabet, and B. Ghanem. SGAS: sequential greedy architecture search. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020b.

W. Li, X. Liu, and H. Bilen. Universal representation learning from multiple domains for few-shot classification. In *International Conference on Computer Vision (ICCV)*, 2021.

W. Li, X. Liu, and H. Bilen. Cross-domain few-shot learning with task-specific adapters. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *International Joint Conference on Natural Language Processing, (IJCNLP)*, 2021.

Q. Liang and C. Chan. Improved adversarial robustness by hardened prediction. In *International Symposium on Information Theory, (ISIT)*, 2022.

J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2020.

H. Liu, K. Simonyan, and Y. Yang. DARTS: differentiable architecture search. In *International Conference on Learning Representations (ICLR)*, 2019a.

L. Liu, W. L. Hamilton, G. Long, J. Jiang, and H. Larochelle. A universal representation transformer layer for few-shot image classification. In *International Conference on Learning Representations (ICLR)*, 2021.

S. Liu, P. Chen, B. Kailkhura, G. Zhang, A. O. H. III, and P. K. Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *Signal Processing Magazine*, 37, 2020.

X. Liu, M. Cheng, H. Zhang, and C. Hsieh. Towards robust neural networks via random self-ensemble. In *European Conference on Computer Vision (ECCV)*, 2018.

X. Liu, Y. Li, C. Wu, and C. Hsieh. Adv-bnn: Improved adversarial defense through robust bayesian neural network. In *International Conference on Learning Representations (ICLR)*, 2019b.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, and B. Hodjat. Chapter 15 - evolving deep neural networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. Academic Press, 2019.

E. G. Miller, N. E. Matsakis, and P. A. Viola. Learning from one example through shared densities on transforms. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

P. Molchanov, J. Hall, H. Yin, J. Kautz, N. Fusi, and A. Vahdat. LANA: latency aware network acceleration. In *European Conference on Computer Vision (ECCV)*, 2022.

B. Moons, P. Noorzad, A. Skliar, G. Mariani, D. Mehta, C. Lott, and T. Blankevoort. Distilling optimal neural networks: Rapid search in diverse spaces. In *International Conference on Computer Vision (ICCV)*, 2021.

A. Mustafa, S. H. Khan, M. Hayat, R. Goecke, J. Shen, and L. Shao. Adversarial defense by restricting the hidden space of deep neural networks. In *International Conference on Computer Vision (ICCV)*, 2019.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

M. P. Niroomand, L. Dicks, E. O. Pyzer-Knapp, and D. J. Wales. Physics inspired approaches towards understanding gaussian processes. *CoRR*, abs/2305.10748, 2023.

J. W. Paisley, D. M. Blei, and M. I. Jordan. Variational bayesian inference with stochastic search. In *International Conference on Machine Learning (ICML)*, 2012.

T. Pang, K. Xu, and J. Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2020.

N. Papernot, P. D. McDaniel, and I. J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016.

N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Asia Conference on Computer and Communications Security, (AsiaCCS)*, 2017a.

N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Asia Conference on Computer and Communications Security (AsiaCCS)*, 2017b.

E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville. Film: Visual reasoning with a general conditioning layer. In *Conference on Artificial Intelligence (AAAI)*, 2018.

R. Pinot, L. Meunier, A. Araujo, H. Kashima, F. Yger, C. Gouy-Pailler, and J. Atif. Theoretical evidence for adversarial robustness through randomization. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

M. Pintor, L. Demetrio, A. Sotgiu, G. Manca, A. Demontis, N. Carlini, B. Biggio, and F. Roli. Indicators of attack failure: Debugging and improving optimization of adversarial examples. In *ICML Workshop on Adversarial Machine Learning*, 2021.

N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12, 1999.

R. Rade and S.-M. Moosavi-Dezfooli. Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

I. Radosavovic, J. Johnson, S. Xie, W. Lo, and P. Dollár. On network design spaces for visual recognition. In *International Conference on Computer Vision (ICCV)*, 2019.

A. Raghunathan, J. Steinhardt, and P. Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2017.

S. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

S. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. A. Mann. Fixing data augmentation to improve adversarial robustness. *CoRR*, abs/2103.01946, 2021.

J. Requeima, J. Gordon, J. Bronskill, S. Nowozin, and R. E. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations (ICLR)*, 2019.

V. B. S. and R. V. Babu. Single-step adversarial training with dropout scheduling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

T. Saikia, T. Brox, and C. Schmid. Optimized generic feature learning for few-shot classification across domains. *CoRR*, abs/2001.07926, 2020.

C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:60, 2019.

J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

C. Song, K. He, J. Lin, L. Wang, and J. E. Hopcroft. Robust local features for improving the generalization of adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020.

K. Sridhar, O. Sokolsky, I. Lee, and J. Weimer. Improving neural network robustness via persistency of excitation. *CoRR*, abs/2106.02078, 2021.

N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15, 2014.

J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *Transactions on Evolutionary Computation*, 23, 2019.

R. S. Sutton, D. A. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 1999.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola. Rethinking few-shot image classification: A good embedding is all you need? In *European Conference on Computer Vision (ECCV)*, 2020.

F. Tramèr, N. Carlini, W. Brendel, and A. Madry. On adaptive attacks to adversarial example defenses. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P. Manzagol, and H. Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations (ICLR)*, 2020.

E. Triantafillou, H. Larochelle, R. S. Zemel, and V. Dumoulin. Learning a universal template for few-shot dataset generalization. In *International Conference on Machine Learning (ICML)*, 2021.

Y. Tsuzuku, I. Sato, and M. Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

I. Ullah, D. Carrión-Ojeda, S. Escalera, I. Guyon, M. Huisman, F. Mohr, J. N. van Rijn, H. Sun, J. Vanschoren, and P. A. Vu. Meta-album: Multi-domain meta-dataset for few-shot image classification. In *Conference on Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2022.

O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

R. Wang, M. Cheng, X. Chen, X. Tang, and C. Hsieh. Rethinking architecture selection in differentiable NAS. In *International Conference on Learning Representations (ICLR)*, 2021a.

X. Wang and K. He. Enhancing the transferability of adversarial attacks through variance tuning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Y. Wang, W. Chao, K. Q. Weinberger, and L. van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *CoRR*, abs/1911.04623, 2019.

Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations (ICLR)*, 2020a.

Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations (ICLR)*, 2020b.

Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, 53, 2021b.

Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision (ECCV)*, 2016.

D. Wu, S. Xia, and Y. Wang. Adversarial weight perturbation helps robust generalization. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

L. Xiang, L. Dudziak, M. S. Abdelfattah, T. Chau, N. D. Lane, and H. Wen. Zero-cost operation scoring in differentiable architecture search. In *Conference on Artificial Intelligence (AAAI)*, 2023.

C. Xiao, P. Zhong, and C. Zheng. Enhancing adversarial defense by k-winners-take-all. In *International Conference on Learning Representations (ICLR)*, 2020.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

C. Xie, Y. Wu, L. van der Maaten, A. L. Yuille, and K. He. Feature denoising for improving adversarial robustness. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

C. Xu, S. Yang, Y. Wang, Z. Wang, Y. Fu, and X. Xue. Exploring efficient few-shot adaptation for vision transformers. *Transactions on Machine Learning Research (TMLR)*, 2022.

T. Yu, Y. Yang, D. Li, T. M. Hospedales, and T. Xiang. Simple and effective stochastic neural networks. In *Conference on Artificial Intelligence (AAAI)*, 2021.

S. Zagoruyko and N. Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*, 2016.

E. B. Zaken, Y. Goldberg, and S. Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Association for Computational Linguistics (ACL)*, 2022.

A. I. Zayed. The weierstrass transform. In *Handbook of Function and Generalized Function Transformations*. CRC Press, 1996.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014.

H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019a.

J. Zhang, C. Zhao, B. Ni, M. Xu, and X. Yang. Variational few-shot learning. In *International Conference on Computer Vision (ICCV)*, 2019b.

J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. S. Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations (ICLR)*, 2021a.

X. Zhang, D. Meng, H. Gouk, and T. M. Hospedales. Shallow bayesian meta learning for real-world few-shot recognition. In *International Conference on Computer Vision (ICCV)*, 2021b.

Y. Zhang, K. Zhou, and Z. Liu. Neural prompt search. *CoRR*, abs/2206.04673, 2022.

D. Zhou, X. Jin, X. Lian, L. Yang, Y. Xue, Q. Hou, and J. Feng. Autospace: Neural architecture search with less human interference. In *International Conference on Computer Vision (ICCV)*, 2021.

Appendix A

Weight-Covariance Alignment for Adversarially-Robust Neural Networks

A.1 Proof of Theorem 1

Proof. The definition of h can be expanded to

$$h(\vec{x}) = \vec{w}^T f(\vec{x}) + \vec{w}^T \vec{z} + b, \quad \vec{z} \sim \mathcal{N}(0, \Sigma), \tag{A.1}$$

and be reinterpreted as

$$h(\vec{x}) \sim \mathcal{N}(\vec{w}^T f(\vec{x}) + b, \vec{w}^T \Sigma \vec{w}). \tag{A.2}$$

Going further, we can see that the distribution of the margin function is

$$m_h(\vec{x}, y) \sim \mathcal{N}(y(\vec{w}^T f(\vec{x}) + b), \vec{w}^T \Sigma \vec{w}), \tag{A.3}$$

for which the probability of being less than zero is given by the cumulative distribution function for the normal distribution,

$$P(m_h(\vec{x}, y) < 0) = \Phi\left(\frac{-y(\vec{w}^T f(\vec{x}) + b)}{\sqrt{\vec{w}^T \Sigma \vec{w}}}\right). \tag{A.4}$$

From the increasing monotonicity of Φ, we also have that

$$\max_{\delta : \|\vec{\delta}\|_p \leq \epsilon} \Phi\left(\frac{-y(\vec{w}^T f(\vec{x} + \delta) + b)}{\sqrt{\vec{w}^T \Sigma \vec{w}}}\right) = \Phi\left(\frac{\max_{\delta : \|\vec{\delta}\|_p \leq \epsilon} -y(\vec{w}^T f(\vec{x} + \delta) + b)}{\sqrt{\vec{w}^T \Sigma \vec{w}}}\right). \tag{A.5}$$

Suppose the adversarial perturbation, δ, causes the output of the non-stochastic version of h to change by a magnitude of $\Delta_p^{\tilde{h}}(\vec{x}, \epsilon)$. There are a number of ways, such as local Lipschitz constants (Tsuzuku et al., 2018; Gouk and Hospedales,

Table A.1: Values for learning rate and weight decay for all experiments in our ablation study.

Benchmark	Learning rate	Weight decay
CIFAR-10	10^{-2}	10^{-4}
CIFAR-100	10^{-2}	10^{-4}
SVHN	10^{-2}	10^{-4}
FMNIST	10^{-4}	10^{-4}

2020), that can be used to bound the quantity for simple networks. Substituting $\Delta_p^{\tilde{h}}$ into the previous equation yields

$$\max_{\vec{\delta}:\|\vec{\delta}\|_p \leq \epsilon} P(m_h(\vec{x}+\delta, y) \leq 0) \leq \Phi\left(\frac{-y(\vec{w}^T f(\vec{x})+b)+\Delta_p^{\tilde{h}}(\vec{x},\epsilon)}{\sqrt{\vec{w}^T \Sigma \vec{w}}}\right). \tag{A.6}$$

Finally, we know that the difference in probabilities of misclassification when the model is and is not under adversarial attack δ, is given by

$$G_{p,\epsilon}^h(\vec{x}, y) = \max_{\vec{\delta}:\|\vec{\delta}\|_p \leq \epsilon} P(m_h(\vec{x}+\delta, y) \leq 0) - P(m_h(\vec{x}, y) \leq 0). \tag{A.7}$$

Combining Equations A.4 and A.6 with Equation A.7 results in

$$G(\vec{x}, y) \leq \Phi\left(\frac{-y(\vec{w}^T f(\vec{x})+b)+\Delta_p^{\tilde{h}}(\vec{x},\epsilon)}{\sqrt{\vec{w}^T \Sigma \vec{w}}}\right) - \Phi\left(\frac{-y(\vec{w}^T f(\vec{x})+b)}{\sqrt{\vec{w}^T \Sigma \vec{w}}}\right). \tag{A.8}$$

Because the Lipschitz constant of Φ is $\frac{1}{\sqrt{2\pi}}$, we can further bound G by

$$G(\vec{x}, y) \leq \frac{\Delta_p^{\tilde{h}}(\vec{x},\epsilon)}{\sqrt{2\pi \vec{w}^T \Sigma \vec{w}}}. \tag{A.9}$$

□

A.2 Hyperparameters of Experiments

In Table A.1, we provide the hyperparameter setup for all the experiments in our ablation study. Note that we use the same values for both the isotropic and anisotropic variants of our model within the same benchmark. We further clarify that we use a batch size of 128 across all experiments. To choose these values, we split the training data into a training and a validation set and performed grid search. The grid consisted of negative powers of 10 $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ for both hyperparameters.

Table A.2: PGD test scores on CIFAR-10 using WRN-34-10, for different values of attack strength ϵ.

PGD($\epsilon/255$)	Clean	1	2	4	8	16	32	64	128
No Defence	0.97	0.63	0.60	0.26	0.12	0	0	0	0
WCA-Net	0.97	0.80	0.80	0.77	0.73	0.70	0.34	0.10	0

A.3 Larger Architectures

In the main body of the thesis we explore how our method scales with the size of the backbone's architecture by experimenting with LeNet++ (small, 60 thousand parameters) and ResNet-18 (medium, 11 million parameters). In Table A.2 we also provide some experimental results on CIFAR-10 with the much larger Wide-ResNet-34-10 architecture (46 million parameters)

A.4 Enforcing Norm Constraints

In Section 3.2.1 we elaborate on how we use an ℓ^2 penalty to prevent the magnitude of the classifier vectors $\vec{w}$ and covariance matrix Σ from increasing uncontrollably. Another approach for controlling the magnitude of the parameters, is enforcing norm constraints after each gradient descent update, using a projected subgradient method. The projected subgradient method changes the standard update rule of the subgradient method from

$$\vec{\theta}^{(t+1)} \leftarrow \vec{\theta}^{(t)} - \alpha \nabla_{\vec{\theta}} \mathcal{L}(\vec{\theta}^{(t)}), \tag{A.10}$$

to

$$\vec{u}^{(t)} \leftarrow \vec{\theta}^{(t)} - \alpha \nabla_{\vec{\theta}} \mathcal{L}(\vec{\theta}^{(t)}) \tag{A.11}$$

$$\vec{\theta}^{(t+1)} \leftarrow \arg\min_{\vec{v} \in \Omega} \|\vec{v} - \vec{u}^{(t)}\|_2^2, \tag{A.12}$$

where Ω is known as the feasible set. In our case there are three sets of parameters: the feature extractor weights, the linear classifier weights, and the covariance matrix. No projection needs to be applied to the extractor weights, as they are unconstrained. The linear classifier weights have an ℓ^2 constraint on the vector associated with each class, so their feasible set it an ℓ^2 ball—there is a known closed form projection onto the ℓ^2 ball (e.g., Gouk et al., 2021). The feasible set for the covariance matrix is the set of positive semi-definite matrices with bounded singular values. This constraint can be enforced by performing a singular value decomposition on the updated covariance matrix, clipping the values to the appropriate threshold, and reconstructing the new projected covariance matrix (Lefkimmiatis et al., 2013). The final algorithm is given by

$$\begin{align}
Y^{(t)} &\leftarrow \vec{\Sigma}^{(t)} - \alpha \nabla_{\Sigma} \mathcal{L}(\vec{\phi}^{(t)}, \vec{w}^{(t)}, L^{(t)}) \nonumber \\
\vec{u}_i^{(t)} &\leftarrow \vec{w}_i - \alpha \nabla_{\vec{w}_i} \mathcal{L}(\vec{\phi}^{(t)}, \vec{w}^{(t)}, L^{t}) \nonumber \\
\vec{\phi}^{(t+1)} &\leftarrow \vec{\phi}^{(t)} - \alpha \nabla_{\vec{\phi}} \mathcal{L}(\vec{\phi}^{(t)}, \vec{w}^{(t)}, L^{t}) \nonumber \\
\vec{w}_i^{(t+1)} &\leftarrow \frac{1}{\text{max}(1, \frac{\|\vec{u}_i^{(t)}\|_2}{\gamma})} \vec{u}_i^{(t)} \nonumber \\
U^{(t)} S^{(t)} V^{(t)} &\leftarrow Y^{(t)T} Y^{(t)} \tag{A.13} \\
\Sigma^{(t)} &\leftarrow U^{(t)} \tilde{S}^{(t)} V^{(t)} \nonumber \\
L^{(t+1)T} L^{(t+1)} &\leftarrow \Sigma^{(t)}, \tag{A.14}
\end{align}$$

where (A.13) is performing a singular value decomposition, $\tilde{S}$ represents the clipped version of S, and (A.14) is computing the Cholesky decomposition.

Appendix B

Attacking Adversarial Defences by Smoothing the Loss Landscape

B.1 Proof of Theorem 2

Proof. The proof is based on using a Bernstein inequality. Let Z_1, ..., Z_m be independent random variables taking positive values in $[a, b]$, and let $S = \frac{1}{m}\sum_i^m Z_i$. From (Lafferty et al., 2008), Bernstein's inequality tells that

$$P(|S - \mathbb{E}[S]| > t) \leq 2\text{exp}\left(\frac{-mt^2}{2\text{Var}[S] + \frac{2}{3}rt}\right) , \tag{B.1}$$

where $r = b - a$. By setting $\delta = P(|S - \mathbb{E}[S]| > t)$ this can be rearranged to show that, with probability at least $1 - \delta$,

$$|S - \mathbb{E}[S]| \leq \sqrt{\frac{2\text{Var}[S]\text{ln}(1/\delta)}{m}} + \frac{2r\text{ln}(1/\delta)}{3m} . \tag{B.2}$$

The result follows from using $Z_i = \mathcal{L}(h_\theta(X_i), c)$ and upper bounding Var$[S]$ and r. Because h_θ is k-Lipschitz and $\mathcal{L}$ is L-Lipschitz on the co-domain of h_θ, we can say that $\mathcal{L}(h_\theta(\cdot), \cdot)$ is kL-Lipschitz. From this Lipschitz property, we know that $b \leq a + kL$, and therefore $r \leq kL$.

Denote by X_i' and S' random variables that follow the same distribution as X_i and S, respectively. The bound for the variance arises from

$$\begin{aligned}
&\text{Var}[S] && \text{(B.3)}\\
&= \mathbb{E}_S[(\mathbb{E}_{S'}[S'] - S)^2] && \text{(B.4)}\\
&\leq \mathbb{E}_{X_i}\mathbb{E}_{X_i'}\left[\left(\frac{1}{m}\sum_{i=1}^{m}(\mathcal{L}(h_\theta(X_i'), c) - \mathcal{L}(h_\theta(X_i), c))\right)^2\right] && \text{(B.5)}\\
&\leq \mathbb{E}_{X_i}\mathbb{E}_{X_i'}\left[\|X_i' - X_i\|_2^2 k^2 L^2\right] && \text{(B.6)}\\
&= 2k^2L^2d\sigma^2, && \text{(B.7)}
\end{aligned}$$

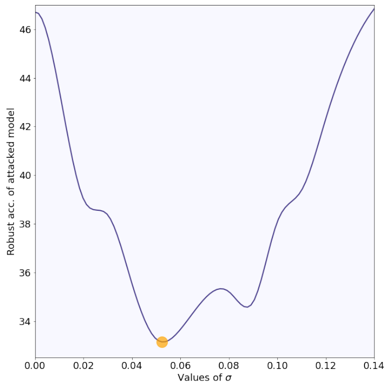


Figure B.1: Sensitivity study of σ. If the value of σ is either too low or too high, the attack is not as effective. The local minima in this curve are caused by randomness and are slightly different in each execution, while the global minima are always in the ballpark of $\sigma = 0.05$.

where the first inequality is due to Jensen's inequality, and the second is from the Lipschitz property of the model. The final equality arises because $X' - X \sim \mathcal{N}(0, 2\sigma^2 I)$, and the expected value of the squared Euclidean norm of a sample from a Gaussian distribution is the trace of the covariance matrix. □

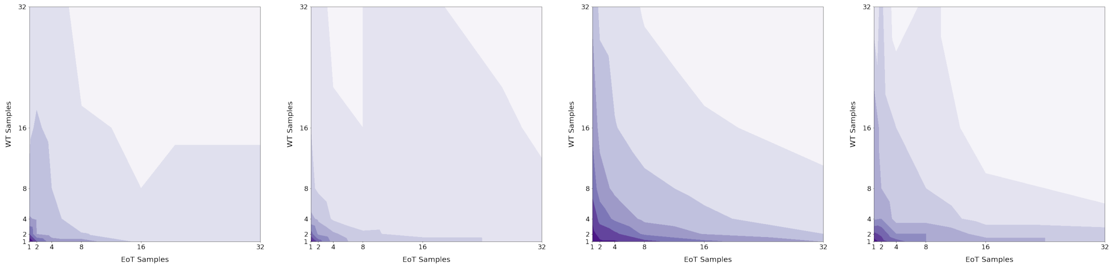
B.2 Experimental Setup: Hyperparameters

For PGD$_{\text{WT}}$, we set an attack strength of $\epsilon = 8/255$ and a step size of $\alpha = 0.01$, as is standard practice. For ZOO$_{\text{WT}}$ we set $k = 100$ and $\alpha = 0.01$. The number of WT samples and EoT iterations in our main experiments are both set to $m = n = 16$. We justify this hyperparameter choice in the analysis of Appendix B.3. Finally, selecting an appropriate value for σ is important. If the value of σ is too high, then the WT samples will be too far from x, lying on points too dissimilar to x to provide an informative gradient signal. If the value of σ is too low, the sampled points will be too close to x, and there will be no smoothing effect. We found that $\sigma = 0.05$ is a suitable value for normalized images, and use it across all experiments. Fig. B.1 summarises our sensitivity study on σ.

It should be mentioned that in the case of AA we do not apply EoT, as it is not a stochastic defence and therefore does not produce stochastic gradients. In addition, all stochastic models evaluated in this paper are retrained, following the instructions in the original published material, when available. As a result, the accuracy scores may not exactly reflect the scores from the original papers.

B.3 Ablation Study: Selection of m and n

We also conduct an experiment using a grid of EoT and WT samples from {1, 2, 4, 8, 16, 32}. Fig. B.2 presents an overhead plot of the resulting network



(a) PNI (He et al., 2019) (b) L2P (Jeddi et al., 2020) (c) SE-SNN (Yu et al., 2021) (d) WCA (Eustratiadis et al., 2021)

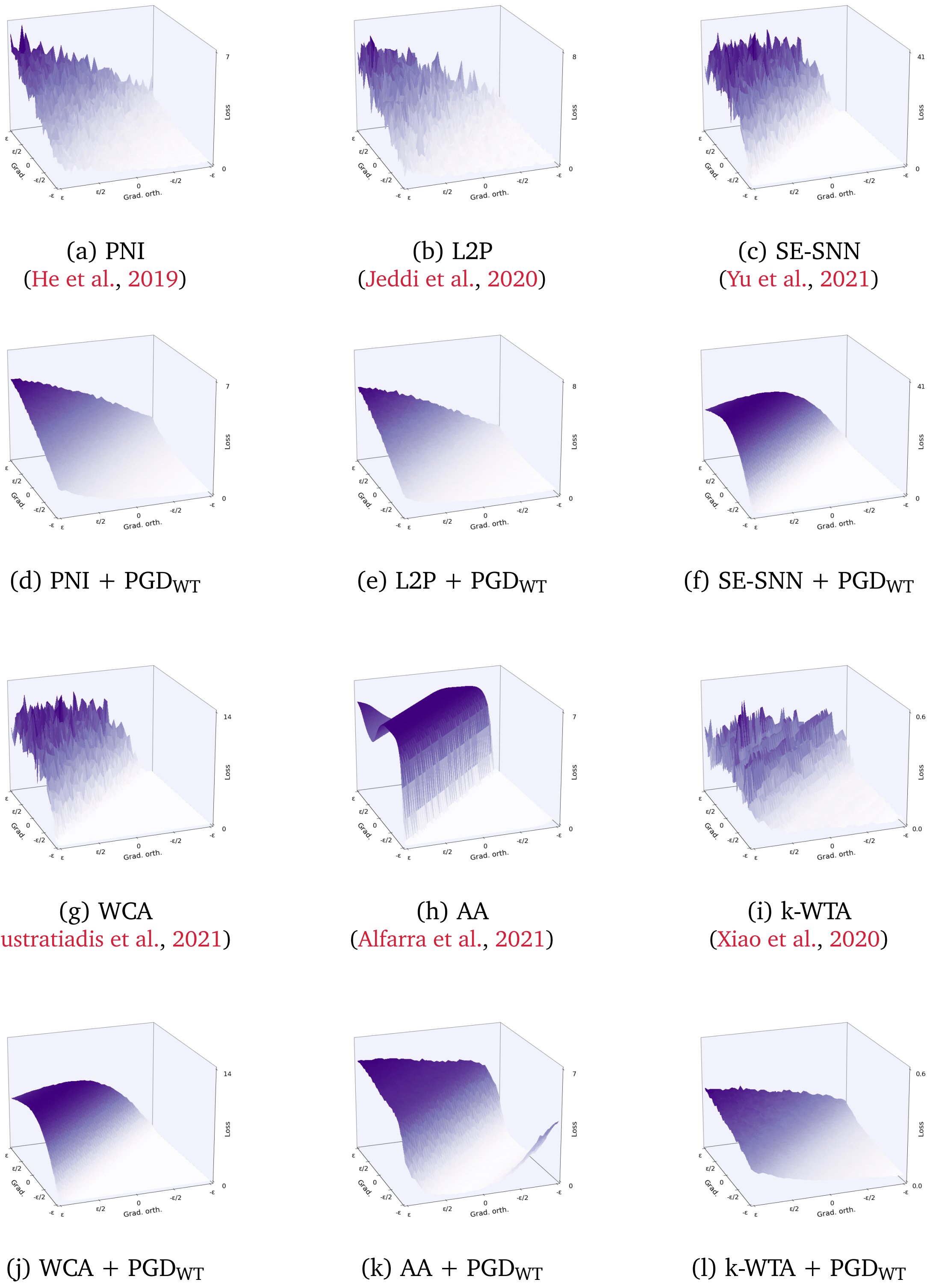
Figure B.2: Analysis of the interaction between WT and EoT on stochastic defences. WT and EoT are complementary. Neither can achieve peak performance alone, and best performance requires combining them (lighter color = lower accuracy).

accuracy as a function of number of samples for each of EoT and WT. Darker colors indicate higher accuracy, starting from the point (1, 1), i.e., 1 iteration of EoT and 1 WT sample (the input image itself). We see that: (i) After (16, 16) the performance of the attack quickly saturates across all defences. This justifies our use of $m = n = 16$ samples in the main experiment. (ii) Even at the limit of 32 samples, neither attack method on its own performs as well as their combination. This shows that simply increasing the number of EoT samples can not replicate the effect of WT (and vice-versa).

B.4 Strong Defences with Smooth Loss Landscapes

In the main body of the thesis, we see the effect of our attack on gradient-obfuscating adversarial defences that construct a noisy loss landscape to confuse the adversary. To further support future adversarial defence research, in this section we want to inform the reader about how the loss landscapes of non-obfuscating defences should look like.

To that end, we choose the 9 highest-scoring adversarial defences from the ℓ_∞ CIFAR-10 leaderboard of the widely used RobustBench (Croce et al., 2021) and visualise their loss landscapes in Fig. B.4. The visualisation method is the same that produced Fig. 2.2; except that none of the defences are stochastic and therefore EoT is not used to obtain better gradient estimates.



(a) PNI (He et al., 2019)

(b) L2P (Jeddi et al., 2020)

(c) SE-SNN (Yu et al., 2021)

(d) PNI + PGD_{WT}

(e) L2P + PGD_{WT}

(f) SE-SNN + PGD_{WT}

(g) WCA (Eustratiadis et al., 2021)

(h) AA (Alfarra et al., 2021)

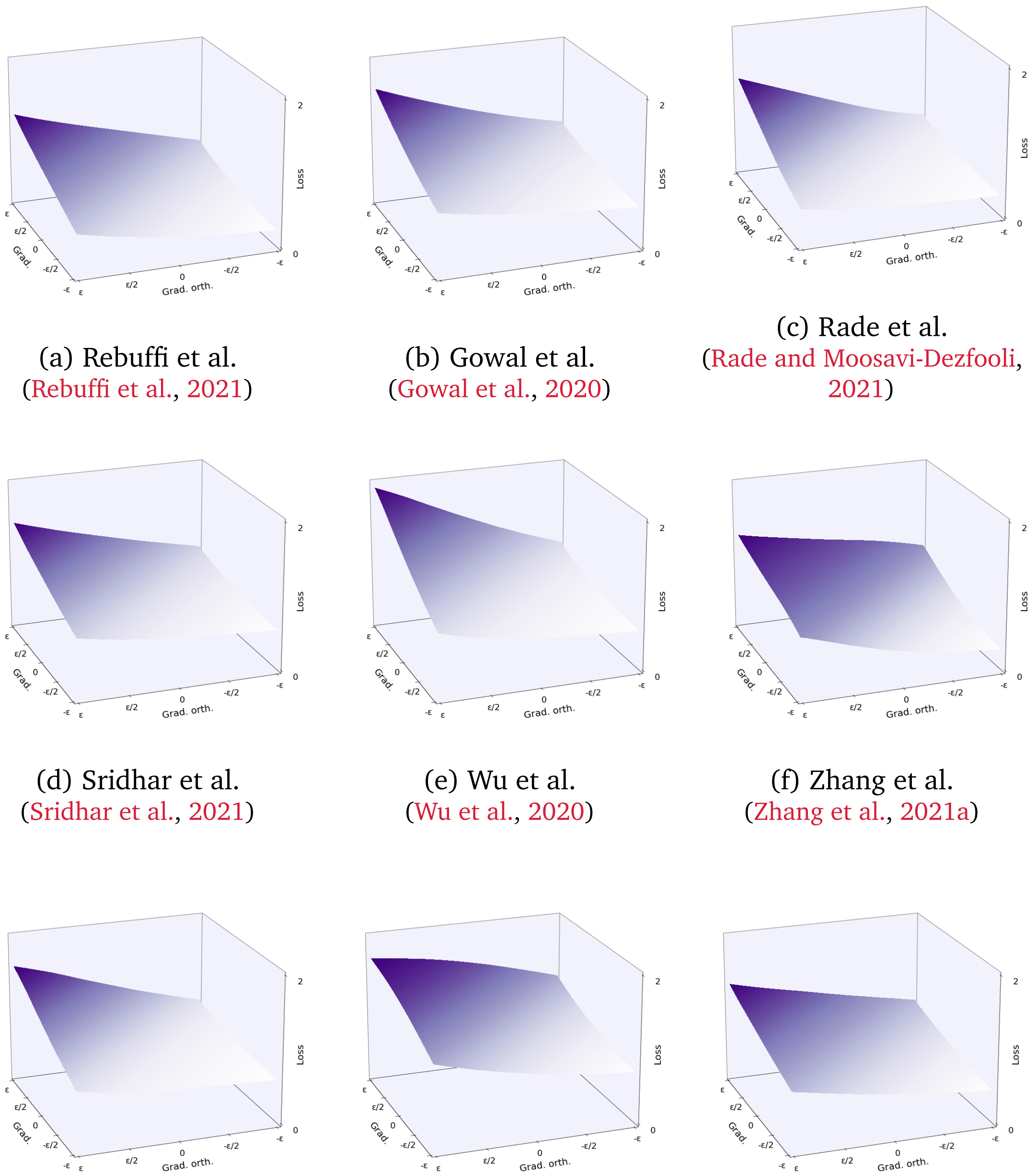
(i) k-WTA (Xiao et al., 2020)

(j) WCA + PGD_{WT}

(k) AA + PGD_{WT}

(l) k-WTA + PGD_{WT}

Figure B.3: Loss landscapes of PNI, L2P, SE-SNN, WCA, AA and k-WTA when under attack by PGD_{WT}. The WT has smoothed the landscapes compared to those shown in Fig. 2.2.



(a) Rebuffi et al. (Rebuffi et al., 2021)

(b) Gowal et al. (Gowal et al., 2020)

(c) Rade et al. (Rade and Moosavi-Dezfooli, 2021)

(d) Sridhar et al. (Sridhar et al., 2021)

(e) Wu et al. (Wu et al., 2020)

(f) Zhang et al. (Zhang et al., 2021a)

(g) Carmon et al. (Carmon et al., 2019)

(h) Wang et al. (Wang et al., 2020b)

(i) Hendrycks et al. (Hendrycks et al., 2019)

Figure B.4: Landscapes of non-obfuscating adversarial defences that score competitively on RobustBench (Croce et al., 2021).

Appendix C

Neural Fine-Tuning Search for Few-Shot Learning

C.1 Hyperparameter Setting

Table C.1 reports the hyperparameters used for all of our experiments. Note the following clarifications:

- "Number of epochs" refers to multiple forward passes of the same episode, while "Number of episodes" refers to the number of episodes sampled in total.
- The batch size is not mentioned, because we only conduct episodic learning, where we do not split the episode into batches, i.e., we feed the entire support and query set into our neural network architectures.
- Learning rate warmup, where applicable, occurs for the first 10% of the episodes.

We further specify something important: While our strongest competitors Li et al. (2022); Xu et al. (2022) tune their learning rates for meta-testing (e.g., TSA uses LR=0.1 for seen domains and LR=1.0 for unseen, and ETT uses a different learning rate per downstream Meta-Dataset domain), we treat meta-testing episodes as completely unknown, and use the same hyperparameters we used on the validation set during search.

C.2 Detailed Ablation Study

Tables C.2 and C.3 provide the exact scores per Meta-Dataset domain that are summarised in Table 6.5 in the main body of the thesis, for single domain and multi-domain FSL respectively.

		ResNet-18			ViT-S	
	Hyperparameter	SDL (MD)	MDL (MD)	MDL (MA)	SDL (MD)	MDL (MD)
	Backbone architecture	URL	URL	Supervised	DINO	DINO
	Adapter architecture	TSA	TSA	TSA	ETT	ETT
TRAIN	Number of episodes	50000	80000	20000	80000	160000
	Number of epochs	1	1	1	1	1
	Optimizer	adadelta	adadelta	adadelta	adamw	adamw
	Learning rate	0.05	0.05	0.05	0.00007	0.00007
	Learning rate schedule	-	-	-	cosine	cosine
	Learning rate warmup	-	-	-	linear	linear
	Weight decay	0.0001	0.0001	0.0001	0.01	0.01
	Weight decay schedule	-	-	-	cosine	cosine
SEARCH	Number of episodes	100	100	100	100	100
	Number of epochs	20	20	20	40	40
	Optimizer	adadelta	adadelta	adadelta	adamw	adamw
	Learning rate	0.1	0.1	0.1	0.000003	0.000003
	Weight decay	0.0001	0.0001	0.0001	0.1	0.1
	Initial population size	64	64	64	64	64
	Top-K crossover	8	8	8	8	8
	Mutation chance	5%	5%	5%	5%	5%
	Top-N paths	3	3	3	3	3
	Diversity threshold	0.4	0.4	0.4	0.2	0.2
TEST	Number of episodes	600	600	1800	600	600
	Number of epochs	40	40	40	40	40
	Optimizer	adadelta	adadelta	adadelta	adamw	adamw
	Learning rate	0.1	0.1	0.1	0.000003	0.000003
	Weight decay	0.0001	0.0001	0.0001	0.1	0.1
	Regulariser strength	0.04	0.04	0.04	-	-

Table C.1: Hyperparameter setting for all experiments presented in Section 6.3, in the main body of the thesis. The notation is as follows: SDL=Single domain learning, MDL=Multi-domain learning, MD=Meta-Dataset, MA=Meta-Album, TRAIN=Supernet training phase, SEARCH=Evolutionary search phase, TEST=Meta-test phase.

	Method	Aircrafts	Birds	Textures	Fungi	ImageNet	Omniglot	QuickDraw	Flowers	CIFAR-10	CIFAR-100	MNIST	MS COCO	Traffic Signs	Average
ResNet-18	$\phi\,,-$	64.5	69.6	71.1	41.2	56.4	74.8	64.2	84.6	75.0	63.9	82.1	55.9	77.7	67.8
	$\phi\,,\alpha$	69.6	67.7	75.0	42.5	59.5	71.3	64.9	88.8	77.4	70.0	90.2	58.4	80.1	70.4
	$\phi',-$	69.9	74.7	73.3	39.5	57.3	71.9	65.4	89.0	76.5	66.3	93.6	54.4	81.4	70.2
	ϕ',α	67.6	69.1	77.0	39.3	59.7	77.8	66.1	87.4	81.7	69.5	91.9	55.1	78.7	70.8
	NFTS-1	73.2	76.5	81.6	42.1	61.3	80.2	66.9	90.0	82.9	68.8	94.0	58.4	80.6	73.6
	NFTS-N	74.9	76.5	81.6	50.5	62.7	80.2	67.2	94.5	83.0	71.5	94.0	59.7	81.9	75.2
ViT-S	$\phi\,,-$	73.4	73.6	81.6	56.3	60.3	69.4	70.8	90.4	70.4	61.5	83.8	60.5	81.7	71.8
	$\phi\,,\alpha$	76.9	83.2	86.7	59.3	63.7	75.8	65.1	89.5	70.7	67.4	81.1	54.8	82.9	73.8
	$\phi',-$	76.8	80.9	85.8	61.4	65.9	73.2	68.5	91.0	69.9	66.1	82.5	57.6	78.8	74.0
	ϕ',α	77.0	83.4	82.4	58.6	66.7	73.1	65.0	95.9	76.7	66.1	87.7	58.7	82.9	74.4
	NFTS-1	83.0	85.5	87.3	62.2	68.8	81.9	72.9	95.3	79.4	72.6	95.2	62.6	87.5	78.7
	NFTS-N	83.0	85.5	87.6	62.2	71.0	81.9	74.5	96.0	79.4	72.6	95.2	62.6	87.9	79.2

Table C.2: Ablation study on Meta-Dataset comparing four special cases of the search space: (i) $\phi, -$: No adaptation, no fine-tuning, (ii) ϕ, α: Adapt all, (iii) $\phi', -$: Fine-tune all, (iv) ϕ', α: Adapt and fine-tune all. NFTS-{1,N} refer to conventional and deferred episode-wise NAS respectively. Single domain setting: Only ImageNet is seen during training and search. Reporting mean accuracy over 600 episodes.

	Method	Aircrafts	Birds	Textures	Fungi	ImageNet	Omniglot	QuickDraw	Flowers	CIFAR-10	CIFAR-100	MNIST	MS COCO	Traffic Signs	Average
ResNet-18	$\phi, -$	64.5	69.6	71.1	41.2	56.4	74.8	64.2	84.6	75.0	63.9	82.1	55.9	77.7	67.8
	ϕ, α	89.3	78.3	76.1	62.7	57.2	93.8	76.0	90.8	77.8	66.1	90.5	56.9	79.5	76.5
	$\phi', -$	90.2	76.7	70.6	63.1	57.8	88.2	79.3	88.9	78.2	68.2	96.1	51.7	82.9	76.3
	ϕ', α	86.1	78.9	77.2	60.5	57.6	94.1	79.5	86.5	81.0	67.2	96.1	52.6	81.8	76.9
	NFTS-1	90.1	82.1	79.9	67.9	61.4	94.3	82.6	92.2	82.4	73.8	95.4	58.1	81.0	80.1
	NFTS-K	90.1	83.8	82.3	68.4	61.4	94.3	82.6	92.2	83.0	75.1	95.4	58.8	81.9	80.7
ViT-S	$\phi, -$	73.4	73.6	81.6	56.3	60.3	69.4	70.8	90.4	70.4	61.5	83.8	60.5	81.7	71.8
	ϕ, α	85.7	84.3	81.8	68.7	70.4	89.1	77.0	90.2	73.5	61.4	82.6	53.7	72.4	77.3
	$\phi', -$	83.0	84.5	81.1	70.9	72.4	88.6	74.6	90.4	75.1	63.5	87.0	54.0	75.5	77.5
	ϕ', α	82.5	85.9	82.7	68.9	73.7	90.4	77.1	94.0	73.4	66.2	85.9	55.9	77.4	78.9
	NFTS-1	89.1	90.3	86.3	75.1	74.6	92.0	80.6	93.5	75.9	70.8	91.3	62.7	87.2	83.1
	NFTS-N	89.1	92.5	86.3	75.1	74.6	92.0	80.6	93.5	75.9	70.8	91.3	62.8	87.2	83.4

Table C.3: Ablation study on Meta-Dataset comparing four special cases of the search space: (i) $\phi, -$: No adaptation, no fine-tuning, (ii) ϕ, α: Adapt all, (iii) $\phi', -$: Fine-tune all, (iv) ϕ', α: Adapt and fine-tune all. NFTS-{1,N} refer to conventional and deferred episode-wise NAS respectively. Multi-domain setting: The first 8 datasets are seen during training and search. Reporting mean accuracy over 600 episodes.