MONASH University

Adversarial Regularisation and Knowledge Distillation for Deep Learning Tasks.

Duc Van Thanh Nguyen

Doctor of Philosophy

A Thesis Submitted for the Degree of Doctor of Philosophy at
Monash University in 2023
School of Information Technology

Copyright notice

Abstract

Deep learning has made remarkable progress in various applications. In this thesis, we aim to advance robust machine learning, deep semi-supervised learning, and model compression tasks. We improve these learning tasks by developing adversarial regularisation and knowledge distillation. The first objective is to gain an understanding of adversarial attacks and knowledge distillation. Through this exploration, we provide valuable interpretations and insights into both the adversarial attack mechanism and the knowledge distillation process. The second objective is to develop and enhance adversarial regularisation and knowledge distillation technique. This involves investigating the principle of adversarial regularisation and knowledge distillation to improve robust deep learning, deep semi-supervised learning and model compression tasks. Our key contributions are as follows:

- We enhance understanding and refine perturbations in adversarial attacks. This research proposes a vulnerability map to identify the most susceptible areas of an image to adversarial perturbations, refining existing attack methods to make them less noticeable while still causing misclassifications.

- We develop a novel knowledge distillation and model interpretation framework for image classification that jointly solves the model interpretation and knowledge distillation. To interpret the teacher model as well as assist the learning of the student, an explainer module is introduced to highlight the regions of an input image that are important for the predictions of the teacher model.

- We contribute to robust deep learning and deep semi-supervised learning tasks by introducing a novel adversarial local distribution (ALD) regularisation. The ALD is defined by a set of all adversarial examples within a ball constraint, which is approximated by Stein Variational Gradient Descent. We illustrate this regularisation is a general form of previous methods (e.g., PGD, TRADES, and VAT), and shows significant performance improvement on the benchmark datasets.

- We extend the ALD for the semi-supervised segmentation task. Cross-adversarial local distribution (Cross-ALD) regularisation is proposed by mixing two ALDs to enhance the smoothness assumption in the semi-supervised segmentation task. The Cross-ALD achieves state-of-the-art performance in medical imaging datasets.

- We extend the ALD for the knowledge distillation task by proposing a teacher adversarial local distribution (TALD). This TALD is used to sufficiently explore the decision boundaries of the teacher using adversarial examples. The student model decision boundaries are then regularised by matching the loss between teacher

and student using these adversarial example inputs. We conducted comprehensive experiments on CIFAR-100 and Imagenet datasets to illustrate this TALD regularisation can be applied to improve the performance of many existing knowledge distillation methods (e.g., KD, FitNet, CRD, VID).

In summary, this thesis advances robust deep learning, deep semi-supervised learning, and model compression tasks by exploiting and developing the principles of knowledge distillation and adversarial regularisation. In the future, we plan to broaden the base of adversarial local distribution regularisation across various deep learning applications.

Declaration

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature: _____

Print Name: Duc Van Thanh Nguyen

Date: 24-Aug-2023

Publications during enrolment

Some parts of this thesis are the extended or modified versions of the conference or journal papers that were published or are under review as follows.

Chapter 3:

- **Thanh Nguyen-Duc**, He Zhao, Trung Le, Paul Montague, Olivier De Vel, Tamas Abraham, and Dinh Phung. "Learning to Attack with Fewer Pixels: A Probabilistic Post-hoc Framework for Refining Arbitrary Dense Adversarial Attacks". Submitted to *Neurocomputing. 2023*

Chapter 4:

- **Thanh Nguyen-Duc**, He Zhao, Jianfei Cai, and Dinh Phung. "MED-TEX: Transfer and Explain Knowledge with Less Data from Pretrained Medical Imaging Models." In proceedings of *the IEEE International Symposium on Biomedical Imaging (ISBI). 2022.*

Chapter 5:

- **Thanh Nguyen-Duc**, Trung Le, He Zhao, Jianfei Cai, and Dinh Phung. "Particle-based Adversarial Local Distribution Regularization." In proceedings of *the Artificial Intelligence and Statistics (AISTAT). 2022.*

Chapter 6:

- **Thanh Nguyen-Duc**, Trung Le, He Zhao, Jianfei Cai, and Dinh Phung. "Cross-adversarial Local Distribution Regularization for semi-supervised medical image segmentation." To be appeared in proceedings of *the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI). 2023.*

Chapter 7:

- **Thanh Nguyen-Duc**, Trung Le, He Zhao, Jianfei Cai, and Dinh Phung. "Adversarial Local Distribution Regularization for Knowledge Distillation." In proceedings of *the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). 2023.*

Acknowledgements

I would like to express my deepest appreciation to all those who provided me with the opportunity to complete this thesis.

First and foremost, I am immensely grateful to my parents (Nguyen Duc Linh and Le Thi Nga) and siblings (Nguyen Thi Ngoc Chanh, Nguyen Thi Ngoc Huong, and Nguyen Duc Van Buu). Your unfaltering faith in my abilities has been a beacon of hope that guided me through the journey of this thesis. Your patience, understanding, and love were constant sources of strength and motivation that propelled me towards the completion of this academic milestone.

I also owe a debt of gratitude to my supervisors (Prof. Dinh Phung, Prof. Jianfei Cai, and Dr. He Zhao), Prof. Roland Bammer, Dr. Trung Le, and fellow friends in research groups, who have been an integral part of this journey. Your unwavering support and encouragement, coupled with the valuable discussions, immensely contributed to the successful completion of this work.

I am deeply appreciative of their time and constructive suggestions from the thesis examiners, A/Prof. Sarah Monazam Erfani and A/Prof. Hady Wirawan Lauw.

Furthermore, this thesis is supported by Monash Central Scholarship, the Australian Defence Science and Technology (DST) Group under the Next Generation Technology Fund (NGTF) scheme, and the Monash International Travel Campus Scholarship.

Finally, I would like to acknowledge the endless support, patience, and kindness that I received from all those who were involved in my journey in ways that words cannot express.

Thank you all for making this journey not just possible, but truly remarkable.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

ADT	Adverasrial Distributional Training
AdvACC	Adversarial Accuracy
ALD	Adversarial Local Distribution
ATD	Adversarial Training Defense
AI	Artificial Intelligence
BIM	Basic Iterative Method
BiGAN	Bidirectional Generative Adversarial Network
CatGAN	Categorical Generative Adversarial Network
CAM	Class Activation Map
CNN	Convolutional Neural Network
Cross-ALD	Cross Adversarial Local Distribution
CE	Cross-Entropy
DNN	Deep Neural Network
DSSL	Deep semi-supervised learning
DetAUC	Detection Area Under Curve
EntMin	Entropy Minimization
XAI	Explainable AI
FGSM	Fast Gradient Sign Method
GAN	Generative Adversarial Network
ICT	Interpolation Consistency Training
JSMA	Jacobian-based Saliency Map Attack
KD	Knowledge distillation
L2X	Learning to Explain
MED-TEX	Medical Transferring and Explaining Framework
MPL	Meta Pseudo Label

PVAR	Pixel Vulnerability Adversary Refinement
PGD	Projected Gradient Descent
RBF	Radial Basic Function
ResNet	Residual Neural Network
SVGDF	Semantic Feature Stein Variational Gradient Decent
SSL	Semi-Supervised Learning
SVGD	Stein Variational Gradient Decent
TALD	Teacher Adversarial Local Distribution
TRADES	Tradeoff-inspired Adversarial Defense via Surrogate-loss
VAT	Virtual Adversarial Training

Notations

$\boldsymbol{x}, \boldsymbol{X}$	Vector and matrix input
$y, \boldsymbol{y}$	Scalar and one-hot vector label
$\mathbb{D}$	Dataset
$f : \mathbb{A} \to \mathbb{B}$	The function f with domain $\mathbb{A}$ and range $\mathbb{B}$
$\ell(f(\boldsymbol{x}), \boldsymbol{y})$	Loss function of $\boldsymbol{x}$ and $\boldsymbol{y}$.
$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim P_{\mathbb{D}}}\ell(f(\boldsymbol{x}), \boldsymbol{y})$	Expectation of ℓ with respect to dataset distribution $P_{\mathbb{D}}$
$H(\mathrm{x})$	Shannon entropy of the random variable x
$I(P, Q)$	Mutual information of P and Q
$D_{\mathrm{KL}}(P, Q)$	Kullback-Leibler divergence of P and Q
$\nabla_{\boldsymbol{x}}\ell$	Gradient of function ℓ with respect to $\boldsymbol{x}$
$\|\|\boldsymbol{x}\|\|_p$	l_p norm of $\boldsymbol{x}$
$\int_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})d\boldsymbol{x}$	Definite integral over the entire domain of $\mathcal{X}$

Chapter 1

Introduction

Deep learning has made significant strides in various domains by delivering state-of-the-art performance across a wide range of applications in recent years, including pattern recognition [12, 13], data mining [14], computer vision [15, 16], natural language processing [17, 18], etc.. However, despite the remarkable successes, deep learning models continue to face challenges related to generalisation, model size, and the need for extensive labelled data. Generalisation refers to a model's capability to adapt and perform effectively on unseen data, leveraging the knowledge gained during the training process. However, achieving generalisation can be a challenging task [19]. In addition, deep learning models contain a significant number of parameters, thereby demanding substantial computational resources, which presents challenges for deployment [3]. These deep models also require a substantial amount of labelled data for effectively training [15, 20].

Adversarial regularisation aims to improve the robustness and generalisation capabilities of deep learning models by training them to withstand specially crafted adversarial perturbations. The perturbations, which are added to natural inputs to generate adversarial examples close/cross to the model's decision boundaries, are often imperceptible to humans. However, adversarial examples can significantly degrade the performance of deep learning models, raising concerns about their reliability and security [21, 22]. By incorporating adversarial regularisation into the training process, we strive to develop models that are robust against adversarial attacks [23–26] and exhibit consistent

performance across diverse input distributions to improve generalisation [27, 28]. Investigating adversarial regularisation allows us to gain valuable insights into the deep learning models and explore potential perspectives for addressing these aforementioned challenges.

Knowledge distillation (KD) [3], on the other hand, is a technique that facilitates the transfer of knowledge from a large model size, computationally expensive model (teacher) to a smaller, more efficient model (student). By leveraging the teacher model's knowledge, the student model can achieve a performance level comparable to its larger teacher while significantly reducing computational and memory requirements [29–31]. The adversarial technique is a promising approach for exploring the teacher model's characteristics (e.g., decision boundaries) that can be effectively transferred to the student model [5]. This technique is particularly relevant in the context of model compression, where the main objective is to develop compact models that retain competitive performance without sacrificing efficiency.

Adversarial regularisation and knowledge distillation are two complementary techniques that have emerged as promising solutions to enhance the generalisation and efficiency of deep learning models. This thesis develops and improves the adversarial regularisation and knowledge distillation techniques to address the challenges, specifically focusing on robust deep learning, deep semi-supervised learning and model compression tasks.

1.1 Aims and Approach

This thesis aims to develop and enhance the adversarial regularisation and knowledge distillation techniques, with the goals of enhancing robust deep learning, semi-supervised learning, and model compression tasks. Specifically, we seek the answers to the following questions:

- Can we develop an understanding and improvement of adversarial perturbations and the knowledge distillation processes?
- Can we develop and enhance the robustness of deep learning models by adversarial regularisation?

- Is it possible to develop and enhance adversarial regularisation for deep semi-supervised learning?
- Can we exploit the principle of knowledge distillation and adversarial regularisation for the model compression task?

Our research approach firstly leverages the mutual information framework to develop an interpretation technique from the information-theoretic perspective for enhancing the understanding of adversarial perturbations, which can improve the existing attack perturbations to make them less detectable by humans while maintaining their effectiveness. This explainable information-theoretic technique is also used to develop an end-to-end framework that addresses the explainability of knowledge distillation in deep learning. Secondly, state-of-the-art deep neural networks are reported to be susceptible to attacks [21, 22]. These attacks add the crafted adversarial perturbations to natural inputs to create adversarial examples (e.g., Fast Gradient Sign Method (FGSM) [22], Projected Gradient Descent (PGD) [23] and Auto-Attack [32]). We adopt a novel probabilistic perspective by defining a distribution for adversarial examples and use this distribution to enhance the robustness of deep learning, semi-supervised learning, and knowledge distillation processes. From the application perspective, we not only apply our approach to normal datasets but also extend it to the more challenging medical domain. These broad applications enable us to tackle complex tasks and demonstrate the versatility of our methods. Our specific aims are:

- To develop an understanding of adversarial perturbations and the knowledge distillation process from the information-theoretic perspective. Through this understanding, this research aims to contribute to the advancement of deep learning models by enhancing the adversarial perturbations and efficiency of the knowledge distillation process.
- To investigate the development and enhancement of the robustness of deep learning models using adversarial regularisation, as well as investigate its potential for deep semi-supervised learning and model compression. We seek to understand the effectiveness of these techniques in improving model performance and generalisation across different learning paradigms. Ultimately, we aim to exploit the principles of knowledge distillation and adversarial regularisation to advance the field of robust deep learning, deep semi-supervised learning and model compression tasks.

1.2 Significance and Contribution

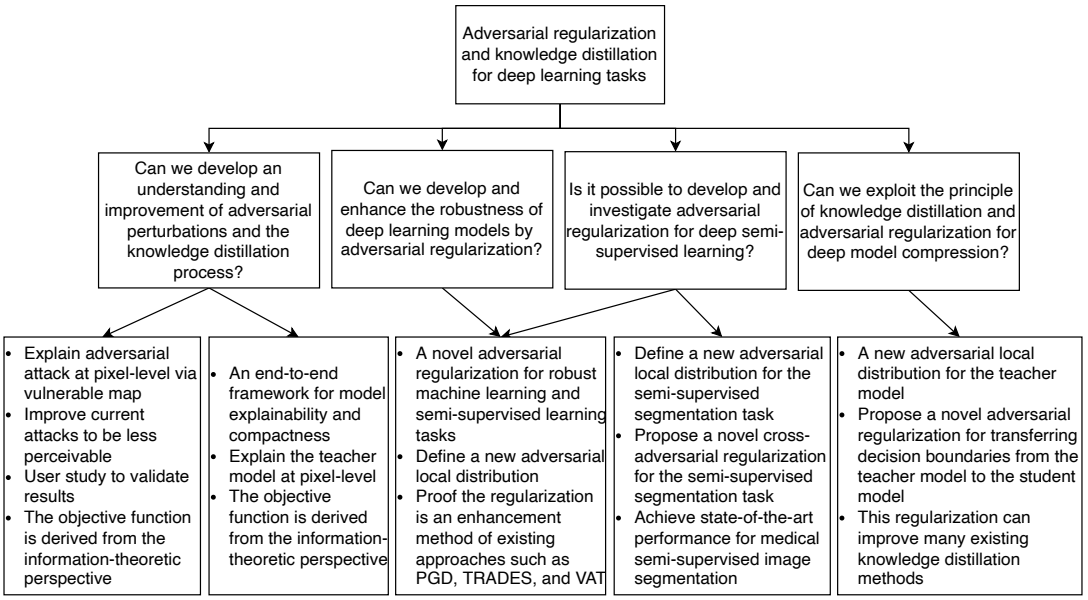


FIGURE 1.1: Thesis significance and contributions.

We summarise the significance and thesis contributions in the Figure 1.1. The theoretical significance of our research encompasses a collection of methods designed to deepen our understanding of adversarial perturbations and knowledge distillation, while also investigating the principles of adversarial regularisation and knowledge distillation to improve various deep learning tasks. On the practical side, this work contributes to enhancing model robustness for defence against adversarial attacks, ensuring safer deployments, significantly lowering costs by reducing the number of parameters in deep models, and developing classification and segmentation models that require fewer labelled data points. This is particularly valuable in medical settings where acquiring labelled data can be costly. In particular, the key contributions of this thesis are:

1. To enhance understanding and refine adversarial perturbations. We propose a vulnerability map to identify the most susceptible areas of an image to adversarial perturbations. This helps us refine existing attack methods, making them less noticeable to humans while keeping their ability to cause misclassifications. We conduct a user study to verify the efficacy of our improved attack method to be less human perceivable than other attacks. Moreover, we create an objective function from the information-theoretic perspective, which provide our research in understanding the principles behind adversarial attacks.

2. To develop an end-to-end technique that addresses the issues of teacher model explainability in knowledge distillation. We propose a novel approach that enables us to explain the teacher at the pixel level, thereby providing a more detailed and understanding of its inner workings. This approach is particularly useful in scenarios where the interpretability of the model is critical, such as in medical imaging. To achieve this, we derive an objective function from the information-theoretic perspective that guides our research efforts and helps us better understand the underlying principles of model explainability and compactness. Utilising this objective function, we optimise the model architecture to achieve high accuracy and interpretability while preserving its compactness.

3. To propose a novel adversarial local regularisation technique to enhance the generalisation of deep learning models. Our method involves the introduction of a novel adversarial local distribution (ALD), which is a set of all possible adversarial examples within a constraint. Based on this adversarial distribution, we introduce our adversarial regularisation which encourages the robustness of deep models by promoting local output distribution smoothness. We theoretically demonstrate that our approach is an enhancement of existing techniques such as FGSM [22], PGD [23] and TRADES [26] in robust deep learning, and Virtual Adversarial Training (VAT) [28] in semi-supervised learning. Our ALD regularisation leads to significant improvements in model performance by achieving state-of-the-art results in both robust deep learning and deep semi-supervised learning tasks on several benchmark datasets.

4. To extend the ALD on enhancing the performance of semi-supervised segmentation tasks in medical imaging. We propose a novel cross-adversarial local distribution (Cross-ALD) that mixes two adversarial local distributions. This distribution is specifically tailored for the semi-supervised segmentation task. Building on this, we introduce a Cross-ALD regularisation technique that further improves performance when compared to existing methods such as MC-Net [33], DTC [34], and SS-Net [9]. Through extensive experimentation, we demonstrate that our proposed method achieves state-of-the-art performance on several benchmark medical datasets.

5. To extend the ALD for the knowledge distillation task, we propose the teacher adversarial local distribution (TALD) for the teacher model that captures the

local properties of the teacher by using adversarial examples. These adversarial examples are informative examples because they are near decision boundaries. We then introduce a novel technique that transfers the decision boundaries learned by the teacher model to the student model using these adversarial examples, thereby improving the knowledge distillation process. Our regularisation approach can be applied to many existing knowledge distillation methods such as FitNet [35], BSS [5], and CRD [10], and leads to significant improvements in the student model accuracy on several benchmark datasets.

1.3 Thesis Structure

The remaining parts of this thesis are structured as follows[1]:

- Chapter 2: *Related Background.* We briefly introduce the essential background knowledge that serves as the foundation for all the research work presented in this thesis.
- Chapter 3: *Learning to Attack with Fewer Perturbations by Refining Adversarial Attacks.* In this chapter, we will explore our first contribution, which focuses on understanding and refining adversarial perturbations.
- Chapter 4: *Transferring and Explaining Knowledge from Pre-trained Teacher Models.* We present our second contribution in this chapter, which emphasises transferring and explaining knowledge from the larger, more complex teacher model to the smaller student model.
- Chapter 5: *Particle-based Adversarial Local Distribution Regularisation.* We discuss our third contribution by introducing a novel adversarial local regularisation technique designed to improve the generalisation of deep learning models in both robust deep learning and deep semi-supervised learning tasks across various benchmark datasets.
- Chapter 6: *Cross-adversarial Local Distribution Regularisation for Semi-supervised Image Segmentation.* In this chapter, we present the fourth contribution. Our

[1] I acknowledge the use of Grammarly, QuillBot and ChatGPT to occasionally assist my writing (e.g., check spelling and grammar). However, I am responsible for writing all information in this thesis.

aim is to expand the adversarial local distribution to enhance the performance of semi-supervised segmentation tasks. To achieve this, we introduce a novel cross-adversarial local distribution (Cross-ALD) that merges two distinct adversarial local distributions.

- Chapter 7: *Adversarial Local Distribution Regularisation for Knowledge Distillation.* In this chapter, we show the fifth contribution, where we propose using ALD for the teacher model to capture its local properties with adversarial examples. We subsequently introduce an innovative technique that transfers the decision boundaries learned by the teacher model to the student model utilising these adversarial examples

- Chapter 8: *Conclusion.* In conclusion, we provide a summary of all the scientific contributions made throughout this thesis and discuss potential future developments in the field.

Chapter 2

Related Background

In this chapter, we provide the background knowledge that forms the foundation for all the work presented in the following chapters. To ensure clarity and simplicity, we will introduce some related background only in specific contexts to cover the scope of our work in this dissertation adequately. We begin with semi-supervised learning, which aims to address the challenge of learning from limited labelled data by utilising both labelled and unlabelled data. This approach aims to improve model performance by utilising the inherent structure present in the data, thereby reducing the reliance on expensive and time-consuming manual labelling (Section 2.1). As deep learning models become increasingly complex and begin to apply to various tasks, the demand for explainable AI grew. We focus on perceptive explainable neural network methods, which provide visual information that can be humanly perceived (Section 2.2). Recently, knowledge distillation has gained traction as a method to transfer knowledge from a larger, more complex model (teacher) to a smaller, more efficient model (student) while retaining performance. This technique allows for the deployment of deep learning models in resource-constrained environments such as edge devices (Section 2.3). Finally, we revisit adversarial machine learning, a research area that explores the vulnerability of deep learning models via adversarial examples. Adversarial examples are instances that cross decision boundaries, causing a well-trained model to produce incorrect classifications or predictions. Therefore, these examples are informative, and can be used to enhance the robust deep learning, deep semi-supervised learning and model compression tasks in Section 2.4.

2.1 Deep Semi-supervised Learning

Deep semi-supervised learning (DSSL) is a rapidly expanding area with numerous practical applications. This section offers the fundamental concepts and recent advancements in deep semi-supervised learning approaches for Chapters 5 and 6, focusing on aspects such as model architecture and unsupervised loss functions. We start by defining the notations for deep semi-supervised learning. Denote $\boldsymbol{x} \in \mathbb{R}^n$ as our n-dimensional input in an input data distribution $P_{\mathbb{X}}$ of a space $\mathcal{X}$. Let $\mathbb{D}_l$ and $\mathbb{D}_u$ be the labelled and unlabelled dataset, respectively, with $P_{\mathbb{D}_l}$ and $P_{\mathbb{D}_u}$ being the corresponding data distribution. The labelled image $\boldsymbol{x}_l$ and segmentation ground-truth $\boldsymbol{y}$ are sampled from the labelled dataset $\mathbb{D}_l$ $(\boldsymbol{x}_l, \boldsymbol{y} \sim P_{\mathbb{D}_l})$, and the unlabelled data sampled from $\mathbb{D}_u$ is $\boldsymbol{x}_u \sim P_{\mathbb{D}_u}$. We form the objective of DSSL is to address the following optimisation problem:

$$\min_{\theta} \mathbb{E}_{(\boldsymbol{x}_l,\boldsymbol{y})\sim P_{\mathbb{D}_l}} \ell_s(\boldsymbol{x}, \boldsymbol{y}; \theta) + \lambda_1 \mathbb{E}_{\boldsymbol{x}_u \sim P_{\mathbb{D}_u}} \ell_u(\boldsymbol{x}_u; \theta) + \lambda_2 \mathbb{E}_{\boldsymbol{x}\sim P_{\mathbb{X}}} \mathcal{R}(\boldsymbol{x}; \theta), \tag{2.1}$$

where ℓ_s denotes the per-sample supervised loss (e.g., cross-entropy for classification, dice loss for image segmentation), ℓ_u is the per-sample unsupervised loss, and $\mathcal{R}$ is the per-sample regularisation (e.g., consistency loss, distributional smoothness). The model is parameterised by θ , and $\lambda_1, \lambda_2 \in \mathbb{R}_{>0}$ denotes the trade-off hyper-parameters. Various selections of unsupervised loss functions and regularisation terms result in different types of semi-supervised models. It is important to note that we do not make a clear distinction between unsupervised losses and regularisation terms. We categorise existing deep semi-supervised learning methods [36] into adversarial generative methods, consistency regularisation methods, pseudo-labelling methods, and hybrid methods.

2.1.1 Generative Adversarial Network

Generative methods are capable of learning the implicit features of data to more effectively model data distributions. They aim to learn the real data distribution from the training dataset and subsequently produce new data according to this distribution. In this section, we explore the deep generative semi-supervised methods, specifically those based on the frameworks of the Generative Adversarial Network (GAN).

A conventional Generative Adversarial Network (GAN) [37] is composed of two main components: a generator, denoted as G, and a discriminator, denoted as D. The generator's objective is to learn a distribution, referred to as P_g, over the data variable $\boldsymbol{x}$, guided by a prior on input noise variables, $P_{\boldsymbol{z}}(\boldsymbol{z})$. The generator G produces fake samples, $G(\boldsymbol{z})$, which are designed to deceive the discriminator D. On the other hand, the role of the discriminator D is to maximise the differentiation between the real training samples $\boldsymbol{x}$ and the fake samples $G(\boldsymbol{z})$. In this arrangement, D and G are engaged in a two-player minimax game, defined by the value function $V(G, D)$.

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim P(\boldsymbol{x})}[logD(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim P_{\boldsymbol{z}}}[log(1 - D(G(\boldsymbol{z})))] \tag{2.2}$$

Given that GANs can learn the distribution of real data from unlabelled samples, they can be employed to help Semi-Supervised Learning (SSL). Various approaches exist for utilising GANs in SSL environments. Drawing inspiration from [36], we cover four primary methods for employing GANs in SSL. These include:

1. Re-using the features derived from the discriminator [38, 39].
2. Using samples generated by GANs to regularise a classifier [40–42].
3. Learning a model of inference [43–45].
4. Using GAN-produced samples as supplementary training data [46–48].

Here, we present well-known works such as Categorical Generative Adversarial Network (CatGAN) [38] for (1), Improved GAN [40] for (2), Bidirectional Generative Adversarial Networks (BiGANs) [43] for (3), and Triple GAN [46] for (4).

The Categorical Generative Adversarial Network (CatGAN) [38] reconfigures the GAN's objective function to consider the mutual information between observed instances and their anticipated categorical class distributions. This approach is focused on learning a discriminator that classifies samples into K categories by assigning a label $\boldsymbol{y}$ to each instance $\boldsymbol{x}$, instead of learning a binary discriminator value function. The CatGAN discriminator loss function includes the supervised loss, which is a cross-entropy term between the predicted conditional distribution $P(\boldsymbol{y}|\boldsymbol{x}, D)$ and the actual label distribution of samples. It consists of three parts: (1) the entropy term $H[P(\boldsymbol{y}|\boldsymbol{x}, D)]$

is to obtain certain category assignment for samples; (2) $H[P(\boldsymbol{y}|G(\boldsymbol{z}), D)]$ for uncertain predictions from generated samples; and (3) the marginal class entropy $H[P(\boldsymbol{y}|D)]$ to uniform usage of all classes.

The Improved GAN [40] concurrently trains the generator and the classifier. The classifier network is designed with $(K+1)$ output units, denoted by $\boldsymbol{y} = \{y_1, y_2, ..., y_K, y_{K+1}\}$. This GAN methodology tackles the (K+1)-class classification problem. The initial K classes are associated with real examples, while the y_{K+1} class consists of the synthetic images generated by the generator G. It introduces refined strategies for training the GANs, specifically feature matching, minibatch discrimination, historical averaging one-sided label smoothing, and virtual batch normalisation. Feature matching is applied specifically for the generator's training. It is trained by minimizing the discrepancy between features of the real and the generated examples $||\mathbb{E}_{\boldsymbol{x}\in\mathbb{X}} D(\boldsymbol{x}) - \mathbb{E}_{z\in P(\boldsymbol{z})} D(G(\boldsymbol{z}))||_2^2$, rather than maximizing the likelihood of its generated examples classified to K real classes. Thus, the training loss function of the classifier is

$$\begin{aligned}\max_D \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim P(\boldsymbol{x},\boldsymbol{y})}[\log P_D(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{y} \leq K)] + \mathbb{E}_{\boldsymbol{x}\sim P(\boldsymbol{x})}[\log P_D(\boldsymbol{y} \leq K|\boldsymbol{x})] \\ + \mathbb{E}_{\boldsymbol{x}\sim P_G}[\log P_D(\boldsymbol{y} = K+1|\boldsymbol{x})],\end{aligned} \tag{2.3}$$

where the first term is the supervised cross-entropy loss. The remaining two terms represent unsupervised losses from unlabelled data and data generated by the model, respectively.

The Bidirectional Generative Adversarial Networks (BiGANs) [43] presents an approach to unsupervised feature learning. Unlike the conventional GAN architecture, BiGAN incorporates an encoder E, which facilitates the conversion of data $\boldsymbol{x}$ into $\boldsymbol{z}'$, thereby forming a data pair $(\boldsymbol{x}, \boldsymbol{z}')$. These data pairs, in combination with those produced by generator G, represent two categories of true and fake data pairs. The objective of the BiGAN discriminator D is to differentiate between these real and fake data pairs. The value function for the training of discriminator becomes

$$\begin{aligned}\min_{G,E}\max_D V(D,E,G) =& \mathbb{E}_{\boldsymbol{x}\sim P_{\mathbb{X}}}[\mathbb{E}_{\boldsymbol{z}\sim P_{\mathbb{E}(\cdot|\boldsymbol{x})}} \log D(\boldsymbol{x},\boldsymbol{z})] \\ &+ \mathbb{E}_{\boldsymbol{z}\sim P(\boldsymbol{z})}[\mathbb{E}_{\boldsymbol{x}\sim P_G(\cdot|\boldsymbol{z})} \log(1 - D(\boldsymbol{x},\boldsymbol{z}))].\end{aligned} \tag{2.4}$$

The Triple GAN [46] is introduced to tackle the challenge arising from the incompatible loss functions of the generator and discriminator in a GAN, which is the impossibility of both the generator and discriminator reaching an optimal state concurrently [40]. The solution proposed by Triple GAN is to introduce a tripartite framework engaging in a three-player game. This structure comprises three components: a generator G that employs a conditional network to generate corresponding fake samples for real labels, classifier C generating pseudo labels for provided real data, and discriminator D distinguishing whether a data-label pair is from the real label dataset or not. The loss of Triple GAN is written as

$$\begin{aligned}\min_{C,G}\max_{D} V(C,G,D) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim P(\boldsymbol{x},\boldsymbol{y})}[\log D(\boldsymbol{x},\boldsymbol{y})] \\ + \lambda_1 \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim P_C(\boldsymbol{x},\boldsymbol{y})}[\log 1 - D(\boldsymbol{x},\boldsymbol{y})] \\ + (1-\lambda_1)\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim P_G(\boldsymbol{x},\boldsymbol{y})}[\log(1 - D(G(\boldsymbol{y},\boldsymbol{z}),\boldsymbol{y})],\end{aligned} \tag{2.5}$$

where the discriminator D obtains label information about unlabelled data from the classifier C, and forces the generator G to generate the realistic image-label samples.

2.1.2 Consistency Regularisation

We introduce the consistency regularisation methods for semi-supervised deep learning. These methods incorporate a consistency regularisation term into the final loss function to enforce the assumed prior constraints. Based on the manifold assumption or the smoothness assumption, consistency regularisation encapsulates a set of methods asserting that data points added perturbations shouldn't change the model's output [49]. Therefore, consistency regularisation is seen as a tool for finding a smooth manifold where the dataset lies by utilising the unlabelled data [50].

The Teacher-Student structure is widely used in consistency regularisation for SSL techniques. The model operates: As a student, the model learns as before, and as a teacher, the model generates targets simultaneously. However, since the model generates these targets, they might not always be correct but are still used for learning by the model's student. The consistency regularisation methods suffer from a bias risk [51] that can be solved by improving the target's quality. Let θ be the weights of the student. We define

the consistency constraints as:

$$\mathbb{E}_{\boldsymbol{x}\in\mathbb{X}}\mathcal{R}(f(\theta,\boldsymbol{x}),T_{\boldsymbol{x}}) \tag{2.6}$$

where $f(\theta,\boldsymbol{x})$ is the prediction from model $f(\theta)$ given input $\boldsymbol{x}$. $T_{\boldsymbol{x}}$ is the consistency target generated by the teacher. $\mathcal{R}(\cdot,\cdot)$ is the distance (e.g., Mean Squared Error (MSE) or KL-divergence) between the student and the teacher outputs. Various consistency regularisation approaches differ in their target generation methods. Several strategies can be explored to improve the quality of these targets.

There are some techniques involving consideration of the Teacher model, rather than merely duplicating the student model. **The Ladder Network** [52, 53], inspired by a deep denoising AutoEncoder, marks the first successful application of a Teacher-Student model to SSL. **The Π-Model** [54] generates two random augmentations of a sample for both labelled and unlabelled data such as randomised data augmentation, dropout, and random maxpooling, process an input sample through the network multiple times, consequently generating varied predictions. **The Temporal Ensembling** [55] shares similarities with the Π-Model, as it establishes a consensus prediction under varying regularisation and input augmentation techniques. This method refines the Π-Model by utilising the Exponential Moving Average (EMA) of predictions from previous epochs to reduce computational overhead. The consistency loss can be defined as:

$$\mathbb{E}_{x\in\mathbb{X}}\mathcal{R}(f(\theta,\boldsymbol{x},\zeta^1),EMA(f(\theta,\boldsymbol{x},\zeta^2), \tag{2.7}$$

where ζ^1 and ζ^2 are injected augmentations such as additive or multiplicative noise.

Mean Teacher [51] employs the EMA of model weights across training steps and improves the development of a more precise model rather than directly relying on output predictions. The Mean Teacher method is composed of two models: the Student and the Teacher. The Student model operates as a typical model, while the Teacher model, mirroring the Student model's architecture, utilises EMA of the Student model's weights. The consistency regularisation is applied between the predictions generated by the Student and Teacher models, as shown in the equation 2.8. After that Dual Student [56] extends the Mean Teacher model by replacing the teacher with another student.

$$\mathbb{E}_{x\in\mathbb{X}}\mathcal{R}(f(\theta,\boldsymbol{x},\zeta^1),f(EMA(\theta),\boldsymbol{x},\zeta^2)). \tag{2.8}$$

2.1.3 Pseudo-labelling Methods

Pseudo-labelling techniques differ from consistency regularisation methods in their foundational principles. While consistency regularisation methods typically rely on the consistency constraints of extensive data transformations, pseudo-labelling methods lean on the high confidence associated with pseudo-labels, which can be incorporated into the training dataset as labelled data. There are two main types: one enhances the entire framework's performance based on the disagreement of views or multiple networks, while the other, self-training, has gained performance thanks to the success of self-supervised learning in the unsupervised domain.

Disagreement-based semi-supervised learning operates on the principle of employing multiple learners for a given task and utilising the discrepancies during the learning process [57]. Under this method design, two or three distinct networks are trained concurrently, with each network assigning labels to the unlabelled samples provided by the others. **The Deep Co-training** [58] operates under the presumption that every data sample $\boldsymbol{x}$ in the dataset embodies two unique and mutually complementary views. Each of these views is deemed sufficient to train an effective classifier. Given this supposition, Co-training engages in learning two separate classifiers tailored to these dual views. Subsequently, these classifiers are employed to predict each view's unlabelled data and assign labels to the most confidently identified candidates for the other model. This process is iteratively repeated until there are no more unlabelled data or a specific condition is satisfied. Let $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ as two different views of data such that $\boldsymbol{x} = (\boldsymbol{v}_1, \boldsymbol{v}_2)$. Co-training assumes that C_1 as the classifier trained on View-1 $\boldsymbol{v}_1$ and C_2 as the classifier trained on View-2 $\boldsymbol{v}_2$ have consistent predictions on $\mathbb{X}$. In the objective function, the Co-training assumption can be modelled as:

$$H(\frac{1}{2}(C_1(\boldsymbol{v}_1) + C_2(\boldsymbol{v}_2))) - \frac{1}{2}(H(C_1(\boldsymbol{v}_1)) + H(C_2(\boldsymbol{v}_2))), \tag{2.9}$$

where $H(\cdot)$ is the entropy. The key to the success of Co-training is that the two views are distinct and complementary. Various other studies have also investigated the implementation of co-training in neural network model training. For instance, [59] regards the RGB and depth of an image as two independent views for object recognition. Subsequently, the Co-training is executed to train two networks based on these two views.

Moreover, in the sentiment classification, [60] treats the original review and the auto-generated anonymous review as two contrasting facets of a single review, subsequently employing the co-training algorithm. Additionally, **Tri-Net** [61] inspired by tri-training [62] uses three classifiers instead of two models in the Co-training. The tri-training process involves learning three classifiers from three distinct training sets, generated through bootstrap sampling. A technique known as output smearing [222] is employed to introduce random noise to the labelled sample, thereby creating different training sets and aiding the learning of the initial three modules. Output smearing [63] is employed to introduce random noise to the labelled sample. This output smearing is used for creating different training sets and aiding the learning of the initial three modules. These three models subsequently predict the pseudo-label for unlabelled data. If the predictions from two modules for the unlabelled instances align, the pseudo-label is deemed confident and stable. This labelled sample is incorporated into the third module's training set, following which the third module is fine-tuned on this expanded training set. Throughout the augmentation process, the three modules progressively become more similar; hence, they are fine-tuned individually on their respective training sets to maintain diversity.

The self-training models employ the model's predictions to generate pseudo-labels for unlabelled data. Essentially, it expands the training dataset by using the currently available labelled data samples to predict the labels of unlabelled data. **Entropy Minimisation (EntMin)** [64] represents an entropy regularisation technique that can facilitate semi-supervised learning by prompting the model to make low-entropy predictions for unlabelled data, subsequently incorporating this unlabelled data into a conventional supervised learning scenario. Theoretically, entropy minimisation helps prevent the decision boundary from intersecting regions with high-density data points, otherwise it would be forced to produce low-confidence predictions for unlabelled data. **Pseudo-label** [65] presents a straightforward and effective method for training neural networks in semi-supervised learning. The networks undergo training using both labelled and unlabelled data concurrently. The model is trained on labelled data in a standard supervised fashion utilising a cross-entropy loss. For unlabelled data, the identical model is employed to generate predictions for a batch of unlabelled samples. The prediction with the highest confidence is termed a pseudo-label, characterised by having the maximum predicted probability. **Noisy Student** [66] introduces a semi-supervised method influenced by the concept of knowledge distillation [3] that uses student models of equal

or larger size. Initially, the teacher EfficientNet [67] model is trained on labelled images, which generates pseudo labels for unlabelled samples. Then, a larger EfficientNet model, acting as a student, is trained on a mix of labelled and pseudo-labelled examples. This mix of instances undergoes data augmentation methods such as RandAugment [68], and the student model incorporates model noise like Dropout and stochastic depth during training. **Meta Pseudo Labels (MPL)** [69] employs a teacher model that assigns distributions to input samples to facilitate the training of the student model. As the student model's training progresses, the teacher model evaluates the student's performance on a separate validation set. The teacher then learns to produce target distributions so that if the student learns from such distributions, the student will achieve good validation performance. **SimCLRv2** [70] adapts the SimCLR [71] model to address semi-supervised learning challenges. Following the paradigm of supervised fine-tuning after unsupervised pre-training, SimCLRv2 leverages unlabelled samples in a task-agnostic manner. It demonstrates that a substantial model (both deep and wide) can be remarkably effective for semi-supervised learning. The SimCLRv2 process can be summarised into three steps: unsupervised or self-supervised pre-training, supervised fine-tuning with 1% or 10% labelled samples, and self-training with task-specific unlabelled examples.

2.1.4 Hybrid Methods

Hybrid methods combine concepts from previously mentioned methodologies like pseudo-labelling, consistency regularisation, and entropy minimisation to boost performance. These hybrid methods also incorporate a technique, namely Mixup [72]. Mixup is a straightforward, data augmentation technique that creates a combination of paired samples and their corresponding labels. Essentially, Mixup generates augmented training examples.

$$\tilde{\boldsymbol{x}} = \lambda \boldsymbol{x}_i + (1-\gamma)\boldsymbol{x}_j, \tilde{\boldsymbol{y}} = \gamma \boldsymbol{y}_i + (1-\gamma)\boldsymbol{y}_j, \tag{2.10}$$

where $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ and $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ are two instances from the training data, and $\gamma \in [0, 1]$. Thus, Mixup extends the training data set by the linear interpolations of samples, which also leads to the linear interpolations of the corresponding labels.

The Interpolation Consistency Training (ICT) [73] is a regularisation technique for semi-supervised learning that promotes consistency between the prediction at an interpolated example of two unlabelled examples and the interpolation of predictions at

those instances. This method is inspired by the low-density separation assumption and the ability of Mixup to create expansive margin decision boundaries. The ICT consistent regularisation is

$$\mathbb{E}_{x\in\mathbb{X}}\mathcal{R}(f(\theta, Mix_\gamma(\boldsymbol{x}_i, \boldsymbol{x}_j)), Mix_\gamma(f(\theta', \boldsymbol{x}_i), f(\theta', \boldsymbol{x}_j))), \tag{2.11}$$

where θ' is a moving average of θ.

MixMatch [74] integrates consistency regularisation and entropy minimisation into a single loss function. This model operates by producing pseudo-labels for each unlabelled instance and then training the original labelled data with the pseudo-labels for the unlabelled data using fully-supervised techniques. This approach is also to craft augmented labelled and unlabelled samples by the Mixup technique. **ReMixMatch** [75] is an expansion of MixMatch, by introducing distribution alignment and augmentation anchoring. Distribution alignment encourages the marginal distribution of aggregated class predictions on unlabelled data close to the marginal distribution of ground-truth labels. Augmentation anchoring replaces the consistency regularisation component of MixMatch.

DivideMix [76] offers a new semi-supervised learning framework to handle learning with noisy labels. The method 'co-divide' trains two networks simultaneously, using a dynamic Gaussian Mixed Model (GMM) to categorise the training set into labelled and unlabelled data. To further address the problem of noisy labels, DivideMix enhances MixMatch [74] through 'co-refinement' and 'co-guessing' strategies in the training process. **FixMatch** [77] is a simplified method that merges the techniques of consistency regularisation and pseudo-labelling. Its contribution lies in blending these two techniques, and the use of a separate weak and strong augmentation in the consistency regularisation approach.

2.2 Perceptive Explainable Deep Learning

Explainable AI (XAI) in deep learning is critical to understanding how these "black box" models make decisions. XAI aids in making the decisions of deep neural networks more perceptible or understandable in the decision-making process, which enhances reliability and accuracy. XAI is an expansive and significant field. In this section,

we introduce the background knowledge for Chapters 3 and 4. We narrow our focus on perceptive explainable deep learning. The term "perceptive" suggests that this approach focuses on making the network's decisions perceivable or understandable to human users. This involves visualizing the activations of the neurons in the network, showing which features the network considers important for its predictions. The visualisation is capable of helping researchers detect erroneous reasoning in classification problems [78]. For example, the predicted class can be perceived by highlighted regions in Figure 2.1. Let denote a model be f, which can make a prediction $\tilde{\boldsymbol{y}} = f(\boldsymbol{X})$ based on the input $\boldsymbol{X} \in \mathbb{R}^{H \times W}$. Perceptive explainable techniques create an important score matrix, denoted as $\boldsymbol{\Theta}(i, j)$. Here, i, j refers to the spatial position of a specific patch, pixel, or feature within $\boldsymbol{X}$. A larger value of $\boldsymbol{\Theta}(i, j)$ usually suggests that the corresponding component in $\boldsymbol{X}$ has a substantial influence on the output prediction. In other words, a higher weight in $\boldsymbol{\Theta}(i, j)$ suggests that the associated component in $\boldsymbol{X}$ contributes significantly to or greatly influences the interpretation of the output prediction.

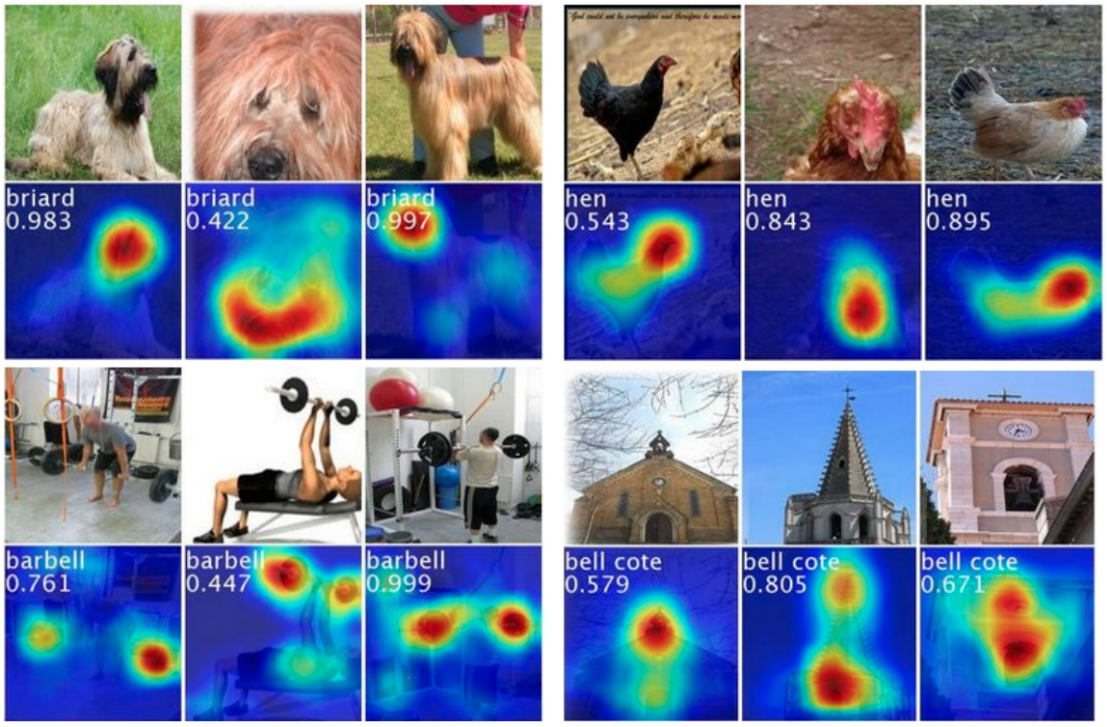


FIGURE 2.1: The Class Activation Maps for four classes that highlight the image regions using learned discriminative features in the classifier [1].

2.2.1 Saliency

Saliency map is a perceptive explainability method that employs feature localisation to identify the most influential features in an input $\boldsymbol{X}$ that significantly contribute to the model's prediction.

The Class Activation Maps (CAM) family is a prime example of this feature localisation method, linking predicted classes to regions in the feature domain that correspond to locations in the input image. Given an input image, the model f generates the feature activation of unit k at spatial location (i, j) in the last convolutional layer, represented as $\boldsymbol{F}_k(i, j)$. The input to softmax of class y is denoted by $\sigma_y = \sum_k w_k^y \boldsymbol{F}_k$, where w_k^y is the weight pertaining to class y and unit k. CAM defines the importance score/heatmap/saliency matrix $\boldsymbol{\Theta}_y$ of class y as:

$$\boldsymbol{\Theta}_y(i, j) = \sum_k w_k^y \boldsymbol{F}_k(i, j), \tag{2.12}$$

The spatial size is generally reduced during the feature extraction process of neural network model f, making $\boldsymbol{\Theta}$ size significantly smaller than $\boldsymbol{X}$. Despite its coarse mapping, CAM can highlight important image locations, as shown in Figure 2.1.

GradCAM [79], generalised of CAM, utilises gradients flowing into the last convolutional layer of a deep learning model to generate a heatmap that visually emphasises the areas of the input that are most important for the model's output prediction. After that, multiple variants of GradCAM have appeared such as GradCAM++ [80], Score-CAM [81], and Smooth-GradCAM++ [82].

2.2.2 Attention Mechanism

Attention mechanism is a common approach for model explainability. The purpose of this mechanism is to automatically learn different weights according to feature content and assign these weights to different feature regions, providing perceptive explainability for decision making. This mechanism first introduces for neural machine translation [83], which aims to transform an input language sequence into another language sequence. The attention mechanism proposed by [84] helps to memorise long source sentences in

neural machine translation, significantly improving model performance. It has been extended in different forms of attention mechanisms such as content-based attention [85], additive [84], dot-product attention [86], and scaled dot-product attention [87]. Given the effectiveness of the attention mechanism, researchers have adopted it into the computer vision domain for tasks such as image captioning [88], supporting classifier decisions [89], and weakly-supervised semantic image segmentation [90]. Especially in the medical image domain, Yang et al. [91] introduced the guided soft attention for histopathology breast cancer detection.

The attention mechanism can be categorised into two types: "soft" attention and "hard" attention. Soft attention makes use of differentiable functions to generate continuous attention weights, while hard attention uses non-differentiable functions to produce binary weights. Soft attention is frequently utilised within the feature domain and has seen extensive application across various fields. For instance, [84] leverages soft attention for natural image captioning, while [92] introduces a guided soft attention mechanism for detecting breast cancer in histopathology images. For hard attention, which typically involves back-propagation through discrete variables, several strategies can be used to make the model differentiable such as the REINFORCE algorithm [93] and the Gumbel-Softmax trick [94–96].

In formal terms, the attention mechanism is tasked with learning the weighted matrix $\boldsymbol{\Theta}$, which is the output of an additional, learnable attention module (f_{att}).

$$\boldsymbol{\Theta}(i,j) = f_{att}(\boldsymbol{F}(i,j)) \tag{2.13}$$

It's important to note that $\boldsymbol{\Theta}(i,j)$ preserves the spatial location (i,j) of the extracted feature $\boldsymbol{F}$. This weighted matrix $\boldsymbol{\Theta}$ is then multiplied element-wise $\odot$ with the features $\boldsymbol{F}(i,j)$ extracted at the last stage of the neural model f to focus on significant reasoning features, as shown in equation Equation 2.14.

$$F_{att} = \boldsymbol{\Theta}(i,j) \odot \boldsymbol{F}(i,j). \tag{2.14}$$

The attention features $\boldsymbol{F}_{att}$ are used for the primary tasks of the model f in place of $\boldsymbol{F}$. Additionally, the attention module f_{att} is trained using standard back-propagation

through the main task. Although attention can highlight semantic regions for better explainability, it is usually trained with the aim of maximizing classification accuracy.

2.2.3 Feature Selection

Feature selection assigns important scores, denoted as $\mathbf{\Theta}$, to each feature directly at the input-data level without any transformation. Consequently, it generates the most meaningful explanation of a model's prediction. **The LIME** method proposed by Ribeiro et al. [97] randomly samples from a density centred at the input to be explained, and then fits a sparse linear model to predict the model outputs for these instances. Shrikumar et al. [98] introduces an approach specifically designed for neural networks that use back-propagation to assign importance scores back to every feature of the input. Lundberg and Lee [99] utilises a sampling-based method called "kernel SHAP" to approximate Shapley values and quantify the importance of a given input's features. The recent **Learning-to-Explain (L2X)** approach [100] trains an explainer to explain the pre-trained teacher model by maximizing the mutual information between selected instance-wise features and the pre-trained model outputs. This feature selection is based on hard attention with the Gumbel-softmax trick.

2.3 Knowledge Distillation

In this section, we introduce the background knowledge for Chapters 4 and 7. Knowledge Distillation (KD) [3] is a process that involves the transfer of knowledge from a complex, pre-trained model (commonly referred to as the "teacher" model, T) to a more compact and lightweight model (known as the "student" model, S). This method is grounded in the idea of enabling the smaller student model to learn from and emulate the behaviour of the larger, more complex teacher model. The student model becomes a valuable asset in situations where there's a need to significantly decrease computational resources or reduce the associated costs during the inference stage. It offers the benefits of decreased computational load, faster execution times, and a lighter footprint, while still aiming to maintain a high level of performance, which it gains from the teacher model's knowledge.

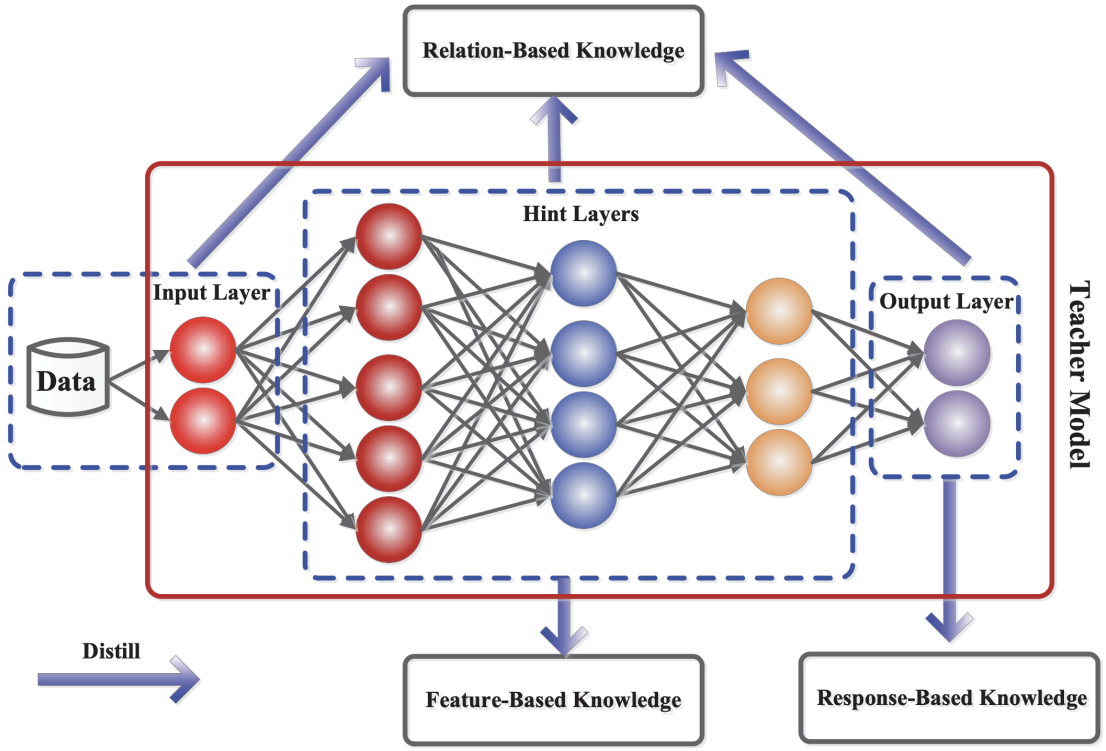


FIGURE 2.2: The schematic illustrations of sources of response-based knowledge, feature-based knowledge and relation-based knowledge in a deep teacher network [2].

In Figure 2.2, knowledge distillation can be classified [2] into the following categories such as response-based knowledge, feature-based knowledge, and relation-based knowledge. Response-based knowledge involves the distillation process where the student model learns to mimic the output layer responses or predictions of the teacher model. In Feature-based knowledge, the student model learns by aligning and adapting to the internal representations (features) learnt by the teacher model. The features might be derived from one or more layers within the teacher model. Relation-based knowledge emphasises on preserving the relations or dependencies between data samples or between features of the data, as learnt by the teacher model.

2.3.1 Response-based Knowledge Distillation

Response-based knowledge typically refers to the neural response of the last output layer of the teacher model. The core concept is to transfer the teacher model's prediction directly to the student model. Despite its simplicity, response-based knowledge distillation is a potent tool for model compression, widely employed across diverse tasks and applications [2]. We denote $\boldsymbol{y}$ and $\boldsymbol{x}$ as the ground-truth label and input in the

dataset $\mathbb{D}$, respectively. Let $\boldsymbol{z}_T$ and $\boldsymbol{z}_S$ be the output logits of the teacher and student, respectively, where $k \in \mathbb{K}$, and the number of labels is K. Given a logit vector $\boldsymbol{z}$, which are the outputs from the deep model's last fully-connected layer, the distillation loss for response-based knowledge can be expressed as:

$$\ell_{ResKD}(\boldsymbol{z}_T, \boldsymbol{z}_S) = \ell_R(\boldsymbol{z}_T, \boldsymbol{z}_S), \tag{2.15}$$

where ℓ_R indicates the divergence loss of logits such as KullbackLeibler divergence loss. The most prevalent form of response-based knowledge applied to classification is often referred to as "soft targets", a concept introduced by [3, 4]. The output of $\boldsymbol{y}_T = \text{softmax}(\boldsymbol{z}_T)$ and $\boldsymbol{y}_S = \text{softmax}(\boldsymbol{z}_S)$ represents the distribution of class probabilities of the teacher and student, respectively. The soft-target is estimated by a temperature softmax function.

$$\sigma_t(\boldsymbol{z}_k) := \text{softmax}_t(\boldsymbol{z}_k) = \frac{\exp(\frac{\boldsymbol{z}_k}{t})}{\sum_{k=1}^{K} \exp(\frac{\boldsymbol{z}_k}{t})}, \tag{2.16}$$

where t is the temperature parameter. When t=1, the equation 2.16 simplifies to the standard softmax function. The knowledge distillation loss is used to match the soft label output from the teacher with the soft prediction from the student, as shown in *Figure* 2.3. Accordingly, the distillation loss for soft logits can be rewritten as:

$$\ell_{ResKD}(\boldsymbol{z}_T, \boldsymbol{z}_S) = \ell_R(\sigma_t(\boldsymbol{z}_T), \sigma_t(\boldsymbol{z}_S)). \tag{2.17}$$

The application of response-based knowledge extends to various forms of model predic-

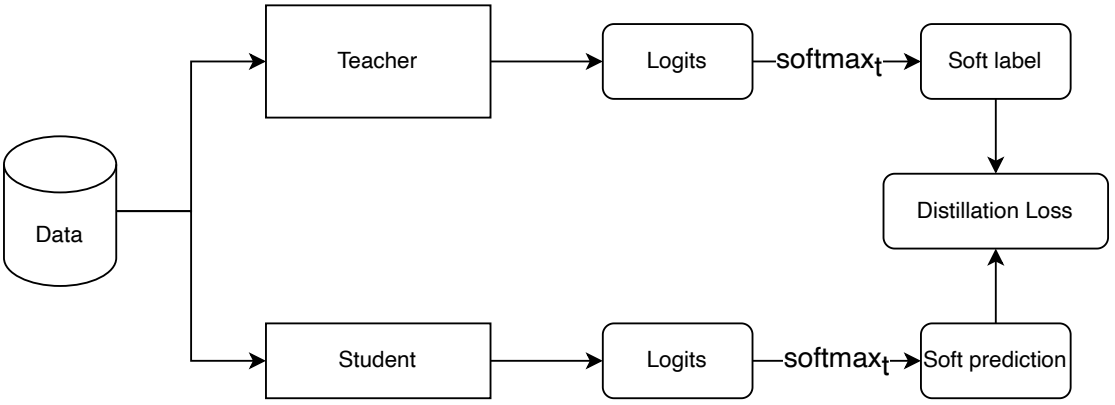


FIGURE 2.3: The response-based knowledge distillation [3, 4].

tions. For instance, in object detection tasks, the response could incorporate the logits

along with the offset of a bounding box, as demonstrated by Chen et al. [101]. In semantic landmark localisation tasks such as human pose estimation, the teacher model's response includes a heatmap for each landmark [102].

2.3.2 Feature-based Knowledge Distillation

Deep neural networks are good at learning numerous levels of feature representation, each with progressively greater levels of abstraction, a concept known as representation learning [29]. Hence, the output from both the last layer and intermediate layers, often called feature maps, can serve as knowledge to guide the student model's training. In particular, feature-based knowledge is derived from the intermediate layers, and serves as an extension of response-based knowledge, especially when training thinner and deeper networks.

The concept of utilising intermediate representations is initially introduced in Fitnets [35], with the intent of enhancing the student model's training process. The primary principle here involves matching the feature activations between the teacher and the student models. Typically, the formula to calculate the distillation loss for feature-based knowledge transfer can be expressed as follows:

$$\ell_{FeaKD}(\boldsymbol{F}_T(\boldsymbol{x}), \boldsymbol{F}_S(\boldsymbol{x})) = \ell_F(\Phi(\boldsymbol{F}_T(\boldsymbol{x})), \Phi(\boldsymbol{F}_S(\boldsymbol{x}))), \tag{2.18}$$

where $\boldsymbol{F}_T(\boldsymbol{x})$ and $\boldsymbol{F}_S(\boldsymbol{x})$ denote the feature maps from intermediate layers of the teacher and student models respectively. Transformation functions, denoted by $\Phi(f_T(\boldsymbol{x}))$ and $\Phi(f_S(\boldsymbol{x}))$, are typically utilised when the teacher and student model feature maps do not share identical shapes. ℓ_F stands for the similarity function, which is utilised to match the feature maps of the teacher and student models.

Inspired by the feature-based knowledge distillation, several methods have been suggested to match the features. The approach [103] introduces an "attention map" drawn from original feature maps as a method of knowledge representation. Expanding on this idea, [104] uses neuron selectivity transfer to further generalise the attention map. The work [105] introduces a method of knowledge transfer that relied on matching the probability distribution in the feature space. Kim et al. [106] puts forth the concept of "factors" as a more digestible form of intermediate representations to simplify teacher

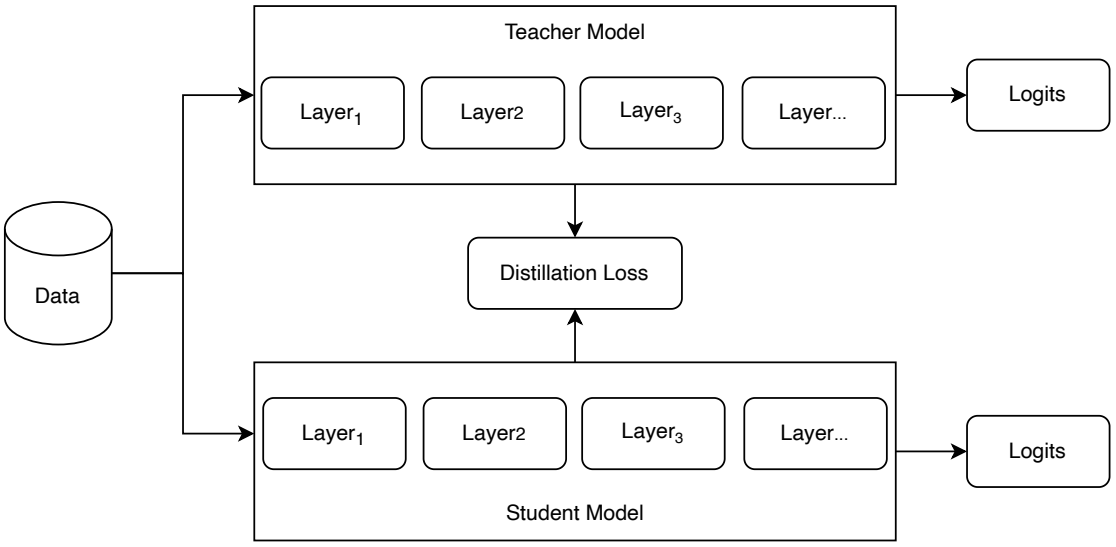


FIGURE 2.4: The feature-based knowledge distillation.

knowledge transfer. A route-constrained hint learning is introduced, aiming to supervise the student using outputs from teacher's hint layers, with a goal to minimise the performance gap between the teacher and the student [107]. The method [5] proposes a method using the activation boundary of hidden neurons for knowledge transfer. Interestingly, Zhou et al. [108] explore parameter sharing of intermediate layers of the teacher model together with response-based knowledge. To align the semantics between teacher and student, [109] suggests a cross-layer knowledge distillation, which adaptively assigns appropriate teacher layers to each student layer through attention allocation.

2.3.3 Relation-based Knowledge Distillation

Both response-based and feature-based knowledge use outputs from specific layers in the teacher model, relation-based knowledge goes a step further by investigating the connections between different layers or data samples.

To explore the relationships among various feature maps from different layers, the distillation loss of relation-based knowledge can be represented by the relationships of these feature maps as follows:

$$\ell_{RelKD}(\boldsymbol{F}_T, \boldsymbol{F}_S) = \ell_{CF}[\Psi_T(\boldsymbol{F}_T^{'}(\boldsymbol{x}), \boldsymbol{F}_T^{''}(\boldsymbol{x})), \Psi_S(\boldsymbol{F}_S^{'}(\boldsymbol{x}), \boldsymbol{F}_S^{''}(\boldsymbol{x}))], \tag{2.19}$$

where $\boldsymbol{F}_T$ and $\boldsymbol{F}_S$ are the feature maps of teacher and student models given input $\boldsymbol{x}$, respectively. Feature map pairs are selected both from the teacher model denoted by

$(\boldsymbol{F}_T^{'}, \boldsymbol{F}_T^{''})$, and the student model denoted by $(\boldsymbol{F}_S^{'}, \boldsymbol{F}_S^{''})$. Ψ_T and Ψ_S are the similarity functions that correlate pairs of feature maps from the teacher and student models, respectively. ℓ_{CF} denotes the function that measures the correlation between the feature maps of the teacher and student models. Yim et al. [110] proposed a technique to investigate the relationships between different feature maps using the Flow of Solution Process (FSP), which is defined by the Gram matrix between two layers. The FSP matrix summarises the relations between various pairs of feature maps, which are computed using the inner products of features derived from the two respective layers. Lee et al. [111] introduced the knowledge distillation via singular value decomposition to extract key information from feature maps. Park et al.[112] used variational informatic knowledge distillation which tries to maximise the mutual information between teacher and student. Multi-head graph-based knowledge distillation was proposed by [113]. The graph knowledge is the intra-data relations between any two feature maps via multi-head attention network. The exploration of pairwise hint information was also considered, with the student model mimicking the mutual information flow from pairs of hint layers of the teacher model [114]. This work highlights the various strategies and techniques in leveraging the relationships between different feature maps in the context of knowledge distillation.

The distillation loss of relation-based knowledge, which is derived from data sample relations, can be expressed as follows:

$$\ell_{RelKD}(\boldsymbol{F}_T, \boldsymbol{F}_S) = \ell_{CF}[\Psi_T(\boldsymbol{F}_T^{'}(\boldsymbol{x}_i), \boldsymbol{F}_T^{''}(\boldsymbol{x}_j)), \Psi_S(\boldsymbol{F}_S^{'}(\boldsymbol{x}_i), \boldsymbol{F}_S^{''}(\boldsymbol{x}_j))], \tag{2.20}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are different input samples. The distilled knowledge contains not only the details of the feature information but also the interrelations of data samples [115, 116]. The knowledge transferred via this instance relationship graph includes details about the instance features, their interrelationships, as well as the transformations occurring across layers in the feature space [117]. The transferring knowledge derived from instance relations was introduced by [116]. Inspired by the manifold learning, the student network is trained using feature embedding, which maintains the similarities of sample features found in the intermediate layers of the teacher networks [118]. The relations between data samples are modelled as probabilistic distributions using feature representations of data, which are then matched through knowledge transfer between the probabilistic distributions of the teacher and student models [105, 119]. Tung et

al. [120] further proposed a similarity-preserving knowledge distillation method that arises from the similar activations of input pairs in the teacher networks. A knowledge distillation method that uses both instance-level information and correlations between instances was proposed by [121]. Recently, contrastive learning [122] and self-supervised learning [123] for knowledge distillation have demonstrated remarkable performance on image classification tasks.

2.4 Adversarial Machine Learning and Regularisation

Trustworthy machine learning is fundamentally focused on the design and creation of models that are reliable. This reliability is measured by the model's robustness under a wide variety of conditions and scenarios, guaranteeing a level of performance that remains consistent. Trustworthy ML applications span across numerous sectors, enabling more reliable and transparent decision-making. For example, trustworthy ML can aid in risk assessment, fraud detection, and investment strategies in the finance sector. In healthcare, these applications can assist in accurate diagnosis and personalised treatment plans, while ensuring the confidentiality of patient data. Recently, deep models are reported to be susceptible to attacks, which use carefully crafted inputs created to mislead a prediction [21, 22]. Therefore, adversarial machine learning is introduced to enhance the security and trustworthiness of machine learning systems. Its primary objective is to investigate and defend against adversarial attacks. The scope of adversarial machine learning can be extended to improve semi-supervised learning and knowledge distillation. In this section, we provide background knowledge for Chapters 3, 5, 6, and 7 by briefly introducing adversarial attacks and adversarial regularisation for model robustness, semi-supervised learning, and knowledge distillation tasks.

2.4.1 Adversarial Attacks

Adversarial attacks exploit the vulnerabilities in the target model to cause incorrect predictions. These attacks generate adversarial examples (such that changes are almost imperceptible to humans) by adding crafted perturbations to natural inputs, as shown in Figure 2.5.

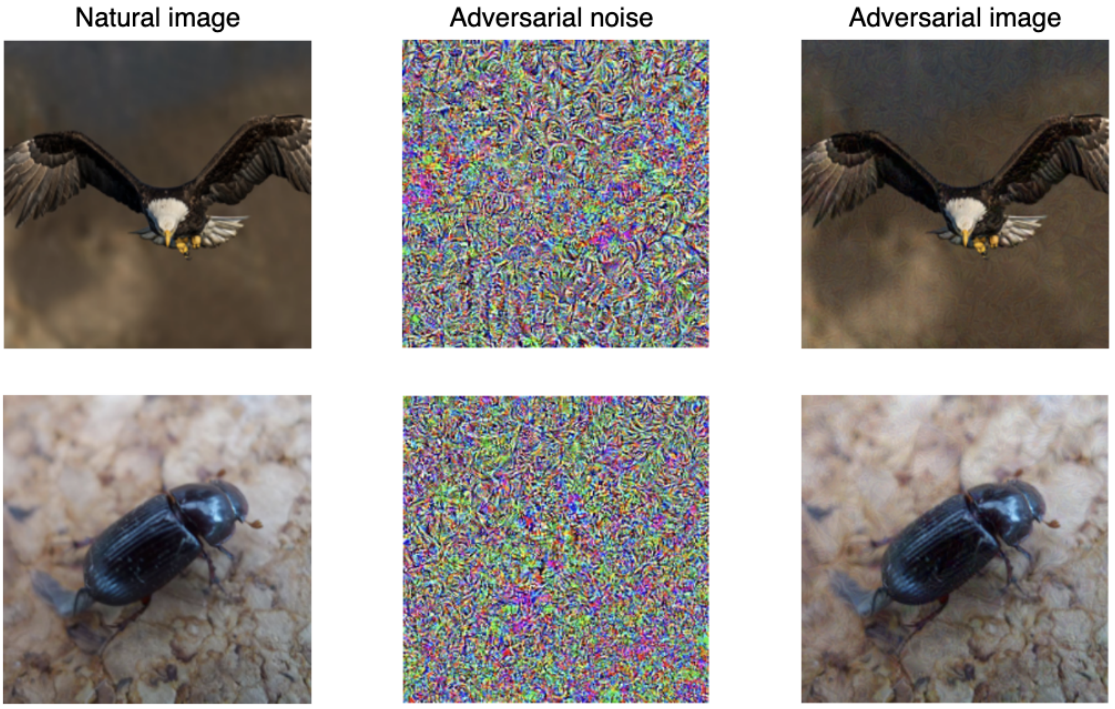


FIGURE 2.5: Original predicted labels are "bald eagle" and "dung beetle" of top and bottom natural input images; by adding adversarial perturbations (or adversarial noises), the model predicts "prairie chicken" and "standard poodle", respectively.

Let $\boldsymbol{x} \in \mathbb{R}^n$ be our natural input data. The adversarial example is denoted by $\boldsymbol{x}^{adv}$. Given an input $(\boldsymbol{x}, y) \sim P_{\mathbb{D}}$ (i.e., the data-label distribution), the adversarial example $\boldsymbol{x}^{adv}$ is defined in [21] as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}^{adv}} \quad d(\boldsymbol{x}^{adv}, \boldsymbol{x}) \\ &\textit{subject to} \quad f(\boldsymbol{x}^{adv}) \neq y, \end{aligned} \tag{2.21}$$

where d is a distance metric to measure the difference between $\boldsymbol{x}^{adv}$ and $\boldsymbol{x}$. The metric is commonly the L^p norm, as defined in Equation 2.22.

$$d(\boldsymbol{x}^{adv}, \boldsymbol{x}) = (\sum_{i=1}^{n} |\boldsymbol{x}'_i - \boldsymbol{x}_i|^p)^{\frac{1}{p}}, \tag{2.22}$$

where n is the number of elements in the inputs, and $\boldsymbol{x}_i$ and $\boldsymbol{x}'_i$ are the i-th element of the original input $\boldsymbol{x}$ and the adversarial example $\boldsymbol{x}^{adv}$, respectively.

Additionally, the work [22] proposed a specific optimisation problem to solve Equation 2.21 as:

$$\begin{aligned} &\max_{\boldsymbol{x}^{adv}} \quad \ell(f(\boldsymbol{x}^{adv}), y) \\ &\textit{subject to} \quad d(\boldsymbol{x}^{adv}, \boldsymbol{x}) \leq \epsilon, \end{aligned} \tag{2.23}$$

where ϵ is a hyper-parameter that controls the maximum perturbation radius constraint. The Equation 2.23 prioritises maximising the prediction loss ℓ for reliably finding successful adversarial examples.

FGSM The most popular way to find the adversarial example is using adversarial direction which leverages gradients to maximise the loss ℓ. Goodfellow et al. [22] proposed the Fast Gradient Sign Method (FGSM), a one-step gradient-based technique, as follows:

$$\boldsymbol{x}^{adv} = \boldsymbol{x} + \epsilon \cdot sign(\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), y)), \tag{2.24}$$

where ϵ is the maximum perturbation allowed, and $sign$ is the sign function. Tramer et al. [124] enhanced the FGSM by incorporating a small random initialisation step prior to enhance the loss function, named R + FGSM. This initialisation assists the method in avoiding the non-smooth vicinity of the input data $\boldsymbol{x}$.

$$\boldsymbol{x}^{adv} = \boldsymbol{x}^{r} + (\epsilon - \lambda_1) \cdot sign(\nabla_{\boldsymbol{x}^{r}} \ell(f(\boldsymbol{x}^{r}), y)), \tag{2.25}$$

where $\boldsymbol{x}^{r} = \boldsymbol{x} + \lambda_1 \cdot sign(\mathcal{N}(0, 1))$ is the input after one small random step. The experiment in [124] showed that incorporating random initialisation markedly improves the performance of R+FGSM over FGSM. This enhancement holds true irrespective of the models being robust or non-robust.

BIM The FGSM modifies inputs by implementing a single large step in the gradient direction that maximises the loss function. However, this strategy might not be adequately effective, especially in scenarios involving complex loss surfaces. A simple improvement uses an iterative process of taking several minor steps. Basic Iterative Method (BIM) was introduced in [125] using multiple gradient update steps.

$$\boldsymbol{x}_0^{adv} = \boldsymbol{x}, \quad \boldsymbol{x}_{t+1}^{adv} = \Pi_{B(\boldsymbol{x},\epsilon)}(\boldsymbol{x}_t^{adv} + \lambda_1 \cdot sign(\nabla_{\boldsymbol{x}_t^{adv}} \ell(f(\boldsymbol{x}_t^{adv}), y)), \tag{2.26}$$

where λ_1 is the step size, and $\Pi_{B(\boldsymbol{x},\epsilon)}$ is the projection operator to keep the adversarial examples in a ball constraint B with radius ϵ. Additionally, the Momentum Iterative Method (MIM) [126] enhances the BIM by integrating momentum into the iterative procedure. This inclusion aids in stabilizing the update directions and expediting the convergence process.

PGD Madry et al. [23] proposed the Projected Gradient Descent (PGD) method, which is a variant of BIM with uniform random initialisation. While BIM and MIM are suffered from the non-smooth vicinity of the complex loss function, PGD can address this issue using the uniform random initialisation to provide a more effective attack.

$$\boldsymbol{x}_0^{adv} = \boldsymbol{x} + \eta, \quad \boldsymbol{x}_{t+1}^{adv} = \Pi_{B(\boldsymbol{x},\epsilon)}(\boldsymbol{x}_t^{adv} + \lambda_1 \cdot sign(\nabla_{\boldsymbol{x}_t^{adv}} \ell(f(\boldsymbol{x}_t^{adv}), y)). \tag{2.27}$$

In the Equation 2.27, PGD method initiates with $\boldsymbol{x}$ with random initialisationη. It then progressively implements multiple update steps, each with a step size λ_1, aiming to find the adversarial examples. The computational cost is the main drawback of PGD when calculating multiple gradient update steps. Thus, several methods have been proposed to accelerate the PGD attack while maintaining its effectiveness [127–129].

2.4.2 Adversarial Regularisation for Deep Learning Tasks

Robust Deep Learning The robustness of deep learning models can be enhanced through the integration of a regularisation loss during training. The most common idea is to use adversarial training proposed by [22]. The adversarial training focuses on improving the resilience of models to adversarial attacks. Adversarial training is a technique that incorporates adversarial examples in the training process to against potential attacks. By subjecting the model to a variety of adversarial examples during training, the model learns to make correct predictions even when faced with intentionally manipulated inputs. This practice aids in enhancing the model's overall robustness. The regularisation of the adversarial training can be defined in the minimax optimisation (Equation 2.28).

$$\min_{\theta} \mathbb{E}_{(\boldsymbol{x},y)\sim P_{\mathbb{D}}} \Big[\max_{\boldsymbol{x}^{adv} \in B(\boldsymbol{x},\epsilon)} \ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta) \Big], \tag{2.28}$$

where $\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta)$ depends on a particular method. For example, FGSM [22], PGD [23] use the cross-entropy loss (CE)

$$\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta) = CE(f_\theta(\boldsymbol{x}^{adv}), y), \tag{2.29}$$

where y is the one-hot ground-truth label of x and $f_\theta(\boldsymbol{x}^{adv})$ is the prediction probabilities. Another example is TRADES [26], which use the Kullback-Leibler divergence loss (D_{KL}) in Equation 5.3

$$\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta) = D_{\mathrm{KL}}(f_\theta(\boldsymbol{x}^{adv}), f_\theta(\boldsymbol{x})). \tag{2.30}$$

Semi-supervised learning In the context of semi-supervised learning, the smoothness assumption is an essential assumption that helps to improve the generalisation learning process from the labelled and unlabelled data. **The Virtual adversarial training (VAT)** technique involves carefully selection of the adversarial noise perturbations instead of Gaussian additive or multiplicative noise. VAT proposed by [27] is a well-known regularisation for semi-supervised learning which can be defined by a min-max optimisation problem similar to the adversarial training defence. This technique aims to generate an adversarial transformation of a sample, which can change the model prediction. Specifically, the adversarial training technique is used to find the optimal adversarial perturbations ζ. The consistency constraint enforces between the model's output of natural sample and the perturbed one.

$$\mathbb{E}_{x\sim P_{\mathbb{X}}}\mathcal{R}(f_\theta(\boldsymbol{x}), (f_\theta(\boldsymbol{x} + \zeta^{adv}). \tag{2.31}$$

In VAT, the adversarial perturbations are added as an additive unit vector to the input or embedding spaces during the adversarial training for enhancing the generalisation performance of semi-supervised learning. **Virtual Adversarial Dropout (VAdD)** [130] integrates adversarial training alongside the Π-Model. VAdD's consistency constraint is calculated from two distinct dropout networks: one network utilises a random dropout mask, while the other employs adversarial training on the optimal dropout network. the output of a neural network with a random dropout mask, and the consistency loss includes adversarial dropout as described:

$$\mathbb{E}_{x\sim P_{\mathbb{X}}}\mathcal{R}(f_\theta(\boldsymbol{x}, \xi^{mask}), f_\theta(\boldsymbol{x}, \xi^{adv})), \tag{2.32}$$

where ξ^{adv} is an adversarial dropout mask, ξ^{mask} is a sampled random dropout mask instance.

Knowledge distillation Exploring the properties of the teacher is the key to improving student performance (e.g., teacher decision boundaries). One decision boundary exploring technique is to leverage adversarial attack methods (e.g., BIM, MIM and PGD), as shown in Figure 2.6. These adversarial examples are informative examples because they are near decision boundaries. The student model is then regularised by matching the loss between teacher and student using these adversarial example inputs. Heo et al.'s paper [5] proposed a **BSS attack** for exploring the teacher's properties using adversarial examples to increase the student's clean input accuracy. The work [131] proposed a noisy feature distillation, a new transfer learning method that improves robustness.

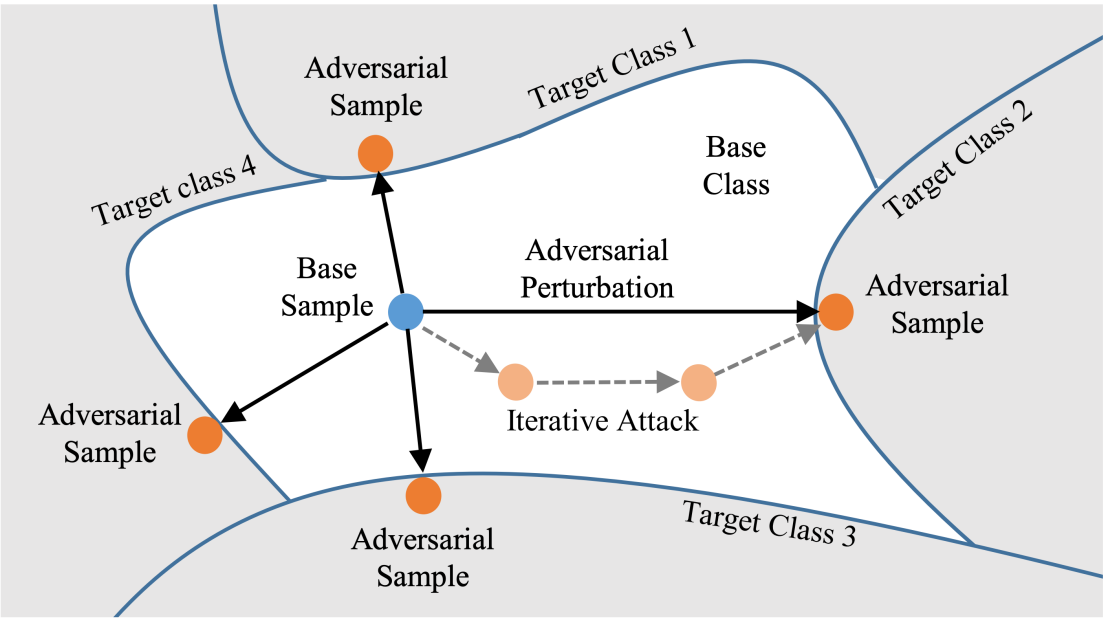


FIGURE 2.6: Iterative to find adversarial examples near decision boundaries [5].

Chapter 3

Learning to Attack with Fewer Perturbations by Refining Adversarial Attacks.

In Chapter 2, we have provided the related background knowledge for the perceptive explainable deep learning and adversarial attack. In this chapter, we introduce our first contribution, which focuses on understanding and refining adversarial perturbations. Deep neural network image classifiers are reported to be susceptible to adversarial evasion attacks, which use carefully crafted images created to mislead a classifier. Many adversarial attacks belong to the category of dense attacks, which generate adversarial examples by perturbing all the pixels of a natural image. To generate sparse perturbations, sparse attacks have been recently developed, which are usually independent attacks derived by modifying a dense attack's algorithm with sparsity regularisation, resulting in reduced attack efficiency. We aim to tackle this task from a different perspective. We provide an understanding of adversarial perturbations by generating a vulnerability map that displays the locations of the most vulnerable pixels to attacks in the input. We then select the most effective perturbations from the ones generated from a dense attack, based on the fact we find that a considerable amount of the perturbations on an image generated by dense attacks may contribute little to attacking a classifier. Accordingly, we propose a probabilistic post-hoc framework that refines given dense attacks by significantly reducing the number of perturbed pixels but keeping their attack

power, trained with mutual information maximisation. Given an arbitrary dense attack, the proposed model enjoys appealing compatibility for making its adversarial images more realistic and less detectable with fewer perturbations. Moreover, our framework performs adversarial attacks much faster than existing sparse attacks.

3.1 Introduction

Recently, Deep Neural Networks (DNNs) have enjoyed great success in many application areas such as computer vision and natural language processing. Nevertheless, DNNs have been demonstrated to be vulnerable to adversarial attacks with samples crafted deliberately to fool DNN classifiers [22, 125, 132–135]. For example, in image classification, an adversarial example (*adv images*) may be perceived as a legitimate data sample in a ground-truth class but misleads a DNN classifier to predict it into a maliciously-chosen target class or any class different from the ground truth.

The most common way to generate adv examples is by adding small perturbations/noise to the pixels of a real image, where two important factors need to be taken into account: perturbation magnitude and location [136]. As adv examples are required to challenge classifiers but be less perceptible to human eyes, the two factors need to be controlled judiciously. There have been two popular categories of adversarial attacks: *dense attacks* and *sparse attacks*. For a dense attack, it tries to apply perturbations with small magnitude and usually cares less about the perturbation locations. Many widely-used dense attacks perturb *all* the pixels of an image, according to the gradient of the classifier loss for each pixel, e.g., in [22, 32, 137–140]. Alternatively, a sparse attack seeks fewer locations (pixels) to perturb. It is shown that by finding appropriate locations, one can change a classifier's prediction of an image by perturbing only a few of its pixels [136, 141–146].

Although sparse attacks can attack with fewer perturbations, their perturbations are usually of a larger magnitude than dense attacks, which may result in more perceptible adversarial examples. One may consider a compromise solution that allows a sparse attack to perturb more (but not all) pixels whilst simultaneously reducing the magnitude of each pixel. Unfortunately, this can be infeasible as a sparse attack requires an individual search process of the locations to perturb for each image, which usually leads to

a much slower attack speed than dense attacks. For many sparse attacks, increasing the number of perturbations will significantly reduce their attack speed [141, 145], making them less useful in practice.

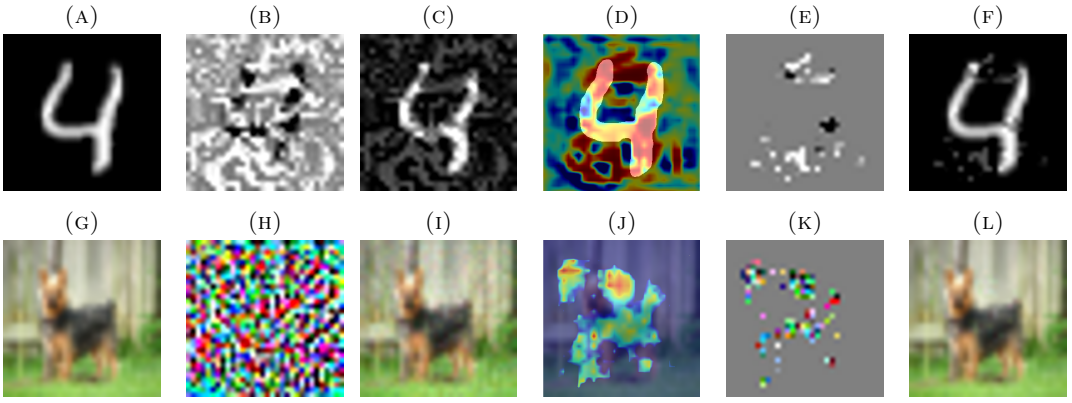


FIGURE 3.1: Demonstration using MNIST [6] and CIFAR10 [7] imagery. (a,g) Natural images (labelled digit "4" and "dog" respectively); (b,h) Adv perturbations generated by PGD; (c,i) Adv images by applying the perturbations (the predictions of the classifier are changed to digit "9" and "deer" respectively); (d, j) the vulnerability heatmaps learned by our method ("hotter" means more vulnerable); (e,k) Selected perturbations of PGD according to the vulnerability heatmaps; (f,l) Refined PGD adv images (the predictions of the classifier are the same as the original PGD). Images have been resized to higher resolutions for better visualisation.

In this chapter, we would like to tackle the task of generating adversarial images (*adv images*) with a fewer number of perturbations from a different perspective than existing sparse attacks. Our general idea is as follows: noting that dense attacks generate dense yet small perturbations, given an adv image with the perturbations generated by a dense attack, can we significantly reduce the number of perturbations by selecting the most effective ones and keep the attack performance as per the original adv image? As dense attacks are usually much faster than sparse ones, with an efficient selection process, we can devise a new framework that attacks with fewer perturbations more efficiently. As distinct from sparse attacks, which use independent methods for integrating the search for perturbation locations and magnitudes, ours is a post-hoc refinement framework for dense attacks, which decomposes the search for magnitude and location to the input dense attacks and the refinement process, respectively.

To further motivate our idea, we demonstrate that a large proportion of the generated perturbations of many dense attacks contribute little to the attack against the classifier. For example, in Figure 3.1 (b,h), we can see that the perturbations generated by Projected Gradient Descent (PGD) [147]) are distributed across all the pixels of the natural

images. However, shown in Figure 3.1 (e, k, f, l), we can fool the classifier's predictions similarly to PGD, with only 10% of its perturbations. The key research question is how to identify the vulnerable pixel locations of an adv image generated by a dense attack. Here we propose to identify an image's *pixel vulnerability* of the target classifier, where perturbations imposed on the more vulnerable pixels are more likely to change the prediction of the classifier. Figure 3.1(d,j) shows the pixel vulnerability heatmaps (we use the words "heatmap" and "map" interchangeably) discovered by our approach from PGD, with which we can choose the perturbations accordingly. This brings two important benefits:

1. With the reduced number of perturbed pixels, we can not only improve the imperceptibility of dense attacks but also significantly lower the accuracy of adv detection methods [148–153] on them.

2. The vulnerability heatmap can greatly help us to get a more intuitive and deeper understanding of the semantic structures of dense attacks. For example, Figure 3.1(d) highlights how the digit "4" is changed to digit "9", which concurs with our visual perception, and in Figure 3.1(j) where PGD tries to add perturbations to generate a pair of "fake" antlers, which fools the classifier to think it as an image of a "deer".

The detailed contributions of this chapter can be summarized as follows:

- We tackle the task of attacking with fewer perturbations with a novel approach, that refines given dense attacks in a post-hoc manner and is different from other sparse attacks. To the best of our knowledge, our methodology is original.

- Given an adv image of a dense attack, we propose to learn its vulnerability map with a DNN, which is used to select the most vulnerable pixels to attack so as to preserve the dense attack's performance but with significantly fewer perturbations. As the proposed framework is trained with adv images generated by dense attacks as input, it can be used to refine an arbitrary dense attack.

- In contrast to input dense attacks, our framework can generally reduce to 70% of the number of perturbations whilst keeping their attack performance as well as significantly reducing their detectability by both human and adversarial detectors.

- Compared with sparse attacks, our method can significantly improve the attack speed with better performance.

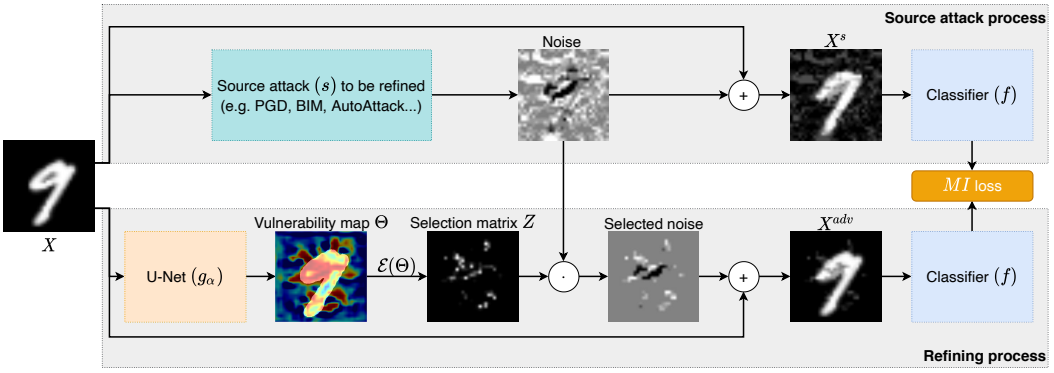


FIGURE 3.2: Overview of our framework. Our method is designed to refine a given source (dense) attack by: 1) invoking the source attack to generate adversarial perturbations/noise, i.e., $\mathbf{X}^s - \mathbf{X}$; 2) taking the natural image $\mathbf{X}$ as input to generate the vulnerability map $\mathbf{\Theta}$; 3) generating the selection matrix $\mathbf{Z}$ according to $\mathbf{\Theta}$; 4) selecting a subset of the perturbations/noise generated by the source attack; 5) generating a new adversarial image $\mathbf{X^{adv}}$ with the selected perturbations.

3.2 Related Work

3.2.1 Dense Attacks

Dense attacks are the most popular category of adv attacks, where all the pixels of a natural image are perturbed, such as in FGSM [22]), PGD [147]), BIM [125], DeepFool [137], CW [138], AutoAttack [32]. Due to the space limit, a comprehensive review of dense attacks is beyond the focus of this chapter. Theoretically, any dense attacks can serve as our source attacks and be refined by our proposed approach, as we only take the generated adversarial images as input for training and attacking.

3.2.2 Sparse Attacks

Existing sparse attacks can be categorised into black-box attacks (such as [142–144, 146]) and white-box attacks, where the latter is more relevant to our approach. As an example of white-box sparse attacks, [138, 145, 154] propose variants of dense attacks that integrate the l_0 constraint to encourage sparsity. Moreover, the Jacobian-based Saliency Map Attack (JSMA) [141] uses the saliency map to determine the pixel positions

to perturb, where perturbations are imposed on the pixels with high saliency. More recently, [136] proposes to factorise adversarial perturbations into two factors, which capture the magnitudes and positions, respectively.

3.2.3 Fundamental Differences between Our Attacks and Sparse Attacks

At the concept level, though both our method and existing sparse attacks fall into the general context of "reducing the number of perturbed pixels for adversarial attacks", *the thinking and methodology of our work are completely different.* Firstly, a sparse attack is an independent attack and many existing sparse attacks are achieved by modifying the inner steps of a dense attack (e.g., adding l_0 regularisation to the CW [138]), while our method is a post-hoc framework that refines dense attacks. As ours does not need to modify the inner steps of dense attacks but uses their output, it can be used to refine an arbitrary dense attack. Secondly, the fundamental difference in the design and thinking results in several unique properties and advantages of our method:

- Sparse attacks focus more on perturbation positions than magnitudes, thus, resulting in fewer but larger perturbations, while ours selects more but much smaller perturbations, which leads to more natural-looking adversarial images (see Figure 3.3 and 3.4).
- Most sparse attacks do complex optimisation for searching the perturbations locations, whiles ours is based on neural networks and has much better efficiency (see Table 3.1).
- With vulnerability maps, our method helps to gain a better and more intuitive understanding of the semantic structures of the source attacks and of classifier vulnerability that further assists the development of robust models (see Figure 3.5).

3.3 Method

3.3.1 Background and Problem Definition

Suppose that a normal image and its ground-truth label are denoted by $\mathbf{X}$ and $\hat{y} \in \{1, \cdots, K\}$, respectively, where K is the number of unique image labels. We consider a pre-trained neural network classifier f, which takes $\mathbf{X}$ as input and outputs a probability vector over K labels, i.e., $\boldsymbol{y} = f(\mathbf{X})$. Usually, for a well-trained classifier, we have $\hat{y} = \arg\max \boldsymbol{y}$. An adversarial attack is to find perturbations ζ (with the same dimensions as $\mathbf{X}$) and add them to $\mathbf{X}$ to give the adv example, $\mathbf{X}^{adv}$, which is expected to mislead the classifier to fail to identify the true label. Importantly, the adv image shall be inside an ϵ-ball around $\mathbf{X}$, i.e., $\|\zeta\| < \epsilon$ where $\|\cdot\|$ can be the l_∞ norm in accordance with [140, 147] or other norms. Finding the value of ζ can be formulated as the following optimisation problem: $\max_{\|\zeta\|<\epsilon} \ell(f(\mathbf{X}+\zeta), \hat{y})$, where $\ell(f(\mathbf{X}), \hat{y})$ denotes the loss function (e.g., cross-entropy loss) of the classifier f given input $\mathbf{X}$ and its true label $\hat{y}$.

Recall that our goal is to refine the adv images of a given dense attack, which we call the *source attack*. We assume an image $\mathbf{X}$ consists of $H \times W$ pixels, where each pixel x_{ij} can have either one channel (greyscale images) or multiple channels (colour images)[1]. Given $\mathbf{X}$ and a target classifier f, we can apply the source attack s to generate the source adversarial image, $\mathbf{X}^s$: $\mathbf{X}^s = s(\mathbf{X})$, whose predicted label distribution is $\boldsymbol{y}^s = f(\mathbf{X}^s)$. We also assume that the source attack can challenge the classifier well, i.e., $\arg\max \boldsymbol{y^s} \neq \arg\max \boldsymbol{y}$ and its perturbations fall into the ϵ-ball. Note that we only need the output of the source attack and need no knowledge of its implementation details.

Next, we introduce a vulnerability map $\boldsymbol{\Theta} \in \mathbb{R}^{H \times W}$, generated from a neural network g parameterised by α taking $\mathbf{X}$ as input: $\boldsymbol{\Theta} = g_\alpha(\mathbf{X})$. Specifically, θ_{ij} indicates the vulnerability of pixel x_{ij}. Based on $\boldsymbol{\Theta}$, we would like to develop a procedure $\mathcal{E}$ (detailed in Section 3.3.3) that discretely selects a subset of the most vulnerable pixels to perturb. Specifically, a draw from $\mathcal{E}$ is a binary selection matrix: $\mathbf{Z} \sim \mathcal{E}(\boldsymbol{\Theta}) \in \{0, 1\}^{H \times W}$, where $z_{ij} = 1$ indicates pixel x_{ij} is selected and vice versa. Given $\mathbf{X}^s$ and $\mathbf{Z}$, we construct a new adv example by imposing the source attack's perturbations only on the selected

[1] We view a multi-channel image as a matrix as we are only interested in the pixel coordinates in lieu of the values of the pixels.

pixels:

$$\mathbf{X}^{adv} = \mathbf{X} + \mathbf{Z} \odot (\mathbf{X}^s - \mathbf{X}), \tag{3.1}$$

where $\odot$ indicates the element-wise product and $\zeta = \mathbf{X}^s - \mathbf{X}$ consists of the perturbations from the source attack. In this way, $\mathbf{X}^{adv}$ is constructed by imposing a subset of perturbations from the source adv image $\mathbf{X}^s$ on the natural image $\mathbf{X}$. Finally, we can describe our research problem as: *Given* $\mathbf{X}$ *and* $\mathbf{X}^s$*, finding the optimal* $\mathbf{\Theta}$ *(i.e.,* $\mathbf{Z}$*) to generate a new adversarial image* $\mathbf{X}^{adv}$ *in the* ϵ*-ball, which can attack as well as the source adversarial image* $\mathbf{X}^s$. We illustrate the research question and our proposed method in Figure 3.2. More details of our proposed framework will be elaborated on in the following sections.

3.3.2 Training Objective

Given the problem definition, we introduce our learning algorithm derived from an information-theoretic perspective. Recall that our objective is to let $\mathbf{X}^{adv}$ challenge the classifier f similarly to $\mathbf{X}^s$. From the probabilistic perspective, if we view $\mathbf{X}^{adv}$ as a random variable parameterised by $\mathbf{\Theta}$, then the objective can be achieved by learning $\mathbf{\Theta}$ to push $\mathbf{X}^{adv}$ close to $\boldsymbol{y}^s$ (the predicted label distribution of the source attack). The mutual information (MI) is a widely-used measure of dependence between two random variables, which captures how much knowledge of one random vector reduces the uncertainty about the other [155]. Therefore, we can formulate the above problem as the following maximisation of the mutual information:

$$\ell = \max_{\mathbf{Z} \sim \mathcal{E}(\mathbf{\Theta})} I\left(\boldsymbol{y}^s, \mathbf{X}^{adv}\right). \tag{3.2}$$

As $\mathbf{X}^s$ comes from the source attack, it already satisfies the constraint of the ϵ-ball. Therefore, there is no need to consider this constraint when learning $\mathbf{Z}$. In Equation 3.3, we show that Equation 3.2 can be written as:

$$\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{\mathbf{X}^s,\mathbf{Z}}\left[\mathbb{E}_{\mathbf{X}^{adv}}\left[\sum_{k=1}^{K} p(y_k|\mathbf{X}^s)\log p(y_k|\mathbf{X}^{adv})\right]\right]\right]. \tag{3.4}$$

Specifically, we have:

$$
\begin{aligned}
I(\boldsymbol{y}^s, \mathbf{X}^{adv}) &= \int p(\boldsymbol{y}^s, \mathbf{X}^{adv}, \mathbf{X}^s, \mathbf{Z}, \mathbf{X}) \log \frac{p(\boldsymbol{y}^s, \mathbf{X}^{adv})}{p(\boldsymbol{y}^s)p(\mathbf{X}^{adv})} \mathrm{d}\boldsymbol{y}^s \mathrm{d}\mathbf{X}^{adv} \mathrm{d}\mathbf{X}^s \mathrm{d}\mathbf{Z} \mathrm{d}\mathbf{X} \\
&= \mathbb{E}_{\mathbf{X}} \left[\mathbb{E}_{\mathbf{X}^s|\mathbf{X},\mathbf{Z}|\mathbf{X}} \left[\mathbb{E}_{\mathbf{X}^{adv}|(\mathbf{X}^s,\mathbf{Z})} \left[\int p(\boldsymbol{y}^s|\mathbf{X}^s) \log \frac{p(\boldsymbol{y}^s|\mathbf{X}^{adv})}{p(\boldsymbol{y}^s)} \mathrm{d}\boldsymbol{y}^s \right] \right] \right] \\
&= \mathbb{E}_{\mathbf{X}} \left[\mathbb{E}_{\mathbf{X}^s|\mathbf{X},\mathbf{Z}|\mathbf{X}} \left[\mathbb{E}_{\mathbf{X}^{adv}|(\mathbf{X}^s,\mathbf{Z})} \left[\sum_{k=1}^{K} p(y_k|\mathbf{X}^s) \log p(y_k|\mathbf{X}^{adv}) \right] \right] \right] + \text{constant}, \qquad (3.3)
\end{aligned}
$$

where $p(\boldsymbol{y}^s, \mathbf{X}^{adv}, \mathbf{X}^s, \mathbf{Z}, \mathbf{X}) = p(\boldsymbol{y}^s|\mathbf{X}^s)p(\mathbf{X}^{adv}|(\mathbf{X}^s, \mathbf{Z}))p(\mathbf{X}^s|\mathbf{X})p(\mathbf{Z}|\mathbf{X})p(\mathbf{X})$.

- $\mathbf{X}^s|\mathbf{X} := s(\mathbf{X})$, which is the process of generating adversarial images from the source attack.
- $\mathbf{Z}|\mathbf{X} := \mathbf{Z} \sim \mathcal{E}(\mathbf{\Theta}), \mathbf{\Theta} = g_\alpha(\mathbf{X})$, which is the proposed process of generating the vulnerability map and selection matrix.
- $\mathbf{X}^{adv}|(\mathbf{X}^s, \mathbf{Z}) := \mathbf{X} + \mathbf{Z} \odot (\mathbf{X}^s - \mathbf{X})$, which is the process of imposing the selected perturbations on the natural image.
- $p(y_k|\mathbf{X}^s) \propto f(\mathbf{X}^s)_k$, and $p(y_k|\mathbf{X}^{adv}) \propto f(\mathbf{X}^{adv})_k$, which correspond to the prediction processes of the source and our new adv images, respectively.

To summarise, the learning process of our refinement framework can be described as: for one input image $\mathbf{X}$, we first apply the source attack to get $\mathbf{X}^s$, we then sample the selection matrix $\mathbf{Z}$; then we accordingly generate $\mathbf{X}^{adv}$; $\mathbf{X}^s$ and $\mathbf{X}^{adv}$ are fed into the classifier to get their predicted label distributions; finally, we minimise the cross-entropy between the two label distributions.

3.3.3 Construction of Selection Process

Now we introduce the details of the process $\mathcal{E}$ that generates $\mathbf{Z}$. Inspired by [100], among all the HW pixels of an image, we maximally select $M = \lceil \beta HW \rceil$ pixels to impose on the perturbations from the source attack where $\beta \in (0, 1)$ is the proportion of the maximally selected pixels. Next, we introduce a matrix version of the categorical distribution parameterised by $\mathbf{\Theta}$, a draw of which picks one pixel among the HW pixels:

$\mathbf{U} \sim \text{Categorical}(\mathbf{\Theta})$. Here $\mathbf{U} \in \{0,1\}^{H \times W}$ is a one-hot binary matrix, where the entry selected by the categorical distribution is turned on and all the others are zeros[2].

To maximally select M pixels, we draw $\mathbf{U}$ M times and choose the pixels that are selected at least once:

$$\begin{aligned} \mathbf{U}^{<m>} &\sim \text{Categorical}(\mathbf{\Theta}) \quad \text{for } m = 1 \text{ to } M, \\ \mathbf{Z} &= \max_m \mathbf{U}^{<m>}, \end{aligned} \tag{3.5}$$

where $\mathbf{U}^{<m>}$ denotes the output matrix of the m^{th} draw and $\max_m \mathbf{U}^{<m>}$ denotes the element-wise maximisation, i.e., $z_{ij} = \max_m u_{ij}^{<m>}$.

Recall that we aim to train the neural network parameter α to optimise Equation 3.2. To use back-propagation through the categorical random variables, we utilise the Concrete distribution [95, 156] to approximate the categorical distribution: $\tilde{\mathbf{U}} \sim \text{Concrete}(\log \mathbf{\Theta})$, where $\tilde{\mathbf{U}} \in \mathbb{R}_+^{H \times W}$ is the continuous relaxation of the one-hot matrix $\mathbf{U}$ in Equation 3.5. Specifically, we have:

$$\tilde{u}_{ij} = \frac{\exp\left(\log(\theta_{ij} + \gamma_{ij})/\tau\right)}{\sum_{i'j'}^{H \times W} \exp\left(\log(\theta_{i'j'} + \gamma_{i'j'})/\tau\right)}, \tag{3.6}$$

where γ_{ij} is from a Gumbel$(0,1)$ distribution: $\gamma_{ij} = -\log(-\log \nu_{ij}), \nu_{ij} \sim \text{Uniform}(0,1)$, and τ is the "temperature". With the Concrete distribution, we have:

$$\tilde{\mathbf{U}}^{<m>} \sim \text{Concrete}(\log \mathbf{\Theta}) \quad \text{for } m = 1 \text{ to } M, \tag{3.7}$$

$$\tilde{\mathbf{Z}} = \max_m \tilde{\mathbf{U}}^{<m>}, \tag{3.8}$$

where $\tilde{\mathbf{Z}}$ is an approximation to $\mathbf{Z}$ for back-propagation in the learning phase.

3.3.4 Implementation and Learning Algorithm

Here we give a further introduction to the implementation of our framework. Specifically, the only learnable component is the neural network $g_\alpha(\cdot)$, which takes $\mathbf{X}$ as input and outputs the vulnerability map $\mathbf{\Theta}$. In general, there is no limitation of implementation

[2] An ordinary categorical distribution's parameter is a probability vector and it outputs a one-hot vector. A more precise notation should be $\mathbf{U} \sim$ reshape_to_matrix(Categorical(softmax(reshape_to_vector($\mathbf{\Theta}$)). We abbreviate it to assist the readability.

Algorithm 3.1: Learning to Attack with Fewer Perturbations algorithm

input : Classifier f, To-be-refined source attack s, Input image collection $\mathbb{X}$, Max proportion of vulnerable pixels β

output: Parameter of the neural network α

while *Not converged* **do**
  Sample a batch of input images $\mathbb{X}_{Batch}$;
  forall $\mathbf{X} \in \mathbb{X}_{Batch}$ **do**
    Generate source adversarial image $\mathbf{X}^s = s(\mathbf{X})$;
    for $m = 1$ *to* M **do**
      Draw $\tilde{\mathbf{U}}^{<m>}$ by Equation 3.7;
    Calculate $\tilde{\mathbf{Z}}$ by Equation 3.8;
    Generate final adversarial image $\mathbf{X}^{adv} = \mathbf{X} + \tilde{\mathbf{Z}} \odot (\mathbf{X}^s - \mathbf{X})$;
    Calculate learning loss ℓ by Equation 3.2;
    Compute gradients of ℓ in terms of α;
  Average noisy gradients of batch samples;
  Update α by stochastic gradient steps;

of $g_\alpha(\cdot)$. As $\mathbf{\Theta}$ has the same dimensions with $\mathbf{X}$, inspired by the rich works in image segmentation and saliency detection, we empirically find that U-Net [157] yields good performance. Finally, we elaborate on the training process in Algorithm 3.1. After the neural network g is trained, given an input image $\mathbf{X}$, we first get $\mathbf{\Theta}$, then select from the perturbations of the source attack, and finally use Equation 3.1 to generate the refined adv image $\mathbf{X}^{adv}$.

3.4 Experiments

In this section, we conduct comprehensive experiments to evaluate the performance of our framework, named **PVAR** (**P**ixel **V**ulnerability **Ad**versary **R**efinement), on attacking image classifiers. Here we focus on untargeted white-box attacks, though our approach has the potential to be adapted to targeted or black-box settings.

3.4.1 Evaluation Metrics

We use five sets of different metrics for comprehensive evaluations:

- We use the classification accuracy on the adversarial images (**AdvAcc**) generated from the test images of a dataset, to test whether PVAR preserves the source attack's performance with fewer perturbations.

- To verify the assumption that the refined attack is less easily detected than its source counterpart. We build an adversarial detector following [148, 151] to predict whether an image is natural or adversarial. For an attack method, we split the test set of a dataset in half, denoted as $\mathbb{X}_1$ and $\mathbb{X}_2$, respectively. Given an attack, we build the training/test sets for the detector as $\{\mathbb{X}_1; \mathbb{X}_1^{adv}\}$ and $\{\mathbb{X}_2; \mathbb{X}_2^{adv}\}$. That is to say, a detector is specifically trained to an attack. For example, PGD, PVAR for PGD with $\beta = 0.1$, and PVAR for PGD with $\beta = 0.3$ are three different attacks and their detectors are trained separately). The detector consists of a binary classifier with three advanced features carefully designed for adversarial detection: classifier confidence, kernel density (K-density), and the entropy of normalised non-maximal elements (non-ME). The former two features are introduced in [148] and the latter is from [151]. As shown in Table 2 of [151], using non-ME only already achieves a very high DetAUC. Our detector is able to obtain stronger performance by combining multiple features. Given an attack, we train the detector in its own training set and report the Area Under Curve (**DetAUC**) on its own test set.

- To compare attack efficiency for each method, we also report the **averaged running time** of generating an adversarial image.

- To measure *how our refined attacks generate less perceivable adversarial images to human*, we report the averaged l_2 **distances** between them for each attack method that measure the closeness between natural images and their adversarial examples, i.e., a smaller the l_2 distance indicates that an adversarial example is closer to its natural counterpart, thus, it is less perceivable.

- To qualitatively study "how less perceivable", we conduct a user study and report the **perceivableness** of attacks. Finally, We note that from an attacker's perspective, *the lower the five metrics are, the better.*

3.4.2 Experimental Settings

Here we consider the MNIST [6], CIFAR10 [7] and ImageNet [158] datasets, where the pixel values of the images are normalised between 0 and 1.

Settings of classifiers We use several widely-used pre-trained covoluntional neural network models as the target classifiers: LeNet [159] for MNIST with 0.995 natural

accuracy; ResNet32-v1 [16] and ResNet56-v2 [160] for CIFAR10 with 0.932 and 0.921 accuracy, respectively; DenseNet169 [161] for ImageNet reached 0.762 top-1 and 0.932 top-5 accuracy.

Settings of source attacks We select three popular dense attacks as the example source attacks to be refined by our PVAR: PGD [147], BIM [125] and AutoAttack [32], where PGD and BIM are implemented in Foolbox [162] and AutoAttack is implemented in https://github.com/fra31/auto-attack. We set the attack strength $\epsilon = 0.3$ for MNIST, and $\epsilon = 0.03$ for CIFAR10 and ImageNet. For other settings, we follow the standard ones. Note that the source attack is not limited to the above three and can be an arbitrary dense attack.

Settings of PVAR For the neural network g, we leverage a symmetric architecture for the U-Net [157] used in our model to generate the vulnerability map $\mathbf{\Theta}$. The U-Net consists of a down-sampling encoder and an up-sampling decoder. Specifically, the encoder is composed of 4 blocks each of which has:

- 3x3 convolution layer with batch normalisation and ReLU activation
- 3x3 convolution layer with batch normalisation and ReLU activation
- 3x3 convolution layer with batch normalisation and ReLU activation
- 2x2 max-pooling

The numbers of convolutional filters in the 4 blocks of the encoder are $[32, 64, 128, 256]$, respectively.

The decoder is similarly composed of 4 blocks, each of which has:

- 2x2 up-sampling layer using nearest neighbours
- Concatenation with the feature maps output by the corresponding block of the encoder
- 3x3 convolution layer with batch normalisation and ReLU activation
- 3x3 convolution layer with batch normalisation and ReLU activation

The numbers of convolutional filters in the 4 blocks of the decoder are $[256, 128, 64, 32]$, respectively.

We vary the parameter that controls the proportion of the maximally selected pixels, β. The temperature of the Concrete distribution is empirically set to 1.0. We implement PVAR in Python with TensorFlow, trained by Rectified Adam [163] with learning rate 0.001 and batch size 32. We train the model for 300 iterations maximally and terminate the training if the loss stops dropping for 15 continuous iterations. We use the datasets' standard training set to train PVAR.

Settings of other baselines Besides the three source attacks that serve as natural baselines for the PVAR refined counterparts, we compare with several state-of-the-art sparse attacks including CornerSearch [146] (a stronger baseline than CW-l_0 [138])[3], JSMA [141][4], and SparseFool [145] (a stronger baseline than the one-pixel attack [144]) implemented in Foolbox [162]. For the sparse attacks in our experiments, we use the default settings except the ones mentioned below:

- JSMA [141]: We set the attack strength ϵ to 0.3 for MNIST and 0.03 for CIFAR10/ImageNet, which is the same as the setting of the source attacks of PVAR. For the γ parameter, which controls the number of pixels to attack, we gradually increase its value until JSMA's AdvACC is under 0.01. The specific setting of γ on MNIST/CIFAR/ImageNet is 0.3/0.3/0.1 respectively.

- SparseFool [145]: We set the attack strength ϵ to 0.3 for MNIST and 0.03 for CIFAR10/ImageNet, which is the same as the setting of the source attacks of PVAR. For the λ parameter, which controls the number of pixels to attack, we gradually increase its value until SparseFool's AdvACC is under 0.01. The specific settings of λ on MNIST/CIFAR/ImageNet are 2.0/2.0/6.0 respectively.

- CornerSearch [146]: We set $K_{max} = \lceil 0.7HW \rceil$, where H and W are the height and width of the images, respectively. We tune $\kappa = 1.5$ for MNIST and $\kappa = 1.0$ for CIFAR10 to make its AdvACC value under 0.01.

[3]Implemented in https://github.com/fra31/sparse-imperceivable-attacks
[4]Implemented in https://github.com/gongzhitaao/tensorflow-adversarial

TABLE 3.1: Comparison with sparse attacks. We keep AdvAcc roughly the same for all the attacks and compare their DetAUC, l_2, and speed. Best results are in boldface.

Attack	AdvAcc	DetAUC	l_2	Attack Speed (secs per image)
MNIST (LeNet)				
JSMA	0.007	0.88	7.21	0.03
SparseFool	0.001	0.83	5.75	0.22
CornerSearch	0.003	0.93	**3.59**	2.76
AutoAttack (dense attack as reference)	0.000	0.86	6.48	-
Our refined AutoAttack	0.001	**0.78**	3.97	**0.008**
CIFAR10 (ResNet32)				
JSMA	0.002	0.95	1.59	0.18
SparseFool	0.003	0.73	1.91	3.09
CornerSearch	0.006	0.83	0.90	1.18
BIM (dense attack as reference)	0.000	0.83	0.97	-
Our refined BIM	0.000	**0.68**	**0.81**	**0.02**
CIFAR10 (ResNet56)				
JSMA	0.005	0.87	1.24	0.39
SparseFool	0.003	0.76	5.53	10.41
CornerSearch	0.002	0.79	2.01	1.92
BIM (dense attack as reference)	0.000	0.91	0.78	-
Our refined BIM	0.004	**0.65**	**0.57**	**0.03**
ImageNet (DenseNet169)				
JSMA	0.030	-	8.30	365.86
SparseFool	0.000	-	9.15	32.92
BIM (dense attack as reference)	0.000	-	6.61	-
Our refined BIM	0.000	-	**3.70**	**0.06**

3.4.3 Results of Refinement of Dense Attacks

We first show AdvAcc, DetAUC, and l_2 distance on MNIST, CIFAR10, and ImageNet in Figure 3.6 and Figure 3.7, and Figure 3.8, respectively. For MNIST and CIFAR10, we use all the test images, while for ImageNet, we randomly sample 1,000 images from the test set. As the detector [148, 151] is originally proposed for MNIST and CIFAR10, and performs poorly on ImageNet, we omit its performance on ImageNet. We divide l_2 on MNIST and ImageNet by 10 to show it in a similar range with DetAUC and AdvAcc. We make the following remarks on the results:

- It can be observed that all the dense attacks (i.e., $\beta = 1.0$) achieve good attack performance, whose AdvAcc is zero. However, they can be easily identified by the detector, shown in the high DetAUC scores. Especially for CIFAR10, the scores are generally greater than 0.95.
- In most cases, our refinement framework can reduce at least 70% (i.e., $\beta = 0.3$) of the perturbations of the dense attacks but keep the AdvAcc nearly zero. This demonstrates that by wisely choosing the perturbations on the most vulnerable pixels, our framework can significantly reduce the number of perturbations without sacrificing the dense attacks' power. Moreover, PVAR shows its effectiveness in refining all the three attacks.
- Reducing the number of perturbations brings us the obvious advantage of the refined attacks over the source ones in terms of l_2 and DetAUC, which correspond to adversarial attacks' detectability by human and detection algorithms, respectively. For example, for AutoAttack on CIFAR10 with ResNet56, the refined AutoAttack with $\beta = 0.3$ significantly reduces the original AutoAttack's l_2 from 1.27 to 0.70 and DetAUC from 0.94 to 0.74, while keeping the AdvAcc the same.

3.4.4 Results of Comparison with Sparse Attacks

Here we generate adv images from the test images of each dataset, with various sparse attacks and our refined dense attacks. Unlike our approach, for some of the compared sparse attacks, controlling the exact number of perturbed pixels is infeasible. For fair comparison with our method, we gradually increase the regularisation parameters of the

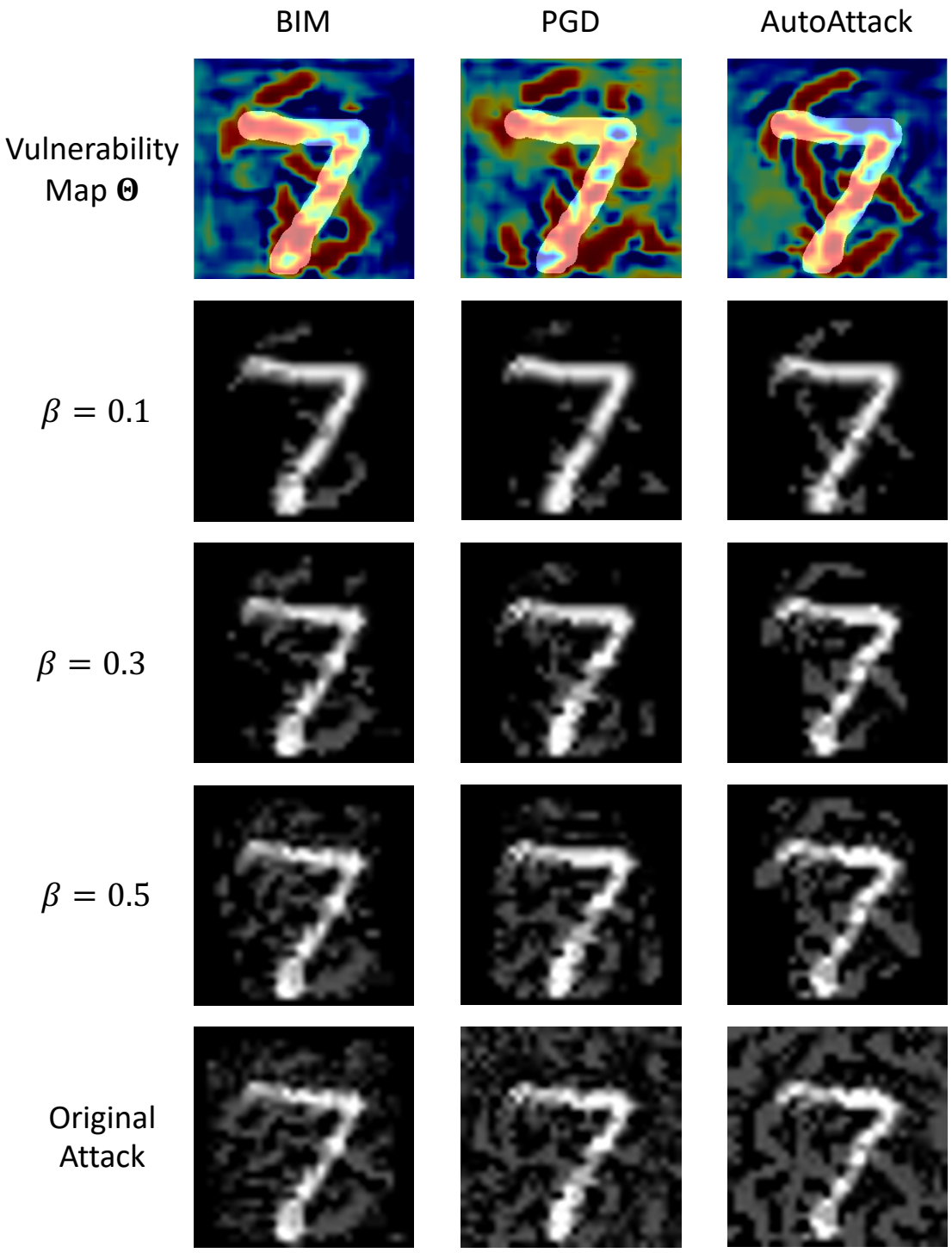


FIGURE 3.3: Visualisation of the source and refined attacks by PVAR on MNIST.

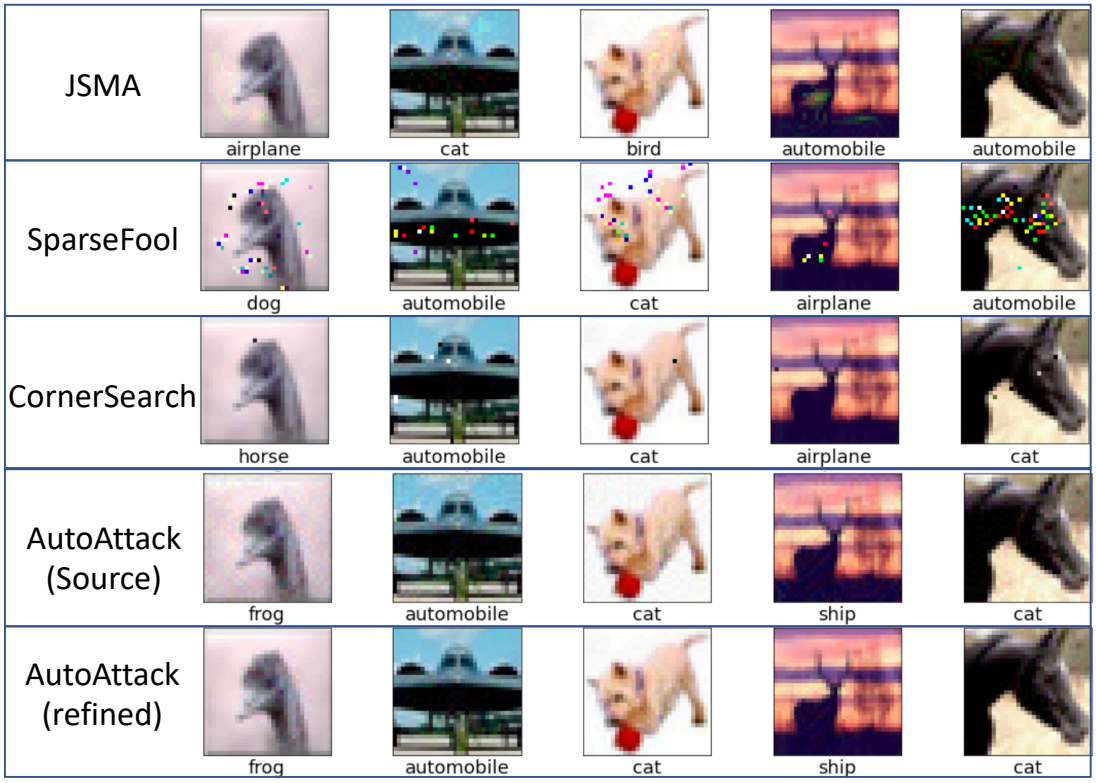


FIGURE 3.4: Sampled adversarial images on CIFAR10 with ResNet56 as the target classifier. All the attacks achieve similar AdvAcc as shown in Table 3.1. The labels below the images are the predicted ones of the classifier.

sparse attacks until their AdvAcc value on the training set is below 0.01[5]. We use the same way of selecting $\beta = 0.3$ for our PVAR. The results are shown in Table 3.1. It can be observed that with similar attack performance, our refined attacks can achieve significantly lower DetAUC and l_2. Although sparse attacks can usually use fewer perturbations, the perturbation magnitudes can be large, resulting in larger DetAUC and l_2. On the other hand, ours is choosing from the perturbations from the source attacks, whose magnitudes are originally very small. We also report the average speed for attacking on one image. All the methods run on the same machine with an NVIDIA TITAN RTX GPU. PVAR clearly shows its faster attack speed, as it is several orders of magnitude faster than the optimisation-based sparse attacks. Even on high-resolution images like ImageNet, our approach can generate an adv image in less than 0.1 sec. Note that our attack time is the sum of those of the source attack and refinement processes, both of which can be done efficiently.

[5]We are unable to further reduce AdvAcc for JSMA on ImageNet lower than 0.03 due to its efficiency.

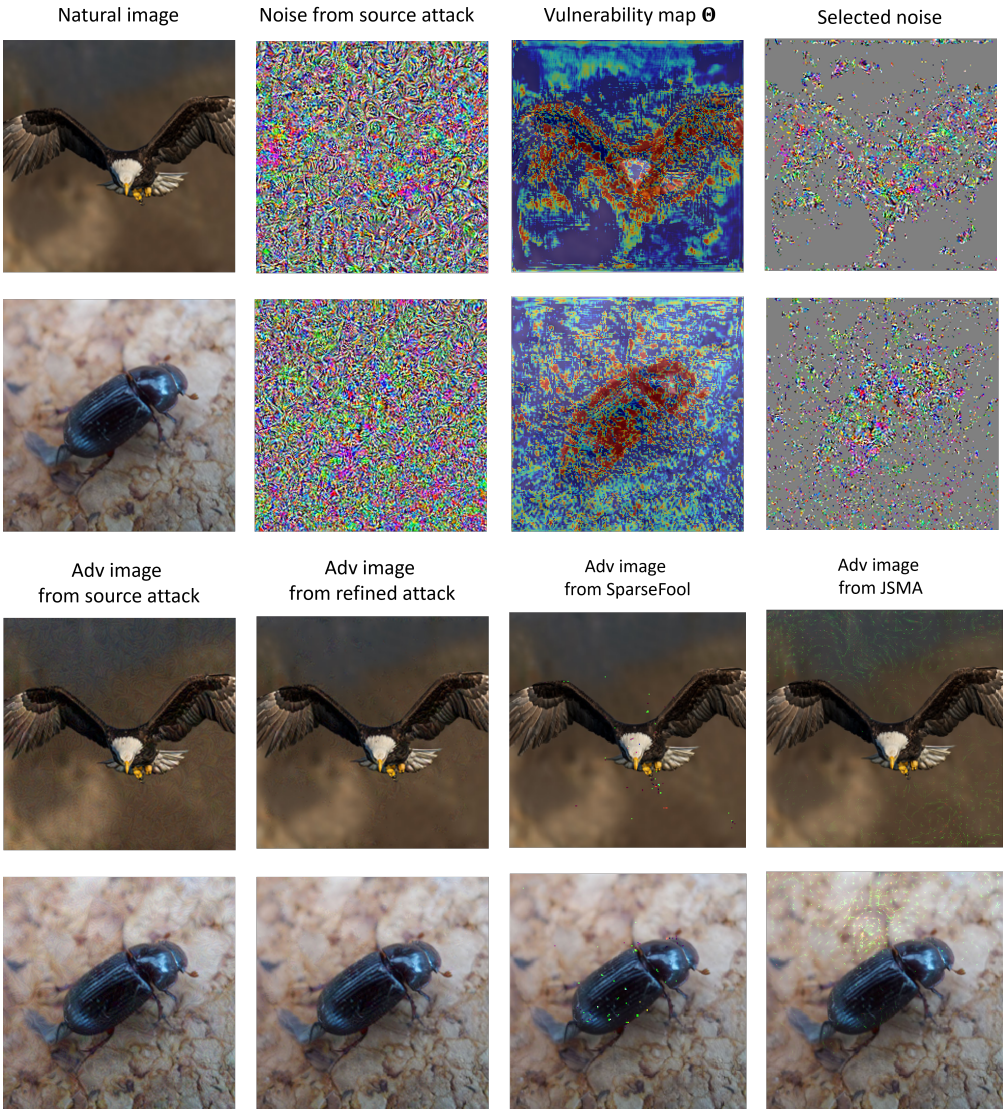


FIGURE 3.5: Top two rows: We sample two images from ImageNet, "bald eagle" and "dung beetle"; then use BIM as the source attack to generate the adversarial noise, which changes the predictions of the images to 'dung beetle" and "standard poodle", respectively; We use PVAR with $\beta = 0.3$ to get the vulnerability map and then generate the selected noise. Bottom two rows: Adversarial images generated from the source attack, our refined attack, JSMA and SparseFool, respectively.

TABLE 3.2: User study of perceivableness.

	CIFAR10	ImageNet
BIM	76.66%±5.56	82.68%±3.66
Our refined BIM	23.33%±5.56	17.32%±3.66
PGD	77.32%±3.66	81.34%±5.60
Our refined PGD	22.68%±3.66	18.66%±5.60
AutoAttack	78.68%±8.70	77.34%±7.60
Our refined AutoAttack	21.32%±8.70	22.66%±7.60

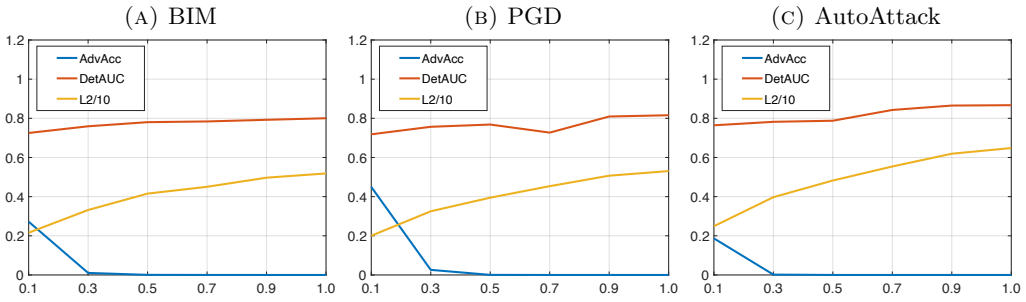


FIGURE 3.6: Refinement of various source attacks by PVAR on MNIST with LeNet as the classifier. The horizontal axis indicates the value of β. When $\beta = 1.0$, it shows the performance of the source attack without any refinement, i.e., all the perturbations are applied. "L2/10" means that we divide l_2 by 10 to show it in the similar range with DetAUC and AdvAcc.

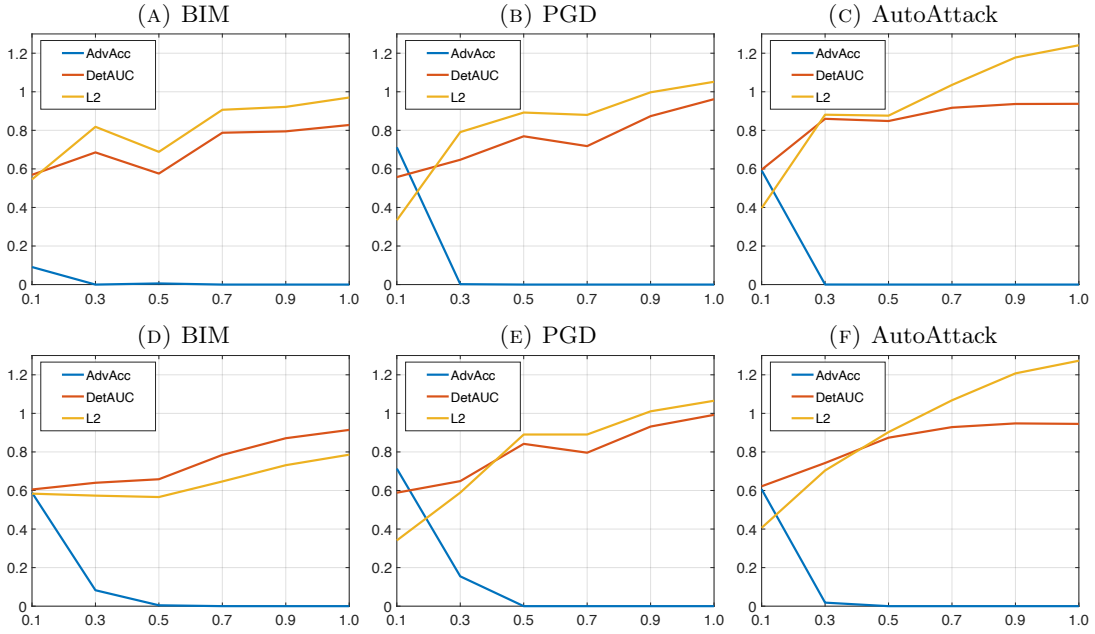


FIGURE 3.7: Refinement of various source attacks by PVAR on CIFAR10. First row: ResNet32 as the classifier. Second row: ResNet56 as the classifier. The meaning of β is the same as in Figure 3.6.

3.4.5 User Study

For each dataset, we sampled 15 natural images and generated adversarial images with the source and our refined attacks. For each pair of two adversarial images, we asked each subject person to select *the one that is with more perceivable perturbations*. We distributed the questionnaire through the Google Forms system, allowing participants to independently complete it. We gathered responses from 15 individuals, comprising both students and researchers from our university. We reported the percentages of the choices averaged over the images for each attack, which are used as the score of

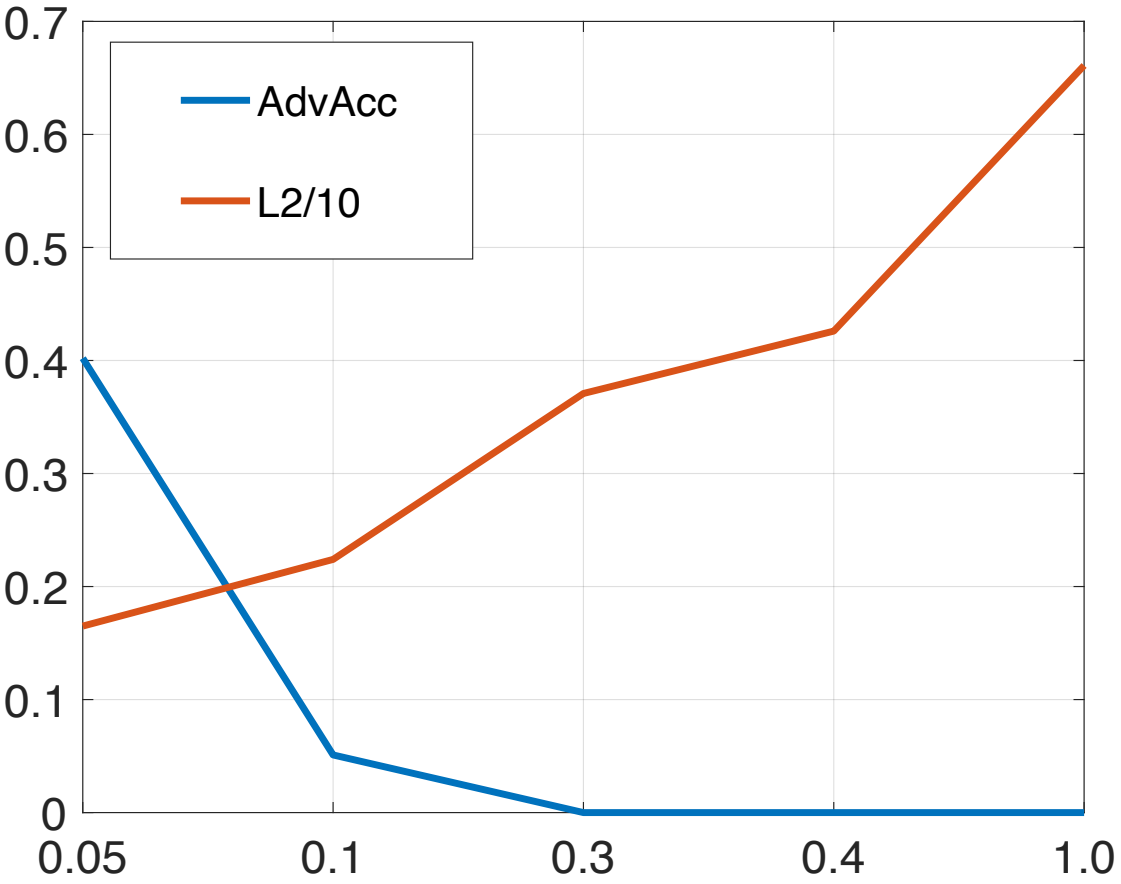


FIGURE 3.8: Refinement of BIM by PVAR on ImageNet with DenseNet169 as the classifier. The meaning of β is the same as in Figure 3.6.

"perceivableness". *A lower score indicates that an adversarial image is less perceivable to human, i.e., a better result.* The results are shown in Table 3.2. It can be seen that our method significantly reduces the "the visual impact of the attacks."

3.4.6 Transferability of Vulnerability Maps

In this experiments, we use a source attack to train our model and use the trained model to refine the generate perturbations generated from the same source attack in the testing phase. To study the transferability of the vulnerability maps learned by our model across different attacks, we use a different attack to generate perturbations in the testing phase. Table 3.3 shows the AdvAcc results on CIFAR10. It can be observed that the vulnerability maps learned by our model from one dense attack can generalise to refine other attacks.

TABLE 3.3: Transferability of vulnerability maps for ResNet32 on CIFAR10. AdvAcc is reported. "Attacks for training" means that the source attack that we use for training our model (i.e., the U-Net, $g_\alpha(\cdot)$) and "Attacks for testing" means that after $g_\alpha(\cdot)$ is trained, we use another attack to serve as the source attack to generate perturbations and then use the trained model to refine them in the testing phase. We set $\beta = 0.3$.

Attacks for training \ Attacks for testing	PGD	AT	BIM
PGD	0.002	0.000	0.001
AT	0.016	0.000	0.013
BIM	0.000	0.000	0.000

3.4.7 Visualisation on MNIST, CIFAR, And ImageNet

Figure 3.3 shows the vulnerability map and source/refined attack of an MNIST digit 7. In Figure 3.4, we qualitatively compare the adv images of our refined AutoAttack with other sparse attacks. One can observe that sparse attacks add sparse but large perturbations to natural images, which can be more easily detected by human eyes. For the original AutoAttack, although its noise is small, it spreads over the image. By removing ineffective perturbations of AutoAttack, our refined version looks significantly less noisy. Finally, we show the visualisation of ImageNet in Figure 3.5, similarly to that in Figure 3.1. It can be observed that the vulnerability maps intuitively help us understand the semantic structures of adversarial attacks. Compared with the adv images of JSMA and SparseFool, the perturbations of our refined adv images are clearly less perceptible. More visualisations are provided in the appendix.

3.4.8 Visualisation on Medical Images

To further show the interpretability of our learned vulnerability maps in medical applications, we conduct the experiment on a fundus dataset[6] with normal or abnormal[7] class. On the fundus dataset, we first learn a classifier to classify whether a fundus image is normal or abnormal and then we learn to use PVAR with PGD as the source attack

[6] Cao Thang Eye Hospital + https://ichallenge.baidu.com

[7] Myopia is an eye disease that causes distant objects to be blurry.

by attacking the classifier. For the settings of the experiment, we train a classifier with ResNet50 as the backbone that achieves 0.93 accuracy on the fundus test set. The UNet architecture and other settings are similar to the ImageNet experiment.

In Figure 3.9, we show the visualisation of the vulnerability maps learned by our method. Specifically, we can observe that many fundus images are diagnosed as abnormal due to the lesions around the lens as shown in the abnormal samples. If we look at the vulnerability maps learned by PVAR of the normal samples, it can be seen that the areas around the lens are more vulnerable (i.e., the heatmap is hotter). That is to say, if we perturb the hotter areas, it is more likely to "flip" the prediction of the classifier on a normal sample to an abnormal one.

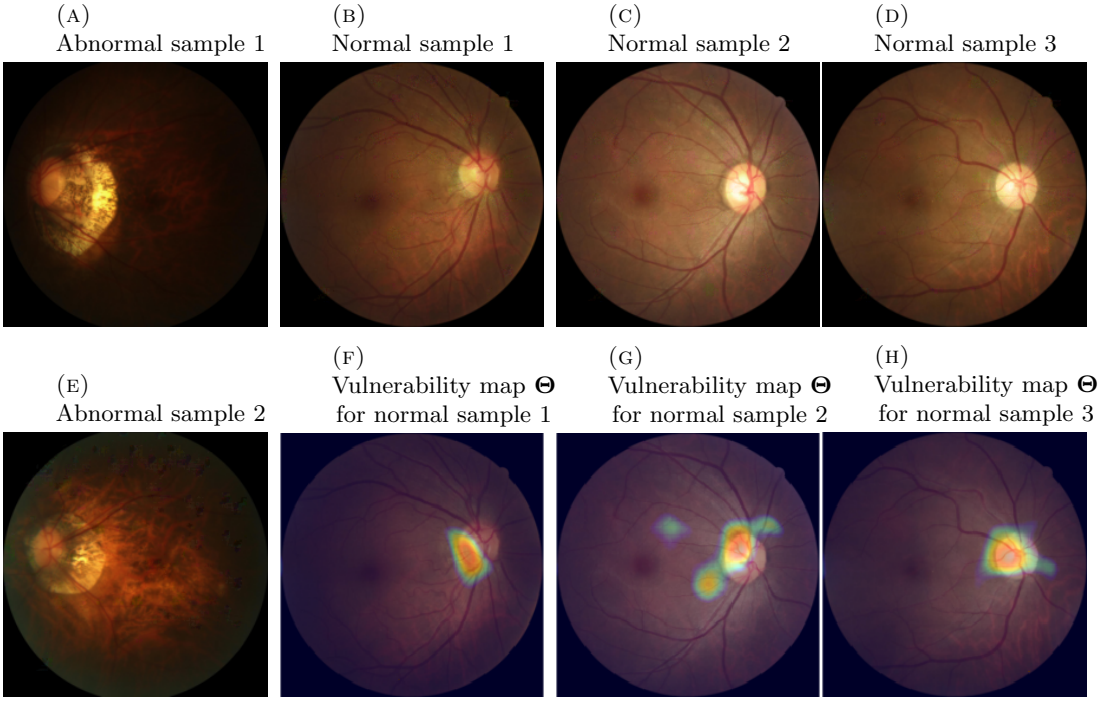


FIGURE 3.9: Visualisation on the medical dataset. (a) and (c) are examples of abnormal fundus images. (b) and (f), (c) and (g), (d) and (h) are normal samples of the fundus images and the corresponding vulnerability maps learned by PVAR.

3.5 Experiments for Adversarially Trained Models

Previously, experiments are done with undefended classifiers. Although our method is not designed to attack adversarially trained (adv-trained) classifiers, it is interesting to get a better understanding by conducting experiments with defended models.

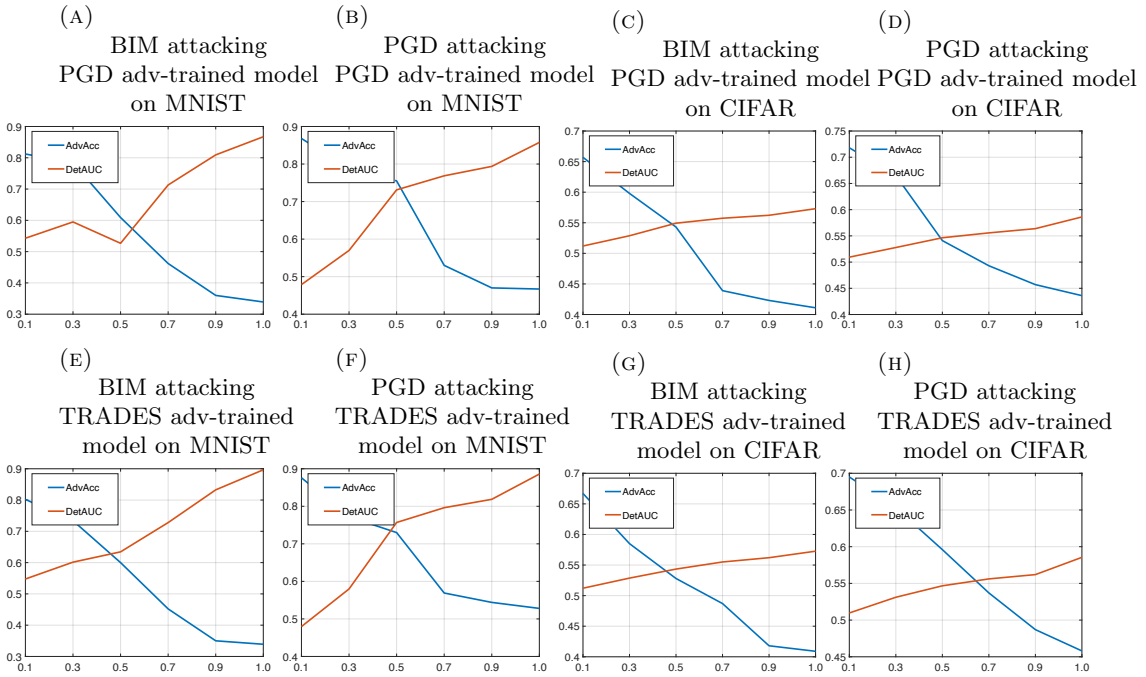


FIGURE 3.10: Refinement of various source attacks by PVAR with adv-trained classifiers.

Here we first adv-train LeNet on MNIST and ResNet32 on CIFAR with two widely used adv-training methods: PGD-Adv [22, 147] and TRADES-adv [26][8]. After the classifiers are adversarially trained, we use PVAR with PGD and BIM as source attacks to attack them, following the settings of the previous experiments except that the classifiers are replaced.

The AdvACC and DetAUC results are shown in Figure 3.10. In the undefended cases, our method can achieve nearly 100% attack performance (i.e., reducing AdvACC to nearly zero) with only 30% perturbations or less than the source attacks. For both PGD-Adv and TRADES trained classifiers, it is harder to select fewer perturbations to achieve the same attack performance compared with the case of undefended classifiers. We believe that it is understandable. Specifically, a deep neural network is a high-dimensional nonconvex function with a large amount of local minima [164], which is an important reason that small or sparse adversarial perturbations can attack well. This is also the key factor that our method can select fewer perturbations than the source attacks. If a classifier is adv-trained, its loss surface will be much smoother [165], which makes it harder to find the local minima with less perturbations. On MNIST, it can

[8]Our implementation is on top of the adversarial toolbox of Trusted-AI https://github.com/Trusted-AI/adversarial-robustness-toolbox.

be observed that reducing the number of perturbations significantly reduces DetAUC. However, this is less significant on CIFAR.

3.6 Conclusion

Summary In this chapter, we have tackled the task of attacking with fewer perturbations in a novel way, where we have proposed to refine given dense attacks by reducing their perturbations. Our idea is inspired by the observation that by carefully choosing the perturbations of a dense attack on the generated interpretation vulnerable map. Accordingly, we have proposed a probabilistic post-hoc framework that first uses a U-Net to learn the vulnerability map of a natural image and then selects from source attacks' perturbations based on the map. The framework is trained by the maximisation of the mutual information. Our method can be applied to refine an arbitrary dense attack by removing 70% of its perturbations in general. The refined attack keeps the same attack power as the original attack, but its adversarial images are significantly less detectable and closer to their natural counterparts. Compared with sparse attacks, our refined attacks usually enjoy a smaller magnitude of perturbations, leading to better l_2 and DetAUC scores. In addition, our method can generate adv images much more efficiently than sparse attacks.

Limitations We address the following limitations of the current method:

- Our method is a learning framework that learns a UNet as the backbone. Although we show that the UNet is generalisable to different source attacks, it might be less generalisable in terms of datasets. We believe that a better-designed backbone architecture or training algorithm might solve this limitation.
- Although our method works well in attacking undefended classifiers, it is less effective when the classifier is adversarially trained. We believe that it is hard for sparse attacks to attack adv-trained classifiers in general, while a more comprehensive study is needed in future work.
- We believe that the chapter can inspire new thinking, and our study on vulnerability maps can assist the development of robust models by better protecting vulnerable pixels, which can be done in future work.

Chapter 4

Transferring and Explaining Knowledge from Pre-trained Teacher Models

In this chapter, we leverage the information-theoretic framework from Chapter 3 to develop a novel knowledge distillation and model interpretation framework for image classification that jointly solves the model interpretation and knowledge distillation. Our proposed approach involves creating a smaller, student model, requiring less data and distilled knowledge, from a larger, pre-trained teacher models. To interpret the teacher model as well as assist the learning of the student, an explainer module is introduced to highlight the regions of an input medical image that are important for the predictions of the teacher model. Furthermore, the joint framework is trained by a principled way derived from the information-theoretic perspective. Our framework performance is demonstrated by the comprehensive experiments on the knowledge distillation and model interpretation tasks compared to state-of-the-art methods on a medical imaging Fundus disease dataset.

4.1 Introduction

Recently, advanced machine learning methods such as Convolutional Neural Networks (CNNs) [159] have shown remarkable performance in the medical imaging domain such

as U-Net [157] for image segmentation, ResNet [16] for image classification and medical image reconstruction [166]. A practical scenario of medical image classification applications [167] is considered, where a central hospital headquarter gathers data from multiple local branches in Figure 4.1(a). The headquarter has developed a large CNN model for disease classification with excellent performance trained on a big dataset, which is the global model to be distributed to the branches. Given the limited computation, a branch wants to develop a customised smaller model using its local data. The branch cannot access the big dataset of the headquarter because of privacy and sensitivity concerns. To assist the development of the local model, the knowledge from the global model is transferred to the local one [3]. For the medical domain, model interpretation is highly desirable. Therefore, the local model should have two capabilities: explaining the global model and transferring the knowledge of the global model to the local model with its local data only.

Model perceptive interpretation is defined by the ability to identify the areas of an input image that are important to the prediction of the classifier. Neural saliency such as Grad-CAM [79] is used to locate feature that contributes the most to the classification output. Feature selection [100], hard attention [96] and soft attention [88] are used to generate different weights for different features. However, they are not designed for explaining a pre-trained global model. The recent Learning-to-Explain (L2X) [100] trains an explainer to explain a pre-trained global model by maximising the mutual information between selected instance-wise features and the teacher outputs. L2X does not address the issue of lack of large training data and its effectiveness on high-resolutional image classification has not been confirmed.

Knowledge distillation is a process of transferring knowledge from the complicated global model (called teacher) to a smaller lighter-weighted one (called student). The small student model can significantly reduce the deployment cost of the local branch. KD was first introduced by Hinton et al. [3] to distill knowledge from the distribution of class probabilities predicted by the teacher model. Recently, Ahn et al. [112] exploited the information-theoretic perspective as maximising the mutual information between the teacher and the student in order to transfer knowledge named (VID). In the medical domain, Wang et al. [168] used KD to train a student model that speeds up the inference time of a 3D neuron segmentation model. However, these previous approaches do not

consider to interpreting the complicated teacher model. Thus, in this chapter, we propose an end-to-end framework to address the above two requirements simultaneously, i.e. we aim to learn a small medical image classification model with less training data but better interpretability. Here we define the interpretability as the ability to identify the areas of an input image that are important to the prediction of the classifier.

Figure 4.1(a) gives an overview of the proposed framework, which consists of a teacher T that is the pre-trained globe model, a learnable "student" S that is the local model extracting the teacher's knowledge, and an explainer E that explains the teacher. The student is expected to be significantly smaller than the teacher to reduce the computational cost. Specifically, given an input image, the explainer highlights the important pixels for the decision of the teacher and suppresses the unimportant pixels, which addresses the interpretability aim. Then, the explainer facilitates the learning of the student by providing it a simplified input image. In this way, the student does not need to learn from the scratch, but focuses on the important regions that the explainer explains, and at the same time the teacher's knowledge is transferred to the student, which addresses the aim of training a small model. Interestingly, the above two aims can be jointly achieved by optimising a joint training objective derived from an information-theoretic perspective by pushing the output of the last layer and intermediate layers of the student close to those of the teacher (see Section 4.3.2).

It is noteworthy that since the explainer selects important image regions, our work appears to be similar to weakly supervised image segmentation with only image-level annotations [91, 169–171]. The fundamental difference is that weakly supervised image segmentation is for the purpose of generating best segmentation with only image-level labels, while our partial goal is to identify the most important image regions w.r.t the teacher's prediction, with the other aim of learning a small student model.

Our contributions can be summarised as follows:

- We propose a new end-to-end Transferring and EXplaining framework for MEDical imaging (MED-TEX) from a pre-trained global model, which combines knowledge distillation and model interpretation. To our knowledge, our approach is novel in solving two important issues in medical imaging in a joint framework: the lack of training data and the lack of interpretation. Existing methods only focus on either of them.

- We develop a joint training objective for our framework, derived from an information-theoretic perspective. Specifically, we introduce to maximise the mutual information between not only the output layers but also the intermediate layers of the student and the teacher, which is both theoretically and practically appealing.

- Extensive experimental results demonstrate that our proposed method achieves better performance on the evaluations of both knowledge distillation and model interpretability. Our approach outperforms many others in identifying important image regions, including soft attention [84, 88], hard attention[96], learning to explain [100], and Grad-CAM [79].

4.2 Related Work

In this section, we review related work in the medical domain including knowledge distillation, model interpretation, and image segmentation with only image-level annotations.

Knowledge distillation This is a process of transferring knowledge from a complicated pre-trained model (teacher) to a smaller lighter-weighted one (student). The student is particularly useful in the cases where computational resources and deployment costs need to be significantly reduced at the inference stage. KD was introduced originally by Hinton et al. [3] to extract knowledge from the distribution of class probabilities predicted by the teacher model. There are also some attempts to apply KD in the medical imaging domain. For example, Wang et al. [168] used KD to train a student model that speeds up the inference time of a 3D neuron segmentation model. The work of [172] leveraged KD for brain lesion segmentation with soft labels by dilating mask boundaries. Transferring knowledge from multiple sources to promote lung pattern analysis was introduced by Christodoulidis et al. [173]. KD was also explored for improving unpaired multi-modal segmentation in [174]. Compared with these existing KD methods, our framework can not only transfer knowledge from the teacher, but also interpret the teacher's behaviours by introducing the explainer, which is critical to medical imaging applications.

Model interpretation The rapid growth of machine learning in many applications leads to a strong requirement for model interpretation, especially in the medical imaging domain. Soft attention can also be applied for weakly supervised image segmentation [91]. Although attention can highlight semantic regions, it is usually trained for

the purpose of maximising the classification accuracy, not for explaining the pre-trained teacher model in our setting. The recent Learning-to-Explain (L2X) approach [100] is the most relevant one. It trains an explainer to explain the pre-trained teacher model by maximising mutual information between selected instance-wise features and the teacher outputs. Its feature selection is based on hard attention with Gumbel-softmax trick. Compared with [100], our method generates soft attention in the pixel domain instead of the feature domain of input images. Moreover, our method also learns a smaller student model with only local data, with information distillation from intermediate layers, which is not considered in L2X.

Image segmentation with only image-level annotation Our pixel selection for model explanation essentially generates some segmentation results. This is related to weakly supervised image segmentation with only image-level annotations. Both CAM and attentions have been applied for weakly supervised image segmentation in medical imaging domain. For example, Izadyyazdanabadi et al. [169] applied CAM for diagnostic brain tumor segmentation in confocal laser endomicroscopy glioma images and the work from Feng et al. [170] introduced a coarse image segmentation followed by a fine instance-level segmentation. Rajpurkar et al.[171] also used CAM for chest X-ray segmentation. In the work [175], both "hard" and "soft" attentions are used for robust brain magnetic resonance image segmentation for hydrocephalus patients. In contrast, our method is mainly designed to identify the most important regions of an input image to the teacher's prediction but not to generate best segmentation, although those important regions are highly overlapped with segmentation because of the well-trained teacher's behaviours. For example, when the teacher predicts a certain disease, our method is trained to detect which parts of the image that cause the disease based on the prediction of the teacher. Moreover, our goal is to train a smaller student model that can achieve similar classification performance as the teacher with only local data, while simultaneously be able to explain the teacher via pixel selection, which is expected to match segmentation to a certain extent.

4.3 Method

In this section, we introduce the details of our Transferring And Explaining Knowledge From Pre-trained Models (TEX) which includes a fixed pre-trained teacher and two

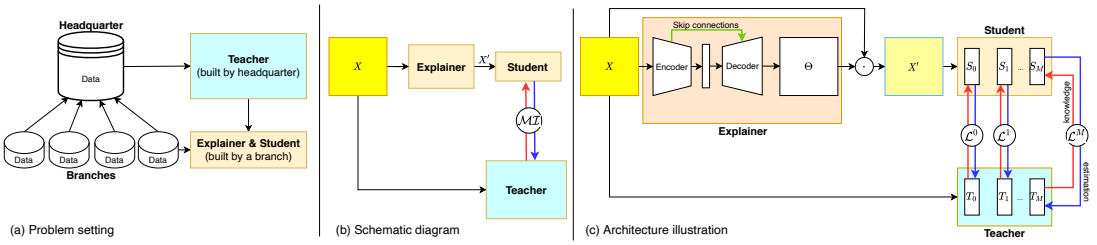


FIGURE 4.1: (a) Problem setting: a headquarter gathers data from multiple branches to produce a shared cumbersome teacher. A branch builds a local small and interpretable model. (b) An overview of our framework: fixed pre-trained teacher, learnable explainer and learnable student. The explainer explains to the student by producing a simplified $\mathbf{X}'$ from input $\mathbf{X}$. The knowledge from teacher is transferred to the student by maximising the mutual information (I). (c) The detailed architecture.

trainable modules called explainer and student, as illustrated in Figure 4.1. Recalling the hospital example in the introduction, suppose that a CNN-based classifier (teacher, T) is pre-trained to classify images which is usually a cumbersome model in order to adapt to the large-scale dataset from the headquarter. The student S is another CNN-based classifier that can be more than a hundred times smaller to significantly reduce computational complexity. It is noteworthy that the dataset used for training the teacher may not be accessible to us due to sensitivity or privacy. With a raw input image, $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ (C, H, W are the channels, height, and width of the image, respectively), the explainer E inspired by the U-Net [157] architecture produces selection scores $\mathbf{\Theta}$, which give high scores for the important pixels for the decision of the teacher and low scores for the unimportant ones. In our framework, $\mathbf{\Theta}$ has the same size to $\mathbf{X}$ and is element-wise multiplied by $\mathbf{X}$ to get a simplified the input image, denoted by $\mathbf{X}'$. This $\mathbf{X}'$ then is input to the student S to perform predictions. Our goal is training the student to mimic the behaviours of the teacher by pushing teacher's outputs from the last and intermediate layers close to student's outputs while the explainer makes use of $\mathbf{\Theta}$ guide the student by simplifying input $\mathbf{X}$ into $\mathbf{X}'$, as illustrated in Figure 4.1. The architecture details of the teacher, student and explainer will be elaborated on later in Section 4.4.

4.3.1 Proposed Framework

Here we denote the teacher's and student's predicted distributions over the labels as $\boldsymbol{y}^T \in \Delta^K$ and $\boldsymbol{y}^S \in \Delta^K$, respectively, where K is the number of labels and Δ^K denotes the K dimensional simplex. We assume there are M layers of the CNNs of the teacher and the student, where the first to $(M-1)^{\text{th}}$ layers are convolutional layers (or block convolution

layers) and the last one is a fully connected layer. These predicted distributions are from the output (M^{th}) layers of the teacher and the student. We further have $\boldsymbol{y}^T = T(\mathbf{X})$ (i.e., $\mathrm{p}\, y_k^T|\mathbf{X} \propto T(\mathbf{X})_k$), $\mathbf{X}'|\mathbf{X} = E(\mathbf{X})$, and $\boldsymbol{y}^S|\mathbf{X}' = S(\mathbf{X}')$ (i.e., $\mathrm{q}\, y_k^S|\mathbf{X}' \propto S(\mathbf{X})_k$). With these notations, we can formulate our preliminary goals of explaining and extracting the teacher's knowledge to the student as the following loss derived from mutual information (See the derivation in Equation 4.10.

$$\ell^M = \min_{E,S} -\mathbb{E}_{\mathbf{X}}\Big[\mathbb{E}_{\mathbf{X}'|\mathbf{X}}\Big[\mathbb{E}_{\boldsymbol{y}^T|\mathbf{X}'}[\log q(\boldsymbol{y}^T|\mathbf{X}')]\Big]\Big], \tag{4.1}$$

where q corresponds to our student, acting as the variational distribution in the deviation of mutual information in Section 4.3.2.

Essentially, Equation 4.1 can be understood as minimising the cross-entropy loss between the outputs of the teacher and the student, and generating $\mathbf{X}'$ by element-wise multiplication between $\mathbf{X}$ and $\boldsymbol{\Theta}$, aiming to push the predictions of the student close to those of the teacher, with the help from the explainer:

$$\ell^M = \min_{E,S} -\mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{\mathbf{X}'|\mathbf{X}}\left[\sum_k^K \mathrm{p}\, y_k^T|\mathbf{X} \log \mathrm{q}\, y_k^S|\mathbf{X}'\right]\right]. \tag{4.2}$$

Next, we introduce the detailed construction of the explainer. Specifically, given an input image $\mathbf{X}$, the explainer generates an importance score for each of its pixels. The higher the important score is, the more important the corresponding pixel is to the prediction of the teacher. All the importance scores form the importance map[1], denoted as $\boldsymbol{\Theta} \in [0,1]^{C\times H\times W}$. In this way, the output of the explainer can be expressed as

$$\mathbf{X}' = \boldsymbol{\Theta} \odot \mathbf{X}, \tag{4.3}$$

where $\odot$ is the element-wise multiplication.

We construct the explainer E with a neural network inspired by U-Net [157], which can output a high resolution probability map (typically same size as input), denoted as $\boldsymbol{\Theta}|\mathbf{X} = U_E(\mathbf{X})$.

Specifically, the explainer produces $\boldsymbol{\Theta}$ scores for both channels and spatial locations of $\mathbf{X}$. The channel selection is via a fully connected layer with sigmoid activation function,

[1]For each pixel, we consider its position as well as its channels to be with different importance scores.

which takes the output of the explainer's encoder. The channel selection component is especially beneficial for medical images with multiple channels. The spatial selection is the output from the decoder of the explainer, where the last layer is a 1×1 convolution layer with sigmoid activation, as shown in Figure 4.1(b).

Note that in the loss of Equation 4.2, the student only learns from the predictions (i.e., the final output layer) of the teacher. Although our ultimate goal is to let the student generate the same predictions of the teacher, the knowledge in the intermediate layers of the teacher can also be informative to the learning of the student[35].

Inspired by the idea of knowledge distillation in [3, 35, 112], we therefore introduce an additional loss to maximise the mutual information between the outputs of each i^{th} intermediate layer of the teacher ($T^i(\mathbf{X})$) and the student ($S^i(\mathbf{X}')$):

$$\ell^i = \min_{E,S} -\mathbb{E}_{\mathbf{X}} \left[\mathbb{E}_{\mathbf{X}'|\mathbf{X}} \left[\log r(T^i(\mathbf{X})|S^i(\mathbf{X}')) \right] \right], \tag{4.4}$$

where $r(T^i(\mathbf{X})|S^i(\mathbf{X}'))$ is a variational distribution used for approximating $\mathrm{p}\left(T^i(\mathbf{X})|S^i(\mathbf{X}')\right)$, which is derived from information-theoretic perspective (see Equation 4.12).

Recall that the output of the i^{th} layer of the teacher is a $C^i \times H^i \times W^i$ feature map (note that the output of the i^{th} layer of the student is of the same spatial dimension but with a smaller number of channels). Following [112], we model $T^i(\mathbf{X})$ as the following Gaussian distribution conditioned on $S^i(\mathbf{X}')$:

$$r\left(T^i(\mathbf{X})|S^i(\mathbf{X}')\right) \sim \prod_{c=1,h=1,w=1}^{C^i,H^i,W^i} \mathcal{N}\left(\mu^i(S^i(\mathbf{X}'))_{c,h,w}, \sigma_c^{i^2}\right), \tag{4.5}$$

where μ^i is a subnetwork with 1×1 convolutional layers to match the channel dimensions between $T^i(\mathbf{X})$ and $S^i(\mathbf{X}')$, $\mu^i_{c,h,w}$ is a single output unit, and $\sigma_c^{i^2}$ is the learnable parameter specific to each channel at the i^{th} layer. For $\sigma_c^{i^2}$, we exploit the softplus function $\sigma_c^{i^2} = \log(1 + e^{\alpha_c^i}) + \epsilon$ where α_c^i is a learnable parameter and ϵ is used for numerical stability.

With Equation 4.5, we can write Equation 4.4 as:

$$\ell^i = \min_{E,S} \mathbb{E}_{\mathbf{X}} \Bigg[\mathbb{E}_{\mathbf{X}'|\mathbf{X}} \Bigg[\sum_{c=1,h=1,w=1}^{C^i,H^i,W^i} \log \sigma_c^i + \frac{(T^i(\mathbf{X})_{c,h,w} - \mu^i(S^i(\mathbf{X}'))_{c,h,w})^2}{2\sigma_c^{i2}} + \text{const.} \Bigg] \Bigg]. \tag{4.6}$$

Finally, the overall loss function of our framework can be written as

$$\ell = \ell^M + \lambda \sum_{i=1}^{M-1} \ell^i, \tag{4.7}$$

where λ is the weight of the losses of the intermediate layers.

4.3.2 Derivation from Information-theoretic Perspective

Previously, we have shown that the objective function of our proposed framework has intuitive interpretations. Here we additionally demonstrate that the objective function can be derived in a theoretical way with mutual information, which is a widely-used measure of the dependence between two random variables and captures how much knowledge of one random variable reduces the uncertainty about the other [176]. In particular, we note: minimising the training losses in Equation 4.2 and Equation 4.4 are equal to maximising the following mutual information: $I(\mathbf{X}'; \boldsymbol{y}^T)$ and $I(T^i(\mathbf{X}); S^i(\mathbf{X}'))$, respectively.

$$\max_{E,S} \; I(\mathbf{X}'; \boldsymbol{y}^T) + \lambda \sum_{i=1}^{M-1} I(T^i(\mathbf{X}); S^i(\mathbf{X}')). \tag{4.8}$$

Given the definition of mutual information, the first term of Equation 4.8 can be derived as:

$$\begin{aligned} I(\mathbf{X}'; \boldsymbol{y}^T) &= H(\boldsymbol{y}^T) - H(\boldsymbol{y}^T|\mathbf{X}') \\ &= \mathbb{E}_{\mathbf{X}} \mathbb{E}_{\mathbf{X}'|\mathbf{X}} \mathbb{E}_{\boldsymbol{y}^T|\mathbf{X}'} [\log p(\boldsymbol{y}^T|\mathbf{X}')] + Const. \end{aligned} \tag{4.9}$$

In general, it is impossible to compute expectations under the conditional distribution of $p(\boldsymbol{y}^T|\mathbf{X}')$. Hence, we define a variational distribution $q(\boldsymbol{y}^T|\mathbf{X}')$ that approximates

$p(\boldsymbol{y}^T|\mathbf{X}')$:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{y}^T|\mathbf{X}'}[\log p(\boldsymbol{y}^T|\mathbf{X}')] \\
&\quad = \mathbb{E}_{\boldsymbol{y}^T|\mathbf{X}'}[\log q(\boldsymbol{y}^T|\mathbf{X}')] \\
&\qquad + \mathbb{D}_{KL}[q(\boldsymbol{y}^T|\mathbf{X}')||p(\boldsymbol{y}^T|\mathbf{X}')] \\
&\quad \geq \mathbb{E}_{\boldsymbol{y}^T|\mathbf{X}'}[\log q(\boldsymbol{y}^T|\mathbf{X}')],
\end{aligned} \tag{4.10}$$

where $\mathbb{D}_{KL}$ is the Kullback–Leibler divergence and equality holds if and only if $q(\boldsymbol{y}^T|\mathbf{X}')$ and $p(\boldsymbol{y}^T|\mathbf{X}')$ are equal in distribution. Note that it is not hard to show that our student corresponds to the variational distribution q.

For the second term of Equation 4.8, we have:

$$\begin{aligned}
&I(T^i(\mathbf{X}); S^i(\mathbf{X}')) = H(T^i(\mathbf{X})) - H(T^i(\mathbf{X})|S^i(\mathbf{X}')) \\
&=\mathbb{E}_{\mathbf{X}}\mathbb{E}_{\mathbf{X}'|\mathbf{X}}\mathbb{E}_{T^i|\mathbf{X},S^i|\mathbf{X}'}[\log p(T^i(\mathbf{X})|S^i(\mathbf{X}'))] + Const
\end{aligned} \tag{4.11}$$

Given Equation 4.11, we can derive the following formula, similar to Equation 4.10:

$$\begin{aligned}
&\mathbb{E}_{T^i|\mathbf{X},S^i|\mathbf{X}'}[\log p(T^i(\mathbf{X})|S^i(\mathbf{X}'))] \\
&\quad \geq \mathbb{E}_{T^i|\mathbf{X},S^i|\mathbf{X}'}[\log r(T^i(\mathbf{X})|S^i(\mathbf{X}'))],
\end{aligned} \tag{4.12}$$

where r is the variational distribution to approximate the conditional distribution.

By using the two variational distributions q and r, the problem Equation 4.8 can be relaxed to Equation 4.13, i.e. maximising the variational lower bounds.

$$\max_{E,S} \ \mathbb{E}[\log q(\boldsymbol{y}^T|\mathbf{X}')] + \lambda \sum_{i=1}^{M-1} \mathbb{E}[\log r(T^i(\mathbf{X})|S^i(\mathbf{X}'))]. \tag{4.13}$$

4.4 Experiments

In this section, we present the experiments conducted on real-world datasets to examine the performance of the proposed MED-TEX against the state-of-the-art methods.

Table 4.1: Teacher and student model architecture.

Teacher	Student
3×3 conv, 32, pad=1; ReLU [b1]	3×3 conv, 2, pad=1; ReLU [b1]
2×2 max pooling	2×2 max pooling
3×3 conv, 64, pad=1; ReLU [b2]	3×3 conv, 4, pad=1; ReLU [b2]
2×2 max pooling	2×2 max pooling
3×3 conv, 128, pad=1; ReLU [b3]	3×3 conv, 8, pad=1; ReLU [b3]
2×2 max pooling	2×2 max pooling
3×3 conv, 256, pad=1; ReLU [b4]	3×3 conv, 16, pad=1; ReLU [b4]
fully connected layer	fully connected layer
softmax	softmax

Table 4.2: Explainer model architecture.

Encoder
2× (3×3 conv, 32, pad=1; Batch Norm; ReLU) [e0]
2×2 max pooling; 2× (3×3 conv, 16, pad=1; batch norm; ReLU) [e1]
2×2 max pooling; 2× (3×3 conv, 8, pad=1; batch norm; ReLU) [e2]
2×2 max pooling; 2× (3×3 conv, 4, pad=1; batch norm; ReLU) [e3]
2×2 max pooling; 2× (3×3 conv, 2, pad=1; batch norm; ReLU) [e4]
2×2 max pooling; 2× (3×3 conv, 2, pad=1; batch norm; ReLU) [e5]
Decoder
2×2 nearest upsample; 2× (3×3 conv, 2, pad=1; Batch Norm; ReLU) [d4]
concatenate [d4, e4]; 2×2 nearest upsample; 2× (3×3 conv, 4, pad=1; Batch Norm; ReLU) [d3]
concatenate [d3, e3]; 2×2 nearest upsample; 2× (3×3 conv, 8, pad=1; Batch Norm; ReLU) [d2]
concatenate [d2, e2]; 2×2 nearest upsample; 2× (3×3 conv, 16, pad=1; Batch Norm; ReLU) [d1]
concatenate [d1, e1]; 2×2 nearest upsample; 2× (3×3 conv, 32, pad=1; Batch Norm; ReLU) [d0]
1×1 conv, 1, pad=0; sigmoid [output]

4.4.1 Architectures and Settings of MED-TEX

For the teacher and student, we adopt a deep architecture with 4 block CNN layers, shown in Table 4.1. It is important to note that with less number of filters, the size of the student model is much (226 times) smaller than the teacher, i.e., 1.7k parameters of the student versus 390.5k parameters of the teacher. We pre-trained the teacher on the training set, which achieves 96.33% accuracy, 0.964 precision, 0.963 recall and 0.96 F1 score on the testing data.

For the explainer, we adopt the U-Net architecture [157], which takes an image $\mathbf{X}$ as input and outputs the selection score $\mathbf{\Theta}$, shown in Table 4.2. Note that of $\mathbf{\Theta} \in \mathbb{R}^{C\times H\times W}$ can be decomposed into a $1\times H\times W$ tensor that models the spatial selection and a $C\times 1$ tensor that models the channel selection. The spatial selection tensor is the output from the decoder and the channel selection tensor is generated by passing the output of the encoder (i.e., [e5] in Table 4.2) through a fully connected neural network with sigmoid activation function. Finally, $\mathbf{\Theta}$ is obtained by matrix multiplication between the spatial and channel selection tensors.

There are four ℓ^i losses for the intermediate layers of the teacher and student, i.e., convolutional blocks b1, b2, b3 and b4 in Table 4.1. Each ℓ^i consists of a subnetwork μ^i and learnable scalar α_c^i. The subnetwork of μ^i consists of a 1×1 convolutional layer (with 16, 32, 64 and 128 numbers of filters respectively), ReLU activation, and a 1×1 convolutional layer (with 32, 64, 128 and 256 numbers of filters respectively). We empirically set $\lambda = 0.001$.

4.4.2 Datasets

We conducted our experiment on a Fundus dataset collected from two sources: Baidu iChallenge[2] and Cao Thang International Eye Hospital (CTEH)[3]. The dataset consists of two kinds of Fundus images: the ones with pathological myopia[4] (abnormal images) and the normal ones without the disease.

Figure 4.2 shows the data processing procedure. Given a raw image with pathological myopia, it was firstly preprocessed by cropping off the background. Next, the lesion

[2]https://ichallenge.baidu.com
[3]http://cteyehospital.com
[4]It is an eye disease that causes distant objects to be blurry.

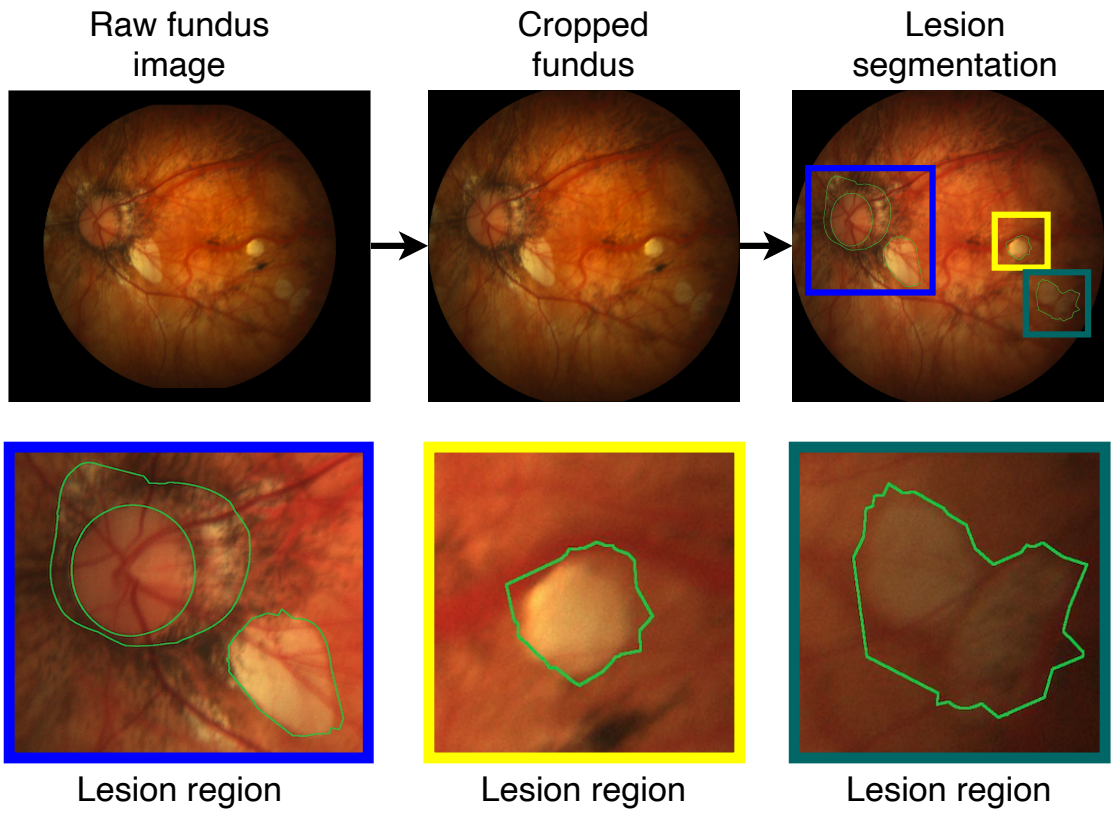


FIGURE 4.2: An example in Fundus dataset. Fine-grained lesion regions are inside the contour of the three images in the 2nd row, which are the zoom-in versions of the three lesion regions identified in the 3rd column of the 1st row.

regions of the image were identified and segmented by medical experts. For normal images, they were collected from the CTEH electronic health record. Finally, we have 1873 images in total, which consists of 1073 normal and 800 abnormal images. For the abnormal images, there are 200 of them with fine-grained lesion segmentations.

Originally, these images are in various sizes, so we rescaled all of them to $3\times256\times256$ (3 is the number of channels) and normalised their values in the range between 0 and 1. We split the dataset into the training (773 normal and 500 abnormal images) and testing (300 normal and 300 abnormal images) sets. It is noteworthy that all the 200 images with lesion segmentations are in the testing set. To mimic the case where we have less data to learn from and explain the teacher, we further reduce the number of training images for MED-TEX, i.e., 25%, 50% and 100% training images are used, denoted as Fundus-25%, Fundus-50% and Fundus-100%, respectively.

TABLE 4.3: Abbreviation of the compared methods

Method	Abbreviation
Resnet18 + hard attention using Gumbel-softmax [96]	Hard attention
Resnet18 + soft attention using [84, 88]	Soft attention
Resnet18 + patch-based image selection using Gumbel-softmax [100]	L2X
Resnet18 + grad class activation map [1, 79]	Grad-CAM
Our student without explainer	Student (only)
Our Transfer and EXplain framework using only explain ℓ^M (Equation 4.1) loss	MED-EX
Our full Transfer and EXplain framework	MED-TEX

4.4.3 Compared Methods

There is no existing method with the same problem setting as ours. So we compare our MED-TEX with the representative model interpretation methods that can be adapted to our scenario, including "hard" attention using Gumbel-softmax trick [96], "soft" attention [84, 88], Grad-CAM [1, 79] and L2X [100]. In particular, for all the compared methods, we leveraged ResNet18 [16] without fully connected layers for feature extraction, which are further input into those methods to generate model interpretations. The "hard" attention using Gumbel-softmax trick [96] discretely samples the feature domain extracted by ResNet18, followed by a fully connected layer with softmax to produce predictions. In the same context, "soft" attention [84, 88] and Grad-CAM [1, 79] also perform feature selection on the feature domain. All these three models are trained by the cross-entropy loss with the labels being generated by the teacher model. We adapt L2X [100], which has not been carefully studied for image classification, by using ResNet18 for the explainer and a similar student architecture as ours. Note that L2X is trained by minimising ℓ^M only, without the intermediate losses.

To demonstrate the effectiveness of knowledge transformation of the intermediate layers, we compare MED-TEX with its variant without information transfer losses (Equation 4.4), denoted as MED-EX. The loss for training MED-EX is the same to L2X, but

the ways of constructing the explainer are totally different in the two models. To illustrate the importance of the explainer, we also consider another variant, i.e., the student without the explainer, which was trained on the raw input image X. We summarise all these comparison methods and their abbreviations in Table 4.3. All models are trained by using Adam with 0.001 learning rate and a batch size of 64 on an NVIDIA RTX Titan GPU with 24GB memory.

4.4.4 Evaluation Metrics

Post-hoc metric To evaluate our MED-TEX, we use post-hoc metric [100] which compares the predictive distributions of the student given $\mathbf{X}'$ and the teacher given $\mathbf{X}$. In other words, we compute the accuracy, precision, recall and F1 score of the outputs from different methods against the output of the pre-trained teacher on the testing dataset.

Intersection over Union (IoU) We compare Intersection over Union (IoU) between the highlighted image regions and the ground-truth lesion segmentation of abnormal images. Note that hard attention, soft attention, Grad-CAM, and L2X output patch-based region selection maps (the ResNet18 feature extraction outputs a feature map of 8×8 spatial size each of which is corresponding to a 32×32 region in image domain), while our MED-EX and MED-TEX give pixel-level selection scores. For a better comparison, we rank feature scores and select the number of pixels corresponding to the top K highest scores (e.g., top$K \in \{k \times 32 \times 32 \mid k = 1, 2, 3, 4, 5, 6\}$):

$$IoU_{topK} = 2\frac{\mathbf{\Theta}_{topK} \bigcap \mathbf{X}_{lesion}}{\mathbf{\Theta}_{topK} \bigcup \mathbf{X}_{lesion}}, \tag{4.14}$$

where $\mathbf{\Theta}_{topK}$ indicates the selected pixels corresponding to the top$K$ feature scores and $\mathbf{X}_{lesion}$ denotes ground-truth lesion segmentation pixels.

4.4.5 Results

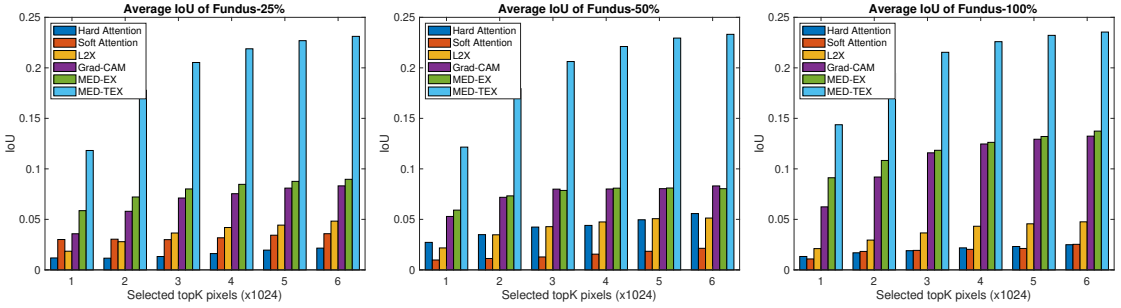
In order to compare our MED-TEX to other methods, we first use post-hoc metric [100]. Our method consistently outperforms hard attention using Gumbel-softmax trick [96], soft attention [84, 88], Grad-CAM [1, 79], and learning to explain [100] in term of both accuracy and F1 score, as shown in Table 4.4. Especially, for our proposed method,

TABLE 4.4: Post-hoc evaluation on the Fundus dataset.

	Fundus-25%				Fundus-50%				Fundus-100%			
Method	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Hard attention	0.895	0.984	0.849	0.912	0.918	0.974	0.896	0.934	0.928	0.955	0.933	0.944
Soft attention	0.895	0.982	0.852	0.912	0.915	0.937	0.93	0.933	0.948	0.975	0.943	0.959
L2X	0.863	0.839	0.974	0.901	0.931	0.964	0.927	0.945	0.94	0.94	0.981	0.961
Grad-CAM	0.891	0.959	0.867	0.911	0.921	0.964	0.911	0.937	0.921	0.918	0.963	0.940
Student (only)	0.863	0.903	0.813	0.856	0.90	0.916	0.879	0.897	0.927	0.960	0.890	0.923
MED-EX	0.908	0.958	0.894	0.925	0.938	0.978	0.924	0.950	0.951	0.994	0.93	0.961
MED-TEX	**0.915**	0.961	0.904	**0.933**	**0.955**	0.984	0.946	**0.964**	**0.975**	0.989	0.972	**0.98**

TABLE 4.5: Average IoU evaluation when top$K$ is equal to the number of ground-truth lesion pixels for every individual image.

	Fundus-100%	Fundus-50%	Fundus-25%
MED-EX	0.091	0.06	0.058
MED-TEX	0.405	0.313	0.304



(A) Fundus-25% (B) Fundus-50% (C) Fundus-100%

FIGURE 4.3: Average IoU evaluation among various methods at different topKs.

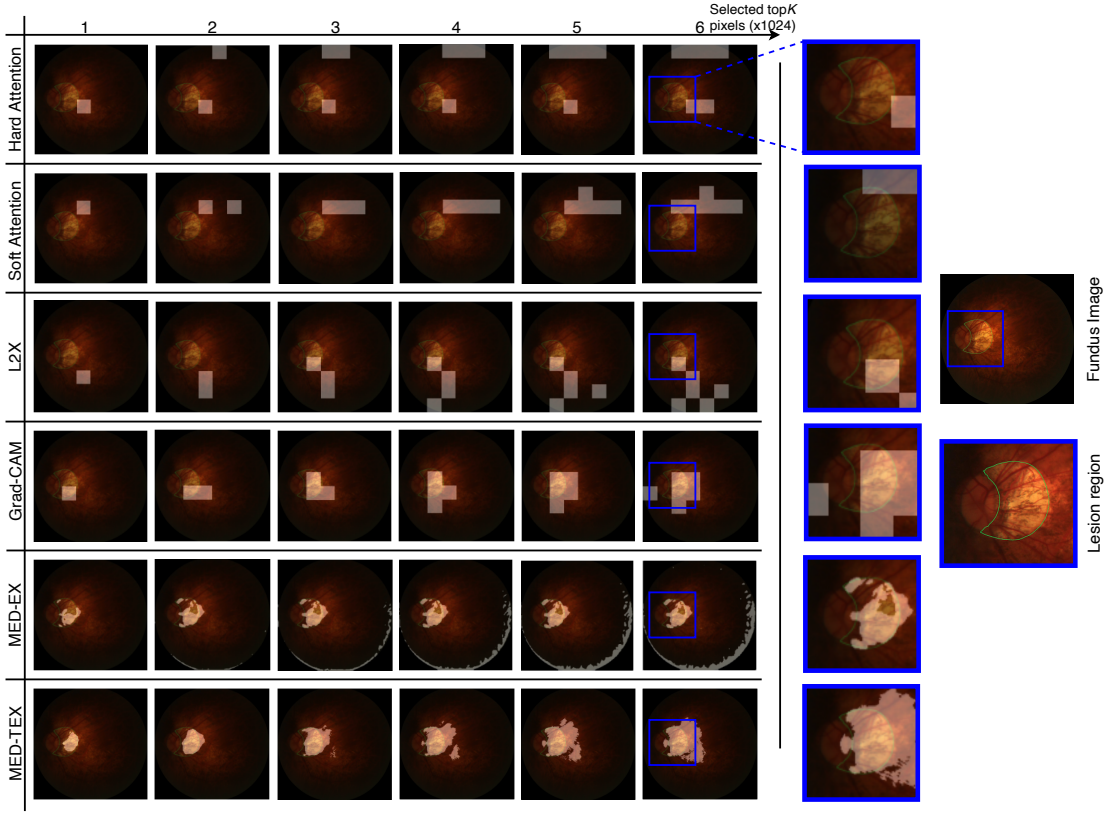


FIGURE 4.4: Visualisation results of top$K$ highlighted image regions of different methods trained on Fundus-50%, compared with the ground-truth lesion segmentation (specified by the green contour). While hard attention [96], soft attention [84, 88], Grad-CAM [79], and L2X [100] output patch-based region selection maps, our MED-EX and MED-TEX give pixel-level selection scores which is more accurate and fine-grained than others.

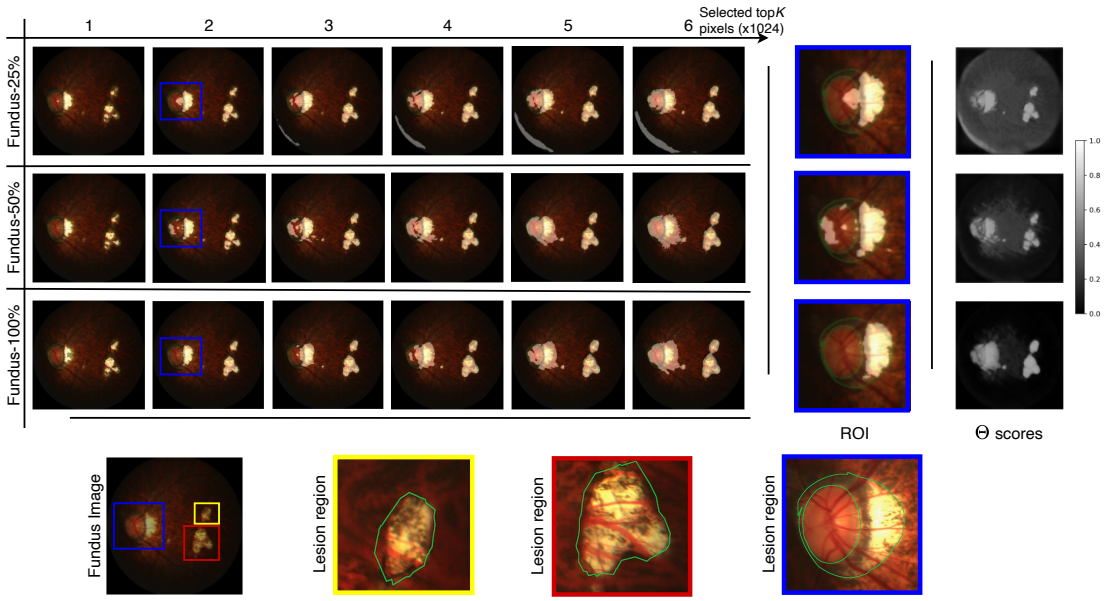
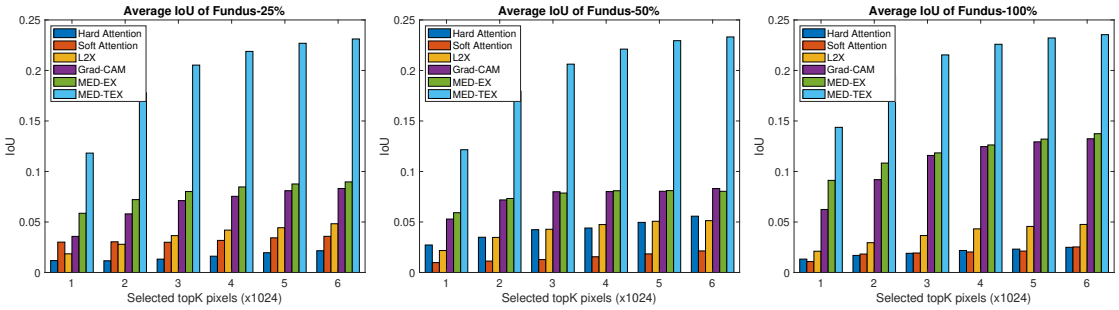


FIGURE 4.5: Visualisation results of top$K$ highlighted image regions of different methods with different numbers of training data, compared with the ground-truth lesion segmentations (specified by the green contours). Feature selection scores Θ are plotted in heatmaps on the right.

TABLE 4.6: Average IoU evaluation when top$K$ is equal to the number of ground-truth lesion pixels for every individual image.

	Fundus-100%	Fundus-50%	Fundus-25%
MED-EX	0.091	0.06	0.058
MED-TEX	0.405	0.313	0.304



(A) Fundus-25% (B) Fundus-50% (C) Fundus-100%

FIGURE 4.6: Average IoU evaluation among various methods at different topKs.

MED-TEX, it achieves reasonably good results on approximating the teacher with only 25% data. In addition, when the full dataset is used, MED-TEX reaches 0.98 F1 score, meaning that the student can perform nearly as well as the teacher on the image classification task.

If we compare MED-TEX with its variant, Student (only), it can be observed that Student (only) trained directly from raw input images cannot perform well. This suggests

that the explainer with feature selection at pixel-level plays a central role to guide the student to achieve better performance.

Figure 4.6 shows the IoU results in bar charts. We can see that our MED-TEX achieves significantly higher IoU than others on the fundus dataset. Specifically, MED-TEX performs approximately 2× better than MED-EX and Grad-CAM and more than 4× better than others. We also observe that even our approach is trained with small amount data of fundus dataset, the IoU is still remaining relatively high (e.g., when top$K$ is equal to 1024, MED-TEX achieves 0.118 IoU in fundus-25%). In addition, we further evaluate the performance of MED-EX and MED-TEX when top$K$ is equal to the number of ground-truth lesion pixels for each individual image. Table 4.6 reports the average IoU results, where MEX-TEX achieves 0.4, 0.31 and 0.3 for Fundus-100%, Fundus-50% and Fundus-25%, respectively.

Figure 4.4 shows the visualisation results of top$K$ highlighted image regions of different methods. Hard attention [96], soft attention [84, 88], Grad-CAM [1, 79], and L2X [100] can only give patch-based region selection maps, while our MED-EX and MED-TEX produce pixel-level selection scores. It can be seen from Figure 4.4 that our method on an abnormal fundus image highlights the regions that well match the ground-truth lesion segmentations. In general, MED-EX and MED-TEX produce more accurate and fine-grained lesion segmentations on the pixel level than those segmentations on the feature level in the other methods. Moreover, MED-TEX clearly outperforms MED-EX due to the use of the intermediate knowledge distillation losses.

We also qualitatively evaluate our method with different proportions of the training data in Figure 4.5, where the example fundus image has multiple lesion regions (red, blue and yellow regions). We can see that MED-TEX is able to precisely point out these regions, even trained with less data. With more training data used, our method gradually improves the quality and accuracy of identifying the lesion regions. Finally, we also evaluate MED-TEX using Tiny ImageNet dataset with the same settings, whose results are shown in the appendix. It can also observed the same trend that our method consistently outperforms the other compared approaches.

TABLE 4.7: Post-hoc evaluation on the Tiny ImageNet dataset

	25%		50%		100%	
Method	Acc	F1	Acc	F1	Acc	F1
Hard attention	0.92	0.923	0.907	0.901	0.966	0.968
Soft attention	0.913	0.91	0.935	0.94	0.953	0.955
L2X	0.833	0.828	0.86	0.859	0.87	0.87
Grad-CAM	0.90	0.906	0.92	0.918	0.953	0.953
Student (only)	0.686	0.711	0.80	0.779	0.826	0.839
MED-EX	0.913	0.912	0.933	0.932	0.953	0.955
MED-TEX	**0.927**	**0.928**	**0.94**	**0.943**	**0.967**	**0.969**

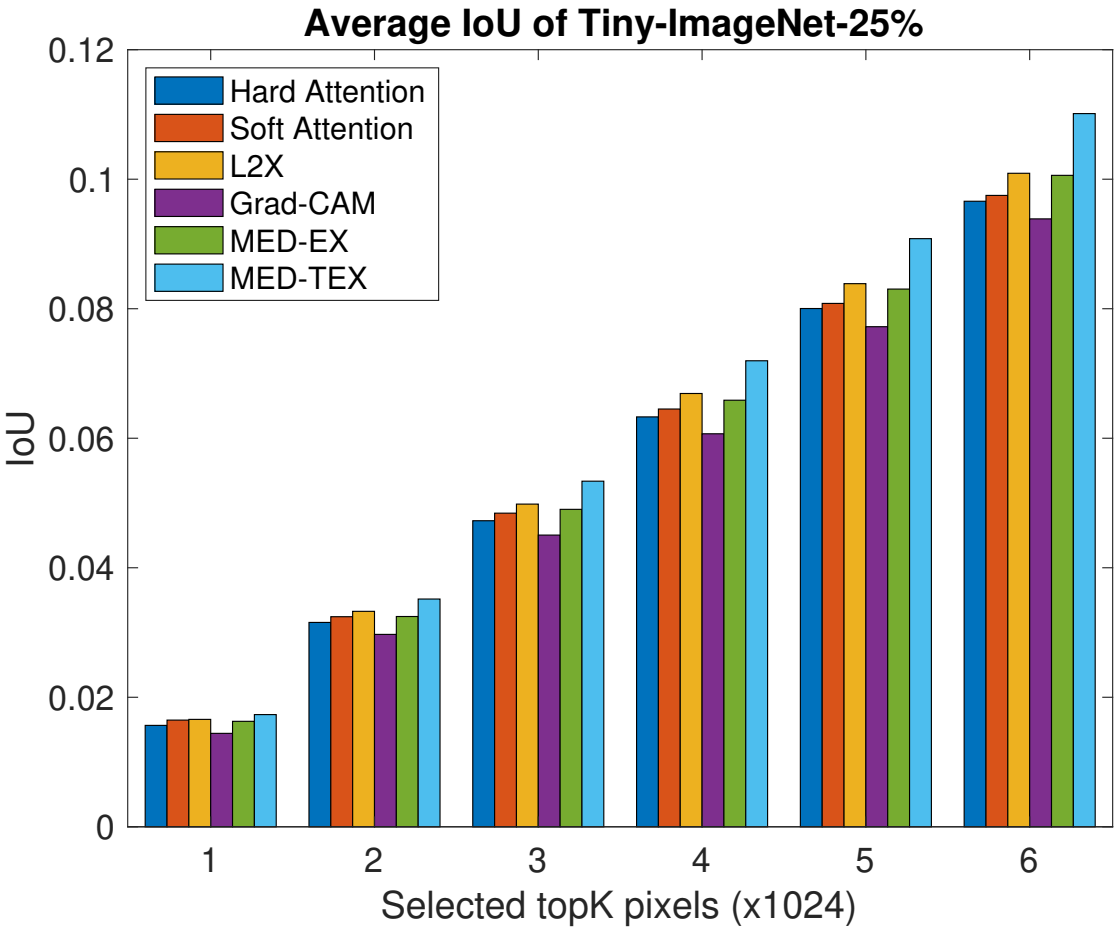


FIGURE 4.7: Average IoU evaluation among various methods.

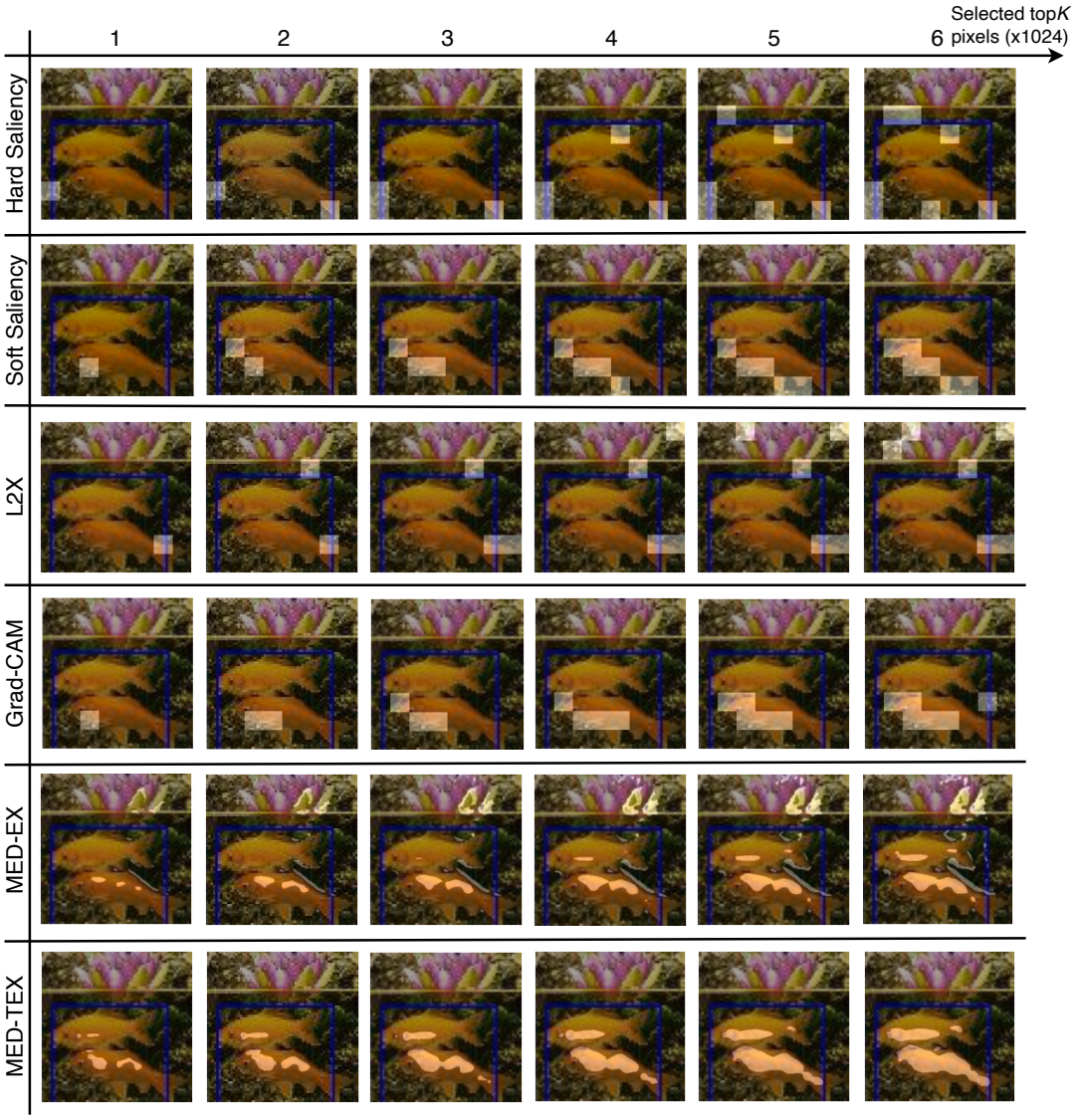


FIGURE 4.8: Visualisation results on the example golden fish image.

4.5 Additional Experiments on Tiny ImageNet Dataset

In this section, we demonstrate that our framework works not only on medical imaging datasets but also on natural imaging datasets. In addition to the Fundus dataset, we also conduct our experiments on the Tiny ImageNet dataset[5]. We select 500 golden fish images (425 for training and 75 for testing) and 500 jellyfish images (also 425 for training and 75 for testing). All the images are with labelled bounding boxes identifying the regions of the fishes. Examples of the images are shown in Figure 4.8 and 4.9. The teacher model is then trained by 850 images and tested with 250 images that reaches 95.4% accuracy (0.947 precision, 0.956 recall and 0.954 F1 score). In the post-hoc

[5]https://www.kaggle.com/c/tiny-imagenet

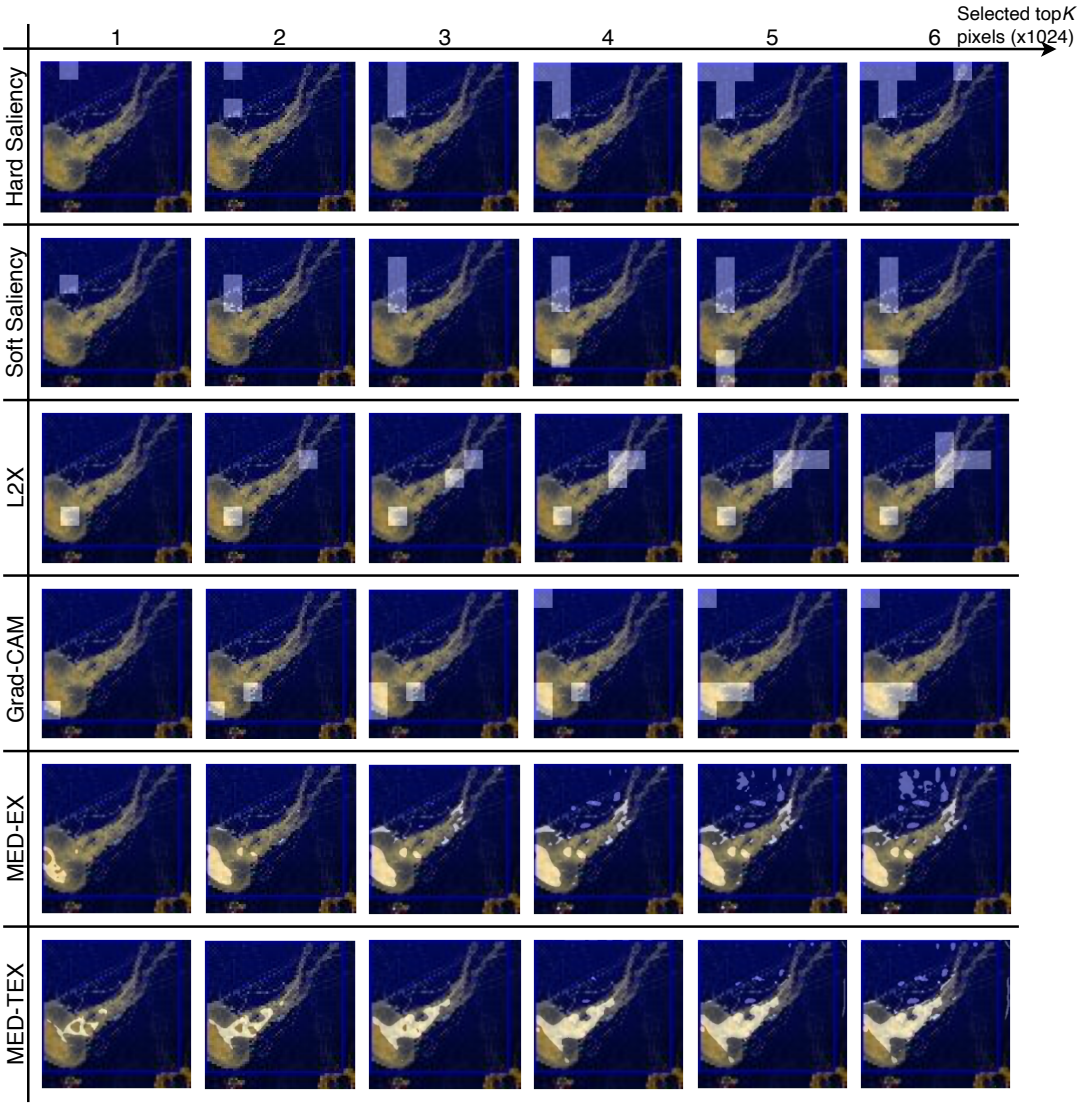


FIGURE 4.9: Visualisation results on the example jelly fish image.

evaluation, MED-TEX gives higher F1 score and accuracy than the other methods. There is also a significant improvement of IoU with less data as shown in Figure 4.7. We qualitatively evaluate MED-TEX by visualizing the demo images (golden fish and jelly fish) and highlighted the regions with different top$K$s trained by Tiny ImageNet 100%, as shown in Figure 4.8 and 4.9, respectively. It can be seen that our approaches are able to locate the fishes more precisely than other methods.

4.6 Conclusion

In this chapter, we have introduced our novel framework MED-TEX, which is a joint knowledge distillation and model interpretation framework that learns the significantly

smaller student (compared to the teacher) and explainer models by leveraging the knowledge only from the pre-trained teacher model. With the proposed framework, the student is trained with less data to learn from the knowledge of the pre-trained teacher with the assistance of the explainer designed to highlight the important image areas to the teacher's predictions. The output of the explainer can also be used as low-level strong annotations trained by high-level weak ones (teacher's knowledge). In addition, to train the framework, we have proposed to maximise the mutual information between the intermediate and output layers of the student and teacher, which forms a novel training objective of our framework. In our experiments, we show that MED-TEX outperforms several widely-used knowledge distillation and model interpretation techniques, including: soft attention [84, 88], hard attention[96], L2X [100], Grad-CAM [1, 79] on the Fundus dataset in terms of both quantitative and qualitative evaluations.

Chapter 5

Particle-based Adversarial Local Distribution Regularisation

In Chapters 3 and 4, we have demonstrated the first thesis objective to gain interpretations and insights into both adversarial attack mechanisms and knowledge distillation techniques. In this chapter, we explore the principle of adversarial regularisation. Adversarial training defence (ATD) and virtual adversarial training (VAT) are the two most effective regularisation methods to improve model robustness against attacks and model generalisation. While ATD is usually applied in robust machine learning, VAT is used in semi-supervised learning. The adversarial local distribution (ALD) is defined by a set of all adversarial examples within a ball constraint given a natural input. The ALD is efficiently approximated by Stein Variational Gradient Descent. We illustrate this novel adversarial local distribution regularisation is a general form of previous methods (e.g., PGD, TRADES, and VAT). We conduct comprehensive experiments on MNIST and CIFAR10 to illustrate that our method outperforms well-known methods such as PGD, TRADES and ADT in robust machine learning, VAT in semi-supervised learning.

5.1 Introduction

Generalisation is defined by model's ability to react to unseen input data, which is one of the most challenging problems in machine learning. For examples, the model should be robust to adversarial example inputs from attacks. The model from semi-supervised

learning applications should not be overfitted to finite training data samples in order to generalise well on unseen data. State-of-the-art deep neural networks are reported to be susceptible to attacks [21, 22]. These attacks add crafted perturbations to clean inputs to create adversarial examples (e.g., Fast Gradient Sign Method (FGSM) [22], Projected Gradient Descent (PGD) [23] and Auto-Attack [32]. The most common way to find the perturbations is using adversarial direction which leverages gradients to maximise the loss of a model on a particular input while keeping the size of the perturbation smaller than a specified amount referred to a radius constraint epsilon. Due to the threats, many methods have been proposed for robust regularisation using adversarial examples to defence such as [23–26]. In addition, overfitting problem occurs when the model performs well on training dataset with low error but the true expected error (test error) is large. Regularisation is the most common way to reduce the gap between the training error and the test error in real world applications. In terms of using adversarial examples as regularisation to improve model generalisation, VAT has been introduced by [27] to tackle the problem which promotes the smoothness of model output distribution named local distributional smoothness. This regularisation shows its effectiveness to reduce overfitting and improve generalisation in semi-supervised learning [28]. Then it is adopted to regularise source and target models in domain adaption in order to boost the generalisation on the target domain [177].

Among defence techniques, adversarial training defence (ATD) is one of the most effective [140, 178]. However, ATD heavily relies on attack algorithms to find adversarial samples during the training, which shows poor generalisation for other unseen attacks [179]. Moreover, a single attack algorithm can only create one adversarial sample in a run, which could be insufficient to completely explore the space of possible perturbations. Even PGD attacks with random initialisation can also lie together and lose diversity [180]. Recently, [181] proposed a technique to form the perturbation distributions named (ADT) to improve model robustness. However, ADT makes a strong assumption that the distribution of perturbations follows certain parameterisation forms (e.g., Gaussian distributions).

Training a deep learning model that can generalise well on unseen data is a challenging problem. When neural networks have a lot of parameters to be tuned by finite training samples, overfitting is a common problem. Several regularisation techniques have been proposed to overcome the overfitting such as l_2 weight decay and Dropout [182]. In

this chapter, we focus on using adversarial examples to reduce overfitting and improve model generalisation. Virtual adversarial training (VAT) introduced local distribution smoothness based regularisation [28]. However, the drawback of this technique is that VAT cannot explore well the local distribution and generate diverse adversarial examples (see more details about VAT in following sections).

In this chapter, we introduce a novel regularisation method using adversarial examples to overcome the drawbacks of previous approaches. Our contributions are summarised as follows:

- We propose an adversarial local distribution based regularisation to encourage model generalisation. The adversarial local distribution is defined by a set of all adversarial examples within a ball constraint that can maximise loss function given a natural input. We also show that this regularisation is a general form of well-known previous approaches such PGD [23], TRADES [26], and VAT [28]).
- We sufficiently approximate the adversarial local distribution without any assumptions using a multiple particle-based search named Stein Variational Gradient Decent (SVGD) [183]. The SVGD can create more diverse adversarial examples which significantly help to improve model performance.
- We show that our method can be adapted well to various applications such as semi-supervised learning, and robust machine learning. We conduct comprehensive experiments on MNIST, and CIFAR10 datasets to demonstrate that our method outperforms previous well-known approaches in the above applications, such as PGD [23], TRADES [26], ADT [181] in robust machine learning, VAT [28] in semi-supervised learning.

5.2 Related Work

In Chapter 2, we have briefly introduced the adversarial regularisation for deep learning tasks. In this section, we discuss the details of existing adversarial regularisation drawbacks. Adversarial training defence (ATD) is one of the most effective techniques to protect deep neural networks from attacks [140, 178]. ATD can be formulated as a

minmax optimisation[23]. While the inner maximisation of ATD tries to find an adversarial example within a ball constraint that maximises the classification loss given a natural input, the outer minimisation aims to train a robust classifier using the generated adversarial examples. In order to solve the inner maximisation problem of ATD, previous works usually used a specific attack algorithm to find adversarial examples such as FGSM [22], PGD [23] and TRADES [26]. The quality of ATD significantly depends on the strength of injected perturbations of adversarial examples. For example, ATD uses non-iterative method FGSM [22], which cannot robust to iterative PGD [23] attack. Previous work proposed by [140] suggests that the adversarial training defence with PGD can perform well against attacks. Therefore, many works attempt to improve ATD with PGD such as [184–188]. Recently, contrastive learning [189] and ensemble method [190] have been used to archive state-of-the-art performance. However, these methods only generate only one adversarial example, which could be insufficient to explore entire space of possible perturbations. Moreover, the work proposed by [180] shows even the attacks with random initialisation can also lie together and lose diversity that reduce the quality of ATD. Recently, [181] proposed a technique to form the perturbation distribution named adversarial distributional training (ADT), where the inner maximisation aims to find adversarial distribution for each natural input. However, ADT makes a strong assumption that the perturbation distribution follows Gaussian distribution. This assumption could be insufficient in practice. Therefore, our method addresses a strong diversity of adversarial examples and sufficiently forms the adversarial distribution without any assumption.

Virtual adversarial training (VAT) proposed by [27] is a well-known regularisation for semi-supervised learning [28] which can be defined by a minmax optimisation problem similar to the adversarial training defence. The inner maximisation of VAT aims to find an adversarial example that maximises KL divergence loss between model outputs of a natural input and the adversarial example input. The outer minimisation aims to smooth the local distribution output of a model given a natural input to reduce overfitting and improve generalisation. This technique is called local distribution smoothness based regularisation. Similar to ATD, VAT cannot sufficiently explore the local distribution. It is worth to note that there is a strong connection between ATD and VAT (e.g., solving minmax optimisation problem and KL divergence loss). For example, TRADES [26] used in ATD solves the mimax problem and leverages KL divergence loss to solve the inner

maximisation. In this chapter, we form a generalisation regularisation for both ATD and VAT.

Semi-supervised learning is a method to machine learning that combines a small amount of labelled data with a large unlabelled data during training. There are several approaches proposed to solve this problem such as entropy minimisation [191], pseudo-labeling [65], MixMatch [74] and VAT [28]. In this chapter, we focus on approaches using adversarial examples to improve performance by smoothing the model output distribution such as VAT [27, 28].

5.3 Method

In this section, we first recall the minmax optimisation problem of adversarial training defence (ATD) [23] and virtual adversarial training (VAT) [28]. We then formulate a novel adversarial local distribution which is a general distribution for ATD and VAT. The adversarial local distribution (ALD) is efficiently approximated without any consumption by using multiple particle-based Stein Variational Gradient Descent (SVGD) [183]. We also show that our method can be adapted in defending against adversarial attacks, semi-supervised learning and domain adaption.

5.3.1 Minmax Optimisation of ATD and VAT

ATD and VAT have a common minmax optimisation problem but aim to achieve different goals. For example, ATD is used to improve the adversarial robustness of models, while VAT is applied to improve the performance of semi-supervised learning. Let $\boldsymbol{x} \in \mathbb{R}^n$ be our n-dimensional natural input data in a space $\boldsymbol{X}$. Given an input $(\boldsymbol{x}, y) \sim P_{\mathbb{D}}$ (i.e., the data-label distribution), we denote $B_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}^{adv} \in \boldsymbol{X} : ||\boldsymbol{x}^{adv} - \boldsymbol{x}||_p \leq \epsilon\}$ is the ball constraint around the natural sample $\boldsymbol{x}$ with a radius ϵ with respect to a norm $||\cdot||_p$. Given a classifier f_θ parameterised by θ, we define the minmax optimisation problem [23] as

$$\min_{\theta} \mathbb{E}_{(\boldsymbol{x},y)\sim P_{\mathbb{D}}} \Big[\max_{\boldsymbol{x}^{adv} \in B_\epsilon(\boldsymbol{x})} \ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta) \Big], \tag{5.1}$$

where $\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta)$ depends on a particular method. For example, FGSM [22], PGD [23] use the cross-entropy loss (ℓ_{CE})

$$\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta) = \ell_{CE}(f_\theta(\boldsymbol{x}^{adv}), y), \tag{5.2}$$

where y is the ground-truth label of $\boldsymbol{x}$ and $f_\theta(\boldsymbol{x}^{adv})$ is the prediction probabilities. Another example is TRADES [26] and VAT [28], which use the Kullback-Leibler divergence loss (D_{KL}) in Equation 5.3

$$\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta) = D_{\mathrm{KL}}(f_\theta(\boldsymbol{x}^{adv}), f_\theta(\boldsymbol{x})). \tag{5.3}$$

5.3.2 Adversarial Local Distribution Regularisation

Recall that the maximisation problem in Equation 5.1 is usually solved by the relevant methods such as FGSM, PGD, TRADES, and VAT. However, these methods only find one adversarial example $\boldsymbol{x}^{adv}$ given a natural input $\boldsymbol{x}$. In this section, we introduce our proposed adversarial local distribution (ALD) regularisation. ALD regularisation considers an adversarial local distribution $P_\theta(\boldsymbol{x}^{adv}|\boldsymbol{x}, y)$ within a ball constraint B_ϵ which is relevant the the loss function $\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta)$ as shown in (5.4).

$$P_\theta(\boldsymbol{x}^{adv}|\boldsymbol{x}, y) := \frac{e^{\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta)}}{\int_{B_\epsilon(\boldsymbol{x})} e^{\ell(\boldsymbol{x}^{adv\prime}, \boldsymbol{x}, y; \theta)} d\boldsymbol{x}^{adv\prime}} = \frac{e^{\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta)}}{Z(\boldsymbol{x}, y; \theta)}, \tag{5.4}$$

where $P_\theta(\cdot|\boldsymbol{x}, y)$ is the conditional local distribution over $B_\epsilon(\boldsymbol{x})$ and $Z(\boldsymbol{x}, y; \theta)$ is a normalisation function. Instead of solving directly the inner maximisation as in the aforementioned approaches, we sample a set of adversarial examples or particles from this local distribution with the aim to reach its modes and avoid the particle collapse to increase the particle diversity. We note that depending on the loss function ℓ, y could be the one-hot ground-truth label of x or the prediction probabilities $f_\theta(x)$.

Given Equation 5.4, we propose the adversarial local distributional regularisation term at the position x

$$\begin{aligned} \mathcal{R}(\theta, \boldsymbol{x}, y) :=\ & \mathbb{E}_{\boldsymbol{x}^{adv} \sim P_\theta(\cdot|\boldsymbol{x}, y)}[\log P_\theta(\boldsymbol{x}^{adv}|\boldsymbol{x}, y)] \\ =\ & -H(P_\theta(\cdot|\boldsymbol{x}, y)), \end{aligned} \tag{5.5}$$

where H indicates the entropy of a given distribution.

For $\boldsymbol{x}$ and y, when minimising $\mathcal{R}(\theta, \boldsymbol{x}, y)$ or equivalently $-H(P_\theta(\cdot|\boldsymbol{x}, y))$ w.r.t. θ, we point-wisely maximise $H(P_\theta(\cdot|\boldsymbol{x}, y))$, which is equivalent to encourage $P_\theta(\cdot|\boldsymbol{x}, y)$ to be more uniform distribution. This further enforces $\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta) = \ell(\boldsymbol{x}^{adv'}, \boldsymbol{x}, y; \theta) = c(x, y; \theta)$, where $\boldsymbol{x}^{adv}, \boldsymbol{x}^{adv'} \sim P_\theta(\cdot|\boldsymbol{x}, y)$. In other words, it implies that $\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}, y; \theta)$ is close to a constant $c(\boldsymbol{x}, y; \theta)$ and smooth over $\boldsymbol{x}^{adv} \in B_\epsilon(\boldsymbol{x})$. Therefore, minimising the adversarial local distribution regularisation loss leads to an enhancement in the model output smoothness (i.e., the classifier does not change outputs with any input $\boldsymbol{x}^{adv} \in B_\epsilon(\boldsymbol{x})$) to encourage model robustness. As demonstrated later, not only strengthens the model robustness in adversarial defence, this also encourages the model generalisation in semi-supervised learning settings.

At this outset, it is worth noting that when sampling only one adversarial example from $P_\theta(\cdot|\boldsymbol{x}, y)$, the Equation 5.5 reduces to FGSM, PGD, TRADES, and VAT respectively (see our asymptotic analysis when we assume using the RBF kernel and consider the behaviours when letting the kernel width $\sigma \to 0$ or ∞ below).

5.3.3 Multiple Particle-based Search to Approximate the Adversarial Local Distribution

In Equation 5.4, $Z(\boldsymbol{x}, y; \theta)$ is intractable to find, we thus use a particle-based method to sample $\boldsymbol{x}^{adv}{}_1, \boldsymbol{x}^{adv}{}_2, \ldots, \boldsymbol{x}^{adv}{}_N \sim P_\theta(\cdot|\boldsymbol{x}, y))$, where N is the number of samples (or *adversarial particles*) to solve the optimisation problem of finding $P_\theta(\cdot|\boldsymbol{x}, y)$. Here we show that our method can sufficiently explore the adversarial local distribution more efficiently compared to previous methods (e.g., FGSM, PGD, TRADES, ADT, and VAT).

Stein Variational Gradient Decent (SVGD) [183] is a particle-based inference method using a functional gradient descent to approximate a ground-truth distribution without explicit parametric assumptions. To this end, SVGD is leveraged to be our solver to approximate the adversarial local distribution $P_\theta(\cdot|\boldsymbol{x}, y)$. The core idea is to find a set of adversarial particles to approximate the local distribution using Alg. 5.1. More specifically, a set of adversarial particles $\{\boldsymbol{x}^{adv}{}_1, \boldsymbol{x}^{adv}{}_2, \ldots, \boldsymbol{x}^{adv}{}_N\}$ is initialised by adding uniform noises, then projected onto the ball B_ϵ. Furthermore, these adversarial particles are then iteratively updated as well as projecting onto the ball B_ϵ (line 4 in Alg. 5.1) until

Algorithm 5.1: Approximating the conditional adversarial local distribution given $\boldsymbol{x}$ by using Stein Variational Gradient Decent

Input: A natural sample $(\boldsymbol{x}, y) \sim P_{\mathbb{D}}$; N number of adversarial particles; ϵ for the constraint B_ϵ; r normalisation function; η initial noise factor; τ step size updating; L number of iterations; $\mathcal{K}$ kernel function

Output: Set of adversarial particles $\{\boldsymbol{x}^{adv}{}_1, \boldsymbol{x}^{adv}{}_2, \ldots, \boldsymbol{x}^{adv}{}_N\} \sim P_\theta(\cdot|\boldsymbol{x}, y)$

1 Initialise a set of N particles and project to the B_ϵ constraint $\{\boldsymbol{x}^{adv}{}_i \in \mathbb{R}^n, i \in \{1, 2, \ldots, n\} | \boldsymbol{x}^{adv}{}_i = \prod_{B_\epsilon}(\boldsymbol{x} + \eta * Uniform\_noise)\}$;

2 **for** $l = 1$ *to* L **do**

3 **for** *each particle* $\boldsymbol{x}^{adv}{}_i^{(l)}$ **do**

4 $\boldsymbol{x}^{adv}{}_i^{(l+1)} = \prod_{B_\epsilon}\left(\boldsymbol{x}^{adv}{}_i^{(l)} + \tau * r\big(\phi(\boldsymbol{x}^{adv}{}_i^{(l)})\big)\right)$;

5 where $\phi(\boldsymbol{x}^{adv}) = \frac{1}{N}\sum_{j=1}^{N}[\mathcal{K}(\boldsymbol{x}^{adv}{}_j^{(l)}, \boldsymbol{x}^{adv})\nabla_{\boldsymbol{x}^{adv}{}_j^{(l)}} \log P(\boldsymbol{x}^{adv}{}_j^{(l)}|\boldsymbol{x}, y) + \nabla_{\boldsymbol{x}_j^{(l)}}\mathcal{K}(\boldsymbol{x}^{adv}{}_j^{(l)}, \boldsymbol{x}^{adv})]$;

6 return $\{\boldsymbol{x}^{adv}{}_1^{L}, \boldsymbol{x}^{adv}{}_2^{L}, \ldots, \boldsymbol{x}^{adv}{}_N^{L}\}$;

reaching the termination condition. Note that $\mathcal{K}$ is a positive definite kernel for which in our experiments, we use radial basic function (RBF) kernel defined in Equation 5.6, where the kernel width σ is empirically set by proportional to the number of particles N (i.e., $\sigma = 10^{1-N}$). Additionally, two terms of ϕ (line 5 in Alg. 5.1) have different roles: (i) the first one enforces the particles move towards to the high density areas of $P_\theta(\cdot|\boldsymbol{x}, y)$ and (ii) the second one prevents all the particles to collapse into local modes of $P_\theta(\cdot|\boldsymbol{x}, y)$.

$$\mathcal{K}(\boldsymbol{x}^{adv}, \boldsymbol{x}) = \exp\left\{\frac{-||\boldsymbol{x}^{adv} - \boldsymbol{x}||^2}{2\sigma^2}\right\}. \tag{5.6}$$

5.3.4 Asymptotic Analysis of Adversarial Local Distribution Approximation

Considering the RBF kernel, the update function ϕ can be rewritten as

$$\begin{aligned}\phi(\boldsymbol{x}^{adv}) = \frac{1}{N}\sum_{j=1}^{N}\Big[&\mathcal{K}(\boldsymbol{x}^{adv}{}_j^{(l)}, \boldsymbol{x}^{adv})\nabla_{\boldsymbol{x}^{adv}{}_j^{(l)}}\ell(\boldsymbol{x}^{adv}{}_j^{(l)}, \boldsymbol{x}, y; \theta) \\ &-\mathcal{K}(\boldsymbol{x}^{adv}{}_j^{(l)}, \boldsymbol{x}^{adv})\frac{(\boldsymbol{x}^{adv}{}_j^{(l)} - \boldsymbol{x}^{adv})}{\sigma^2}\Big].\end{aligned} \tag{5.7}$$

When $\sigma \to \infty$, it is obvious that

$$\phi(\boldsymbol{x}^{adv}) \to \frac{1}{N}\sum_{j=1}^{N}\nabla_{\boldsymbol{x}^{adv}{}_j^{(l)}}\ell(\boldsymbol{x}^{adv}{}_j^{(l)}, \boldsymbol{x}, y; \theta). \tag{5.8}$$

Therefore, our approach reduces exactly to FGSM, PGD, TRADES, and VAT with n independent particles, where in the update quantity is the average of the gradients at each particle as shown in Equation 5.10. Evidently, in the update rule in Equation 5.10, there does not exist any term that promotes the particle diversity. In addition, when using a single particle (i.e., $n = 1$), our approach under its asymptotic case reduces exactly to the aforementioned approaches.

Particularly, in our update formula in Equation 5.13, the first term encourages the particles to seek the optimal values of the loss surface as in FGSM, PGD, TRADES, and VAT, while the second term plays a role of a repulsive term to push the particles away for enhancing the particle diversity. The reason is that when ${\boldsymbol{x}^{adv}}_j^{(l)}$ moves closer to $\boldsymbol{x}^{adv}$, the weight $\mathcal{K}({\boldsymbol{x}^{adv}}_j^{(l)}, \boldsymbol{x}^{adv})$ becomes larger to push them further away from each other.

We present the asymptotic analysis when $\sigma \to 0$. Considering the RBF kernel, the update function ϕ can be rewritten as

$$\begin{aligned}\phi(\boldsymbol{x}^{adv}) = \frac{1}{N}\sum_{j=1}^{N}\Big[&\mathcal{K}({\boldsymbol{x}^{adv}}_j^{(l)}, \boldsymbol{x}^{adv})\nabla_{{\boldsymbol{x}^{adv}}_j^{(l)}}\ell({\boldsymbol{x}^{adv}}_j^{(l)}, \boldsymbol{x}, y; \theta) \\ &- \mathcal{K}({\boldsymbol{x}^{adv}}_j^{(l)}, \boldsymbol{x}^{adv})\frac{({\boldsymbol{x}^{adv}}_j^{(l)} - \boldsymbol{x}^{adv})}{\sigma^2}\Big].\end{aligned} \tag{5.9}$$

When $\sigma \to 0$, it is obvious that

$$\phi(\boldsymbol{x}^{adv}) \to \frac{1}{N}\sum_{j=1}^{N} 1_{\boldsymbol{x}^{adv}={\boldsymbol{x}^{adv}}_j^{(l)}}\nabla_{{\boldsymbol{x}^{adv}}_j^{(l)}}\ell({\boldsymbol{x}^{adv}}_j^{(l)}, \boldsymbol{x}, y; \theta), \tag{5.10}$$

where 1_A is the indicator function which returns 1 if A is true and 0 if otherwise. Here we note that we have used the following equations in the above derivation.

$$\lim_{\sigma\to 0}\mathcal{K}({\boldsymbol{x}^{adv}}_j^{(l)}, \boldsymbol{x}^{adv})\frac{({\boldsymbol{x}^{adv}}_j^{(l)} - \boldsymbol{x}^{adv})}{\sigma^2} = 0. \tag{5.11}$$

$$\lim_{\sigma\to 0}\mathcal{K}({\boldsymbol{x}^{adv}}_j^{(l)}, \boldsymbol{x}^{adv}) = 0 \tag{5.12}$$

if $\boldsymbol{x}^{adv} \neq {\boldsymbol{x}^{adv}}_j^{(l)}$.

$$\lim_{\sigma\to 0}\mathcal{K}({\boldsymbol{x}^{adv}}_j^{(l)}, \boldsymbol{x}^{adv}) = 1 \tag{5.13}$$

if $\boldsymbol{x}^{adv} = \boldsymbol{x}^{adv}{}_j^{(l)}$.

Therefore, the update amount $\phi(\boldsymbol{x}^{adv})$ in Equation 5.10 reduces to only one gradient. It is evident that when $N = 1$, our approach reduces exactly to PGD, TRADES, or VAT.

5.4 Robust Learning and Semi-supervised Learning

In this section, we adapt our adversarial local distribution regularisation Equation 5.5 to various applications such as robust machine learning, semi-supervised learning. The previous methods (e.g., PGD, TRADES or VAT) can be addressed as sampling only one adversarial particle in $P_\theta(\cdot|\boldsymbol{x}, y)$. More specifically, our approach with a single particle under its asymptotic setting can reduce to these methods as analysed in the previous section.

We now illustrate how to apply our adversarial local distribution regularisation to specific problems such as semi-supervised learning, robust machine learning. Generally, our adversarial local distribution (ADL) regularisation can be applied to a data example x with or without label to make its local vicinity smoother. More specifically, if $\boldsymbol{x}$ has a label y, we can flexibly apply $\mathcal{R}(\theta, \boldsymbol{x}, y)$ as in Equation 5.2 or $\mathcal{R}(\theta, \boldsymbol{x}, f_\theta(\boldsymbol{x}))$ as in Equation 5.3. In contrast, if $\boldsymbol{x}$ is an unlabelled data example, we can use $\mathcal{R}(\theta, \boldsymbol{x}, f_\theta(\boldsymbol{x}))$ as in Equation 5.3. We hence can apply our ALD regularisation to both labelled and unlabelled portions in the semi-supervised setting, and labelled dataset in robust machine learning. Moreover, our ADL regularisation for each data example x can be estimated conveniently by sampling a set of adversarial particles $\{\boldsymbol{x}^{adv}{}_1, \boldsymbol{x}^{adv}{}_2, \ldots, \boldsymbol{x}^{adv}{}_n\} \sim P_\theta(\cdot|\boldsymbol{x}, y)$ using Alg. 5.1.

For semi-supervised learning, $(\boldsymbol{x}_l, y) \sim P_{\mathbb{D}_l}$ and $\boldsymbol{x}_u \sim P_{\mathbb{D}_u}$, where $\mathbb{D}_l$ and $\mathbb{D}_u$ is the labelled and unlabelled dataset respectively. Based on the VAT loss, the loss function is adapted to semi-supervised learning using cross-entropy loss (ℓ_{CE}) and adversarial local distribution regularisations weighted by λ_1 and λ_2, as shown in (6.8).

$$\begin{aligned}\min_\theta \Big\{ &\mathbb{E}_{(\boldsymbol{x}_l, y)\sim P_{\mathbb{D}_l}} \Big[\ell_{CE}(f_\theta(\boldsymbol{x}_l), y) + \lambda_1 \mathcal{R}(\theta, \boldsymbol{x}_l, f_\theta(\boldsymbol{x}_l))\Big] \\ &+ \lambda_2 \mathbb{E}_{\boldsymbol{x}_u \sim P_{\mathbb{D}_u}} \Big[\mathcal{R}(\theta, \boldsymbol{x}_u, f_\theta(\boldsymbol{x}_u))\Big] \Big\},\end{aligned} \tag{5.14}$$

where $\mathcal{R}(\theta, \boldsymbol{x}, f_\theta(\boldsymbol{x}))$ is relevant to the loss in Equation 5.3.

For robust machine learning, $(\boldsymbol{x}, y) \sim P_{\mathbb{D}}$, where $\mathbb{D}$ is the dataset. The loss function is adapted to this problem using cross-entropy loss (ℓ_{CE}) and adversarial local distribution regularisation weighted by λ. Based on PGD or TRADES, we can adapt the loss function

$$\min_{\theta} \left\{ \mathbb{E}_{(\boldsymbol{x},y)\sim P_{\mathbb{D}}} \Big[\ell_{CE}(f_\theta(\boldsymbol{x}), y) \Big] + \lambda R(\theta, \boldsymbol{x}, \tilde{y}) \Big] \right\}, \tag{5.15}$$

where in $\mathcal{R}(\theta, \boldsymbol{x}, \tilde{y})$, we set $\tilde{y} = y$ (cf. Equation 5.2) for our PGD version and $\tilde{y} = f_\theta(\boldsymbol{x})$ (cf. Equation 5.3) for our TRADES version.

5.5 Experiments

In this section, we conducted several comprehensive experiments using MNIST [159], and CIFAR10 [7] datasets. We first analyse the adversarial particles generated by SVGD compared to adversarial examples from PGD with random initialisations. We compare the performance of our method to several well-known approaches such as PGD [23], TRADES [26], ADT [181] in robust machine learning, VAT [28] in semi-supervised learning.

5.5.1 Diversity of Adversarial Particles vs. PGD Random Initialisation

General setup The pretrained model of MNIST and CIFAR10 used in this experiment is LeNet [159] and ResNet18 [16] respectively. The LeNet achieves 0.99 accuracy on MNIST, while ResNet18 achieves 0.93 accuracy on CIFAR10. We fix all pre-trained models in order to generate adversarial examples using PGD with random initialisation and adversarial particles using our method. We set the same ϵ (e.g., 0.1 for MNIST, 8/255 for CIFAR10) and number of iterations $N = 200$. Note that all adversarial examples and particles fool the classifiers with 1.0 confidence.

Experimental setup In Figure 5.1, we generate 3 adversarial examples with random initialisations using PGD for an MNIST image (e.g., digit 7). Given the same image,

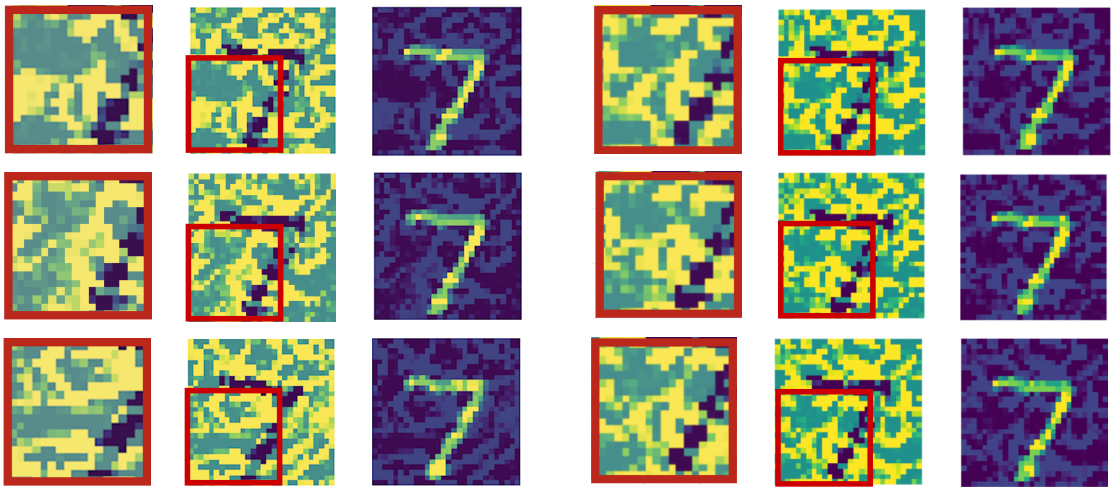
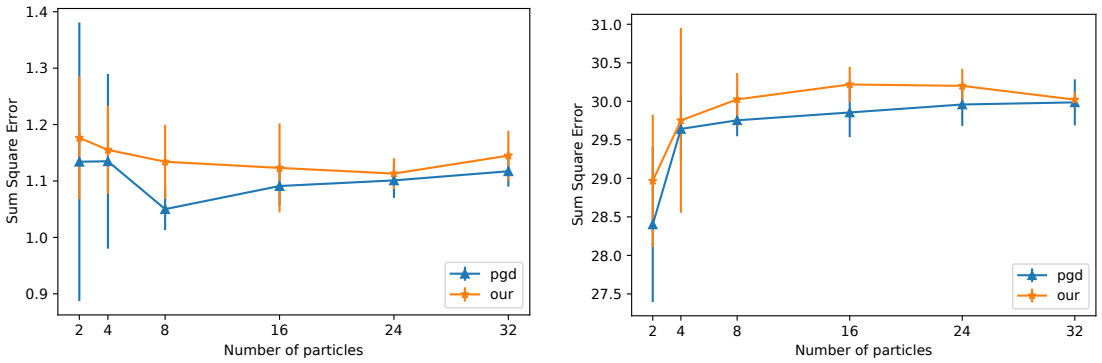


FIGURE 5.1: Comparison of three adversarial examples generated by (a) our method with SVGD and (b) PGD with random initialisation. The first, second and third column of each sub-figure is region of interest of adversarial perturbations (ROIs), adversarial perturbations and adversarial particles respectively.



(A) MNIST (B) CIFAR10

FIGURE 5.2: Diversity comparison of our method and PGD with random initialisation using sum of square error (SSE). The figure illustrates the average of mean (point) and standard deviation (bar) of the three different runs.

we also sample 3 adversarial particles in the adversarial local distribution $P_\theta(\cdot|\boldsymbol{x})$ using SVGD. In Figure 5.2, we generate adversarial examples using PGD with random initialisations and our SVGD method with different numbers of particles for the MNIST and CIFAR10 datasets. We then calculate sum squared error (SSE) between these particles to evaluate their diversity. At each setting of the number of particles, we run 3 times to calculate the average of the means and standard deviations of SSE.

Results Recall that in Alg. 5.1, SVGD is designed to generate diverse adversarial particles from $P_\theta(\cdot|\boldsymbol{x})$ because the first and second terms in ϕ enforce the particles to stay in the high density areas and avoid collapsing into local modes, respectively. Previous

methods such as PGD, TRADES and VAT reply on random initialisation to generate different adversarial examples in $P_\theta(\cdot|\boldsymbol{x}, y)$. Therefore, these previous methods can lie together and lose diversity [180]. Thus, adversarial examples generated from the previous methods are not diverse enough to improve the model performance compared to our method. As seen in Figure 5.1, our method can have significantly different noise patterns (the middle column) compared to PGD especially in the regions of interests in the first column. In Figure 5.2, the figure further shows that our method can generate more diverse samples with bigger SSE compared to PGD with random initialisations in both MNIST and CIFAR10 datasets.

5.5.2 Semi-supervised Learning

Datasets and General setups In order to conduct the experiment with different numbers of labelled samples, we select 300 and 500 labelled samples from the MNIST training dataset and the rest of training samples are unlabelled samples, denoted by MNIST-300 and MNIST-500 respectively. We also select 1000 and 4000 labelled samples from the CIFAR10 training dataset and the rest of the training samples are unlabelled samples, denoted by CIFAR10-1000 and CIFAR10-4000 respectively. For MNIST,

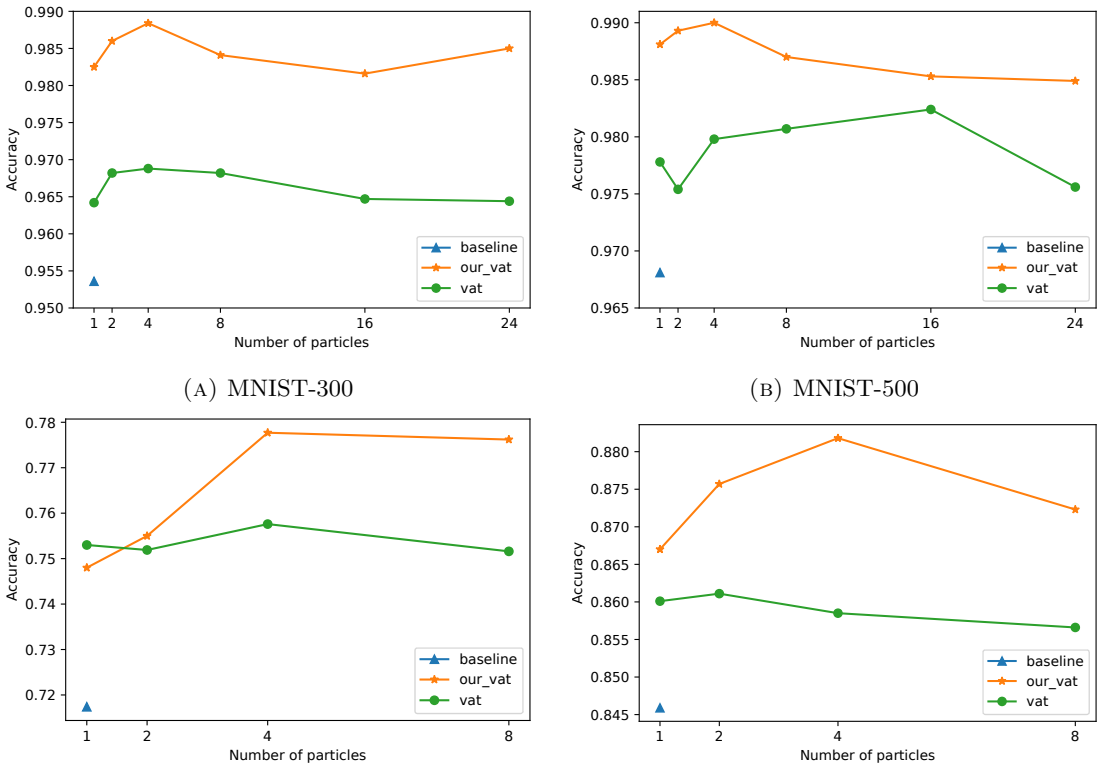
- We select 300 and 500 labelled samples from the MNIST training dataset and the rest of training samples (59700 and 59500) are unlabelled samples, denoted by MNIST-300 and MNIST-500 respectively. Test set consists of 10000 images. All images are scaled from 0 to 1.
- We set $\epsilon = 0.01$, τ=0.01, $r = l_2$ normalisation, $\eta = 10$, $\lambda_1 = \lambda_2$=30 and $N = 1$.
- We use LeNet architecture [159] trained by 400 epochs using SGD optimiser with initial learning rate = 0.1, momentum = 0.9, batch size = 128 and cosine annealing learning rate scheduling between 1e-4 and 0.1.

For CIFAR10,

- We select 1000 and 4000 labelled samples from the CIFAR10 training dataset and the rest of training samples (49000 and 46000) are unlabelled samples, denoted by CIFAR-1000 and CIFAR10-4000 respectively. Test set consists of 10000 images.

All images are scaled using mean = [0.4914, 0.4822, 0.4465], std = [0.2023, 0.1994, 0.2010]

- We set ϵ = 5e-4 , τ=0.01, $r = l_2$ normalisation, $\eta = 10$, $\lambda_1 = \lambda_2$=30 and $N = 1$.
- We use Conv-Large architecture [28] trained by 600 epochs using SGD optimiser with initial learning rate = 0.1, momentum = 0.9, batch size = 128 and cosine annealing learning rate scheduling between 1e-4 and 0.1.



(A) MNIST-300 (B) MNIST-500

(C) CIFAR10-1000 (D) CIFAR10-4000

FIGURE 5.3: Performance comparison of semi-supervised learning using MNIST with LeNet (first row) and CIFAR10 with Conv-Large (second row). (a) MNIST-300 and (b) MNIST-500 use 300 and 500 labelled data of MNIST training set respectively and the rest of training set as unlabelled data . (c) CIFAR10-1000 and (b) CIFAR10-4000 use 1000 and 4000 labelled data of CIFAR10 training data respectively and the rest of training set as unlabelled data. The baseline model is trained by using only labelled data.

We use the LeNet [159] architecture for MNIST and Conv-Large architecture following VAT [28] for CIFAR10. We train both of models 500 epochs using the SGD optimiser[1].

Experimental setup. We set up two experiments for semi-supervised learning using the MNIST and CIFAR10 datasets. The first experiment is the performance comparison between our method and VAT. VAT can generate different adversarial examples using

[1]Based on https://github.com/iBelieveCJM/Tricks-of-Semi-supervisedDeepLeanring-Pytorch

TABLE 5.1: Performance comparison between our method and VAT using mixup technique for all adversarial particles in mini-batch on Conv-Large architecture.

n particle(s)	1	2	4	8
VAT	0.8601	0.8611	0.858	0.856
Our	0.867	0.876	0.883	0.872
VAT + Mixup	0.870	0.887	0.9013	0.893
Our + Mixup	**0.913**	**0.925**	**0.930**	**0.927**

random initialisation; therefore, we compare between VAT and our method at different number of adversarial particles n. Note that we can encourage the regularisation strength of both VAT and our method by increasing number of adversarial particles n. The second experiment is to leverage the Mixup [72] technique to encourage global smoothness in the case of CIFAR10-4000 with Conv-Large architecture in the work [28]. Mixup [72] is a data augmentation technique which generates new training samples by weighted combinations of random image pairs from the training data. We apply Mixup to all adversarial particles within mini-batch to encourage global smoothness, while the local smoothness is enforced by the adversarial local distribution regularisation.

Results Our method can significantly outperform VAT on both MNIST and CIFAR10, as shown in Figure 5.3. When the number of adversarial particles n=4, our method reaches 0.779 and 0.883 accuracy, while VAT with random initialisations achieves only 0.757 and 0.858 accuracy in case of CIFAR10-1000 and CIFAR10-4000 respectively. Our method increases 6% and 3.6% accuracy compared to the baseline models as shown in Figure 5.3c and 5.3d. By increasing the number of particles, we accordingly increase the regularisation strength of our model. It is as expected that over regularisation may hurt the performance. Therefore, we observe the dropping accuracy at n=24 for MNIST and n=8 for CIFAR10. However, in these cases, our method can still outperform VAT.

In Table 5.1, Mixup with mini-batch to encourage global smoothness can significantly improve the accuracy of both VAT and our method. Our method achieves 0.93 accuracy which outperforms VAT because our method can generate more diverse adversarial particles.

5.5.3 Robust Machine Learning

Datasets and General setups The MNIST and CIFAR10 datasets are used in this experiment. For each dataset, all images are scaled from 0 to 1 and we split 1000 samples from the training set as the validation set. For MNIST,

- We select 59000 images for training, 1000 images for validation and 10000 images for testing. All images are scaled from 0 to 1.
- We set $\epsilon = 0.3$, τ=0.01, $r = l_{inf}$ normalisation, η = 1e-3, λ=1 and $N = 40$.
- We use LeNet architecture [159] trained by 100 epochs using SGD optimiser with initial learning rate = 0.01, momentum = 0.9, batch size = 100 and learning rate decay (0.1) scheduling at [75, 90] epoch.

For CIFAR10,

- We select 59000 images for training, 1000 images for validation and 10000 images for testing. All images are scaled from 0 to 1.
- We set $\epsilon = 0.031$, τ=0.007, $r = l_{inf}$ normalisation, η = 1e-3 and $N = 10$. PGD and Our_PGD use λ=1, while TRADES and Our_TRADES use λ=6.
- We use ResNet18 architecture [16] trained by 100 epochs using SGD optimiser with learning rate = 0.1, momentum = 0.9, weight decay = 5e-4, batch size = 100 and learning rate decay (0.1) scheduling at [75, 90] epoch.

We also use LeNet [159] architecture for MNIST and ResNet18 [16] architecture for CIFAR10[2].

Experimental setup PGD [23] and TRADES [26] are two well-known defence techniques in adversarial training defence. While PGD uses the cross-entropy loss (Equation 5.2), TRADES uses the KL divergence loss (Equation 5.3). Recall that our method can be applied with any loss function; therefore, we compare our method with the CE loss (denoted by Our_PGD) vs. PGD and our method with the KL divergence loss (denoted by Our_TRADES) vs. TRADES. We also compare our method with adversarial distributional training [181] (ADT) such as ADT-EXP and ADT-EXPAM, which assume that the adversarial distribution explicitly follows normal distribution.

[2]Based on https://github.com/tuananhbui89/Adversarial-Divergence-Reduction

We evaluate natural accuracy and robust accuracy at different numbers of adversarial particles in Figure 5.4 and 5.5. Natural accuracy is the accuracy of a model with natural inputs, while robust accuracy shows the robustness of a model against adversarial examples generated by attacks. PGD [23] is widely used to attack models because of its effectiveness and stability. We use PGD with the large number of iterations N=200 iteration steps (PGD-200) as a major metric to draw robust accuracy of Figure 5.4 and 5.5. We also use advanced attacks to evaluate the models to evaluate the model robustness against various attacks such as Auto-Attack [32] and B&W attack [192].

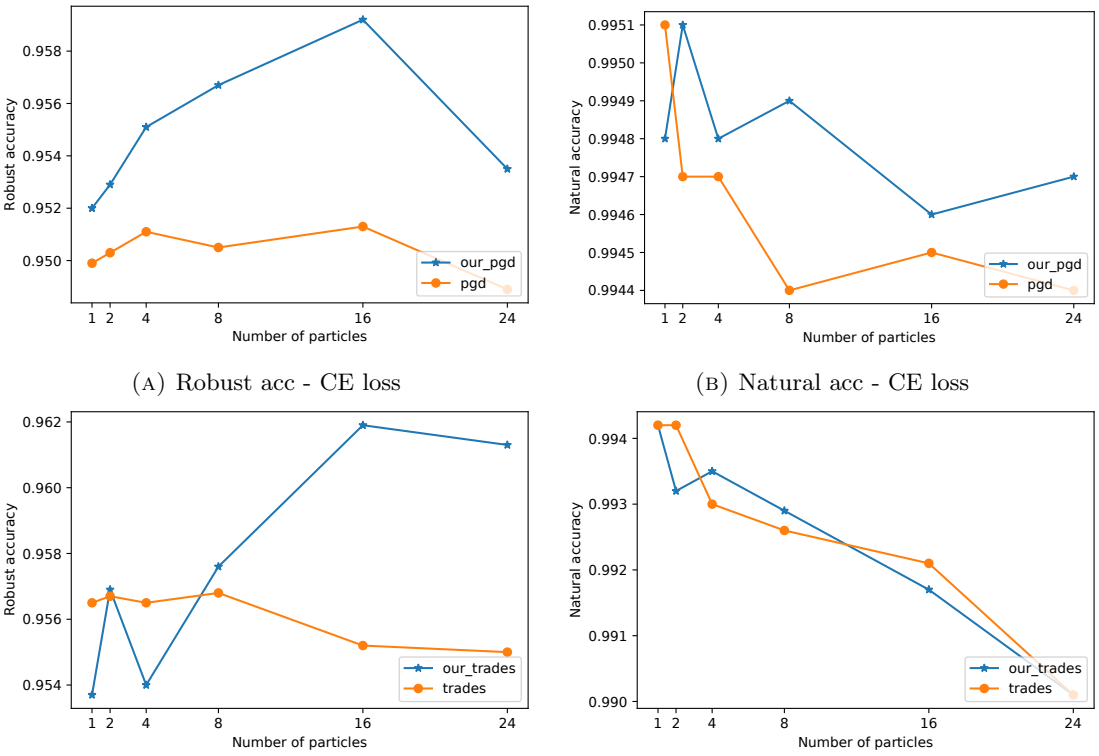
Results As can be seen in Figure 5.4 and 5.5, our method outperforms PGD and TRADES in terms of robust accuracy with the increased the number of adversarial particles. All of the four methods, PGD, TRADES, Our_PGD and Our_TRADES decrease natural accuracy with the overly enforced regularisation strength when the the number of adversarial particles is set to a too large number. This trade-off between natural and robust accuracy is inline with the study in [26]. However, our method can achieve higher natural accuracy than others at the same number of particles.

We illustrate additional natural accuracy in the Table 2 of main chapter. As can be seen, Our_PGD can achieve the highest natural accuracy. Our_TRADES is more robust against various attacks but we trade off natural accuracy. This trade-off between natural and robust accuracy is inline with the study in [26].

TABLE 5.2: Robust and natural accuracy comparison using CIFAR10 with ResNet18.

Method	Natural accuracy
ADT-EXP	0.83
ADT-EXPAM	0.84
PGD	0.852
Our_PGD	**0.857**
TRADES	0.834
Our_TRADES	0.778

In Table 5.3, Our_PGD and Our_TRADES can consistently outperform standard PGD and TRADES against various attacks respectively. ADT has better robust accuracy than Our_PGD in Auto-Attack and B&W attack but Our_TRADES achieves the best performance. Here we emphasise that our method does not assume a particular parameterised of the adversarial local distribution, which is more flexible than ADT.



(A) Robust acc - CE loss

(B) Natural acc - CE loss

(C) Robust acc - KL loss

(D) Natural acc - KL loss

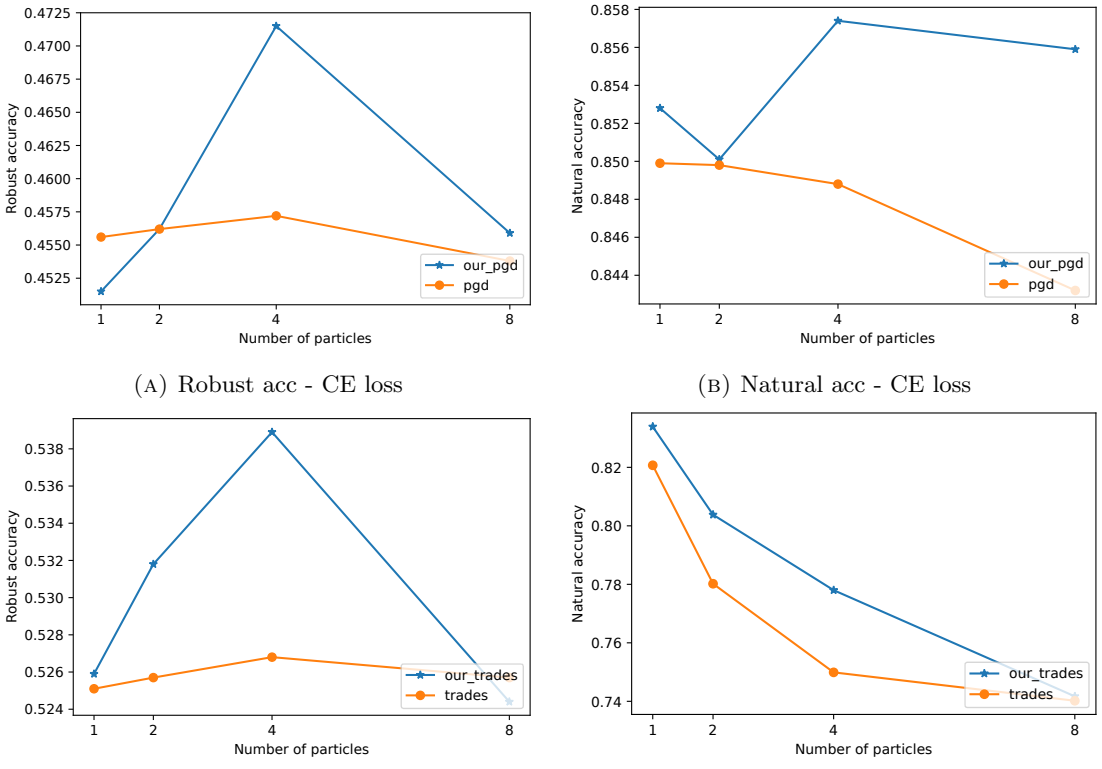
FIGURE 5.4: Robust accuracy against PGD-200 and natural accuracy comparison using MNIST with LeNet architecture.

TABLE 5.3: Robust accuracy comparison using CIFAR10 with ResNet18.

Method	Robust accuracy		
	PGD-200	Auto-Attack	B&B
ADT-EXP	0.458	0.458	0.465
ADT-EXPAM	0.461	0.445	0.458
PGD	0.455	0.419	0.426
Our_PGD	0.471	0.436	0.44
TRADES	0.525	0.483	0.487
Our_TRADES	**0.539**	**0.501**	**0.506**

5.5.4 Running Time

General setup PGD, TRADES and VAT have different Pytorch implementations. Therefore, we adapt our method to these individual code base. We observe the running time on our workstation machine with a TITAN V GPU, 16 cores CPU and 64GB of RAM. In addition, PGD, TRADES and VAT implementations do not optimise for multiple adversarial examples. Thus, we only compare running time per epoch when the number of particles n=1 in Figure 5.6. We also illustrate the running of our method



(A) Robust acc - CE loss

(B) Natural acc - CE loss

(C) Robust acc - KL loss

(D) Natural acc - KL loss

FIGURE 5.5: Robust accuracy against PGD-200 and natural accuracy comparison using CIFAR10 with ResNet18 architecture.

at different numbers of adversarial particles in Figure 5.7.

Results Due to the overhead of kernel computation, the running time of our method is slightly bigger than PGD, TRADES and VAT, as shown in Figure 5.6. However, with the efficient implementation on GPUs, the running time per epoch of our method scales linearly with the number of adversarial particles, as shown in Figure 5.7.

5.6 Conclusion

In this chapter, we have introduced a novel adversarial local distribution regularisation technique that extends and improves on previous methods (e.g., FGSM, PGD, TRADES, and VAT). In our method, SVGD is used to approximate the adversarial local distribution by using more diverse adversarial particles. We adapt our method to a wide range of applications where better generalisation is needed, such as semi-supervised learning, and robust machine learning. Comprehensive experiments show that our method can significantly outperform many widely-used regularisation approaches used in the above

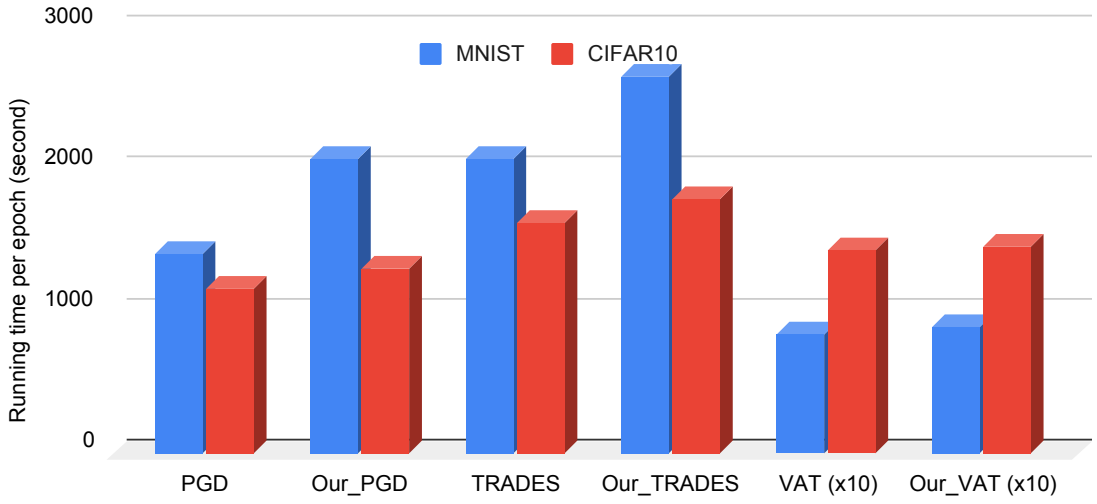
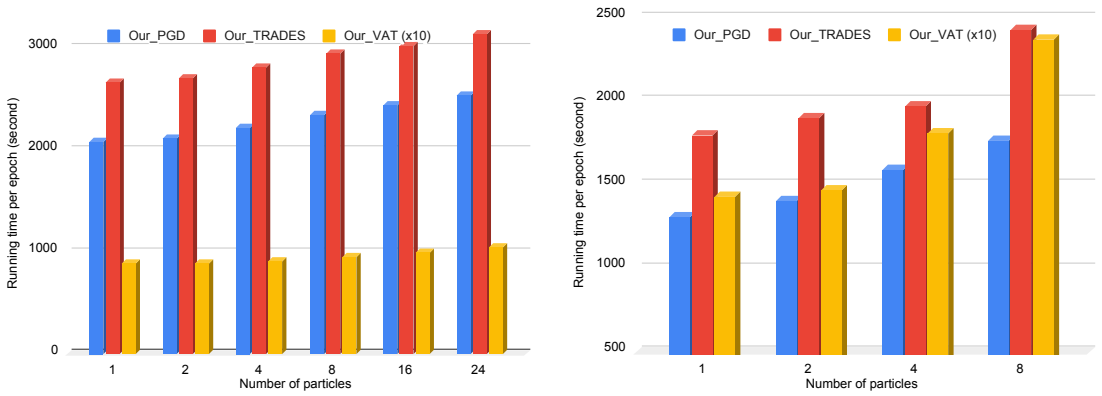


FIGURE 5.6: Running time per epoch of compared methods on MNIST and CIFAR10.



(A) MNIST (B) CIFAR10

FIGURE 5.7: Running time per epoch of our method at different number of adversarial particles on MNIST and CIFAR10.

applications, such as PGD, TRADES, and ADT in robust machine learning, VAT in semi-supervised learning.

5.7 Appendix - Stein's Method

In this section, we provide the discussion based on the research works [183, 193, 194]. Stein's method [195] is a general method in probability theory, which determines the bound of the distance between any two distributions.

5.7.1 Preliminary

Let $\mathcal{X}$ be a subset of $\mathbb{R}^n$, and Ξ: $\mathcal{X} \to \mathbb{R}$ be a smooth function. We assume that $r(\boldsymbol{x})$ is a continuously differentiable function (also called smooth) supported on $\mathcal{X} \subset \mathbb{R}^n$.

Stein class If the function Ξ is belong to the the Stein class of the $r(\theta)$ when it satisfies

$$\int_{\boldsymbol{x}\in\mathcal{X}} \nabla_{\boldsymbol{x}} \Xi(\boldsymbol{x}) r(\boldsymbol{x}) d\boldsymbol{x} = 0. \tag{5.16}$$

We denote $\boldsymbol{\Xi}(\boldsymbol{x}) = [\Xi_1(\boldsymbol{x}), \Xi_2(\boldsymbol{x}), \ldots, \Xi_N(\boldsymbol{x})]$ is a vector-valued function in the Stein class if and only if Ξ_i is in the Stein class of the r for all $i \in \{1, 2, \ldots, N\}$.

Score function The score function of r is defined as follows:

$$s_r := \nabla_{\boldsymbol{x}} \log r(\boldsymbol{x}) = \frac{\nabla_{\boldsymbol{x}} r(\boldsymbol{x})}{r(\boldsymbol{x})}. \tag{5.17}$$

In practice, we usually have an unnormalised $\hat{r}(\boldsymbol{x})$ rather than the normalised version $r(\boldsymbol{x})$. Therefore, we can derive the score function

$$s_{\tilde{r}} := \nabla_{\boldsymbol{x}} \log \tilde{r}(\boldsymbol{x}) = \frac{\nabla_{\boldsymbol{x}} \tilde{r}(\boldsymbol{x})}{\tilde{r}(\boldsymbol{x})} = \frac{\nabla_{\boldsymbol{x}} r(\boldsymbol{x})}{r(\boldsymbol{x})} = s_r. \tag{5.18}$$

Stein's operator The linear Stein operator ($\boldsymbol{A}_r$) for the r acting on the function Ξ is defined as

$$\boldsymbol{A}_r \Xi(\boldsymbol{x}) = s_r(\boldsymbol{x}) \Xi(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} \Xi(\boldsymbol{x}). \tag{5.19}$$

Stein's identity The $r(\boldsymbol{x})$ is a smooth density function supported on $\mathcal{X}$, and the function Ξ is in the Stein class of the r, then the Stein's identity yields (see the proof in [193]):

$$\mathbb{E}_r[\boldsymbol{A}_r \Xi(\boldsymbol{x})] = \mathbb{E}_r[(\boldsymbol{s}_r(\boldsymbol{x})) \Xi(\boldsymbol{x})) + \nabla_{\boldsymbol{x}} \Xi(\boldsymbol{x}))] = 0. \tag{5.20}$$

Now we assume $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ are two smooth density functions both supported on $\mathcal{X}$, and Ξ is a smooth function in the Stein class of p. The expectation is related to the discrepancy between two distributions, as we can demonstrate through the following

derivation (see the details in [194]):

$$\begin{aligned}\mathbb{E}_q[\boldsymbol{A}_p\Xi(\boldsymbol{x})] &= \mathbb{E}_q[\boldsymbol{A}_p\Xi(\boldsymbol{x})] - \mathbb{E}_q[\boldsymbol{A}_q\Xi(\boldsymbol{x})] \\ &= \mathbb{E}_q[(s_q(\boldsymbol{x}) - s_p(\boldsymbol{x}))\Xi(\boldsymbol{x})] \\ &= \mathbb{E}_q[\Xi(\boldsymbol{x})\nabla \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}].\end{aligned} \tag{5.21}$$

We can get a different value for $\mathbb{E}_q[\boldsymbol{A}_p\Xi(\boldsymbol{x})]$ corresponding to each choice of the function Ξ.

Stein discrepancy The Stein discrepancy is defined between two distributions with smooth densities (p and q) .

$$\mathbb{S}(q,p) = \max_{\Xi\in\mathcal{F}}(\mathbb{E}_q[\boldsymbol{A}_p\Xi(\boldsymbol{x})])^2, \tag{5.22}$$

where $\mathcal{F}$ is a function class.

Furthermore, we also define the Stein discrepancy in case $\boldsymbol{\Xi}$ is the vector-valued function as follows:

$$\mathbb{S}(q,p) = \max_{\boldsymbol{\Xi}\in\boldsymbol{\mathcal{F}}}(\mathbb{E}_q[tr(\boldsymbol{A}_p\boldsymbol{\Xi}(\boldsymbol{x}))])^2. \tag{5.23}$$

The selection of $\mathcal{F}$ should meet two criteria: firstly, it must be broad enough to cover the functions that can effectively distinguish between p and q based on the value of $\mathbb{E}_q[\boldsymbol{A}_p\Xi(\boldsymbol{x})]$, and secondly, it needs to guarantee that the optimisation Equation 5.22 process is tractable and capable of being solved.

Reproducing kernel Hilbert space We denote $\mathcal{K}(\boldsymbol{x},\boldsymbol{x}')$ as the positive semi-definite kernel in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. Each function, Ξ, within the RKHS is calculated: $\Xi(\boldsymbol{x}) = \sum_i v_i\ \mathcal{K}(\boldsymbol{x},\boldsymbol{x}')|v_i \in \mathbb{R}; \boldsymbol{x}_i \in \mathcal{X}$ for all i.

The inner product of two functions $\Xi(\boldsymbol{x}) = \sum_i v_i\ \mathcal{K}(\boldsymbol{x},\boldsymbol{x}')$ and $\Theta(\boldsymbol{x}) = \sum_j w_j\ \mathcal{K}(\boldsymbol{x},\boldsymbol{x}')$ is defined as $\langle\Xi,\Theta\rangle_{\mathcal{H}_\mathcal{K}} = \sum_{ij} v_i w_j \mathcal{K}(\boldsymbol{x},\boldsymbol{x}')$ and the norm of Ξ is defined as $||\Xi||_{\mathcal{H}_\mathcal{K}} = \sqrt{\langle\Xi,\Xi\rangle_{\mathcal{H}_\mathcal{K}}}$.

Reproducing property Any function Ξ in the RKHS $\mathcal{H}_\mathcal{K}$ satisfies the reproducing property: $\Xi(\boldsymbol{x}) = \langle\Xi(\cdot),\mathcal{K}(\boldsymbol{x},\cdot)\rangle_{\mathcal{H}_\mathcal{K}}$ and $\nabla_{\boldsymbol{x}}\Xi(\boldsymbol{x}) = \langle\Xi(\cdot),\nabla_{\boldsymbol{x}}\mathcal{K}(\boldsymbol{x},\cdot)\rangle_{\mathcal{H}_\mathcal{K}}$.

Kernelised Stein discrepancy We denote $\mathcal{F}$ as the unit ball in a $\mathcal{H}$ whose positive semi-definite kernel is $\mathcal{K}(\cdot,\cdot)$. The kernelised Stein discrepancy between two distributions with smooth densities p and q is defined as:

$$\mathbb{S}(q,p) = \max_{\Xi\in\mathcal{H}} \Big\{ (\mathbb{E}_q[\boldsymbol{A}_p \Xi(\boldsymbol{x})])^2, ||\Xi||_{\mathcal{H}} \leq 1 \Big\}. \tag{5.24}$$

According to [183], let $\alpha_{q,p}(\cdot) = \mathbb{E}_{\boldsymbol{x}\sim q}[\boldsymbol{A}_p \mathcal{K}(\boldsymbol{x},\cdot)]$, the optimal solution of the kernelised Stein discrepancy in Equation 5.24 (see the proof in [194]) is $\Xi^* = \frac{\alpha_{q,p}}{||\alpha_{q,p}||_{\mathcal{H}}}$ and the value of the kernelised Stein discrepancy is

$$\mathbb{S}(q,p) = ||\alpha_{q,p}||^2_{\mathcal{H}}. \tag{5.25}$$

5.7.2 Stein Variational Gradient Descent

Stein Variational Gradient Descent (SVGD) [183] is a general technique that is used to approximate a complex target probability distribution by particles. SVGD requires only any unnormalised version of the target probability density. Given the target distribution P with probability density function $p(\boldsymbol{x})$, SVGD works by starting with an initial distribution Q_0 with probability density function $q_0(\boldsymbol{x})$ and then progressively adjusting this distribution to more closely approximate the target distribution P. This adjustment is accomplished by applying a series of transformations, which iteratively move the particles towards the target distribution.

Now, we tackle the issue of determining an approximation for a target distribution $p(\boldsymbol{x})$ using the approximate distribution $q(\boldsymbol{x})$. SVGD approximates the target distribution via a set of particles $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$, where N is the number of particles. The particles are firstly initialised by sampling from an initial distribution $q_0(\boldsymbol{x})$ that is usually simple and easy to sample. Generally, our objective is to learn a set of transformations, denoted as $\hat{\Gamma}$, that transports the initial density q_0 towards the target density p. SVGD does the transportation through a series of iterative invertible transformations: $\hat{\Gamma}(\boldsymbol{x}) = \Gamma^L(\Gamma^{L-1}(\Gamma^{\cdots}(\Gamma^1(\boldsymbol{x}))))$. Each transformation pushes the distribution $q_{[\Gamma]}$ to closer to the target distribution (Γ is one of the mappings $\Gamma^1, \Gamma^2, \ldots, \Gamma^L$). According to SVGD [183], it proposed to use the perturbation of mapping as $\Gamma(\boldsymbol{x}) = \boldsymbol{x} + \tau\boldsymbol{\Xi}(\boldsymbol{x})$, where $\Xi(\boldsymbol{x})$ is a smooth function in the function class $\mathcal{F}$ and τ is a small real number.

Let $\Gamma_{\#q}$ be the push-forward measure of the distribution q via the mapping Γ. In order to find optimal transformation Γ, we need to solve the optimisation problem:

$$\min_{\Xi \in \mathcal{F}} D_{\mathrm{KL}}(\Gamma_{\#q}, p), \tag{5.26}$$

where $D_{\mathrm{KL}}(\Gamma_{\#q}, p)$ is the KL divergence from distribution $\Gamma_{\#q}$ to the target distribution p. In the work [183], it was proved that

$$\nabla_\tau D_{\mathrm{KL}}(\Gamma_{\#q}, p)|_{\tau=0} = -\mathbb{E}_q(tr(\boldsymbol{A}_p \boldsymbol{\Xi}(\boldsymbol{x})). \tag{5.27}$$

We obtain the steepest descent that maximises the negative gradient of KL divergence in Equation 5.27 when $\boldsymbol{\Xi}^*(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{x}\sim q}[\boldsymbol{A}_p \mathcal{K}(\boldsymbol{x}, \cdot)]$ (by applying Equation 5.25). As this transformation is iteratively applied, the Kullback-Leibler (KL) divergence between the approximated distribution and the target distribution progressively reduces, provided the step size τ is adequately small. This implies that our approximation is continually improving with each step. We can see that this procedure can work regardless of the choice of the initial approximate density q_0.

In a more technical sense, SVGD operates in the following way (refer to Algorithm 1 in [183] for a more understanding). Initially, we draw a collection of particles $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ from a simple initial approximate distribution characterised by the density q_0. Subsequently, all these particles are transported iteratively, guided by the following set of equations:

$$\boldsymbol{\Xi}(\cdot) = \frac{1}{N} \sum_{n=1}^{N} [\mathcal{K}(\boldsymbol{x}_n^l, \cdot) \nabla_{\boldsymbol{x}_n^l} \log p(\boldsymbol{x}_n^l) + \nabla_{\boldsymbol{x}_n^l} \mathcal{K}(\boldsymbol{x}_n^l, \cdot)]. \tag{5.28}$$

We update the particle using $\boldsymbol{x}_n^{l+1} = \boldsymbol{x}_n^l + \tau \boldsymbol{\Xi}^*(\boldsymbol{x}_n^l)$, where τ is the update step size and $\boldsymbol{x}_n^l$ is the particle $\boldsymbol{x}_n$ at the l^{th} update step.

Unnormalised probability density function In the Equation 5.28, the target probability density only appears in the term $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$, which is the score function Equation 5.17. The score function of a probability density p is equivalent to the score function of its unnormalised density $\tilde{p}$ in Equation 5.18. Therefore, SVGD only requires an unnormalised version of the target probability density.

Particle update term properties In the particle updating: (i) the first one enforces the particles move towards to the high density areas of $p(\boldsymbol{x})$ and (ii) the second one prevents all the particles from collapsing into local modes of $p(\boldsymbol{x})$. The two properties are very interesting properties of SVGD.

Chapter 6

Cross-adversarial Local Distribution Regularisation for Semi-supervised Image Segmentation

Semi-supervised segmentation is a technique where a model is trained to segment objects of interest in images with limited annotated data. Existing semi-supervised segmentation methods are usually based on the smoothness assumption. This assumption implies that the model output distributions of two similar data samples are encouraged to be invariant. In other words, the smoothness assumption states that similar samples (e.g., adding small perturbations to an image) should have similar outputs. We improve the ALD regularisation in Chapter 5 for the semi-supervised image segmentation by introducing a novel cross-adversarial local distribution (Cross-ALD) regularisation. The Cross-ALD further enhances the smoothness assumption for the semi-supervised image segmentation task. In this chapter, we concentrate on medical image segmentation, for which obtaining annotations is both challenging and costly. We conducted comprehensive experiments that the Cross-ALD archives state-of-the-art performance against many recent methods on the medical public LA and ACDC datasets.

6.1 Introduction

Medical image segmentation is a critical task in computer-aided diagnosis and treatment planning. It involves the delineation of anatomical structures or pathological regions in medical images, such as magnetic resonance imaging (MRI) or computed tomography (CT) scans. Accurate and efficient segmentation is essential for various medical applications, including tumour detection, surgical planning, and monitoring disease progression. However, manual medical imaging annotation is time-consuming and expensive because it requires domain knowledge from medical experts. Therefore, there is a growing interest in developing semi-supervised learning that leverages both labelled and unlabelled data to improve the performance of image segmentation models [36, 196].

Existing semi-supervised segmentation methods exploit smoothness assumption, e.g., the data samples that are closer to each other are more likely to have the same label. In other words, the smoothness assumption encourages the model to generate invariant outputs under small perturbations. We have seen such perturbations being added to natural input images at data-level [9, 28, 77, 197, 198], feature-level [199–202], and model-level [34, 203–206]. Among them, virtual adversarial training (VAT) [28] is a well-known one which promotes the smoothness of the local output distribution using adversarial examples. The adversarial examples are near decision boundaries generated by adding adversarial perturbations to natural inputs. However, VAT can only create one adversarial sample in a run, which is often insufficient to completely explore the space of possible perturbations (see Section 6.2.1). In addition, the adversarial examples of VAT can also lie together and lose diversity which significantly reduces the quality of adversarial examples [180, 207]. Mixup regularisation [72] is a data augmentation method used in deep learning to improve model generalisation. The idea behind mixup is to create new training examples by linearly interpolating between pairs of existing examples and their corresponding labels, which has been adopted in [74, 75, 77] to semi-supervised learning. The work [208] suggests that Mixup improves the smoothness of the neural function by bounding the Lipschitz constant of the gradient function of the neural networks. However, we show that mixing between more informative samples (e.g., adversarial examples near decision boundaries) can lead to a better performance enhancement compared to mixing natural samples (see Section 6.3.3).

In this chapter, we propose a novel cross-adversarial local distribution regularisation for semi-supervised medical image segmentation for smoothness assumption enhancement. Our contributions are summarised as follows:

- To overcome the VAT's drawback, we formulate an adversarial local distribution (ALD) with Dice loss function that covers all possible adversarial examples within a ball constraint.
- To enhance the smoothness assumption, we propose a novel cross-adversarial local distribution regularisation (Cross-ALD) to encourage the smoothness assumption, which is a random mixing between two ALDs.
- We also propose a sufficient approximation for the Cross-ALD by a multiple particle-based search using the semantic feature Stein Variational Gradient Decent (SVGDF), an enhancement of the vanilla SVGD [183].
- We conduct comprehensive experiments on ADCD [209] and LA [210] datasets, showing that our Cross-ALD regularisation achieves state-of-the-art performance against existing solutions [9, 28, 33, 34, 203, 204, 206].

6.2 Method

In this section, we begin by revisiting the minmax optimisation problem of virtual adversarial training (VAT)[28]. Given an input, we then formulate a novel adversarial local distribution (ALD) with Dice loss, which benefits the medical semi-supervised image segmentation problem specifically. Next, a cross-adversarial local distribution (Cross-ALD) is constructed by randomly combining two ALDs. We approximate the ALD by a particle-based method named semantic feature Stein Variational Gradient Descent (SVGDF). Considering the resolution of medical images are usually high, we enhance the vanilla SVGD [183] from data-level to feature-level, which is named SVGDF. We finally provide our regularisation loss for semi-supervised medical image segmentation.

6.2.1 The Minimax Optimisation of VAT

To facilitate a better understanding of our proposed approach, we revisit the minimax optimisation of Virtual Adversarial Training (VAT) and the terminologies introduced in

Chapter 5. Let $\mathbb{D}_l$ and $\mathbb{D}_u$ be the labelled and unlabelled dataset, respectively, with $P_{\mathbb{D}_l}$ and $P_{\mathbb{D}_u}$ being the corresponding data distribution. Denote $\boldsymbol{x} \in \mathbb{R}^d$ as our d-dimensional input in a space $\boldsymbol{X}$. The labelled image $\boldsymbol{x}_l$ and segmentation ground-truth $\boldsymbol{y}$ are sampled from the labelled dataset $\mathbb{D}_l$ $(\boldsymbol{x}_l, \boldsymbol{y} \sim P_{\mathbb{D}_l})$, and the unlabelled image sampled from $\mathbb{D}_u$ is $\boldsymbol{x} \sim P_{\mathbb{D}_u}$.

Given an input $\boldsymbol{x} \sim P_{\mathbb{D}_u}$ (i.e., the unlabelled data distribution), let us denote the ball constraint around the image $\boldsymbol{x}$ as $B_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}^{adv} \in \boldsymbol{X} : ||\boldsymbol{x}^{adv} - \boldsymbol{x}||_p \leq \epsilon\}$, where ϵ is a ball constraint radius with respect to a norm $||\cdot||_p$, and $\boldsymbol{x}^{adv}$ is an adversarial example[1]. Given that f_θ is our model parameterised by θ, VAT [28] trains the model with the loss of ℓ_{vat} that a minimax optimisation problem:

$$\ell_{vat} := \min_{\theta} \mathbb{E}_{\boldsymbol{x} \sim P_{\mathbb{D}_u}} \Big[\max_{\boldsymbol{x}^{adv} \in B_\epsilon(\boldsymbol{x})} D_{\mathrm{KL}}(f_\theta(\boldsymbol{x}^{adv}), f_\theta(\boldsymbol{x})) \Big], \tag{6.1}$$

where D_{KL} is the Kullback-Leibler divergence. The inner *maximisation problem* is to find an adversarial example near decision boundaries, while the *minimisation problem* enforces the local smoothness of the model. However, VAT is insufficient to explore the set of of all adversarial examples within the constraint B_ϵ because it only find one adversarial example $\boldsymbol{x}^{adv}$ given a natural input $\boldsymbol{x}$. Moreover, the works [180, 207] show that even solving the *maximisation problem* with random initialisation, its solutions can also lie together and lose diversity, which significantly reduces the quality of adversarial examples.

6.2.2 Adversarial Local Distribution with Dice Loss

In order to overcome the drawback of VAT, we introduce our proposed adversarial local distribution (ALD) with Dice loss function instead of D_{KL} in [28, 207]. ALD forms a set of all adversarial examples $\boldsymbol{x}^{adv}$ within the ball constraint given an input $\boldsymbol{x}$. Therefore, the distribution can help to sufficiently explore all possible adversarial examples. The adversarial local distribution $P_\theta(\boldsymbol{x}^{adv}|\boldsymbol{x})$ is defined with a ball constraint B_ϵ as follow:

$$P_\theta(\boldsymbol{x}^{adv}|\boldsymbol{x}) := \frac{e^{\ell_{Dice}(\boldsymbol{x}^{adv}, \boldsymbol{x}; \theta)}}{\int_{B_\epsilon(\boldsymbol{x})} e^{\ell_{Dice}(\boldsymbol{x}^{adv\prime}, \boldsymbol{x}; \theta)} d\boldsymbol{x}^{adv\prime}} = \frac{e^{\ell_{Dice}(\boldsymbol{x}^{adv}, \boldsymbol{x}; \theta)}}{Z(\boldsymbol{x}; \theta)}, \tag{6.2}$$

[1] A sample generated by adding perturbations toward the adversarial direction.

where $P_\theta(\cdot|\boldsymbol{x})$ is the conditional local distribution, and $Z(\boldsymbol{x};\theta)$ is a normalisation function. The ℓ_{Dice} is the Dice loss function as shown in Equation 6.3

$$\ell_{Dice}(\boldsymbol{x}^{adv}, \boldsymbol{x}; \theta) = \frac{1}{K}\sum_{k=1}^{K}[1 - \frac{2||p_\theta(\hat{\boldsymbol{y}}_k|\boldsymbol{x}) \cap p_\theta(\tilde{\boldsymbol{y}}_k|\boldsymbol{x}^{adv})||}{||p_\theta(\hat{\boldsymbol{y}}_k|\boldsymbol{x}) + p_\theta(\tilde{\boldsymbol{y}}_k|\boldsymbol{x}^{adv})||}], \tag{6.3}$$

where k is the number of classes. $p_\theta(\hat{\boldsymbol{y}}_k|\boldsymbol{x})$ and $p_\theta(\tilde{\boldsymbol{y}}_k|\boldsymbol{x}^{adv})$ are the predictions of input image $\boldsymbol{x}$ and adversarial image $\boldsymbol{x}^{adv}$, respectively.

6.2.3 Cross-adversarial Distribution Regularisation

Given two random samples $\boldsymbol{x}_i, \boldsymbol{x}_j \sim P_{\mathbb{D}}$ $(i \neq j)$, we define the cross-adversarial distribution (Cross-ALD) denoted $\tilde{P}_\theta$ as shown in Equation 6.4

$$\tilde{P}_\theta(\cdot|\boldsymbol{x}_i, \boldsymbol{x}_j) = \gamma P_\theta(\cdot|\boldsymbol{x}_i) + (1-\gamma)P_\theta(\cdot|\boldsymbol{x}_j) \tag{6.4}$$

where $\gamma \sim \text{Beta}(\alpha, \alpha)$ for $\alpha \in (0, \infty)$, inspired by [72]. The $\tilde{P}_\theta$ is the Cross-ALD distribution, a mixture between the two adversarial local distributions.
Given Equation 6.4, we propose the Cross-ALD regularisation at two random input images $\boldsymbol{x}_i, \boldsymbol{x}_j \sim P_{\mathbb{D}}$ $(i \neq j)$ as

$$\mathcal{R}(\theta, \boldsymbol{x}_i, \boldsymbol{x}_j) := \mathbb{E}_{\tilde{\boldsymbol{x}}^{adv} \sim \tilde{P}_\theta(\cdot|\boldsymbol{x}_i, \boldsymbol{x}_j)}[\log \tilde{P}_\theta(\tilde{\boldsymbol{x}}^{adv}|\boldsymbol{x}_i, \boldsymbol{x}_j)] = -H(\tilde{P}_\theta(\cdot|\boldsymbol{x}_i, \boldsymbol{x}_j)), \tag{6.5}$$

where H indicates the entropy of a given distribution.

When minimising $\mathcal{R}(\theta, \boldsymbol{x}_i, \boldsymbol{x}_j)$ or equivalently $-H(P_\theta(\cdot|\boldsymbol{x}_i, \boldsymbol{x}_j))$ w.r.t. θ, we encourage $P_\theta(\cdot|\boldsymbol{x}_i, \boldsymbol{x}_j)$ to be closer to a uniform distribution. This implies that the outputs of $f(\tilde{\boldsymbol{x}}^{adv}) = f(\tilde{\boldsymbol{x}}^{adv'}) =$ a constant c, where $\tilde{\boldsymbol{x}}^{adv}$,$\tilde{\boldsymbol{x}}^{adv'} \sim \tilde{P}_\theta(\cdot|\boldsymbol{x}_i, \boldsymbol{x}_j)$. In other words, we encourage the invariant model outputs under small perturbations. Therefore, minimising the Cross-ALD regularisation loss leads to an enhancement in the model smoothness. While VAT only enforces local smoothness using one adversarial example, Cross-ALD further encourages smoothness of both local and mixed adversarial distributions to improve the model generalisation.

6.2.4 Multiple Particle-based Search to Approximate the Cross-ALD Regularisation

In Equation 6.2, the normalisation $Z(\boldsymbol{x};\theta)$ in the denominator term is intractable to find. Therefore, we propose a multiple particle-based search method named SVGDF to sample $\boldsymbol{x}^{adv(1)}, \boldsymbol{x}^{adv(2)}, \ldots, \boldsymbol{x}^{adv(N)} \sim P_\theta(\cdot|\boldsymbol{x}))$. N is the number of samples (or *adversarial particles*). SVGDF is used to solve the optimisation problem of finding a target distribution $P_\theta(\cdot|\boldsymbol{x}))$. SVGDF is a particle-based Bayesian inference algorithm that seeks a set of points (or particles) to approximate the target distribution without explicit parametric assumptions using iterative gradient-based updates. Specifically, a set of adversarial particles ($\boldsymbol{x}^{adv(n)}$) is initialised by adding uniform noises, then projected onto the ball B_ϵ. These adversarial particles are then iteratively updated using a closed-form solution (Equation 6.6) until reaching termination conditions (e.g., number of iterations).

$$
\begin{aligned}
\boldsymbol{x}^{adv(n),(l+1)} &= \prod_{B_\epsilon}\Big(\boldsymbol{x}^{adv(n),(l)} + \tau * \big(\phi(\boldsymbol{x}^{adv(n),(l)})\big)\Big) \\
s.t.\ \phi(\boldsymbol{x}^{adv}) &= \frac{1}{N}\sum_{j=1}^{N}[\mathcal{K}(\Phi(\boldsymbol{x}^{adv(j),(l)}), \Phi(\boldsymbol{x}^{adv}))\nabla_{\boldsymbol{x}^{adv(j),(l)}} \log P(\boldsymbol{x}^{adv(j),(l)}|\boldsymbol{x}) \\
&\quad + \nabla_{\boldsymbol{x}^{(j),(l)}}\mathcal{K}(\Phi(\boldsymbol{x}^{adv(j),(l)}), \Phi(\boldsymbol{x}^{adv}))],
\end{aligned}
\tag{6.6}
$$

where $\boldsymbol{x}^{adv(n),(l)}$ is a n^{th} adversarial particle at l^{th} iteration ($n \in \{1, 2, ..., N\}$, and $l \in \{1, 2, ..., L\}$ with the maximum number of iteration L). $\prod_{B_\epsilon}$ is projection operator to the B_ϵ constraint. τ is the step size updating. k is the radial basis function (RBF) kernel $\mathcal{K}(\boldsymbol{x}^{adv}, \boldsymbol{x}) = \exp\left\{\frac{-||\boldsymbol{x}^{adv}-\boldsymbol{x}||^2}{2\sigma^2}\right\}$. Φ is a fixed feature extractor (e.g., encoder of U-Net/V-Net). While vanilla SVGD [183] is difficult to capture the semantic meaning of high-resolution data because of calculating RBF kernel ($\mathcal{K}$) directly on the data-level, we use the feature extractor Φ as a semantic transformation to further enhance the SVGD algorithm performance for medical imaging, as shown in Alg. 6.1. Moreover, the two terms of ϕ in Equation 6.6 have different roles: (i) the first one encourages the adversarial particles to move towards the high density areas of $P_\theta(\cdot|\boldsymbol{x})$ and (ii) the second one prevents all the particles from collapsing into the local modes of $P_\theta(\cdot|\boldsymbol{x})$ to enhance diversity (e.g., pushing the particles away from each other).

SVGDF approximates $P_\theta(\cdot|\boldsymbol{x}_i)$ and $P_\theta(\cdot|\boldsymbol{x}_j)$ in Equation 6.4, where $\boldsymbol{x}_i, \boldsymbol{x}_j \sim P_{\mathbb{D}_u}$ $(i \neq j)$. We form sets of adversarial particles as $\mathbb{D}_{adv}|\boldsymbol{x}_i = \{\ \boldsymbol{x}_i^{adv(1)}, \boldsymbol{x}_i^{adv(2)}, \ldots, \boldsymbol{x}_i^{adv(N)}\}$ and

$\mathbb{D}_{adv}|\boldsymbol{x}_j = \{\boldsymbol{x}_j^{adv(1)}, \boldsymbol{x}_j^{adv(2)}, \ldots, \boldsymbol{x}_j^{adv(N)}\}$. The problem (6.5) can then be relaxed to

$$\begin{aligned}\mathcal{R}(\theta, \boldsymbol{x}_i, \boldsymbol{x}_j) := \mathbb{E}_{\boldsymbol{x}_i^{adv(n)} \sim P_{\mathbb{D}_{adv}|\boldsymbol{x}_i}, \boldsymbol{x}_j^{adv(m)} \sim P_{\mathbb{D}_{adv}|\boldsymbol{x}_j}} \Big[\ell_{Dice}(\tilde{x}', \tilde{x}; \theta)\Big] \\ s.t. : \tilde{x}' = \gamma \boldsymbol{x}_i^{adv(n)} + (1-\gamma)\boldsymbol{x}_j^{adv(m)};\ \tilde{x} = \gamma \boldsymbol{x}_i + (1-\gamma)\boldsymbol{x}_j,\end{aligned} \tag{6.7}$$

where $\gamma \sim \text{Beta}(\alpha, \alpha)$ for $\alpha \in (0, \infty)$.

Algorithm 6.1: Approximating the adversarial local distribution (ALD) given $\boldsymbol{x}$ by using semantic feature Stein Variational Gradient Decent (SVGDF).

Input: A natural sample $\boldsymbol{x} \sim P_{\mathbb{D}_u}$; n number of adversarial particles; ϵ for the constraint B_ϵ; r is ℓ_2 normalisation function; η initial noise factor; τ step size updating; L number of iterations; $\mathcal{K}$ is RBF kernel function. Φ is a semantic feature extractor.

Output: Set of adversarial particles $\{\boldsymbol{x}^{adv}{}^1, \boldsymbol{x}^{adv}{}^2, \ldots, \boldsymbol{x}^{adv}{}^N\} \sim P_\theta(\cdot|\boldsymbol{x})$

Initialise a set of n particles and project to the B_ϵ constraint $\{\boldsymbol{x}^{adv}{}_i \in \mathbb{R}^d, i \in \{1, 2, \ldots, N\} | \boldsymbol{x}^{adv}{}^i = \prod_{B_\epsilon}(\boldsymbol{x} + \eta * Uniform\_noise)\}$;

for $l = 1$ *to* L **do**

  for *each particle* $\boldsymbol{x}^{adv}{}^{i,(l)}$ **do**

    $\boldsymbol{x}^{adv}{}^{i,(l+1)} = \prod_{B_\epsilon}\Big(\boldsymbol{x}^{adv}{}^{i,(l)} + \tau * r\big(\phi(\boldsymbol{x}^{adv}{}^{i,(l)})\big)\Big)$;

    where

    $\phi(\boldsymbol{x}^{adv}) = \frac{1}{N}\sum_{j=1}^{N}[\mathcal{K}(\Phi(\boldsymbol{x}^{adv}{}^{j,(l)}), \Phi(\boldsymbol{x}^{adv}))\nabla_{\boldsymbol{x}^{adv}{}^{j,(l)}} \log P(\boldsymbol{x}^{adv}{}^{j,(l)}|\boldsymbol{x}) + \nabla_{\boldsymbol{x}^{j,(l)}}\mathcal{K}(\Phi(\boldsymbol{x}^{adv}{}^{j,(l)}), \Phi(\boldsymbol{x}^{adv}))]$;

return $\{\boldsymbol{x}^{adv}{}^{1,L}, \boldsymbol{x}^{adv}{}^{2,L}, \ldots, \boldsymbol{x}^{adv}{}^{N,L}\}$;

6.2.5 Cross-ALD Regularisation Loss in Medical Semi-supervised Image Segmentation

In this chapter, the overall loss function ℓ_{total} consists of three loss terms. The first term is the dice loss, where labelled image $\boldsymbol{x}_l$ and segmentation ground-truth $\boldsymbol{y}$ are sampled from labelled dataset $\mathbb{D}_l$. The second term is a contrastive learning loss for inter-class separation ℓ_{cs} proposed by [9]. The third term is our Cross-ALD regularisation, which is an enhancement of ℓ_{vat} to significantly improve the model performance.

$$\begin{aligned}\ell_{total} := \min_\theta\ \mathbb{E}_{(\boldsymbol{x}_l, \boldsymbol{y}) \sim P_{\mathbb{D}_l}}\big[l_{Dice}(\boldsymbol{x}_l, \boldsymbol{y}; \theta)\big] + \lambda_{cs}\ \mathbb{E}_{\boldsymbol{x}_l \sim P_{\mathbb{D}_l}, \boldsymbol{x} \sim P_{\mathbb{D}_u}}\big[\ell_{cs}(\boldsymbol{x}_l, \boldsymbol{x})\big] \\ + \lambda_{Cross-ALD}\ \mathbb{E}_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \sim P_{\mathbb{D}_u}}\big[R(\theta, \boldsymbol{x}_i, \boldsymbol{x}_j)\big],\end{aligned} \tag{6.8}$$

where λ_{cs} and $\lambda_{Cross-ALD}$ are the corresponding weights to balance the losses. Note that our implementation is replacing ℓ_{vat} loss with the proposed Cross-AD regularisation in SS-Net code repository[2] [9] to reach the state-of-the-art performance.

6.3 Experiments

In this section, we conduct several comprehensive experiments using the ACDC[3] dataset [209] and the LA [4] dataset [210] for 2D and 3D image segmentation tasks, respectively. For fair comparisons, all experiments are conducted using the identical setting, following [9]. We evaluate our model in challenging semi-supervised scenarios, where only 5% and 10% of the data are labelled and the remaining data in the training set is treated as unlabelled. The Cross-ALD uses the U-Net [157] and V-Net [211] architectures for the ACDC and LA dataset, respectively. We compare the diversity between the adversarial particles generated by our method against vanilla SVGD and VAT with random initialisation in Section 6.3.1 . We then illustrate the Cross-AD outperforms other recent methods on ACDC and LA datasets in Section 6.3.2. We show ablation studies in Section 6.3.3.

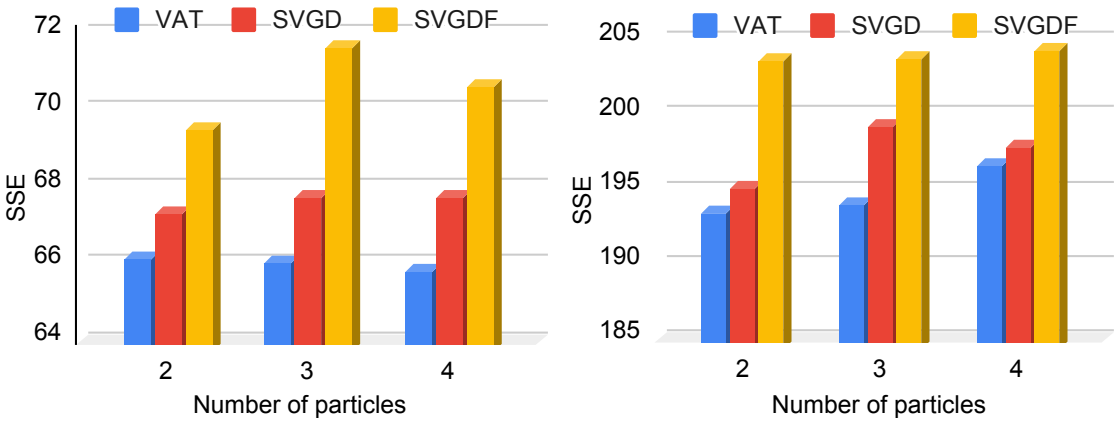
Dataset	Description	Processing code
ACDC	The Automated Cardiac Diagnosis Challenge (ACDC) was created from real clinical exams acquired at the University Hospital of Dijon. Acquired data were fully anonymized and handled within the regulations set by the local ethical committee of the Hospital of Dijon (France). The dataset has a fixed data split, with 70, 10, and 20 patients' scans allocated for training, validation, and testing, respectively.	https://github.com/HiLab-git/SSL4MIS/tree/master/data/ACDC
LA	The Left Atrium (LA) dataset comprises 100 gadolinium-enhanced MRI scans, which have been divided into a fixed split of 80 samples for training and 20 samples for testing.	https://github.com/yulequan/UA-MT/tree/master/data

FIGURE 6.1: Public ACDC and LA datasets.

6.3.1 Diversity of Adversarial Particle Comparison

Settings We fixed all the decoder models (U-Net for ACDC and V-Net for LA). We run VAT with random initialisation and SVGD multiple times to produce adversarial

[2] https://github.com/ycwu1997/SS-Net
[3] https://www.creatis.insa-lyon.fr/Challenge/acdc/databases.html
[4] http://atriaseg2018.cardiacatlas.org



(A) ACDC (B) LA

FIGURE 6.2: Diversity comparison of our SVGDF, SVGD and VAT with random initialisation using sum of square error (SSE) of ACDC and LA datasets.

examples, which we compared to the adversarial particles generated using SVGDF. SVGDF is the proposed algorithm, which leverages feature transformation to capture the semantic meaning of inputs. Φ is the decoder of U-Net in ACDC dataset, while Φ is the decoder of V-Net in LA dataset. We set the same radius ball constraint, updating step, etc.. We randomly pick three images from the datasets to generate adversarial particles. To evaluate their diversity, we report the sum squared error (SSE) between these particles. Higher SSE indicates more diversity, and for each number of particles, we calculate the average of the mean of SSEs.

Results Note that the advantage of SVGD over VAT is that the former generates diversified adversarial examples because of the second term in Equation 6.6 while VAT only creates one example. Moreover, vanilla SVGD is difficult to capture the semantic meaning of high-resolution medical imaging because it calculates kernel $\mathcal{K}$ on image-level. In Figure 6.2, our SVGDF produces the most diverse particles compared to SVGD and VAT with random initialisation.

6.3.2 Performance Evaluation on The ACDC and LA Datasets

Settings We use the metrics of Dice, Jaccard, 95% Hausdorff Distance (95HD), and Average Surface Distance (ASD) to evaluate the results. We compare our Cross-ALD to six recent methods including UA-MT [206] (MICCAI'19), SASSNet [203] (MICCAI'20), DTC [34] (AAAI'21) , URPC [204] (MICCAI'21) , MC-Net [33] (MICCAI'21), and SS-Net [9] (MICCAI'22). The loss weights $\lambda_{Cross-ALD}$ and λ_{cs} are set as an iteration

TABLE 6.1: Performance comparisons with six recent methods on ACDC dataset. All results of existing methods are used from [9] for fair comparisons.

Method	# Scans used		Metrics				Complexity	
	Labelled	Unlabelled	Dice(%)↑	Jaccard(%)↑	95HD(voxel)↓	ASD(voxel)↓	Para.(M)	MACs(G)
U-Net	3(5%)	0	47.83	37.01	31.16	12.62	1.81	2.99
U-Net	7(10%)	0	79.41	68.11	9.35	2.7	1.81	2.99
U-Net	70(All)	0	91.44	84.59	4.3	0.99	1.81	2.99
UA-MT [206]	3 (5%)	67(95%)	46.04	35.97	20.08	7.75	1.81	2.99
SASSNet [203]			57.77	46.14	20.05	6.06	1.81	3.02
DTC [34]			56.9	45.67	23.36	7.39	1.81	3.02
URPC [204]			55.87	44.64	13.6	3.74	1.83	3.02
MC-Net [33]			62.85	52.29	7.62	2.33	2.58	5.39
SS-Net [9]			65.82	55.38	6.67	2.28	1.83	2.99
Cross-ALD (**Ours**)			**80.6**	**69.08**	**5.96**	**1.9**	1.83	2.99
UA-MT [206]	7 (10%)	63(90%)	81.65	70.64	6.88	2.02	1.81	2.99
SASSNet [203]			84.5	74.34	5.42	1.86	1.81	3.02
DTC [34]			84.29	73.92	12.81	4.01	1.81	3.02
URPC [204]			83.1	72.41	4.84	1.53	1.83	3.02
MC-Net [33]			86.44	77.04	5.5	1.84	2.58	5.39
SS-Net [9]			86.78	77.67	6.07	**1.4**	1.83	2.99
Cross-ALD (**Ours**)			**87.52**	**78.62**	**4.81**	1.6	1.83	2.99

TABLE 6.2: Performance comparisons with six recent methods on LA dataset. All results of existing methods are used from [9] for fair comparisons.

Method	# Scans used		Metrics				Complexity	
	Labelled	Unlabelled	Dice(%)↑	Jaccard(%)↑	95HD(voxel)↓	ASD(voxel)↓	Para.(M)	MACs(G)
V-Net	4(5%)	0	52.55	39.6	47.05	9.87	9.44	47.02
V-Net	8(10%)	0	82.74	71.72	13.35	3.26	9.44	47.02
V-Net	80(All)	0	91.47	84.36	5.48	1.51	9.44	47.02
UA-MT [206]	4 (5%)	76(95%)	82.26	70.98	13.71	3.82	9.44	47.02
SASSNet [203]			81.6	69.63	16.16	3.58	9.44	47.05
DTC [34]			81.25	69.33	14.9	3.99	9.44	47.05
URPC [204]			82.48	71.35	14.65	3.65	5.88	69.43
MC-Net [33]			83.59	72.36	14.07	2.7	12.35	95.15
SS-Net [9]			86.33	76.15	9.97	2.31	9.46	47.17
Cross-ALD (**Ours**)			**88.62**	**79.62**	**7.098**	**1.83**	9.46	47.17
UA-MT [206]	8 (10%)	72(90%)	87.79	78.39	8.68	2.12	9.44	47.02
SASSNet [203]			87.54	78.05	9.84	2.59	9.44	47.05
DTC [34]			87.51	78.17	8.23	2.36	9.44	47.05
URPC [204]			86.92	77.03	11.13	2.28	5.88	69.43
MC-Net [33]			87.62	78.25	10.03	1.82	12.35	95.15
SS-Net [9]			88.55	79.62	**7.49**	1.9	9.46	47.17
Cross-ALD (**Ours**)			**89.92**	**81.78**	7.65	**1.546**	9.46	47.17

dependent warming-up function [55], and number of particles $N = 2$. All experiments are conducted using the identical settings in the Github repository[5] [9] for fair comparisons.

Results Recall that our Cross-ALD generates diversified adversarial particles using SVGDF compared to vanilla SVGD and VAT, and further enhances the smoothness of cross-adversarial local distributions. In Table 6.1 and 6.2, the Cross-ALD can significantly outperform other recent methods with only 5%/10% labelled data training based on the four metrics. Especially, our method impressively gains 14.7% and 2.3% Dice score higher than state-of-the-art SS-Net using 5% labelled data of ACDC and LA, respectively. Moreover, the visualised results of Figure 6.3 show Cross-ALD can segment the most organ details compared to other methods.

6.3.3 Ablation Study

TABLE 6.3: Ablation study on ACDC and LA datasets.

Dataset	Method	# Scans used		Metrics			
		Labelled	Unlabelled	Dice(%)↑	Jaccard(%)↑	95HD(voxel)↓	ASD(voxel)↓
ACDC	U-Net	4(5%)	0	47.83	37.01	31.16	12.62
	RanMixup	4 (5%)	76(95%)	61.78	51.69	8.16	3.44
	VAT			63.87	53.18	7.61	3.38
	VAT + Mixup			66.23	56.37	7.18	2.53
	SVGD			66.53	58.09	6.41	2.4
	SVGDF			73.15	61.71	6.32	2.12
	SVGDF + ℓ_{cs}			74.89	62.61	6.52	2.01
	Cross-ALD (**Ours**)			**80.6**	**69.08**	**5.96**	**1.9**
LA	V-Net	3(5%)	0	52.55	39.6	47.05	9.87
	RanMixup	3 (5%)	67(95%)	79.82	67.44	16.52	5.19
	VAT			82.27	70.46	13.82	3.48
	VAT + Mixup			83.28	71.77	12.8	2.63
	SVGD			84.62	73.6	11.68	2.94
	SVGDF			86.3	76.17	10.01	2.11
	SVGDF + ℓ_{cs}			86.55	76.51	9.41	2.24
	Cross-ALD (**Ours**)			**87.52**	**78.62**	**4.81**	**1.6**

Settings We use the same network architectures and parameter settings in Section 6.3.2, and train the models with 5% labelled training data of ACDC and LA. We illustrate that crossing adversarial particles is more beneficial than random mixup between natural inputs (RanMixup [72]) because these particles are near decision boundaries. Recall that our SVGDF is better than VAT and SVGD by producing more diversified adversarial particles. Applying SVGDF's particles and ℓ_{cs} (SVGDF + ℓ_{cs}) to gain the model performance in the semi-supervised segmentation task, while Cross-ALD efficiently enhances smoothness to significantly improve the generalisation.

Result Table 6.3 shows that mixing adversarial examples from VAT outperforms those from RanMixup. While SVGDF + ℓ_{cs} is better than SVGD and VAT, the proposed

[5]https://github.com/ycwu1997/SS-Net

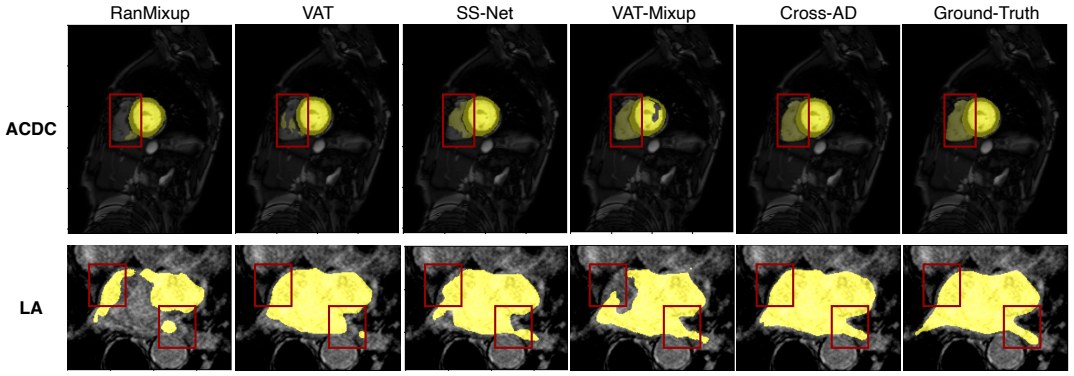


FIGURE 6.3: Visualisation results of several semi-supervised segmentation methods with 5% labelled training data and its corresponding ground-truth on ACDC and LA datasets.

Cross-ALD achieves the most outstanding performance among comparison methods. In addition, our method produces more accurate segmentation masks compared to the ground-truth, as shown in Figure 6.3.

6.4 Adversarial Particle Analysis

TABLE 6.4: We study the number of adversarial particles that affect to the model performance.

Dataset	# Particles	# Scans used		Metrics			
		Labelled	Unlabelled	Dice(%)↑	Jaccard(%)↑	95HD(voxel)↓	ASD(voxel)↓
ACDC	1	3 (5%)	67(95%)	76.59	65.73	8.44	2.21
	2			**80.6**	**69.08**	5.96	**1.9**
	3			80.36	68.05	**5.61**	2.07
	4			77.86	65.49	6.16	2.14
LA	1	4 (5%)	76(95%)	86.83	77.03	5.5671	1.993
	2			87.52	**78.62**	**4.81**	**1.6**
	3			**87.71**	78.44	5.204	1.9216

Settings We study the number of adversarial particles that affect to the model performance. The settings are kept similar to Section 6.3.2 except the number of particles N. We set N in $\{1, 2, 3, 4\}$ for ACDC dataset and N in $\{1, 2, 3\}$ for LA dataset.

Results Note that we use cross-adversarial particles to enhance smoothness. Therefore, by increasing the number of particles, we accordingly increase the regularisation strength. The model performance increases by increasing N. However, it is as expected that over regularisation may hurt the performance when $N = 4$ in ACDC dataset[6]. With more adversarial particles, our method may have increased training time. However, the

[6]We cannot run N=4 in 3D segmentation task of LA dataset with the equal batch size due to the limitation of GPU memory.

model inference time remains unchanged because we do not use these particles during the inference stage.

6.5 Conclusion

In this chapter, we have introduced a novel cross-adversarial local distribution (Cross-ALD) regularisation that extends and overcomes drawbacks of VAT and Mixup techniques. In our method, SVGDF is proposed to approximate Cross-ALD, which produces more diverse adversarial particles than vanilla SVGD and VAT with random initialisation. We adapt Cross-ALD to semi-supervised medical image segmentation to achieve start-of-the-art performance on the ACDC and LA datasets compared to many recent methods such as VAT [28], UA-MT [206], SASSNet [203], DTC [34], URPC [204] , MC-Net [33], and SS-Net [9].

Chapter 7

Adversarial Local Distribution Regularisation for Knowledge Distillation

In Chapters 3 and 4, we introduced the information-theoretic approach to deepen our understanding of adversarial attacks and knowledge distillation. Then, in Chapters 5 and 6, we proposed the adversarial local distribution (ALD) regularisation for enhancing robust machine learning and semi-supervised learning. In this chapter, we explore the properties of the teacher as the key to improving student performance (e.g., teacher decision boundaries). One decision boundary exploring technique is to leverage adversarial attack methods, which add crafted perturbations within a ball constraint to natural inputs to create attack examples of the teacher called adversarial examples. These adversarial examples are informative examples because they are near decision boundaries. We formulate a teacher adversarial local distribution, a set of all adversarial examples within the ball constraint given an input. This distribution is used to sufficiently explore the decision boundaries of the teacher by covering the full spectrum of possible teacher model perturbations. The student model is then regularised by matching the loss between teacher and student using these adversarial example inputs. We conducted comprehensive experiments on CIFAR-100 and Imagenet datasets to illustrate this teacher adversarial local distribution regularisation (TALD) can be applied to improve the performance of many existing knowledge distillation methods (e.g., KD, FitNet, CRD, VID,

FT, etc.).

7.1 Introduction

Transferring knowledge from an excessive deep learning model (teacher) to a lighter model (student) is known as knowledge distillation (KD). The light-weight student is advantageous when deployment costs need to be lowered due to the devices' constrained computing and memory capabilities. Hinton et al. [3] originally introduced the objective of KD loss, which minimises the KL divergence between the teacher and student outputs. This KD loss extracts knowledge from the teacher's class probabilities with the temperature softmax to guide the training of the student. Therefore, the student network is developed to be a better classifier than the student developed without KD loss.

Many studies improve the KD loss [3] for matching the teacher and student outputs such as label smoothing [30], virtual teacher [212], and decouple KD loss [31]. Moreover, deep learning models are well-learned multiple levels of feature representation [29]. Feature-based KD is adopted in the works [35, 105, 213], in which the teacher provides intermediate representations and hints for training the student. These previous approaches attempt to manipulate various network components to enhance the knowledge distillation process. The work [214] shows that input samples close to the classifier's decision boundaries affect performance more than samples further from it, which can help to regularise the student [5]. Therefore, we can effectively transfer the teacher properties to the student by utilizing informative samples near the teacher decision boundaries.

One strategy for exploring decision boundaries is utilizing adversarial attack approaches. The adversarial attack [5, 22, 23, 26, 32] transports natural inputs to the model's decision boundaries by iteratively adding specially designed perturbations inside of a ball constraint to produce adversarial examples. Although finding a decision boundary is not the primary objective of an adversarial attack, they are closely related to each other [215]. Moreover, the vanilla adversarial attacks [5, 22, 23, 26, 32, 216, 217] can only create one adversarial example, which may not be enough to examine the full spectrum of possible teacher model perturbations. The works [180, 218] also show how attacks with random

initialisation can lie together and lose diversity, which reduces the quality of adversarial examples.

In this chapter, we introduce a *teacher adversarial local distribution* (TALD) regularisation for knowledge distillation, which can be used to improve many existing KD approaches. Our contributions are summarised as follows:

- We explore the teacher decision boundaries by introducing the teacher adversarial local distribution, a set of all adversarial examples within the constraint given an input that maximises a teacher loss function.
- We find TALD by using a multiple particle-based search named Stein Variational Gradient Decent (SVGD) [183]. The SVGD sufficiently approximates the TALD without any assumptions and creates more diverse adversarial examples. The student decision boundaries are then regularised by matching the loss between teacher and student using these adversarial example inputs.
- We show that our method can be adapted well to various existing methods. We conduct various experiments on CIFAR-100 and ImageNet to demonstrate our TALD regularisation can improve the performance of many existing methods such as KD [3], FitNet [35], AT [219], SP [120], CC [121], VID [112], RKD [116], PKT [105], AB [5], FT [106], NST [104], CRD [10], SSKD [123], and HSAKD[11].

7.2 Related Work

Knowledge distillation Hinton et al. [3] originally introduced knowledge distillation (KD) which extracts knowledge from class probabilities of a large deep learning model (teacher) to a lighter model (student). Many approaches have adopted the class-probability knowledge transfer perspective of KD [3] to improve model compression such as class-distance loss [30], label smoothing [220], adaptive regularisation [221], virtual teacher [212], and decoupling KD loss [31]. In addition, deep learning models are well-learned in multiple levels of feature representation [29]. Romero et al. exploited the intermediate teacher representation that the student model is trained by matching the teacher responses from multiple hidden layers named FitNets [35]. Various approaches have been proposed inspired by [35] to significantly improve the student model. The

teacher feature maps are selected using the attention mechanism to omit redundant knowledge transfer from the teacher to student [103, 104]. The knowledge from the teacher can distil to the student and be explained [222]. Zhou et al. [108] explored parameter sharing of intermediate layers of the teacher model. Cross-layer knowledge distillation [213] adaptively assigns proper teacher layers for each student layer via attention allocation that matches the semantics between teacher and student. The work [223] proposed an efficient use of the pre-trained teacher's intermediate representations. Contrastive learning loss was proposed by Tian et al. [10] to capture correlations and higher order output dependencies. A hierarchical self-supervised learning technique was proposed in the work [11] to improve the student. However, these above approaches have not considered the teacher decision boundary perspective. In this chapter, we introduce the regularisation using the teacher decision boundary information to add an additional help to enhance the teacher property transfer to improve the student. Our regularisation loss only requires the teacher to be differentiable and has no additional learnable module. Therefore, we can easily add the regularisation loss to many existing KD methods.

Adversarial attack State-of-the-art deep neural networks are reportedly vulnerable to attacks [21, 22]. Fast Gradient Sign Method (FGSM) [22], Projected Gradient Descent (PGD) [147], and Auto-Attack [32] are a few examples of adversarial attacks that add specially crafted perturbations to natural inputs to produce adversarial examples. The most popular technique for finding perturbations is using gradients to maximise a model's loss on given a natural input while limiting the perturbation size smaller than a specified amount referred to a radius constraint epsilon. In other words, the adversarial attacks find a path to transport natural inputs to cross model decision boundaries, which means to fool the model prediction. Due to the threats, many methods have been proposed for defence techniques using adversarial examples such as [23–26, 224, 225]. Recently, the works [181, 218] proposed adversarial distribution training to improve the model robustness. In knowledge distillation, exploring the properties of the teacher (e.g., decision boundaries) is the key to improving student performance. Therefore, we leverage the attack to explore the teacher decision boundaries using generated adversarial examples. These adversarial examples are then used to regularise the student.

Knowledge distillation using adversarial attacks Many previous approaches use knowledge distillation for transferring robustness from a well-defended teacher to a student. Robustness transfer from a robust teacher to a student using KD loss [3] technique

was proposed by the work [226]. Chan et al. [227] trained a student model's input gradients to match those of the robust teacher to gain robustness. In addition, the work [131] proposed a noisy feature distillation, a new transfer learning method that improves robustness. Other works [228, 229] used contrastive learning loss to transfer robustness. These above distillation approaches only attempt to distil robustness to defend from adversarial attacks. Heo et al.'s paper [5] proposed a BSS attack for exploring the teacher's properties using adversarial examples to increase the student's natural input accuracy. This BSS can only produce one adversarial example, which insufficiently explores the full spectrum of possible teacher model perturbations [180, 218]. In this chapter, our approach sufficiently explores teacher's properties (e.g., decision boundaries) using the teacher adversarial local distribution (TALD). The student is then regularised by TALD regularisation to improve natural input accuracy.

7.3 Method

In this section, we introduce our teacher adversarial local distribution (TALD) regularisation that can be used to improve the performance of many previous knowledge distillation methods (e.g., KD, FitNet, CRD, SSKD, etc.). We denote the large classifier teacher model by T with parameters θ_T. The teacher is pre-trained and fixed. The student is a smaller model which needs to be trained with help from the teacher. The smaller student model is S parameterised by θ_S. Let input $\boldsymbol{x} \in \mathbb{R}^d$ be our d-dimensional natural input data in a space $\boldsymbol{X}$, and $(\boldsymbol{x}, y) \sim P_{\mathbb{D}}$ is our data-label distribution.

7.3.1 Teacher Adversarial Local Distribution

We use adversarial examples, which are near decision boundaries, to explore teacher decision boundary properties called teacher adversarial examples. The student decision boundaries are then regularised by matching the teacher loss and student loss given these input examples. These adversarial inputs can be found by attacking the teacher model. The attack adds crafted perturbations within a ball constraint to natural input $\boldsymbol{x}$, which maximises the teacher loss function ℓ to generate adversarial examples $\boldsymbol{x}^{adv}$. We denote the ball constraint of $\boldsymbol{x}^{adv}$ by $B_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}^{adv} \in \boldsymbol{X} : ||\boldsymbol{x}^{adv} - \boldsymbol{x}||_p \leq \epsilon\}$, where $\boldsymbol{x}^{adv}$ is

adversarial example, and ϵ is a ball constraint radius with respect to a norm $||\cdot||_p$. The teacher attack can be defined by the maximisation optimisation in Equation 7.1.

$$\boldsymbol{x}^{adv} = \underset{\boldsymbol{x}^{adv} \in B_\epsilon(\boldsymbol{x})}{\arg\max}\ \ell(\boldsymbol{x}^{adv}, \boldsymbol{x}; \theta_T), \tag{7.1}$$

where ℓ is the Kullback-Leibler divergence loss (D_{KL}) $D_{\mathrm{KL}}(T(\boldsymbol{x}^{adv}), T(\boldsymbol{x}))$ [26]. However, vanilla attack methods [5, 23, 26, 32] can only create one adversarial example, which could be insufficient to explore entire space of possible teacher model perturbations. In addition, the works [180, 218] illustrate even attacks with random initialisation can also lie together and lose diversity which reduces the quality of adversarial examples.

We propose to improve the vanilla adversarial attack optimisation (Equation 7.1) with a teacher adversarial local distribution (TALD), which captures the distribution of all teacher adversarial examples around natural input $\boldsymbol{x}$ within the constraint $B_\epsilon(\boldsymbol{x})$, as shown in Equation 7.2.

$$\begin{aligned} P_{\theta_T}(\boldsymbol{x}^{adv}|\boldsymbol{x}) : &= \frac{e^{\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}; \theta_T)}}{\int_{B_\epsilon(\boldsymbol{x})} e^{\ell(\boldsymbol{x}^{adv\prime}, \boldsymbol{x}; \theta_T)} d\boldsymbol{x}^{adv\prime}} \\ &= \frac{e^{\ell(\boldsymbol{x}^{adv}, \boldsymbol{x}; \theta_T)}}{Z(\boldsymbol{x}; \theta_T)}, \end{aligned} \tag{7.2}$$

where $P_{\theta_T}(\cdot|\boldsymbol{x})$ is the teacher conditional adversarial local distribution over $B_\epsilon(\boldsymbol{x})$, and a normalisation function is $Z(\boldsymbol{x}; \theta_T)$. Here we show that the TALD can sufficiently represent the entire space of possible teacher perturbations.

7.3.2 TALD Approximation using Multiple Particle-based Search

In this chapter, we leverage a multiple particle-based search method named Stein Variational Gradient Descent (SVGD) [183] to find the TALD $P_{\theta_T}(\cdot|\boldsymbol{x})$ because finding the denominator $Z(\boldsymbol{x}; \theta_T)$ term in the Equation 7.2 is intractable. SVGD is a Bayesian inference algorithm that seeks a set of points (or particles) to approximate the target distribution using iterative gradient-based updates. It has a simple form that closely mimics the typical gradient descent for optimisation. This makes SVGD highly flexible and scalable, and can be easily combined with various state-of-the-art techniques

responsible for the success of gradient optimisation. While Markov chain Monte Carlo (MCMC) is often slow and has difficulty reaching convergence, SVGD efficiently approximates the target distribution by using an off-the-shelf optimisation solver and is easily applicable to large datasets. It also enforces diversity of particles and works without explicit parametric assumptions in its solution, demonstrating better than other particle-based SGLD [230] and parametric-based method (with strong assumptions such as the target distribution follows Gaussian distributions) [181].

We denote $\boldsymbol{x}_1^{adv}, \boldsymbol{x}_2^{adv}, \boldsymbol{x}_3^{adv}, \ldots, \boldsymbol{x}_N^{adv} \sim P_{\theta_T}(\cdot|\boldsymbol{x})$, where $\boldsymbol{x}_i^{adv}$ is a i^{th} teacher adversarial example (named *teacher adversarial particle*), and $N = |\{1, 2, \ldots, N\}|$ is the number of adversarial examples. Here we show that our method can sufficiently explore the teacher decision boundaries by using multiple adversarial particles compared to vanilla attacking methods [5, 23, 26, 32]. SVGD is used to find a set of teacher adversarial particles to approximate the teacher adversarial local distribution $P_{\theta_T}(\cdot|\boldsymbol{x})$. First, the particles $\{\boldsymbol{x}_1^{adv}, \boldsymbol{x}_2^{adv}, \boldsymbol{x}_3^{adv}, \ldots, \boldsymbol{x}_N^{adv}\}$ are initialised by adding uniform noises to $\boldsymbol{x}$ constrained within the $B_\epsilon(\boldsymbol{x})$. They are then iteratively updated as well as projected to $B_\epsilon(\boldsymbol{x})$ until reaching the termination conditions (line 4 in Alg. 7.1). The normalisation function $Z(\boldsymbol{x}; \theta_T)$ is estimated based on the number of particles (N), which is implicitly demonstrated in the mean operator of line 5 - Alg. 7.1. Moreover, the two terms of line 5 in Alg. 7.1 have different major roles: (i) the first one transports the adversarial particles more toward to the high density areas of $P_{\theta_T}(\cdot|\boldsymbol{x})$ and (ii) the second term prevents all particles from collapsing into local modes of $P_{\theta_T}(\cdot|\boldsymbol{x})$ (e.g., pushing the particles away for enhancing the particle diversity). We empirically use l_2 normalisation (l_2), and radial basis function kernel $\mathcal{K}(\boldsymbol{x}', \boldsymbol{x}) = \exp\left\{\frac{-||\boldsymbol{x}'-\boldsymbol{x}||^2}{2\sigma^2}\right\}$ with σ=1e-3 in this chapter. We show that our method in a generalisation of previous attacks when $N = 1$ from an asymptotic analysis of adversarial local distribution approximation in Chapter 5.

7.3.3 Teacher Adversarial Local Distribution (TALD) Regularisation

In this section, we propose our Teacher Adversarial Local Distribution (TALD) regularisation. Recall that we form the TALD which is approximated using the adversarial particles generated by SVGD. We now illustrate how to use the teacher adversarial particles for knowledge distillation. We propose the TALD regularisation term (ℓ_{TALD})

Algorithm 7.1: Stein Variational Gradient Descent solver to approximate the teacher adversarial local distribution $P_{\theta_T}(\cdot|\boldsymbol{x})$.

Input: natural example $\boldsymbol{x} \sim P_{\mathbb{D}}$. Number of adversarial particles N. Radius ϵ of the constraint B_ϵ. Normalisation function l_p. Initial noise factor η. Uniform noise $U(-\epsilon, \epsilon)$. Step size updating particles τ. Number of iterations L. Kernel function $\mathcal{K}$.

Initialise a set of N particles and project to the $B_\epsilon(x)$ constraint $\{\boldsymbol{x^{adv}}_i \in \mathbb{R}^n, i \in \{1, 2, \ldots, N\} | \boldsymbol{x^{adv}}_i = \prod_{B_\epsilon}(\boldsymbol{x} + \eta * Uniform\_noise(-\epsilon, \epsilon))\}$;

for $l = 1$ *to* L **do**

  for $i = 1$ *to* N **do**

    $\boldsymbol{x^{adv}}_i^{(l+1)} = \prod_{B_\epsilon}\left(\boldsymbol{x^{adv}}_i^{(l)} + \tau * l_p\big(\phi(\boldsymbol{x^{adv}}_i^{(l)})\big)\right)$;

    where $\phi(\boldsymbol{x^{adv}}) = \frac{1}{N}\sum_{j=1}^{N}\Big[\mathcal{K}(\boldsymbol{x^{adv}}_j^{(l)}, \boldsymbol{x^{adv}})\nabla_{\boldsymbol{x^{adv}}_j^{(l)}} \log P(\boldsymbol{x^{adv}}_j^{(l)}|\boldsymbol{x}) + \nabla_{\boldsymbol{x^{adv}}_j^{(l)}}\mathcal{K}(\boldsymbol{x^{adv}}_j^{(l)}, \boldsymbol{x^{adv}})\Big]$;

return $\{\boldsymbol{x^{adv}}_1^{(L)}, \boldsymbol{x^{adv}}_2^{(L)}, \ldots, \boldsymbol{x^{adv}}_N^{(L)}\}$;

Output: Set of adversarial particles $\boldsymbol{x^{adv}}_1^{(L)}, \boldsymbol{x^{adv}}_2^{(L)}, \ldots, \boldsymbol{x^{adv}}_N^{(L)}\} \sim P_{\theta_T}(\cdot|\boldsymbol{x})$

with respect to the student parameters θ_S at a position $\boldsymbol{x}$ with label y:

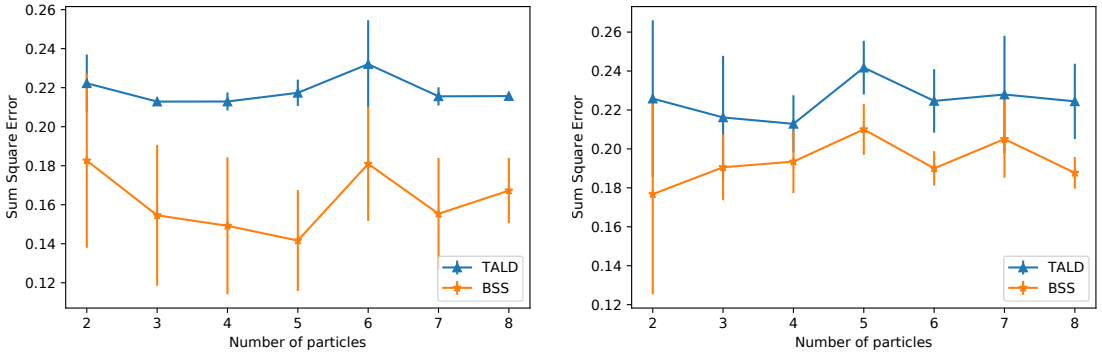
$$\ell_{TALD} := \min_{\theta_S} \mathbb{E}_{\boldsymbol{x}^{adv} \sim P_\theta(\cdot|\boldsymbol{x})} \Big[\|\ell_{CE}(T(\boldsymbol{x}^{adv}), y) - \ell_{CE}(S(\boldsymbol{x}^{adv}), y)\|_2^2\Big], \tag{7.3}$$

where ℓ_{CE} is the cross-entropy loss function.

For each $\boldsymbol{x}$, SVGD samples N adversarial particles from the high density areas of $P_{\theta_T}(\cdot|\boldsymbol{x})$ to sufficiently explore the teacher decision boundaries, while the vanilla attacking methods [5, 23, 26, 32] only generate one adversarial example. We use these adversarial particles to regularise the decision boundaries of the student by matching the cross-entropy loss between the teacher and student model, as shown in Equation 7.3.

Here we show how to apply TALD regularisation to existing knowledge distillation methods. The adversarial particles are generated with the differentiable teacher model and do not need additional learnable modules, as shown in Alg. 7.1. Therefore, we can easily combine to existing knowledge distillation losses, as shown in Equation 7.4.

$$\min_{\theta_S} \mathbb{E}_{(\boldsymbol{x},y)\sim P_{\mathbb{D}}}\Big[\ell_S + \ell_{KD} + \ell_{AM} + \lambda\ell_{TALD}\Big], \tag{7.4}$$



(A) resnet32x4 (B) wrn-40-2

FIGURE 7.1: Diversity comparison of our method and BSS with random initialisation using the sum of square error (SSE) using the pre-trained (a) resnet32x4 and (b) wrn-40-2 architectures. The figure illustrates the average of mean (point) and standard deviation (bar) of the three different inputs from CIFAR-100.

where ℓ_S is the student cross-entropy loss $\ell_{CE}(S(\boldsymbol{x}), y)$. ℓ_{KD} is the original knowledge distillation loss proposed by Hinton et al. [3]. ℓ_{AM} can be an additional loss from other existing methods such as FitNet [35], CRD [10], etc.. λ is the weighted loss hyper-parameter[1].

7.4 Experiments

In this section, we conduct various experiments on CIFAR-100 [7] and ImageNet [231]. In Section 7.4.1, we compare the diversity between the adversarial particles generated by our method and adversarial examples from BSS with random initialisation. We then show that TALD regularisation can improve the performance of many existing methods in Section 7.4.2 and 7.4.3. We evaluate decision boundary similarity between the teacher and student in Section 7.4.4. The effect of the number particles to the student is studied in Section 7.4.5.

7.4.1 Diversity of Teacher Adversarial Particles vs. BSS Random Initialisation

Setting We use pre-trained classifiers (e.g., resnet32x4 and wrn-40-1 architecture) on CIFAR-100 in this experiment. All pre-trained models are fixed. BSS [5] is an attack method that can generate one adversarial example at one run. We run BSS multiple

[1]Note that we ignore other weighted loss hyper-parameters.

times with random initialisation to generate adversarial examples compared to the adversarial particles using our method. We set the same radius ball constraint ϵ, updating step η, and uniform noise factor τ initialisation. Note that all adversarial examples and particles fool the classifiers.

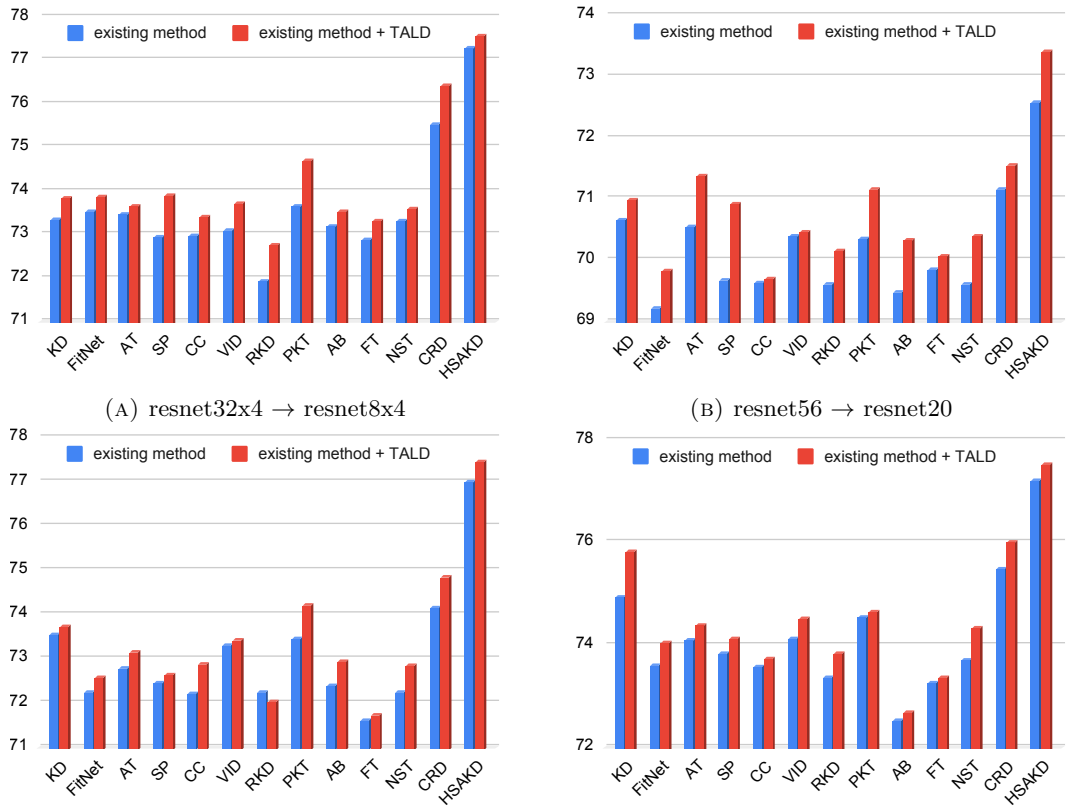
Experimental setup We randomly select three images from CIFAR-100 dataset. Given these inputs, we generate adversarial examples using BSS with random initialisations. The adversarial particles are generated by our method with different numbers of particles, as shown in Figure 7.1. We then calculate the sum squared error (SSE) between these particles to evaluate their diversity. At each setting of the number of particles, we calculate the average of the means and standard deviations of SSE. We set $\epsilon = 0.3$, η =1.0, L=200, τ=0.01.

Result Note that the advantages are illustrated in the Alg. 7.1 where the first and the second term of SVGD can sample in the high density areas and enforce diverse adversarial particles from the local distribution, respectively. Previous attack methods [5, 23, 26, 32] can generate multiple adversarial examples using random initialisation but it can lie together and lose diversity [180]. Therefore, the adversarial particles from our method are diverse. In Figure 7.1, our method has bigger SSE compared to BSS with random initialisation because generated samples are more diverse.

7.4.2 TALD Regularisation with Existing Methods on CIFAR-100

Setting In this experiment, we evaluate TALD regularisation on model compression of a large network (teacher T) to a smaller one (student S). We use CIFAR-100 [7], which contains 50K training images with 500 images per class and 10K test images. We apply our TALD regularisation to improve the performance of existing methods using CIFAR-100. The existing methods is implemented from RepDistiller[2] and HSAKD[3] repositories. Our regularisation is combined with these existing methods without changing parameter settings on CIFAR-100. We set the radius constraint $\epsilon = 0.3$, number of particles $N = 4$, and $\lambda = 0.01$. For CIFAR-100, we set $\epsilon = 0.3$, η=10.0, L=1, N=4, τ=0.1, λ=0.01. All methods used in our experiments are trained by SGD. The learning rate is initialised as 0.05, and decayed it by 0.1 every 30 epochs after the first 150 epochs until the last 240

[2]https://github.com/HobbitLong/RepDistiller
[3]https://github.com/winycg/HSAKD



(A) resnet32x4 → resnet8x4

(B) resnet56 → resnet20

(C) wrn-40-2 → wrn-40-1

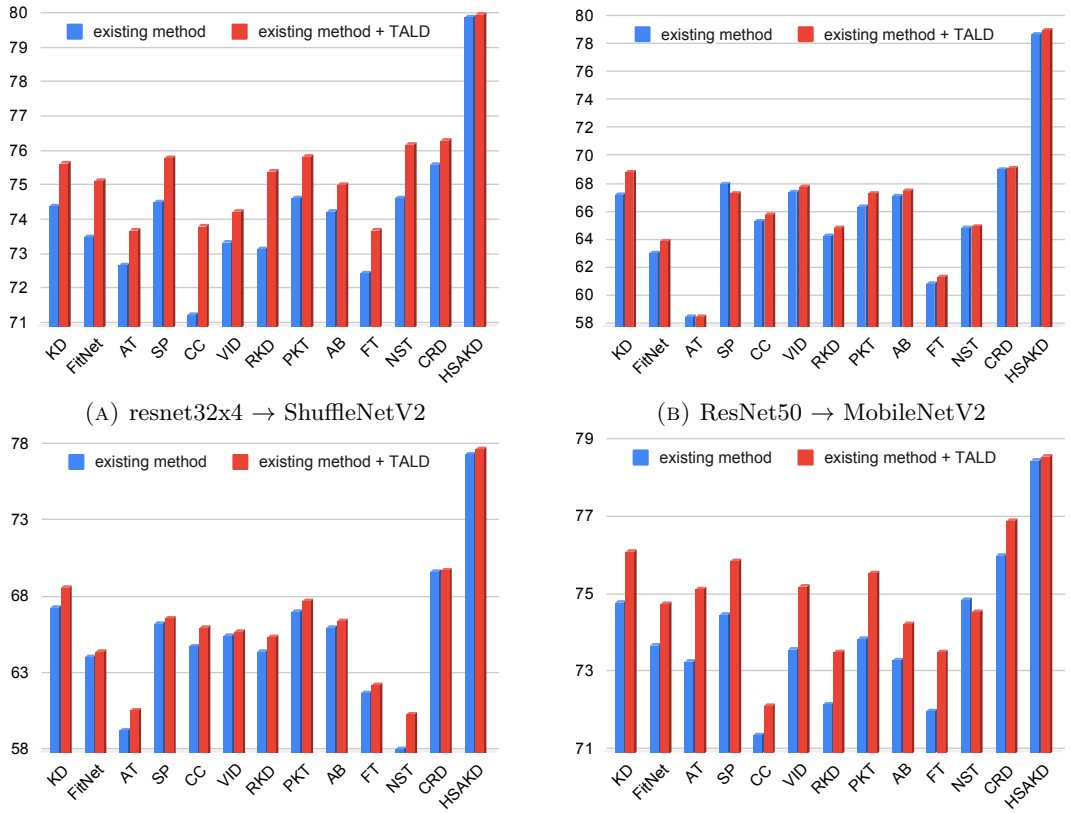
(D) wrn-40-2 → wrn-40-1

FIGURE 7.2: Test accuracy (%) of student networks on CIFAR-100 of a number of distillation methods from teacher to student (teacher → student). *Existing method* denotes a previous distillation method, while *existing method + TALD* is a combination of the respective existing method and our regularisation. All student accuracies of existing methods are used from [10, 11].

epochs. We use a learning rate of 0.01 for MobileNetV2, ShuffleNetV1 and ShuffleNetV2, while 0.05 is optimal for other models. Batch size is 64.

Experimental setup The goal of knowledge distillation is to improve performance of the student S by using the teacher knowledge. In this experiment, all teacher T models are pre-trained on CIFAR-100 and fixed. The accuracy of all models trained on CIFAR-100 with only ℓ_S is shown in Table 7.1. The TALD regularisation is intensively evaluated on many existing methods such as KD [3], FitNet [35], AT [219], SP [120], CC [121], VID [112], RKD [116], PKT [105], AB [5], FT [106], NST [104], CRD [10], BSS [5], and HSAKD[11]. We set up various teacher-student neural network architectures for the same architecture style (Figure 7.2) and across-architecture style (Figure 7.3) knowledge distillation settings. BSS[5] is an attack proposed for the knowledge distillation task. Therefore, we compare BSS with KD to our TALD with KD[3], as shown in Figure 7.4.

Result Recall that the proposed method is an additional regularisation loss, which can



(A) resnet32x4 → ShuffleNetV2

(B) ResNet50 → MobileNetV2

(C) vgg13 → MobileNetV2

(D) wrn-40-2 → ShuffleNetV1

FIGURE 7.3: Test accuracy (%) of student networks on CIFAR-100 of a number of distillation methods for transfer across very different teacher to student architectures (teacher → student). *Existing method* denotes a previous distillation method without TALD regularisation term, while *existing method + TALD* is a combination of the respective existing method and our regularisation. All student accuracies of existing methods are used from [10, 11].

combine with many existing methods. Our regularisation explores the teacher decision boundaries using the teacher adversarial particles, then enforces decision boundary matching between the teacher and student loss. In Figure 7.2, the teacher and student are from the same architectural style. When adding our TALD loss, we consistently improve test accuracy. In the context of transfer across very different teacher and student, we also increase the performance of existing methods in Figure 7.3. Additionally, our method (KD+TALD) outperforms the adversarial approach BSS [5] for distillation (KD+BSS) shown in Figure 7.4.

7.4.3 TALD Regularisation with Existing Methods on ImageNet

Setting In this experiment, TALD regularisation is evaluated on a large-scale ImageNet [231] dataset (1.2 million for training and 50K for validation images with 1K

Architecture	Accuracy (%)
wrn-40-2	75.61
wrn-40-1	71.98
wrn-16-2	73.26
resnet56	72.34
resnet20	69.06
resnet32x4	79.42
resnet8x4	72.5
ShuffleNetV1	70.5
MobileNetV2	64.6
ResNet50	79.34

TABLE 7.1: Test accuracy (%) of different pre-trained model architectures on CIFAR-100. Note that all test accuracies are used from [10, 11].

classes). We adopt the implementation of existing methods from Torchdistill[4][8], ResNet-34 as the teacher and ResNet-18 as the student. The ResNet-34 and ResNet-18 architectures are released by the PyTorch team. We keep all original settings of [8] and set our TALD following $\lambda = 0.001$, number of particles $N = 4$, and the radius constraint $\epsilon = 0.3$. For ImageNet, we use settings from config files of Torchdistill[5]. We set $\epsilon = 0.3$, η =5.0, L=1, N=4, τ=0.2, λ=0.01.

Experiment setup We illustrate the performance of TALD regularisation by compressing the teacher ResNet-34 to the student ResNet-18. The teacher is fixed and pre-trained with 73.31% accuracy. The base student trained on ImageNet without distillation methods achieves 69.75% accuracy. We combine our method to improve the implemented existing methods such as KD [3], AT [219], FT [106], CRD [10], and SSKD [123].

Result We calculate the accuracy of students on 50K validation images. In Figure 7.5, the all student accuracies are used from the implementation of the Torchdistill repository [8]. As can be seen, our TALD regularisation can consistently improve the accuracy of the ResNet-18 student on top of respecting existing methods such as KD [3], AT [219], FT [106], CRD [10], and SSKD [123].

[4]https://github.com/yoshitomo-matsubara/torchdistill
[5]github.com/yoshitomo-matsubara/torchdistill/tree/main/torchdistill

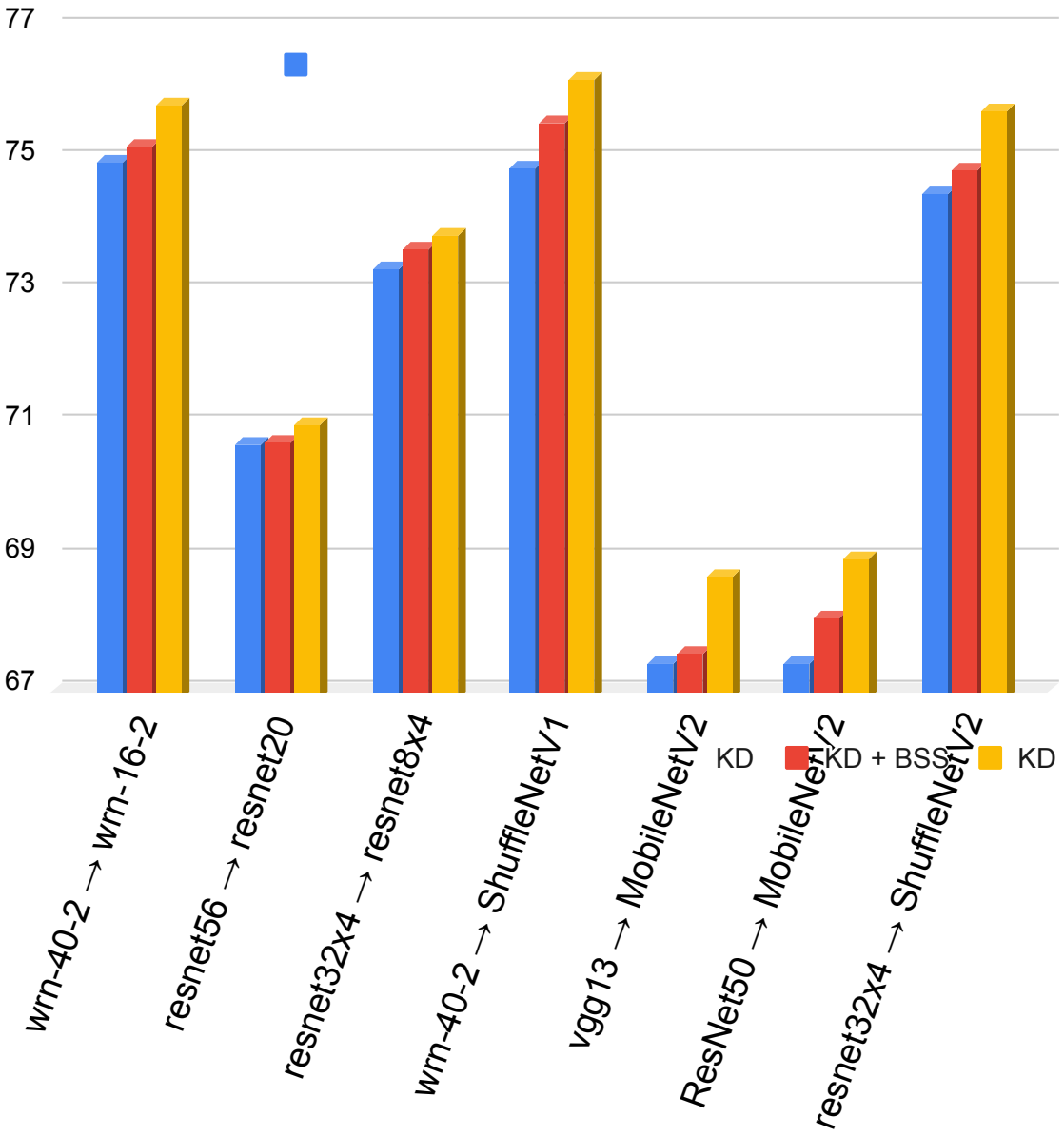


FIGURE 7.4: Test accuracy (%) of student networks on CIFAR-100 of KD, KD +BSS, and KD + TALD for transfer various teacher and student architectures (teacher → student).

7.4.4 Decision Boundary Similarity Evaluation

Metrics for similarity of decision boundaries To verify our TALD regularisation, we use metrics proposed by Heo et al. [5] to measure the similarity between the decision boundaries of two classifiers (e.g., teacher and student in the knowledge distillation task). The metrics are calculated using BSS attack. For each data point $\boldsymbol{x}$, BSS attacks the teacher and student to generate teacher $\boldsymbol{x}^T_{adv}$ and student $\boldsymbol{x}^S_{adv}$ adversarial example, respectively. We then obtain the perturbation vector of teacher ($\boldsymbol{v}^T = \boldsymbol{x}^T_{adv} - \boldsymbol{x}$) and student ($\boldsymbol{v}^S = \boldsymbol{x}^S_{adv} - \boldsymbol{x}$). Since the perturbation vector is obtained by the attacking path

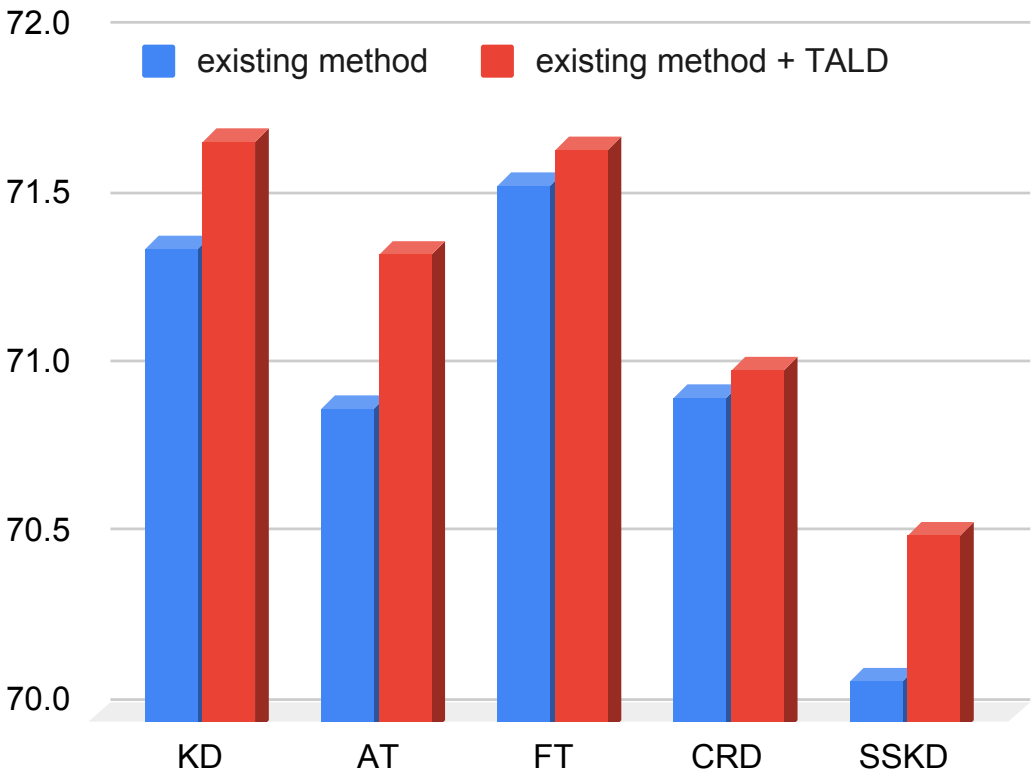
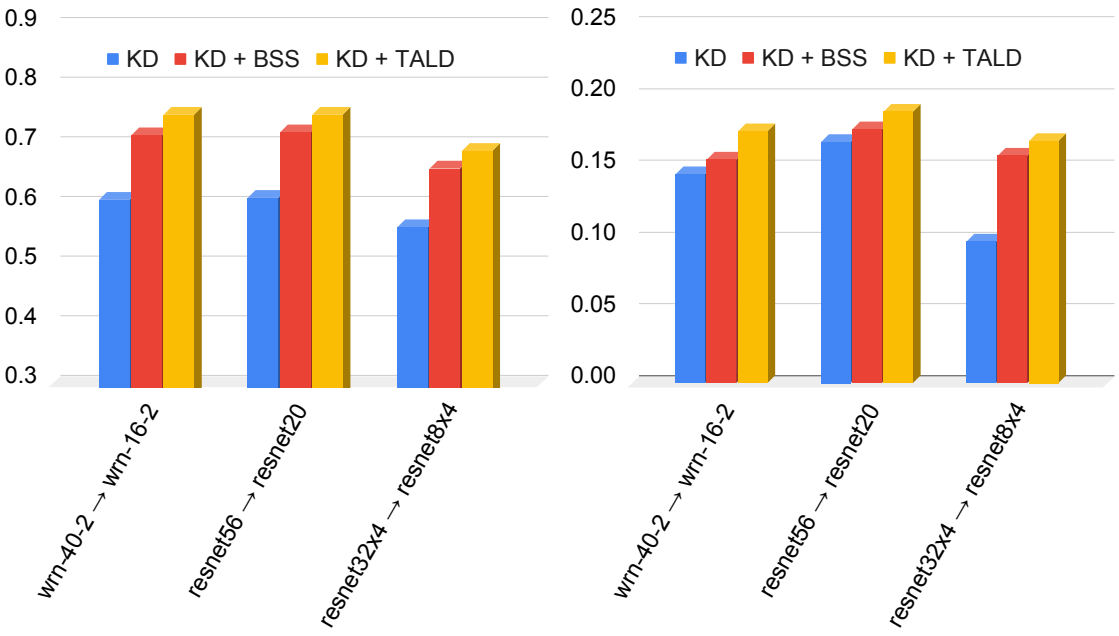


FIGURE 7.5: Accuracy (%) of ResNet-18 student on validation ImageNet dataset (ResNet-34 → ResNet-18). All student accuracies of existing methods are used from [8].

from natural sample $\boldsymbol{x}$ to model decision boundaries, we compare the *Magnitude Similarity (MagSim)* and *Angle similarity (AngSim)* of the two vectors. *MagSim* represents the similarity with respect to the distance from the natural sample $\boldsymbol{x}$ to the decision boundary, while *AngSim* reflects it with respect to the path direction from the natural sample $\boldsymbol{x}$ to the decision boundary. These two metrics have values in the range of [0,1] and higher values represent more similar decision boundaries. Please refer to the work [5] for more information.

Setup We use pre-trained teachers and distilled students using CIFAR-100 from Section 7.4.2. Our baseline is KD without regularisation. Our method is compared to KD + BSS, which uses adversarial examples to support student decision boundaries. We calculate *MagSim* and *AngSim*, as shown in Figure 7.6.

Result Recall that KD method does not have decision boundary regularisation, while KD + BSS insufficiently explores the teacher perturbations [180]. In Figure 7.6, our KD + TALD method can consistently improve decision boundary matching based on



(A) MagSim (B) AngSim

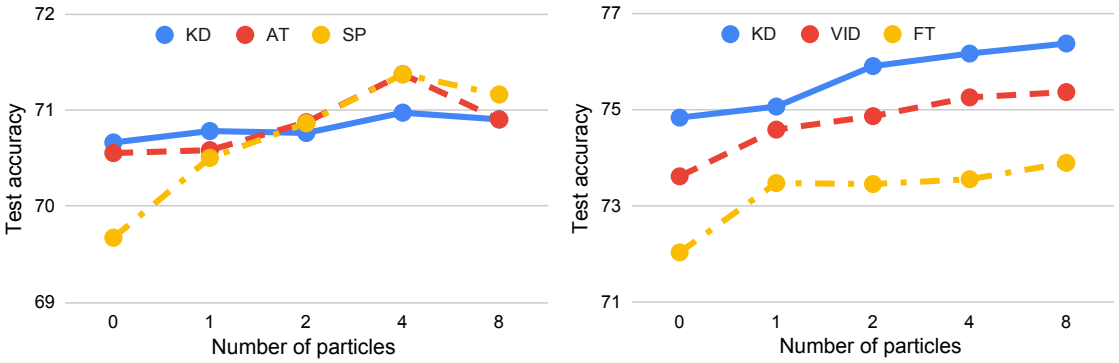
FIGURE 7.6: Evaluation on decision boundary similarity between teacher and student (teacher → student) using *Magnitude Similarity (MagSim)* and *Angle similarity (AngSim)*. These two metrics have values in the range of [0,1] and higher values represent more similar decision boundaries.

MagSim and *AngSim* metrics with various architectures such as wrn-40-2 → wrn-16-2, resnet56 → resnet20, and resnet32x4 → resnet8x4.

7.4.5 Teacher Adversarial Particle Analysis

Setting We study the number of teacher adversarial particles that affect the performance of the student on CIFAR-100. We perform the model compression task (teacher → student) on the same architecture style (teacher: resnet56 → student: resnet20) and very different architecture style (teacher: wrn-40-2 → student: ShuffleNetV1) with different knowledge distillation methods. The implementation adopts the RepDistill, and all parameters are kept similar to Section 7.4.2 settings except the number of particles N.

Experiment setup We change the number of teacher adversarial particles N in {0, 1, 2, 4, 8}. When $N = 0$ implies that we do not use the TALD regularisation. We study our method using different knowledge distillation methods with different N such as KD [3], AT [219] and SP [120] for resnet56 → resnet20, and KD [3], VID [112] and FT [106] for wrn-40-2 → ShuffleNetV1.



(A) resnet56 → resnet20 (B) wrn-40-2 → ShuffleNetV1

FIGURE 7.7: Test accuracy (%) of the students when distilling from teacher to student (teacher → student) at different number of teacher adversarial particles $N \in \{0, 1, 2, 4, 8\}$. When $N = 0$ implies that we do not use TALD regularisation.

Result Note that we approximate the teacher adversarial local distribution $P_{\theta_T}(\cdot|\boldsymbol{x})$ using the particles. Thus, by increasing the number of particles, we accordingly increase the regularisation strength of the student model. Figure 7.7 shows that the test accuracy can be improved by increasing N from 0 to 4. It is as expected that over regularisation may hurt the performance when $N = 8$ on Figure 7.7(a). However, our method can still outperform existing methods without TALD regularisation ($N = 0$) in these cases.

7.4.6 Running Time Analysis

We agree that our method needs a tradeoff between training time and student performance. By using a few adversarial particles, we improve the student performance at the cost of increasing the training time. However, the student inference time remains unchanged compared to existing KD methods because we do not use these particles during the student inference stage. In other words, our method aligns with the major aims of KD (e.g., reduce inference time and increase the performance of the student). We also show the training time comparison between BSS and ours in Table 7.2.

BSS (Minutes)	Our method (Minutes)			
	K=1	K=2	K=4	K=8
19.1	22.01	26.35	32.35	49.01

TABLE 7.2: Running time per epoch on CIFAR-100.

7.5 Conclusion

In this chapter, we have introduced a novel teacher adversarial local distribution (TALD) regularisation that can adapt well to improve on many existing methods such as KD [3], FitNet [35], AT [219], SP [120], CC [121], VID [112], RKD [116], PKT [105], AB [5], FT [106], NST [104], CRD [10],and HSAKD[11]. In the proposed method, we form the teacher adversarial local distribution for exploring the teacher's properties (e.g., decision boundaries). Our strategy uses SVGD to estimate the adversarial local distribution using more diverse adversarial particles. We intensively conduct experiments on CIFAR-100 and ImageNet where the TALD consistently improves the performance of many existing knowledge distillation methods. By using a few adversarial particles, we improve the student performance at the cost of increasing the training time. In the future, we would like to reduce the TALD running time and use the targeted attack perspective using TALD.

Chapter 8

Conclusion and Future Work

8.1 Contributions

The first objective of this thesis is to develop a comprehensive understanding of adversarial attacks and knowledge distillation techniques. Chapters 3 and 4 aim to provide valuable interpretations and insights into both adversarial attacks and knowledge distillation.

- In Chapter 3, the task of attacking with fewer perturbations was addressed in a novel way. The approach involved refining existing dense attacks by reducing the number of perturbations. The inspiration for this idea came from the observation that carefully selecting perturbations based on the vulnerability map of a natural image can enhance the attack. To achieve this, a probabilistic post-hoc framework was proposed, utilising a U-Net to learn the vulnerability map and selecting perturbations from source attacks based on the map. The framework was trained using mutual information maximisation. The method demonstrated the ability to refine any dense attack by removing approximately 70% of its perturbations while maintaining the same attack power. The resulting refined attacks were found to be significantly less detectable and closer to their natural counterparts.

- In Chapter 4, we introduced MED-TEX, our innovative framework that combines knowledge distillation and model interpretation. MED-TEX enables the training of a significantly smaller student model compared to the teacher model by leveraging

the knowledge from the pre-trained teacher. The framework utilises an explainer model to highlight important image areas for the teacher's predictions, assisting the student in learning from the teacher's knowledge with less data. The explainer's output can also serve as low-level strong annotations trained using high-level weak annotations from the teacher. To train the framework, we propose maximising the mutual information between the intermediate and output layers of the student and teacher, creating a novel training objective. Our experimental results demonstrate that MED-TEX surpasses several widely-used knowledge distillation and model interpretation techniques in performance on the Fundus dataset in terms of both quantitative and qualitative evaluations.

Furthermore, the thesis seeks to enhance adversarial regularisation and improve the efficiency of the knowledge distillation process. This exploration includes investigating the potential of adversarial regularisation and knowledge distillation in robust deep learning, deep semi-supervised learning, and model compression tasks, covered in Chapters 5, 6, and 7.

- In Chapter 5, we proposed a novel adversarial local distribution regularisation technique that improves previous methods, including FGSM, PGD, TRADES, and VAT. Our method utilises SVGD to approximate the adversarial local distribution, employing a diverse set of adversarial particles. This approach is applicable to various domains that require improved generalisation, such as semi-supervised learning and robust machine learning. Through comprehensive experiments, we demonstrate that our method consistently outperforms widely-used regularisation approaches, including PGD, TRADES, and ADT in robust machine learning, as well as VAT in semi-supervised learning.

- In Chapter 6, we introduced a novel regularisation technique called Cross-Adversarial Local Distribution (Cross-ALD), which addresses the limitations of VAT and Mixup methods. Our approach utilises SVGDF to approximate Cross-ALD, generating more diverse adversarial particles compared to vanilla SVGD and VAT with random initialisation. We apply Cross-ALD to semi-supervised medical image segmentation, achieving state-of-the-art performance on the ACDC and LA datasets. Our results surpass many recent methods, including VAT [28], UA-MT [206], SASSNet [203], DTC [34], URPC [204], MC-Net [33], and SS-Net [9].

- In Chapter 7, we constructed the teacher's adversarial local distribution (TALD) to investigate its properties, particularly decision boundaries. Our approach employs SVGD to estimate the TALD, leveraging a diverse set of adversarial particles. Extensive experiments on CIFAR-100 and ImageNet demonstrate the consistent performance improvement of many existing knowledge distillation methods through the use of TALD. However, it should be noted that incorporating a few adversarial particles increases training time.

8.2 Future Research

Overall, the thesis provides an exploration of adversarial regularisation and knowledge distillation, focusing on their applications in robust machine learning, semi-supervised learning, and model compression tasks. This work substantially contributes to this rapidly expanding and vital field of research. It is indeed possible to conceive future research prospects. Here, we outline two particularly promising directions for future exploration.

Various types of adversarial attacks exist, each with its unique objective and approach. Poisoning attacks, for instance, aim to manipulate a model during its training phase, affecting its behaviour when deployed in real-world scenarios [232, 233]. While the poisoning attacks aim to manipulate the model to yield biased or inaccurate predictions, backdoor attacks follow a different objective: they seek to secretly embed a backdoor within the model that can be triggered by a specific input pattern [234, 235]. Another type of attack, the model extraction attacks, also known as model stealing, represents a security threat where the attacker seeks to clone or extract a target model without direct access to the model itself [236, 237]. Lastly, privacy attacks attempt to draw out sensitive information from a target model, posing a severe risk to privacy [238, 239]. Consequently, we could develop and enhance the techniques outlined in Chapters 3 and 4 to better understand and counter these diverse adversarial attacks.

The implementation of our adversarial regularisation makes use of SVGD. This implies the necessity of developing and exploring different variants within the SVGD family. For instance, the work SVGD as Gradient Flow offers a fresh viewpoint on SVGD, framing it

as a gradient flow in the functional space of probability distributions [240]. Stein Variational Adaptive Importance Sampling (SVAIS) further evolves SVGD to cover adaptive importance sampling [241]. Furthermore, our adversarial regularisation is designed for the image domain, thereby demanding a substantial computational effort and prolonged training duration. Recently, SAM [242] tackles the issue of loss landscape smoothness linked with generalisation by introducing minor perturbations at the level of the model parameters instead of the image level. Consequently, we have the potential to broaden our adversarial regularisation approach to involve adjustments at the level of the model parameters.

Bibliography

[1] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of CVPR*, pages 2921–2929, 2016.

[2] Jianping Gou, Baosheng Yu, Stephen John Maybank, and Dacheng Tao. Knowledge distillation: A survey. *arXiv preprint arXiv:2006.05525*, 2020.

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[4] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27, 2014.

[5] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of AAAI*, volume 33, pages 3779–3787, 2019.

[6] Yann LeCun and Corrina Cortes. The MNIST database of handwritten digits. 1998.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[8] Yoshitomo Matsubara. torchdistill: A modular, configuration-driven framework for knowledge distillation. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 24–44. Springer, 2021.

[9] Yicheng Wu, Zhonghua Wu, Qianyi Wu, Zongyuan Ge, and Jianfei Cai. Exploring smoothness and class-separation for semi-supervised medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 13435, pages 34–43. Springer, Cham, 2022.

[10] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

[11] Linhang Cai Chuanguang Yang, Zhulin An and Yongjun Xu. Hierarchical self-supervised augmented knowledge distillation. In *Proceedings of IJCAI*, pages 1217–1223, 2021.

[12] Jayashree Padmanabhan and J Melvin Jose Premkumar. Advanced deep neural networks for pattern recognition: An experimental study. In *Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016)*, pages 166–175. Springer, 2018.

[13] Kyongsik Yun, Alexander Huyen, and Thomas Lu. Deep neural networks for pattern recognition. *arXiv preprint arXiv:1809.09645*, 2018.

[14] Qingchen Zhang, Laurence T Yang, Zhikui Chen, and Peng Li. A survey on deep learning for big data. *Information Fusion*, 42:146–157, 2018.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778, 2016.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Amirsina Torfi, Rouzbeh A Shirvani, Yaser Keneshloo, Nader Tavaf, and Edward A Fox. Natural language processing advancements by deep learning: A survey. *arXiv preprint arXiv:2003.01200*, 2020.

[19] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, 2015.

[21] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[22] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *Proceeding ICLR*, 2014.

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of ICLR*, 2018.

[24] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Proceedings of NeurIPS*, volume 32, 2019.

[25] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of CVPR*, pages 501–509, 2019.

[26] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of ICML*, pages 7472–7482. PMLR, 2019.

[27] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.

[28] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE TPAMI*, 41(8):1979–1993, 2018.

[29] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[30] Seung Wook Kim and Hyo-Eun Kim. Transferring knowledge to smaller network with class-distance loss. In *Proceeding of ICLR*, 2017.

[31] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of CVPR*, pages 11953–11962, 2022.

[32] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceeding of ICML*, pages 2206–2216. PMLR, 2020.

[33] Yicheng Wu, Minfeng Xu, Zongyuan Ge, Jianfei Cai, and Lei Zhang. Semi-supervised left atrium segmentation with mutual consistency training. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 297–306. Springer, 2021.

[34] Xiangde Luo, Jieneng Chen, Tao Song, and Guotai Wang. Semi-supervised medical image segmentation through dual-task consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8801–8809, 2021.

[35] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[36] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[37] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[38] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[39] Emily Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430*, 2016.

[40] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[41] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems*, 30, 2017.

[42] Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki, Jingdong Wang, and Xian-Sheng Hua. Global versus localized generative adversarial nets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1517–1525, 2018.

[43] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[44] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[45] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. *Advances in neural information processing systems*, 30, 2017.

[46] Chongxuan Li, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. *Advances in neural information processing systems*, 30, 2017.

[47] Si Wu, Guangchang Deng, Jichang Li, Rui Li, Zhiwen Yu, and Hau-San Wong. Enhancing triplegan for semi-supervised conditional instance synthesis and classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10091–10100, 2019.

[48] Zhe Gan, Liqun Chen, Weiyao Wang, Yuchen Pu, Yizhe Zhang, Hao Liu, Chunyuan Li, and Lawrence Carin. Triangle generative adversarial networks. *Advances in neural information processing systems*, 30, 2017.

[49] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31, 2018.

[50] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14, 2001.

[51] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[52] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28, 2015.

[53] Mohammad Pezeshki, Linxi Fan, Philemon Brakel, Aaron Courville, and Yoshua Bengio. Deconstructing the ladder network architecture. In *International conference on machine learning*, pages 2368–2376. PMLR, 2016.

[54] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29, 2016.

[55] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[56] Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson WH Lau. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6728–6736, 2019.

[57] Zhi-Hua Zhou and Ming Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24:415–439, 2010.

[58] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[59] Yanhua Cheng, Xin Zhao, Rui Cai, Zhiwei Li, Kaiqi Huang, Yong Rui, et al. Semi-supervised multimodal deep learning for rgb-d object recognition. In *IJCAI*, pages 3345–3351, 2016.

[60] Rui Xia, Cheng Wang, Xinyu Dai, and Tao Li. Co-training for semi-supervised sentiment classification based on dual-view bags-of-words representation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1054–1063, 2015.

[61] W Dong-DongChen and ZH WeiGao. Tri-net for semi-supervised deep learning. In *Proceedings of twenty-seventh international joint conference on artificial intelligence*, pages 2014–2020, 2018.

[62] Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.

[63] Leo Breiman. Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40:229–242, 2000.

[64] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

[65] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Proceedings of ICML workshop*, volume 3, page 896, 2013.

[66] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[67] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR, 2019.

[68] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[69] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11557–11568, 2021.

[70] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[71] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[72] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[73] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145:90–106, 2022.

[74] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

[75] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

[76] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[77] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[78] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096, 2019.

[79] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep

networks via gradient-based localization. In *Proceedings of ICCV*, pages 618–626. IEEE, 2017.

[80] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *Proceedings of WACV*, pages 839–847. IEEE, 2018.

[81] Haofan Wang and Zifan Wang. Mengnan du, fan yang, zijian zhang, sirui ding, piotr mardziel, and xia hu. score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, volume 1, page 3, 2020.

[82] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224*, 2019.

[83] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

[84] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[85] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[86] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[87] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[88] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceeding of ICML*, pages 2048–2057. PMLR, 2015.

[89] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip Torr. Learn to pay attention. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyzbhfWRW.

[90] Tianyi Zhang, Guosheng Lin, Weide Liu, Jianfei Cai, and Alex Kot. Splitting vs. merging: Mining object regions with discrepancy and intersection loss for weakly supervised semantic segmentation. In *Proceeding of EECV*, 2020.

[91] Tianyi Zhang, Guosheng Lin, Jianfei Cai, Tong Shen, Chunhua Shen, and Alex C Kot. Decoupled spatial neural attention for weakly supervised semantic segmentation. *Transactions on Multimedia*, 21(11):2930–2941, 2019.

[92] Heechan Yang, Ji-Ye Kim, Hyongsuk Kim, and Shyam P Adhikari. Guided soft attention network for classification of breast cancer histopathology images. *Transactions on Medical Imaging*, 39(5):1306–1315, 2019.

[93] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[94] Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *Proceedings of NIPS*, pages 3086–3094, 2014.

[95] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[96] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[97] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of SIGKDD*, pages 1135–1144, 2016.

[98] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceeding of ICML*, pages 3145–3153. PMLR, 2017.

[99] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[100] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceeding of ICML*, pages 883–892. PMLR, 2018.

[101] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017.

[102] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3517–3526, 2019.

[103] Nikos Komodakis and Sergey Zagoruyko. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *Proceeding of ICLR*, 2017.

[104] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

[105] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of ECCV*, pages 268–284, 2018.

[106] Jangho Kim, SeoungUK Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *arXiv preprint arXiv:1802.04977*, 2018.

[107] Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. In *Proceedings of ICCV*, pages 1345–1354, 2019.

[108] Guorui Zhou, Ying Fan, Runpeng Cui, Weijie Bian, Xiaoqiang Zhu, and Kun Gai. Rocket launching: A universal and efficient framework for training well-performing light net. In *Proceedings of the AAAI*, volume 32, 2018.

[109] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI*, 2020.

[110] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of CVPR*, pages 4133–4141, 2017.

[111] Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. Self-supervised knowledge distillation using singular value decomposition. In *Proceedings of ECCV)*, pages 335–350, 2018.

[112] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of CVPR*, pages 9163–9171, 2019.

[113] Seunghyun Lee and Byung Cheol Song. Graph-based knowledge distillation by multi-head attention network. *arXiv preprint arXiv:1907.02226*, 2019.

[114] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2339–2348, 2020.

[115] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1285–1294, 2017.

[116] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[117] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7096–7104, 2019.

[118] Hanting Chen, Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Learning student networks via feature embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[119] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Probabilistic knowledge transfer for lightweight deep representation learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[120] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of ICCV*, pages 1365–1374, 2019.

[121] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of ICCV*, pages 5007–5016, 2019.

[122] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *Proceeding of ICLR*, 2020. URL https://openreview.net/forum?id=SkgpBJrtvS.

[123] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. In *Proceeding of ECCV*, pages 588–604. Springer, 2020.

[124] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[125] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *Proceeding of ICLR*, 2017.

[126] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[127] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019.

[128] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[129] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33: 16048–16059, 2020.

[130] Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. Adversarial dropout for supervised and semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[131] Ting-Wu Chin, Cha Zhang, and Diana Marculescu. Renofeation: A simple transfer learning method for improved adversarial robustness. In *Proceedings of CVPR*, pages 3243–3252, 2021.

[132] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceeding of CVPR*, pages 427–436, 2015.

[133] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

[134] Akshay Chaturvedi and Utpal Garain. Mimic and fool: A task-agnostic adversarial attack. *IEEE Transactions on Neural Networks and Learning Systems*, 32(4):1801–1808, 2020.

[135] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360, 2020.

[136] Yanbo Fan, Baoyuan Wu, Tuanhui Li, Yong Zhang, Mingyang Li, Zhifeng Li, and Yujiu Yang. Sparse adversarial attack via perturbation factorization. In *Proceeding of ECCV*, 2020.

[137] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of CVPR*, pages 2574–2582, 2016.

[138] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[139] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In *Proceeding of IJCAI*, pages 3905–3911, 2018.

[140] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceeding of ICML*, pages 274–283, 2018.

[141] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European symposium on security and privacy*, pages 372–387, 2016.

[142] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*, 2016.

[143] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on mnist. In *Proceeding of ICLR*, 2018.

[144] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[145] Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: A few pixels make a big difference. In *Proceeding of CVPR*, pages 9087–9096, 2019.

[146] Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *Proceeding of ICCV*, pages 4724–4732, 2019.

[147] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[148] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[149] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[150] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

[151] Tianyu Pang, Chao Du, Yinpeng Dong, and Jun Zhu. Towards robust detection of adversarial examples. In *Procding of NeurIPS*, pages 4579–4589, 2018.

[152] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, page 107332, 2020.

[153] Akshay Agarwal, Gaurav Goswami, Mayank Vatsa, Richa Singh, and Nalini K Ratha. Damad: Database, attack, and model agnostic adversarial perturbation detector. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[154] Pu Zhao, Sijia Liu, Yanzhi Wang, and Xue Lin. An admm-based universal framework for adversarial attacks on deep neural networks. In *ACMMM*, pages 1065–1073, 2018.

[155] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[156] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with Gumbel-softmax. In *Proceeding of ICLR*, 2017.

[157] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceeding of MICCAI*, pages 234–241. Springer, 2015.

[158] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[159] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[160] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.

[161] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of CVPR*, pages 4700–4708, 2017.

[162] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017.

[163] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[164] Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *COLT*, pages 1246–1257, 2016.

[165] Chen Liu, Mathieu Salzmann, Tao Lin, Ryota Tomioka, and Sabine Süsstrunk. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. *NeurIPS*, 33:21476–21487, 2020.

[166] Tran Minh Quan, Thanh Nguyen-Duc, and Won-Ki Jeong. Compressed sensing mri reconstruction using a generative adversarial network with a cyclic loss. *IEEE TMI*, 37(6):1488–1497, 2018.

[167] Nicola Rieke, Jonny Hancox, Wenqi Li, F. Milletari, H. Roth, Shadi Albarqouni, S. Bakas, M. Galtier, B. Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah J. Sheller, R. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Cardoso. The future of digital health with federated learning. *NPJ Digital Medicine*, 3, 2020.

[168] Heng Wang, Donghao Zhang, Yang Song, Siqi Liu, Yue Wang, Dagan Feng, Hanchuan Peng, and Weidong Cai. Segmenting neuronal structure in 3D optical microscope images via knowledge distillation with teacher-student network. In *Proceedings of ISBI*, pages 228–231. IEEE, 2019.

[169] Mohammadhassan Izadyyazdanabadi, Evgenii Belykh, Claudio Cavallo, Xiaochun Zhao, Sirin Gandhi, Leandro Borba Moreira, Jennifer Eschbacher, Peter Nakaji, Mark C Preul, and Yezhou Yang. Weakly-supervised learning-based feature localization for confocal laser endomicroscopy glioma images. In *Proceedings of MICCAI*, pages 300–308. Springer, 2018.

[170] Xinyang Feng, Jie Yang, Andrew F Laine, and Elsa D Angelini. Discriminative localization in CNNs for weakly-supervised segmentation of pulmonary nodules. In *Proceedings of MICCAI*, pages 568–576. Springer, 2017.

[171] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet:

Radiologist-level pneumonia detection on chest X-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[172] Eytan Kats, Jacob Goldberger, and Hayit Greenspan. Soft labeling by distilling anatomical knowledge for improved ms lesion segmentation. In *Proceedings of ISBI*, pages 1563–1566. IEEE, 2019.

[173] Stergios Christodoulidis, Marios Anthimopoulos, Lukas Ebner, Andreas Christe, and Stavroula Mougiakakou. Multisource transfer learning with convolutional neural networks for lung pattern analysis. *IEEE journal of biomedical and health informatics*, 21(1):76–84, 2016.

[174] Q Dou, Q Liu, PA Heng, and B Glocker. Unpaired multi-modal segmentation via knowledge distillation. *Transactions on Medical Imaging*, 39(7), 2020.

[175] Xuhua Ren, Jiayu Huo, Kai Xuan, Dongming Wei, Lichi Zhang, and Qian Wang. Robust brain magnetic resonance image segmentation for hydrocephalus patients: Hard and soft attention. In *Proceedings of ISBI*, pages 385–389. IEEE, 2020.

[176] Thomas M Cover and Joy A Thomas. Elements of information theory, 2012.

[177] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[178] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. Benchmarking adversarial robustness on image classification. In *Proceedings of CVPR*, pages 321–331, 2020.

[179] Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Improving the generalization of adversarial training with domain adaptation. *arXiv preprint arXiv:1810.00740*, 2018.

[180] Yusuke Tashiro, Yang Song, and Stefano Ermon. Diversity can be transferred: Output diversification for white-and black-box attacks. *Proceedings of NeurIPS*, 33, 2020.

[181] Yinpeng Dong, Zhijie Deng, Tianyu Pang, Jun Zhu, and Hang Su. Adversarial distributional training for robust deep learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Proceedings of NeurIPS*, volume 33, pages 8270–8283, 2020.

[182] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Machine Learning Research*, 15(1):1929–1958, 2014.

[183] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Proceedings of NeurIPS*, volume 29, 2016.

[184] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[185] Quan Hoang, Trung Le, and Dinh Phung. Parameterized rate-distortion stochastic encoder. In Hal Daumé III and Aarti Singh, editors, *Proceedings of ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 4293–4303. PMLR, 13–18 Jul 2020.

[186] Anh Bui, Trung Le, He Zhao, Paul Montague, Olivier deVel, Tamas Abraham, and Dinh Phung. Improving adversarial robustness by enforcing local and global compactness. In *Proceedings of ECCV*, pages 209–223. Springer, 2020.

[187] Anh Tuan Bui, Trung Le, Quan Hung Tran, He Zhao, and Dinh Phung. A unified wasserstein distributional robustness framework for adversarial training. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Dzpe9C1mpiv.

[188] Trung Le, Anh Bui, Tue Le, He Zhao, Paul Montague, Quan Tran, and Phung Dinh. On global-view based defense via adversarial attack and defense risk guaranteed bounds. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

[189] Anh Bui, Trung Le, He Zhao, Paul Montague, Seyit Camtepe, and Dinh Phung. Understanding and achieving efficient robustness with adversarial supervised contrastive learning. *arXiv preprint arXiv:2101.10027*, 2021.

[190] Anh Tuan Bui, Trung Le, He Zhao, Paul Montague, Olivier deVel, Tamas Abraham, and Dinh Phung. Improving ensemble robustness by collaboratively promoting and demoting adversarial robustness. *Proceedings of AAAI*, 35(8):6831–6839, May 2021.

[191] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

[192] Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. *arXiv preprint arXiv:1907.01003*, 2019.

[193] Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pages 276–284. PMLR, 2016.

[194] Nhan Nguyen Trong Dam. *Modelling Data with Stochastic Generative Processes*. PhD thesis, Monash University.

[195] Charles Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proc. Sixth Berkeley Symp. Math. Stat. Prob.*, pages 583–602, 1972.

[196] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020.

[197] Geoff French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. *arXiv preprint arXiv:1906.01916*, 2019.

[198] Xiaomeng Li, Lequan Yu, Hao Chen, Chi-Wing Fu, Lei Xing, and Pheng-Ann Heng. Transformation-consistent self-ensembling model for semisupervised medical image segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):523–534, 2020.

[199] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020.

[200] Zhonghua Wu, Xiangxi Shi, Guosheng Lin, and Jianfei Cai. Learning meta-class memory for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 517–526, 2021.

[201] Yutong Xie, Jianpeng Zhang, Zhibin Liao, Johan Verjans, Chunhua Shen, and Yong Xia. Intra-and inter-pair consistency for semi-supervised gland segmentation. *IEEE Transactions on Image Processing*, 31:894–905, 2021.

[202] Xin Lai, Zhuotao Tian, Li Jiang, Shu Liu, Hengshuang Zhao, Liwei Wang, and Jiaya Jia. Semi-supervised semantic segmentation with directional context-aware consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1205–1214, 2021.

[203] Shuailin Li, Chuyu Zhang, and Xuming He. Shape-aware semi-supervised 3d semantic segmentation for medical images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 552–561. Springer, 2020.

[204] Xiangde Luo, Wenjun Liao, Jieneng Chen, Tao Song, Yinan Chen, Shichuan Zhang, Nianyong Chen, Guotai Wang, and Shaoting Zhang. Efficient semi-supervised gross target volume of nasopharyngeal carcinoma segmentation via uncertainty rectified pyramid consistency. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 318–329. Springer, 2021.

[205] Yingda Xia, Fengze Liu, Dong Yang, Jinzheng Cai, Lequan Yu, Zhuotun Zhu, Daguang Xu, Alan Yuille, and Holger Roth. 3d semi-supervised learning with uncertainty-aware multi-view co-training. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3646–3655, 2020.

[206] Lequan Yu, Shujun Wang, Xiaomeng Li, Chi-Wing Fu, and Pheng-Ann Heng. Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 605–613. Springer, 2019.

[207] Thanh Nguyen-Duc, Trung Le, He Zhao, Jianfei Cai, and Dinh Q Phung. Particle-based adversarial local distribution regularization. In *AISTATS*, pages 5212–5224, 2022.

[208] Prashnna Gyawali, Sandesh Ghimire, and Linwei Wang. Enhancing mixup-based semi-supervised learningwith explicit lipschitz regularization. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1046–1051. IEEE, 2020.

[209] Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, et al. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved? *IEEE transactions on medical imaging*, 37(11):2514–2525, 2018.

[210] Zhaohan Xiong, Qing Xia, Zhiqiang Hu, Ning Huang, Cheng Bian, Yefeng Zheng, Sulaiman Vesal, Nishant Ravikumar, Andreas Maier, Xin Yang, et al. A global benchmark of algorithms for segmenting the left atrium from late gadolinium-enhanced cardiac magnetic resonance imaging. *Medical image analysis*, 67:101832, 2021.

[211] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[212] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of CVPR*, pages 3903–3911, 2020.

[213] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of AAAI*, volume 35, pages 7028–7036, 2021.

[214] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[215] Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 278–287, 2017.

[216] Anh Tuan Bui, Trung Le, Quan Hung Tran, He Zhao, and Dinh Phung. A unified wasserstein distributional robustness framework for adversarial training. In *International Conference on Learning Representations*, 2022.

[217] Anh Tuan Bui, Trung Le, He Zhao, Paul Montague, Olivier deVel, Tamas Abraham, and Dinh Phung. Improving ensemble robustness by collaboratively promoting and demoting adversarial robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6831–6839, 2021.

[218] Thanh Nguyen-Duc, Trung Le, He Zhao, Jianfei Cai, and Dinh Phung. Particle-based adversarial local distribution regularization. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 5212–5224. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/nguyen-duc22a.html.

[219] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[220] Rafael Muller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *Proceeding of NeurIPS*, 2019.

[221] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Shu-Tao Xia. Adaptive regularization of labels. *arXiv preprint arXiv:1908.05474*, 2019.

[222] Thanh Nguyen-Duc, He Zhao, Jianfei Cai, and Dinh Phung. Med-tex: Transfer and explain knowledge with less data from pretrained medical imaging models. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–4. IEEE, 2022.

[223] Ruifei He, Shuyang Sun, Jihan Yang, Song Bai, and Xiaojuan Qi. Knowledge distillation as efficient pre-training: Faster convergence, higher data-efficiency, and better transferability. In *Proceedings of CVPR*, pages 9161–9171, 2022.

[224] Trung Le, Anh Tuan Bui, He Zhao, Paul Montague, Quan Tran, Dinh Phung, et al. On global-view based defense via adversarial attack and defense risk guaranteed bounds. In *International Conference on Artificial Intelligence and Statistics*, pages 11438–11460. PMLR, 2022.

[225] Anh Bui, Trung Le, He Zhao, Paul Montague, Olivier deVel, Tamas Abraham, and Dinh Phung. Improving adversarial robustness by enforcing local and global compactness. In *European Conference on Computer Vision*, pages 209–223. Springer, 2020.

[226] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Proceedings of AAAI*, volume 34, pages 3996–4003, 2020.

[227] Alvin Chan, Yi Tay, and Yew-Soon Ong. What it thinks is important is important: Robustness transfers through input gradients. In *Proceedings of CVPR*, pages 332–341, 2020.

[228] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. *Proceedings of Advances in Neural Information Processing Systems*, 33:16199–16210, 2020.

[229] Tao Bai, Jinnan Chen, Jun Zhao, Bihan Wen, Xudong Jiang, and Alex Kot. Feature distillation with guided adversarial contrastive learning. *arXiv preprint arXiv:2009.09922*, 2020.

[230] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of ICML*, pages 681–688, 2011.

[231] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Processings of CVPR*, pages 248–255. Ieee, 2009.

[232] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

[233] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

[234] T Gu, B Dolan-Gavitt, and SG BadNets. Identifying vulnerabilities in the machine learning model supply chain. In *Proceedings of the Neural Information Processing Symposium Workshop Mach. Learning Security (MLSec)*, pages 1–5, 2017.

[235] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[236] Jean-Baptiste Truong, Pratyush Maini, Robert J Walls, and Nicolas Papernot. Data-free model extraction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4771–4780, 2021.

[237] Hengrui Jia, Christopher A Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. Entangled watermarks as a defense against model extraction. In *USENIX Security Symposium*, pages 1937–1954, 2021.

[238] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021.

[239] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022.

[240] Qiang Liu. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30, 2017.

[241] Jun Han and Qiang Liu. Stein variational adaptive importance sampling. *arXiv preprint arXiv:1704.05201*, 2017.

[242] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

IEEE Requesting permission to reuse content from an IEEE publication

MED-TEX: Transfer and Explain Knowledge with Less Data from Pretrained Medical Imaging Models

Conference Proceedings:
2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)

Author: Thanh Nguyen-Duc

Publisher: IEEE

Date: 28 March 2022

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

Adversarial local distribution regularization for knowledge distillation

Conference Proceedings:
2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)

Author: Thanh Nguyen-Duc

Publisher: IEEE

Date: January 2023

IEEE Requesting permission to reuse content from an IEEE publication

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW