

Constraining Robust Information Quantities Improves Adversarial Robustness

by

Renhao Tan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical & Computer Engineering

Waterloo, Ontario, Canada, 2024

© Renhao Tan 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

It is known that deep neural networks (DNNs) are vulnerable to imperceptible adversarial attacks, and this fact raises concerns about their safety and reliability in real-world applications. In this thesis, we aim to boost the robustness of DNNs against white-box adversarial attacks by defining three information quantities: robust conditional mutual information (CMI), robust separation, and robust normalized CMI (NCMI), which can serve as evaluation metrics of robust performance for a DNN. We then utilize these concepts to introduce a novel regularization method that constrains intra-class concentration and increases inter-class separation simultaneously among output probability distributions of attacked data. Our experimental results demonstrate that our method consistently enhances model robustness against C&W and AutoAttack on CIFAR and Tiny-ImageNet datasets, both with and without additional synthetic data. The results show that our approach enhances the robust accuracy of DNNs by up to 2.66% on CIFAR datasets and 3.49% on Tiny-ImageNet against PGD attacks, and by 1.70% on CIFAR and 1.63% on Tiny-ImageNet against AutoAttack, compared to several state-of-the-art adversarial training methods.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. En-Hui Yang, for his invaluable guidance, countless support, and encouragement throughout the course of my research. His insightful feedback, expertise, and patience have been instrumental in shaping both this thesis and the development of my research. I am truly grateful for the opportunities he has provided, and for his mentorship, which has inspired me to push the boundaries of my academic pursuits.

I would like to extend my sincere thanks to all my former and current lab colleagues and coauthors for their invaluable help, support, and companionship throughout my research journey, especially Shayan Mohajer Hamidi, Linfeng Ye and Likhil Babu Pallatti. Their thoughtful insights, constructive feedback, and collaborative spirit have played a key role in the progress of this work. The numerous discussions, brainstorming sessions, and shared experiences have made this journey not only productive but also enjoyable. I am grateful for the friendship we have built, and for the inspiration each of you has provided. I would like to thank all of them for being an essential part of this experience.

I thank my parents, partner and family for their unconditional love and support throughout my master career. I am grateful to them for always standing on my side during the challenging time.

Dedication

This is dedicated to my loved ones and those provided help to me in the past.

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgements	iv
Dedication	v
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 General Background	1
1.2 Motivation	2
1.3 Contributions	3
2 Related Work	5
2.1 Adversarial Attack	5
2.2 Adversarial Defense	6
2.3 Relationship between Our Approach to Existing Adversarial Defense Techniques	7

3	Robust Information Quantities	9
3.1	Notation	9
3.2	Robust Error Rate	10
3.3	Extensions of CMI, Separation and NCMI to Adversarial Case	11
3.4	Robust CMI, Robust Separation, and Robust NCMI	14
3.5	Connections among Robust Accuracy, Adversarial NCMI, and Robust NCMI	15
4	Methodology	19
5	Experimental Results	25
5.1	General Experimental Settings	25
5.2	Results on Tiny-ImageNet	26
5.3	Results on CIFAR	26
5.4	Robustness against Various Perturbation Budgets	27
5.5	Simplex Visualization	30
5.6	Training Curves	32
5.7	Constraining NCMI over Benign Samples Also Boosts Robustness	34
6	Conclusion	36
	References	37
	APPENDICES	42
A	Proof of Theorem 1	43
B	Proof of Theorem 2	44
C	Proof of Theorem 3	46
D	Hyper-Parameter Settings	48

List of Figures

3.1	Diagram of generating an adversarial sample instance, where $\hat{Y}'$ is the random label predicted by the DNN in response to the adversarial instance. .	10
3.2	Robust accuracy of different learned models against AutoAttack vs the corresponding adversarial NCMI on CIFAR-10, where blue and orange bars represent robust accuracy (left axis) and adversarial NCMI (right axis), respectively, with corresponding specific values shown on top of each bar. . .	16
3.3	Robust accuracy of different learned models against PGD, C&W and AutoAttack vs the corresponding robust NCMI on CIFAR-10, where robust accuracy against PGD, C&W, and AutoAttack are presented as blue, orange and green bars (left axis), respectively. Red bars represent corresponding robust NCMI value (right axis) of each learned model. Specific value of each evaluation metric is shown on top of each bar.	17
5.1	Robust accuracy against PGD attack with various perturbation budgets ϵ on original CIFAR-10 when the underlying DNN model is PreAct-ResNet-18.	30
5.2	Visualization of the output probability simplex on the CIFAR-10 validation set, with varying constraining levels on adversarial CMI (α) and a fixed constraining level on adversarial separation ($\beta = 0$).	31
5.3	Visualization of the output probability simplex on the CIFAR-10 validation set, with varying constraining levels on adversarial separation (β) and a fixed constraining level on adversarial CMI ($\alpha = 1$).	32
5.4	Evaluation metric curves of PreAct-ResNet-18 on CIFAR-10 validation set during training. All metrics are computed over validation data after C&W $_{\infty}$ attack with 40 steps. ‘Baseline’ stands for training with vanilla AT [24], and ‘Ours’ represent vanilla AT with robust NCMI regularization.	33

List of Tables

5.1	Experimental results on Tiny-ImageNet.	26
5.2	Experimental results on original CIFAR- $\{10, 100\}$ datasets without synthetic data, are averaged over 3 runs. '+Ours' indicates the use of our $\hat{I}(\epsilon)$ constrained AT. Better results are in bold	28
5.3	Experimental results on CIFAR- $\{10, 100\}$ datasets with additional 1M synthetic data, are averaged over 3 runs. Datasets are denoted as 'CIFAR- $\{10, 100\}$ S', respectively. '+Ours' indicates the use of our $\hat{I}(\epsilon)$ constrained AT. Better results are in bold	29
5.4	Experimental results of constraining NCMI on benign samples (denoted as '+Clean NCMI') with TRADES on original CIFAR- $\{10, 100\}$ datasets without synthetic data, in comparison with TRADES and $\hat{I}(\epsilon)$ constrained AT (denoted as '+Robust NCMI'), are averaged over 3 runs. Best result of each group are marked in bold	34
5.5	Experimental results of constraining NCMI on benign samples (denoted as '+Clean NCMI') with TRADES on CIFAR- $\{10, 100\}$ datasets with additional 1 million synthetic data, in comparison with TRADES and $\hat{I}(\epsilon)$ constrained AT (denoted as '+Robust NCMI'), are averaged over 3 runs. Datasets are denoted as 'CIFAR- $\{10, 100\}$ S', respectively. Best result of each group are marked in bold	35
D.1	Hyper-parameter settings of $\hat{I}(\epsilon)$ constrained AT on CIFAR- $\{10, 100\}$ datasets. Each parenthesis represent the hyper-parameter value of α and β in our approach, respectively.	49
D.2	Hyper-parameter settings of $\hat{I}(\epsilon)$ constrained AT on Tiny-ImageNet datasets. Each parenthesis represent the hyper-parameter value of α and β in our approach, respectively.	49

Chapter 1

Introduction

1.1 General Background

Despite the remarkable success of deep neural networks (DNNs) in computer vision [18, 12] and natural language processing [36, 6], researchers found that DNNs are vulnerable to adversarial attacks [33, 9]. These attacks generate adversarial samples by adding slight perturbations, imperceptible to human eyes, to the original benign samples to deceive the underlying DNN. This raises concern to apply deep learning (DL) models to safety-critical domains like autonomous driving and medical diagnosis [19, 8].

A simple, yet effective method to train robust DNNs against adversarial attacks is adversarial training (AT) [9, 24]. In AT, the model is trained not on benign samples, but on adversarial ones. This process is formulated as a min-max optimization problem: the inner maximization focuses on generating adversarial samples, while the outer minimization aims to reduce the robust loss associated with these adversarial samples.

Following [24], a significant body of work has focused on improving model robustness, primarily through four approaches: (i) modifying or adding additional regularization terms to the loss function [47, 37, 41], (ii) altering model architecture [43], (iii) applying data augmentation techniques [29], and (iv) utilizing strategies like early stopping and weight averaging [30, 10].

1.2 Motivation

Even with the recent enhancement of AT, the underlying vulnerability of DNNs against adversarial attacks still remains unresolved: for certain data points, especially those with output prediction of natural examples near to the decision boundary, are more susceptible to perturbation and likely to cause misclassification [48].

To address the issue discussed earlier, this thesis analyzes the robustness of DNNs by closely examining their behavior within the output probability distribution space from an information geometric perspective.

First, we observe that, given a specific attack method G , all adversarial samples perturbed by G and corresponding to the same ground truth label form a distinct cluster in the DNN’s adversarial output probability distribution space with respect to attack G . We refer the set of such clusters across all labels as ‘adversarial clusters’. For a DNN to be robust against any adversarial attacks, output probability vectors of samples within the same adversarial cluster should be tightly concentrated even under various adversarial attacks, while sample output probability vectors with different labels should also remain well-separated under different attacks¹.

Inspired by a prior work developed by my supervisor, my colleagues and I [45], where we introduced information-theoretic quantities to improve DNN accuracy in conventional image classification, we extend these concepts to the adversarial domain. Specifically, [45] utilized conditional mutual information (CMI), $I(X; \hat{Y}|Y)$, to measure intra-class concentration, where X is the benign input, $\hat{Y}$ is the predicted label, and Y is the ground truth label.

In the adversarial context, we redefine CMI as $I(X'; \hat{Y}'|Y = y)$ to measure the intra-class concentration of adversarial clusters. Here, X' represents the adversarially perturbed sample, and $\hat{Y}'$ is the predicted label given X' . Averaging this over all class labels y with respect to their distributions $P_Y(y)$ gives $I(X'; \hat{Y}'|Y)$, which quantifies the average concentration of all adversarial clusters around their corresponding centroids.

A small value of $I(X'; \hat{Y}'|Y = y)$ indicates that the output distributions of adversarial samples from a given class are closely concentrated around the centroid of their adversarial cluster. In contrast, a large value represents that these distributions are more dispersed. As adversarial outputs become more tightly clustered around their centroids, $I(X'; \hat{Y}'|Y)$

¹In a similar vein, some prior studies in the literature attempted to train robust DNNs by bounding the polytopes formed by the adversarial cluster [40, 15, 25], where they restricted all crafted adversarial samples with the same class label fall in a certain proximity centered around their benign counterparts, either in output feature or probability distribution space.

decreases, making it harder for an adversarial attack to successfully cause misclassification of a DNN.

In the above case, $I(X'; \hat{Y}'|Y)$ measures the intra-class concentration with respect to a specific adversarial attack approach which crafts perturbed sample x' for each $(x, y) \in \mathcal{D}$ dataset. Since there is a wide range of adversarial attack methods and it is unknown to us which one would be applied to attack the underlying DNN, it is vital to us to consider the worst-case scenario among all possible attacks. Hence, we define the robust CMI of a DNN, which maps an input image x to a probability distribution $f(x)$, as below:

$$I(\epsilon) = \max_{x'} I(X'; \hat{Y}'|Y) \quad (1.1)$$

where the maximization is over all possible adversarial samples x' that generated by any attack method, subject to the constraint $\|x' - x\|_p \leq \epsilon$. With the same intuition, we further extend the concepts of separation and normalized CMI from [45] to the adversarial robustness setting, defining robust separation $\Gamma(\epsilon)$ and robust normalized CMI (NCMI) $\hat{I}(\epsilon)$. More specifically, robust separation stands for the minimum of the inter-class separation among all adversarial clusters across all attack methods that satisfying $\|x' - x\|_p \leq \epsilon$, and robust NCMI indicates the maximum ratio between robust CMI and robust separation over all possible adversarial attacks.

In this thesis, we argue that the concepts of robust CMI, robust separation and robust NCMI can be utilized to measure the robustness of a DNN viewed from the perspective of information geometry. In order to enhance the robustness of a DNN, one can aim to minimize the robust error rate while also minimizing robust CMI and maximizing robust separation at the same time. This promotes intra-class concentration and inter-class separation among perturbed sample outputs, which effectively aligns with the goal of minimizing both robust error rate and robust NCMI simultaneously.

1.3 Contributions

In summary, the contributions of this thesis are listed as follows:

- We extend the concepts of CMI, separation, and normalized CMI in [45] to the adversarial case, and introduce three new information quantities, robust CMI $I(\epsilon)$, robust separation $\Gamma(\epsilon)$, and robust NCMI $\hat{I}(\epsilon)$, to gain insights of the intrinsic mapping structure of DNNs in the context of adversarial robustness.

- We propose a brand new adversarial defense mechanism by jointly minimizing robust NCMI along with existing AT objective functions to enhance model robustness.
- We present a novel alternating learning algorithm to alternatively optimize DNN weight parameters θ and centroids of adversarial clusters.
- We conduct extensive experiments on CIFAR- $\{10, 100\}$ [17] and Tiny-ImageNet [20] datasets. Our results demonstrate that our proposed method boosts adversarial robustness when combined with existing adversarial defense objective functions, both with and without synthetic data.

Chapter 2

Related Work

2.1 Adversarial Attack

Since [33] revealed that deep neural networks (DNNs) are highly vulnerable to adversarial examples, where perturbations are imperceptible to the human eye but cause significant misclassifications, the development of increasingly sophisticated adversarial attack methods has become a focal point of research. [9] introduced the Fast Gradient Sign Method (FGSM), a simple yet effective approach that generates adversarial examples using a single step in the direction of the gradient, bounded by a small perturbation budget. Building on this, subsequent methods have refined the attack process to increase its potency. For example, the R+FGSM [34] attack added a randomization step to the original FGSM, and the Basic Iterative Method (BIM) [19] applied FGSM iteratively with smaller steps, leading to more effective perturbations.

A major breakthrough came with the development of projected gradient descent (PGD) by [24], which improved the success rate of adversarial attacks. PGD initializes adversarial samples randomly within an ϵ -bounded neighborhood of the original input and iteratively applies small perturbations that maximize a given loss function, making it a more powerful attack. This method essentially frames adversarial example generation process as a min-max optimization problem, seeking the worst-case perturbation for a model.

More recent technique, such as AutoAttack [5], has taken a more adaptive and parametric-free approach. This method have been regarded as state-of-the-art adversarial attack, and shown to outperform earlier attacks but at the cost of significantly higher computational time, further pushing the boundaries of both adversarial attack efficacy and computational

expense. In general, an adversarial sample x' is crafted as following:

$$x' = G(x, y) = x + \arg \max_{\|\delta\|_p \leq \epsilon} \mathcal{L}(f_\theta(x + \delta), y), \quad (2.1)$$

where $G(x, y)$ represents the attack function to generate adversarial example x' given benign input x and its corresponding ground truth label y , δ is the perturbation to be added to x to generate adversarial sample x' , $\|\cdot\|_p$ indicates the L_p -norm of a vector, ϵ denotes the maximum perturbation allowed to generate an adversarial sample, and $\mathcal{L}(\cdot, \cdot)$ stands for the objective function, which is typically cross entropy (CE) loss in previous works.

2.2 Adversarial Defense

As a defense mechanism against adversarial attacks, vanilla AT [24] achieved significant success by training deep neural networks (DNNs) with adversarial samples generated through the PGD attack. The simplicity and effectiveness of this method led to its widespread adoption as a foundation for further exploration of robust optimization techniques.

Building on the success of vanilla AT, subsequent studies refined the formulation of adversarial training to improve robustness. [16] introduced adversarial logit pairing (ALP), which encourages the logits of clean and adversarial examples to be similar, thus promoting robustness. [47] proposed TRADES, a method that balances the trade-off between adversarial robustness and accuracy on benign samples by explicitly incorporating a regularization term in the loss function to control the robustness-accuracy trade-off. Similarly, [37] developed MART, which focuses on misclassified examples and adjusts the training process to increase attention to these harder-to-classify adversarial samples. A more recent technique, adversarial weight perturbation (AWP) [41], additionally perturbs model weights other than input samples during training to further enhance robustness by smoothing the loss landscape.

Other lines of research focused on improving adversarial robustness by studying the margin between adversarial samples and decision boundaries. Methods like MMA training [7] and reduced variance adversarial training [27] aimed to increase the margin between adversarial examples and the decision boundaries, thereby improving robustness. [44] explored adaptive margin techniques to achieve a more flexible defense. Moreover, some approaches emphasized weighting adversarial examples based on their vulnerability, particularly those near decision boundaries. For instance, probabilistic margin-based defense [22] and GAIRAT [48] focused on assigning greater weights to adversarial samples close

to the decision boundary, prioritizing the model training on the most vulnerable samples, ultimately improving both robustness and generalization.

Aside from the aforementioned approaches for enhancing adversarial robustness, researchers have also conducted theoretical analyses showing that augmenting training data, whether labeled or unlabeled, can significantly improve the adversarial robustness of DNNs in both supervised and semi-supervised learning settings [31, 2, 35]. Building on this, several studies have explored the use of generative models to create large volumes of high-quality synthetic data for adversarial training [11, 29, 32, 38]. These works demonstrate that incorporating synthetic data helps alleviate robust overfitting and reduces the generalization gap, leading to significantly better performance compared to methods trained solely on the original dataset without additional data.

Together, these advancements reflect a diverse array of strategies aimed at enhancing adversarial training by addressing different aspects of the defense process, whether through loss refinement, margin optimization, instance weighting or utilizing synthetic data.

2.3 Relationship between Our Approach to Existing Adversarial Defense Techniques

Despite the advanced robust performance achieved by the methods discussed in the previous section, they primarily treat each attacked data point individually, failing to consider adversarial examples that share the same ground truth label as part of a cluster. By focusing solely on single adversarial samples, these approaches neglect the potential benefits of viewing adversarial data from a collective, cluster-based perspective.

In our approach, we propose that adversarial robustness can be further improved by treating adversarial examples with the same ground truth label as clusters and introducing constraints to ensure that the worst-case perturbed samples remain concentrated around the centroid of their respective adversarial cluster. By doing so, we aim to maintain the internal input-output mapping structure of each adversarial cluster as compact as possible, meanwhile further separate different clusters as distant as possible, thereby enhancing robustness. This perspective helps mitigate the impact of adversarial attacks, as even the most extreme perturbed samples within a class will stay close to their class’s centroid, making the model more robust against such attacks.

Furthermore, since prior studies have demonstrated that incorporating additional labeled or unlabeled data, as well as synthetic data, can significantly enhance robustness, we

extend our evaluation to datasets with and without synthetic data. The detailed methodology and results of our experiments are presented in the following chapters, showing the effectiveness of our cluster-based approach.

Chapter 3

Robust Information Quantities

In this chapter, we build upon the concepts of CMI, separation, and normalized CMI (NCMI) from [45] and adapt them to the adversarial setting. We introduce three new information measures, namely robust CMI $I(\epsilon)$, robust separation $\Gamma(\epsilon)$, and robust NCMI $\hat{I}(\epsilon)$, to evaluate the robustness of DNNs. We start by introducing the notation that will be used throughout this thesis.

3.1 Notation

For a positive integer K , let $[K] = \{1, \dots, K\}$ denote a set of integers starting from 1 to K . Assume there are C class labels, with $[C]$ indicating the set of all such labels. Let $\mathcal{P}([C])$ denote the set of all probability distributions over $[C]$. We also use $s[i]$ to represent the i -th entry of a probability distribution s . Given any two probability distributions $s_1, s_2 \in \mathcal{P}([C])$, we define CE of s_1 and s_2 as:

$$H(s_1, s_2) = \sum_{i=1}^C -s_1[i] \ln s_2[i] \quad (3.1)$$

and the Kullback-Leibler (KL) divergence between s_1 and s_2 as:

$$D(s_1 \| s_2) = \sum_{i=1}^C s_1[i] \ln \frac{s_1[i]}{s_2[i]}. \quad (3.2)$$

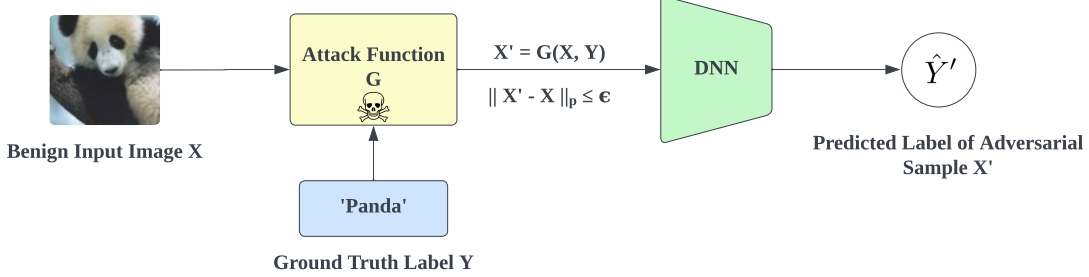


Figure 3.1: Diagram of generating an adversarial sample instance, where $\hat{Y}'$ is the random label predicted by the DNN in response to the adversarial instance.

For any random vector (X, Y) , we use $P_{X,Y}(x, y)$ or $P(x, y)$ to denote its joint distribution, and $P_X(x)$ and $P_Y(y)$ (or simply $P(x)$ and $P(y)$) to denote the marginal distributions of X and Y , respectively. The conditional distribution of Y given $X = x$ is written as $P_{Y|X}(\cdot|x)$, and that of X given $Y = y$ is denoted by $P_{X|Y}(\cdot|y)$. Additionally, denote $\mathbb{E}[\cdot]$ as expectation, and $\mathbb{E}_X[\cdot]$ as expectation with respect to random variable X . To clarify, we use the term ‘budget ϵ ’ to represent the maximum perturbation allowed to generate an adversarial instance x' within the L_p -norm ball centered at the benign sample instance x with a radius of ϵ . Furthermore, we use $G(x, y)$ to represent an attack function which crafts, for each (x, y) , an adversarial instance x' satisfying $\|x' - x\|_p \leq \epsilon$. Note that we used capital case letters like X to denote a random variable, while a lower case letter like x represents an individual sample.

3.2 Robust Error Rate

Given a DNN f mapping from an input image $x \in \mathbb{R}^d$ to a probability distribution $f(x)$, one essential evaluation metric of its robust performance is the robust error rate. Denoting (X, Y) as a random sample pair, indicating the pair of benign input and associated ground truth label, which follows the joint distribution of input and target from a given dataset. We use $\hat{Y}$ to represent the predicted label in response to input X by the DNN with output probability $f(X)[\hat{Y}]$. Conventionally, the error rate of a DNN over benign samples is defined as below:

$$\xi = \Pr(\hat{Y} \neq Y) \quad (3.3)$$

As shown in Fig. 3.1, when the benign input X is perturbed by a specific attack method, resulting in adversarial sample X' , the DNN produces a predicted label $\hat{Y}'$ with probability $f(X')[\hat{Y}']$ given X' . The adversarial error rate of the DNN for this specific attack method is then defined as:

$$\xi' = \Pr(\hat{Y}' \neq Y) \quad (3.4)$$

Since the above formulation restrict the error rate of a DNN to a specific attack method, we need to further consider the error rate of a DNN against all possible adversarial attack methods G subject to the constraint of $\|x' - x\|_p \leq \epsilon$, where ϵ denotes the maximum perturbation allowed to generate an adversarial sample x' . Then, we define the robust error rate of a DNN against all possible attacks as below:

$$\xi_r(\epsilon) = \max_{G: \|x' - x\|_p \leq \epsilon} \Pr(\hat{Y}' \neq Y) \quad (3.5)$$

Given Eq. (3.5), we argue that the robust error rate can be upper bounded by the average CE of conditional probability distribution over ground truth label Y given perturbed input X' , which is the one-hot probability distribution, and DNN's output distribution over perturbed input $f(X')$, as indicated in the theorem below:

Theorem 1 *For any DNN with mapping structure $x \rightarrow f(x)$, any (X, Y) , and any $\epsilon \geq 0$,*

$$\begin{aligned} \xi_r(\epsilon) &\leq \max_{G: \|x' - x\|_p \leq \epsilon} \mathbb{E}_{X'}[H(P_{Y|X'}(\cdot, X'), f(X'))] \\ &= \sum_{(x, y)} P(x, y) \left[\max_{G: \|x' - x\|_p \leq \epsilon} \{-\ln f(x')[y]\} \right] \end{aligned} \quad (3.6)$$

The proof of the above theorem is shown in Appendix A. Suggested by Theorem 1, minimizing the upper bound of the average CE between ground truth one-hot probability vectors and the DNN's output probability distributions for adversarial samples leads to a lower upper bound on the robust error rate, thereby improving robust performance. This also explains the effectiveness of vanilla AT [24] and MART [37], as they fundamentally follow this principle.

3.3 Extensions of CMI, Separation and NCMI to Adversarial Case

In a classification task of a dataset $\mathcal{D}$ with C different classes on benign sample instances x , one can observe that, a DNN with high accuracy, its output probability distributions

over samples with the same ground truth label i form a cluster, as most of them have the highest value at entry i and are relatively similar to each other. We refer to such clusters as benign clusters afterwards. As shown by [45], the centroid s_y of each benign cluster with ground truth label y is defined as:

$$s_y = P_{\hat{Y}|Y}(\cdot|Y = y) = \sum_x P_{X|Y}(x|Y = y) \cdot f(x) = \mathbb{E}_{X|Y}[f(X)|Y = y] \quad (3.7)$$

Then, the CMI $I(X; \hat{Y}|Y = y)$, which quantifies the concentration of sample outputs within the benign cluster associated with label y is defined as:

$$\begin{aligned} I(X; \hat{Y}|Y = y) &= \sum_x P_{X|Y}(x|y) \left[\sum_{i=1}^C P(\hat{Y} = i|x, y) \ln \frac{P_{\hat{Y}|XY}(\hat{Y} = i|x, y)}{P_{\hat{Y}|Y}(i|y)} \right] \\ &= \mathbb{E}_{X|Y} \left[\left(\sum_{i=1}^C f(X)[i] \ln \frac{f(X)[i]}{P_{\hat{Y}|Y}(i|y)} \right) \mid Y = y \right] \\ &= \mathbb{E}_{X|Y} [D(f(X) \| s_y) \mid Y = y] \end{aligned} \quad (3.8)$$

Following the above equation, the averaged CMI across all benign clusters is:

$$\begin{aligned} I(X; \hat{Y}|Y) &= \sum_{y \in [C]} P_Y(y) \cdot I(X; \hat{Y}|Y = y) \\ &= \mathbb{E}_{XY} [D(f(X) \| s_Y)] \end{aligned} \quad (3.9)$$

In [45], the inter-class separation Γ among all benign clusters is defined as:

$$\Gamma = \mathbb{E}[\mathcal{I}_{\{Y \neq V\}} \cdot H(f(X), f(U))] \quad (3.10)$$

Here, (U, V) represents another random sample, independent of (X, Y) , but drawn from the same joint distribution. $\mathcal{I}_{\{\cdot\}}$ denotes an indicator function with respect to a specific condition which returns 1 if the condition holds true and 0 otherwise. Then, the NCMI $\hat{I}(X; \hat{Y}|Y)$ for all benign clusters is defined as the ratio between average CMI across all clusters and Γ :

$$\hat{I}(X; \hat{Y}|Y) = \frac{I(X; \hat{Y}|Y)}{\Gamma} \quad (3.11)$$

For a given dataset $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^n$ and its joint distribution $P_{(X,Y)}$ is unknown and approximated by empirical distribution, one can estimate the above information quantities

defined from Eq. (3.7) to Eq. (3.10) by computing their corresponding sample means:

$$s_y = \frac{1}{|\mathcal{D}_y|} \sum_{j \in \mathcal{D}_y} f(x_j) \quad (3.12)$$

$$I(X; \hat{Y}|Y = y) = \frac{1}{|\mathcal{D}_y|} \sum_{j \in \mathcal{D}_y} D(f(x_j) \| s_y) \quad (3.13)$$

$$I(X; \hat{Y}|Y) = \frac{1}{n} \sum_{j=1}^n D(f(x_j) \| s_{y_j}) \quad (3.14)$$

$$\Gamma = \frac{1}{n^2} \sum_{j=1}^n \sum_{k=1}^n \mathcal{I}_{\{y_j \neq y_k\}} \cdot H(f(x_j), f(x_k)), \quad (3.15)$$

where $\mathcal{D}_y = \{j \mid 1 \leq j \leq n, y_j = y\}$ and $|\mathcal{D}_y|$ denotes the cardinality of $\mathcal{D}_y$.

Now, given a specific attack function G , all benign samples x will be replaced by corresponding adversarial samples $x' = G(x, y)$, and each benign cluster is consequently replaced by its respective adversarial cluster. Then, the information quantities defined in Eq. (3.7) to (3.11) can then be extended to corresponding adversarial case with replacing $X, \hat{Y}, U$ to $X', \hat{Y}', U'$, respectively, where $X' = G(X, Y)$ and $U' = G(U, V)$:

$$s'_y = P_{\hat{Y}'|Y}(\cdot|Y = y) = \mathbb{E}_{X|Y} [f(G(X, Y))|Y = y] \quad (3.16)$$

$$I(X'; \hat{Y}'|Y = y) = \mathbb{E}_{X'|Y} [D(f(X') \| s'_y) \mid Y = y] = \mathbb{E}_{X|Y} [D(f(G(X, Y)) \| s'_y) \mid Y = y] \quad (3.17)$$

$$I(X'; \hat{Y}'|Y) = \mathbb{E}_{X'Y} [D(f(X') \| s'_Y)] = \mathbb{E}_{XY} [D(f(G(X, Y)) \| s'_Y)] \quad (3.18)$$

$$\Gamma' = \mathbb{E} [\mathcal{I}_{\{Y \neq V\}} H(f(X'), f(U'))] = \mathbb{E} [\mathcal{I}_{\{Y \neq V\}} H(f(G(X, Y)), f(G(U, V)))] \quad (3.19)$$

$$\hat{I}(X'; \hat{Y}'|Y) = \frac{I(X'; \hat{Y}'|Y)}{\Gamma'} \quad (3.20)$$

We refer the above information quantities as adversarial centroid, adversarial CMI of adversarial cluster y , average adversarial CMI, adversarial separation, and adversarial NCMI, respectively. Note that given the attack function G , since $P_{\hat{Y}'|XY}(\hat{Y}' = i|x, y) = P_{\hat{Y}'|X'Y}(\hat{Y}' = i|x', y) = f(x')[i]$, it follows that:

$$I(X; \hat{Y}'|Y = y) = I(X'; \hat{Y}'|Y = y) \text{ and } I(X; \hat{Y}'|Y) = I(X'; \hat{Y}'|Y)$$

As we show that $I(X; \hat{Y}'|Y)$ and $I(X'; \hat{Y}'|Y)$ are indeed equivalent, we will address both terms interchangeably hereafter. Again, when the joint distribution $P_{X,Y}$ is unknown and

approximated by the empirical distribution, we can compute the adversarial information quantities defined in Eq. (3.16) to (3.19) as their respective sample means in the same spirit of Eq. (3.12) to (3.15):

$$s'_y = \frac{1}{|\mathcal{D}_y|} \sum_{j \in \mathcal{D}_y} f(x'_j) \quad (3.21)$$

$$I(X'; \hat{Y}' | Y = y) = \frac{1}{|\mathcal{D}_y|} \sum_{j \in \mathcal{D}_y} D(f(x'_j) \| s'_y) \quad (3.22)$$

$$I(X'; \hat{Y}' | Y) = \frac{1}{n} \sum_{j=1}^n D(f(x'_j) \| s'_{y_j}) \quad (3.23)$$

$$\Gamma' = \frac{1}{n^2} \sum_{j=1}^n \sum_{k=1}^n \mathcal{I}_{\{y_j \neq y_k\}} \cdot H(f(x'_j), f(x'_k)), \quad (3.24)$$

3.4 Robust CMI, Robust Separation, and Robust NCMI

As the above adversarial information quantities defined in Eq. (3.16) to (3.20) depend on a specific adversarial attack method G , this means that such metrics are sensitive to the underlying attack method and do not necessarily reflect some generic robustness characteristics of a given DNN. Hence, we introduce a new set of information quantities to capture the worst-case scenarios for a robust DNN in terms of CMI, separation, and NCMI. And these newly defined quantities are independent of the specific attack method used to perturb benign samples, and consider all possible perturbed samples that satisfying the condition $\|x' - x\|_p \leq \epsilon$ at the same time.

Definition 1 *Given a DNN mapping from input x to output probability distribution $f_\theta(x)$ with its weight parameters θ , define its robust CMI, separation, and NCMI over a random sample (X, Y) respectively as follows:*

$$I(\epsilon) = \max_{G: \|x' - x\|_p \leq \epsilon} I(X; \hat{Y}' | Y) \quad (3.25)$$

$$\Gamma(\epsilon) = \min_{G: \|x' - x\|_p \leq \epsilon} \mathbb{E} [\mathcal{I}_{\{Y \neq V\}} H(f_\theta(G(X, Y)), f_\theta(G(U, V)))] \quad (3.26)$$

$$\hat{I}(\epsilon) = \max_{G: \|x' - x\|_p \leq \epsilon} \frac{I(X; \hat{Y}' | Y)}{\mathbb{E} [\mathcal{I}_{\{Y \neq V\}} H(f_\theta(G(X, Y)), f_\theta(G(U, V)))]} \quad (3.27)$$

When (X, Y, θ) is clear from the context, we abbreviate $I(X, Y, \theta, \epsilon)$, $\Gamma(X, Y, \theta, \epsilon)$, and $\hat{I}(X, Y, \theta, \epsilon)$ as $I(\epsilon)$, $\Gamma(\epsilon)$, and $\hat{I}(\epsilon)$, respectively.

Given (X, Y, θ) , $\Gamma(\epsilon)$ can be computed using the standard gradient descent method through backpropagation. By introducing a dummy conditional "backward channel" $q(x|i, y)$ from $\hat{Y}'$ to X conditioned on Y , both $I(\epsilon)$ and $\hat{I}(\epsilon)$ can be computed using an alternating method, as suggested by the following theorem.

Theorem 2 *For any $i, y \in [C]$, let $q(\cdot|i, y)$ denote a dummy conditional distribution in the space of X , which is assumed to be discrete for simplicity. Then, $I(\epsilon)$ and $\hat{I}(\epsilon)$ can be computed by solving the following double maximization problems, respectively:*

$$I(\epsilon) = \max_{\{q(x|i, y)\}} \max_{G: \|x' - x\|_p \leq \epsilon} \sum_{x, y} P(x, y) \left[\sum_{i=1}^C f_\theta(G(x, y))[i] \ln q(x|i, y) - \ln P_{X|Y}(x|y) \right] \quad (3.28)$$

$$= \max_{\{q(x|i, y)\}} \sum_{x, y} P(x, y) \left[\max_{x': \|x' - x\|_p \leq \epsilon} \sum_{i=1}^C f_\theta(x')[i] \ln q(x|i, y) - \ln P_{X|Y}(x|y) \right] \quad (3.29)$$

$$\hat{I}(\epsilon) = \max_{\{q(x|i, y)\}} \max_{G: \|x' - x\|_p \leq \epsilon} \frac{\sum_{x, y} P(x, y) \left[\sum_{i=1}^C f_\theta(G(x, y))[i] \ln q(x|i, y) - \ln P_{X|Y}(x|y) \right]}{\mathbb{E} [\mathcal{I}_{\{Y \neq V\}} H(f_\theta(G(X, Y)), f_\theta(G(U, V)))]} \quad (3.30)$$

where given attack method G , the outer maximization above is achieved when

$$q(x|i, y) = \frac{P_{X|Y}(x|y) f_\theta(x')[i]}{P_{\hat{Y}'|Y}(i|y)}. \quad (3.31)$$

Theorem 2 will be proven in Appendix B. Given $q(x|i, y)$, the inner maximization in Eq. (3.28) to (3.30) can now be computed using the standard gradient descent method via backpropagation. Therefore, we can iteratively compute $I(\epsilon)$ and $\hat{I}(\epsilon)$ by alternating between optimizing $q(x|i, y)$ given G , and optimizing G given $q(x|i, y)$.

3.5 Connections among Robust Accuracy, Adversarial NCMI, and Robust NCMI

As suggested by [45], a strong positive linear relationship exists between NCMI and error rate in benign image classification tasks, particularly within the same model family (e.g., ResNet-18 and ResNet-34). In our AT experiments, we trained several PreAct-ResNet-18

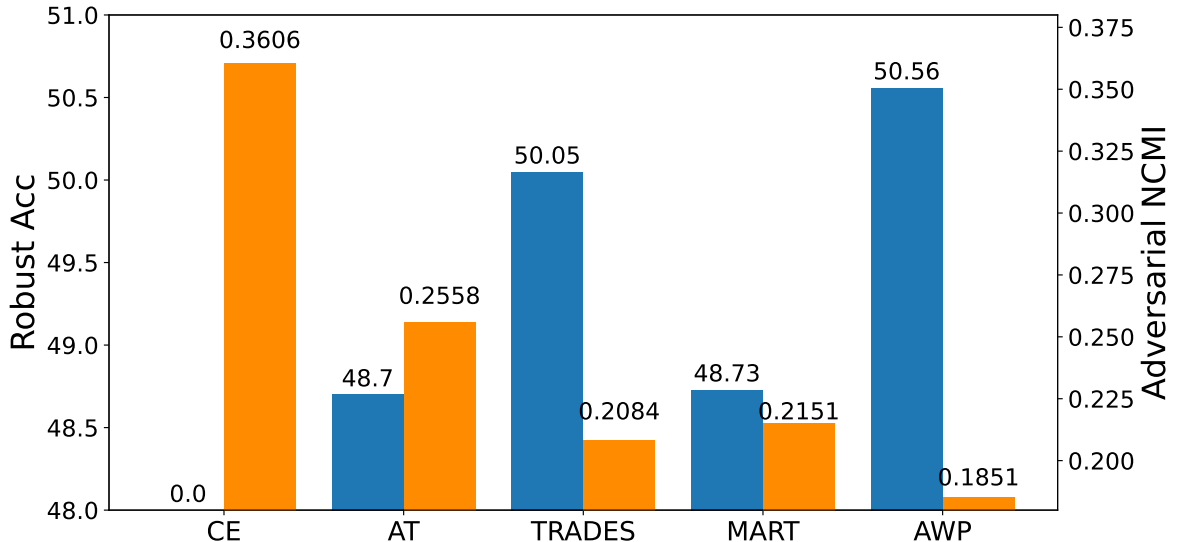


Figure 3.2: Robust accuracy of different learned models against AutoAttack vs the corresponding adversarial NCMI on CIFAR-10, where blue and orange bars represent robust accuracy (left axis) and adversarial NCMI (right axis), respectively, with corresponding specific values shown on top of each bar.

models [13] using various AT methods, including the standard CE method without any adversarial instances (denoted as ‘CE’), on CIFAR-10 dataset. We then evaluated the robust accuracy of these models against AutoAttack [5]. As illustrated in Fig. 3.2, our results showed that robust accuracy is generally inversely proportional to the adversarial NCMI of the output probability distributions when the attack function G is set to AutoAttack, aligning with the findings in [45] for benign image classification.

In order to evaluate the robust NCMI $\hat{I}(\epsilon)$ of a given trained DNN, one should follow Eq. (3.30) to compute the corresponding value over validation data. Given that Eq. (3.30) is formulated as a double maximization, the optimum $\hat{I}(\epsilon)$ can be determined through an alternating algorithm. This process involves the following steps, repeated for five iterations or until $\hat{I}(\epsilon)$ converges: (1) fixing $q(x|i, y)$ and solving for the attack function G over all validation data simultaneously, due to the interdependencies among attacked samples when computing robust separation (denominator in Eq. (3.30)); (2) fixing the attack function G and computing each $q(x|i, y)$ following Eq. (3.30).

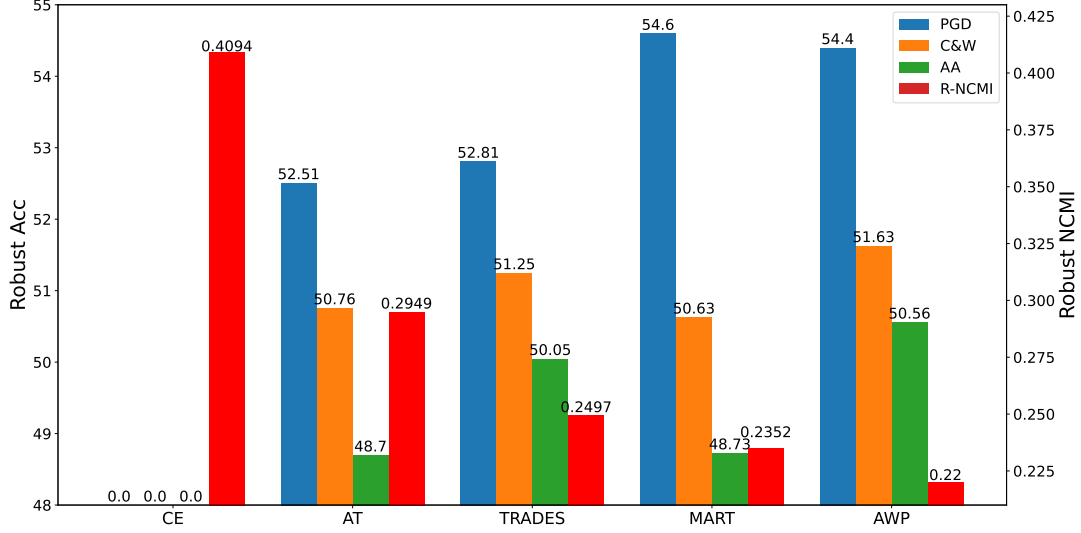


Figure 3.3: Robust accuracy of different learned models against PGD, C&W and AutoAttack vs the corresponding robust NCMi on CIFAR-10, where robust accuracy against PGD, C&W, and AutoAttack are presented as blue, orange and green bars (left axis), respectively. Red bars represent corresponding robust NCMi value (right axis) of each learned model. Specific value of each evaluation metric is shown on top of each bar.

We then reported the corresponding $\hat{I}(\epsilon)$ values for various conventional AT methods in Fig. 3.3, comparing them with robust accuracy against multiple white-box attacks. The relationship between robust accuracy and robust NCMi $\hat{I}(\epsilon)$ is similar to the relationship between robust accuracy and adversarial NCMi shown in Fig. 3.2, where higher robust accuracy corresponds to lower $\hat{I}(\epsilon)$. This suggests that robust NCMi $\hat{I}(\epsilon)$ can serve as an evaluation metric not only for robustness but also for reflecting the intrinsic mapping structure of a DNN.

Furthermore, the value of $\hat{I}(\epsilon)$ in Fig. 3.3 is generally higher than the adversarial NCMi in Fig. 3.2 for each AT method. This difference arises due to that adversarial NCMi is dependent on the specific attack method used to generate perturbed data and measures the ratio between intra-class concentration and inter-class separation based on those specific adversarial samples. In contrast, robust NCMi $\hat{I}(\epsilon)$ reflects the worst-case scenario across all possible attack methods, and does not rely on any single attack function G to quantify

the ratio between intra-class concentration and inter-class separation. As a result, the value of $\hat{I}(\epsilon)$ will generally be greater than adversarial NCMI for a given attack function G , especially when the perturbed data from G do not induce the worst-case scenario in terms of CMI and separation.

Chapter 4

Methodology

Equipped with the definitions of robust information quantities introduced in the previous chapter, we can constrain these metrics when training a robust DNN. Since robust NCMI capture a DNN's intrinsic mapping structure without directly impacting robust accuracy, they can be constrained independently of methods like TRADES [47] and MART [37] that only focus on accuracy. Thus, we constrain robust NCMI $\hat{I}(\epsilon)$ by minimizing CMI and maximizing separation among samples perturbed by an attack function G , optimizing them alongside existing loss functions in AT. In this case, both $\hat{I}(\epsilon)$ and conventional AT loss are small. Let $\mathcal{L}(X, \hat{X}, Y)$ denote a conventional AT loss such as one of those in TRADES and MART, where $\hat{X} = G_1(X, Y)$ is a perturbed sample instance given by the attack function G_1 in the conventional AT loss. The objective function in conventional AT methods indicates below:

$$\min_{\theta} \mathbb{E} \left[\max_{G_1: \|\hat{x}-x\|_p \leq \epsilon} \mathcal{L}(X, \hat{X}, Y) \right], \quad (4.1)$$

In order to further constrain robust NCMI along with robust error rate of a DNN, we try to optimize the following objective function during training:

$$\min_{\theta} \left\{ \mathbb{E} \left[\max_{G_1: \|\hat{x}-x\|_p \leq \epsilon} \mathcal{L}(X, \hat{X}, Y) \right] + \lambda \hat{I}(X, Y, \theta, \epsilon) \right\} \quad (4.2)$$

Most current AT frameworks generate adversarial examples within an L_p -norm ball to maximize an objective function, typically CE loss due to its direct connection to robust accuracy. Following the same intuition, we constrain the upper bound of robust NCMI by generating worst-case adversarial examples using an attack function different from G_1 mentioned earlier. This is achieved through seeking perturbed samples which maximizes

a combined formulation of CMI and separation. Then, we optimize this upper bound alongside the existing adversarial training loss function, as indicated in Eq. (4.2). Due to Eq. (3.30), (3.18) and (3.19), Eq. (4.2) is equivalent to:

$$\min_{\theta} \left\{ \mathbb{E} \left[\max_{G_1: \|\hat{x}-x\|_p \leq \epsilon} \mathcal{L}(X, \hat{X}, Y) \right] + \max_{G: \|x'-x\|_p \leq \epsilon} \left[\alpha \cdot I(X'; \hat{Y}'|Y) - \beta \cdot \Gamma' \right] \right\} \quad (4.3)$$

$$= \min_{\theta} \left\{ \mathbb{E} \left[\max_{G_1: \|\hat{x}-x\|_p \leq \epsilon} \mathcal{L}(X, \hat{X}, Y) \right] + \max_{G: \|x'-x\|_p \leq \epsilon} \left[\alpha \mathbb{E}[D(f_{\theta}(G(X, Y)) \| s'_Y)] \right. \right. \\ \left. \left. - \beta \mathbb{E}[\mathcal{I}_{\{Y \neq V\}} H(f_{\theta}(G(X, Y)), f_{\theta}(G(U, V)))] \right] \right\} \quad (4.4)$$

$$= \min_{\theta} \max_{G_1} \max_G \left\{ \mathbb{E}[\mathcal{L}(X, G_1(X, Y), Y)] + \alpha \mathbb{E}[D(f_{\theta}(G(X, Y)) \| s'_Y)] \right. \\ \left. - \beta \mathbb{E}[\mathcal{I}_{\{Y \neq V\}} H(f_{\theta}(G(X, Y)), f_{\theta}(G(U, V)))] \right\} \quad (4.5)$$

where the above maximization of attack function G_1 is subject to the constraint $\|\hat{x} - x\|_p \leq \epsilon$, and the second maximization denoting as G represents all possible adversarial attacks which generates the worst-case adversarial sample with respect to robust NCM $\hat{I}(\epsilon)$ under constraint $\|x' - x\|_p \leq \epsilon$.

Since the second term in Eq. (4.5) depends on the output probability distributions of all perturbed samples within each adversarial cluster, it is not feasible to directly compute this information quantity in parallel on a GPU with respect to s'_Y . To address this, we introduce dummy variables Q'_y for each label $y \in [C]$, approximating the actual probability distribution of each adversarial centroid s'_y , as suggested by [45]. After introducing Q'_Y , our objective function can be sample additive and then optimized in parallel on GPU, as shown in the following theorem and will be proved in Appendix C:

Theorem 3 For any $\epsilon \geq 0$,

$$\min_{\theta} \left\{ \mathbb{E} \left[\max_{G_1: \|\hat{x}-x\|_p \leq \epsilon} \mathcal{L}(X, \hat{X}, Y) \right] + \max_{G: \|x'-x\|_p \leq \epsilon} \left[\alpha \cdot I(X'; \hat{Y}'|Y) - \beta \cdot \Gamma' \right] \right\} \\ \leq \min_{\theta} \min_{\{Q'_c\}_{c \in [C]}} \max_{G_1, G} \left\{ \mathbb{E}[\mathcal{L}(X, G_1(X, Y), Y)] + \alpha \mathbb{E}[D(f_{\theta}(G(X, Y)) \| Q'_Y)] \right. \\ \left. - \beta \mathbb{E}[\mathcal{I}_{\{Y \neq V\}} H(f_{\theta}(G(X, Y)), f_{\theta}(G(U, V)))] \right\} \quad (4.6)$$

where given (θ, G_1, G) , the minimization over $\{Q'_c\}_{c \in [C]}$ is achieved when

$$Q'_y = s'_y = \mathbb{E}_{X|Y} [f_{\theta}(G(X, Y)) | Y = y]. \quad (4.7)$$

Given $(\theta, \{Q'_c\}_{c \in [C]}, G_1, G)$, the objective function on the right side of Eq. (4.6) is now sample additive, and becomes:

$$\mathcal{J}_{\mathcal{B}}(\theta, \{Q'_c\}_{c \in [C]}, G_1, G) = \left\{ \frac{1}{|\mathcal{B}|} \sum_{(x,y) \in \mathcal{B}} \mathcal{L}(x, G_1(x, y), y) + \frac{1}{|\mathcal{B}|} \mathcal{L}_{\mathcal{B}}(\theta, \{Q'_c\}_{c \in [C]}, G) \right\} \quad (4.8)$$

when $P(x, y)$ is unknown and approximated by the empirical distribution of a mini-batch $\mathcal{B}$, where:

$$\begin{aligned} \mathcal{L}_{\mathcal{B}}(\theta, \{Q'_c\}_{c \in [C]}, G) = & \sum_{(x,y) \in \mathcal{B}} \left[\alpha D(f_{\theta}(G(x, y)) \| Q'_y) \right. \\ & \left. - \frac{\beta}{|\mathcal{B}|} \sum_{(u,v) \in \mathcal{B}} \mathcal{I}_{\{y \neq v\}} H(f_{\theta}(G(x, y)), f_{\theta}(G(u, v))) \right]. \end{aligned} \quad (4.9)$$

In $\hat{I}(\epsilon)$ constrained AT, we solve the optimization problem on the right side of Eq. (4.6):

$$\min_{\theta} \min_{\{Q'_c\}_{c \in [C]}} \max_{G_1, G} \mathcal{J}_{\mathcal{B}}(\theta, \{Q'_c\}_{c \in [C]}, G_1, G) \quad (4.10)$$

when $P(x, y)$ is unknown and approximated by the empirical distribution of a mini-batch $\mathcal{B}$.

Unlike most existing AT methods that generate perturbed samples individually for each mini-batch during training, the optimal attack function $G^*(X, Y)$ defined in Eq. (4.10) crafts adversarial samples x' for the entire mini-batch simultaneously. This is necessary because adversarial separation Γ' measures the CE between pairs of perturbed samples with different ground truth labels, making the attack process interdependent rather than treating each sample in isolation.

Up until now, the remaining problem is how to optimize dummy probability distributions of centroids $\{Q'_c\}_{c \in [C]}$ and DNN parameter θ during training. One feasible way is to update θ and $\{Q'_c\}_{c \in [C]}$ in a fixed order through an alternating algorithm to minimize the objective function in Eq. (4.10). In each mini-batch $\mathcal{B}$, we firstly fix $\{Q'_c\}_{c \in [C]}$ to perform back-propagation to update θ , then we fix θ and update $\{Q'_c\}_{c \in [C]}$ in a weight averaging manner based on the output of perturbed samples following the optimal G^* in the current mini-batch, since computing adversarial samples over the entire dataset is computationally expensive. The pseudo-code is presented in Algorithm 1. We argue that this adversarial centroids updating method only follows the assumption that there are sufficient number of samples from each adversarial cluster within a mini-batch (based on the law of large

Algorithm 1 $\hat{I}(\epsilon)$ Constrained Adversarial Training

Input: DNN f with model parameter θ , hyper-parameters α and β , dataset $\mathcal{D}:(\mathcal{X}, \mathcal{Y})$, number of epochs T , number of classes C , adversarial loss function $\mathcal{L}$, perturbation budget ϵ , EMA factor τ

```
1: for  $i \in [1, \dots, C]$  do
2:   Initialize adversarial centroids:  $Q'_i \leftarrow \frac{1}{|\mathcal{D}_i|} \sum_{j \in \mathcal{D}_i} f_\theta(x_j)$ 
3: end for
4: for  $e \in [1, \dots, T]$  do
5:   for each mini-batch  $\mathcal{B}$  do
6:     [Update DNN parameter  $\theta$ ]:
7:     Determine the optimized  $G_1^*$  by computing a perturbed instance  $\hat{x}$  for each
       sample instance  $(x, y) \in \mathcal{B}$  via PGD attack with 10 steps.
8:     Determine the optimized  $G^*$  by computing all adversarial instances  $x'$  for all
       sample instances  $(x, y) \in \mathcal{B}$  simultaneously all at once in the same manner as PGD
       attack with 10 steps, where the objective function to be maximized in the attack process
       is  $\mathcal{L}_{\mathcal{B}}(\theta, \{Q'_c\}_{c \in [C]}, G)$ .
9:      $x'_j \leftarrow G^*(x_j, y_j)$  for all  $(x_j, y_j) \in \mathcal{B}$ 
10:     $loss \leftarrow \mathcal{J}_{\mathcal{B}}(\theta, \{Q'_c\}_{c \in [C]}, G_1^*, G^*)$ 
11:    Update  $\theta$  w.r.t.  $loss$ .
12:    [Update adversarial centroids]:
13:    for  $i \in [1, \dots, C]$  do
14:      Use adversarial instances  $x'_j$  in line 9 to craft  $\mathcal{B}'_i \leftarrow \{(x'_j, y_j) | (x_j, y_j) \in \mathcal{B}, y_j =$ 
         $i\}$ 
15:      Continue if  $|\mathcal{B}'_i| = 0$ 
16:       $Q'_i \leftarrow \tau Q'_i + (1 - \tau) \frac{1}{|\mathcal{B}'_i|} \sum_{(x'_j, y_j) \in \mathcal{B}'_i} f_\theta(x'_j)$ 
17:    end for
18:  end for
19: end for
20: return DNN  $f_\theta$ 
```

Algorithm 2 Relaxed Alternating Algorithm for $\hat{I}(\epsilon)$ Constrained Adversarial Training

Input: DNN f with model parameter θ , hyper-parameters α and β , dataset $\mathcal{D}:(\mathcal{X}, \mathcal{Y})$, number of epochs T , number of classes C , adversarial loss function $\mathcal{L}$, perturbation budget ϵ

```
1: for  $e \in [1, \dots, T]$  do
2:   [Update DNN parameter  $\theta$ ]:
3:   for each mini-batch  $\mathcal{B} \subset \text{train set } \mathcal{D}_{\text{train}}$  do
4:     Determine the optimized  $G_1^*$  by computing a perturbed instance  $\hat{x}$  for each
     sample instance  $(x, y) \in \mathcal{B}$  via PGD attack with 10 steps.
5:     if  $e > 1$  then
6:       Determine the optimized  $G^*$  by computing all adversarial instances  $x'$  for
       all sample instances  $(x, y) \in \mathcal{B}$  simultaneously all at once in the same manner as
       PGD attack with 10 steps, where the objective function to be maximized in the attack
       process is  $\mathcal{L}_{\mathcal{B}}(\theta, \{Q'_c\}_{c \in [C]}, G)$  in (4.9).
7:        $x'_j \leftarrow G^*(x_j, y_j)$  for all  $(x_j, y_j) \in \mathcal{B}$ 
8:        $loss \leftarrow \mathcal{J}_{\mathcal{B}}(\theta, \{Q'_c\}_{c \in [C]}, G_1^*, G^*)$ 
9:     else
10:       $loss \leftarrow \frac{1}{|\mathcal{B}|} \sum_{(x_j, y_j) \in \mathcal{B}} \mathcal{L}(x_j, G_1^*(x_j, y_j), y_j)$ 
11:    end if
12:    Update  $\theta$  w.r.t.  $loss$ 
13:  end for
14:  if  $e = 1$  then
15:    Initialize adversarial centroids:  $Q'_i \leftarrow \frac{1}{|\mathcal{D}_i|} \sum_{j \in \mathcal{D}_i} f_{\theta}(x_j)$  for  $i \in [1, \dots, C]$ 
16:  end if
17:  [Update adversarial centroids]:
18:  Initialize  $Q'_{i, dummy}$  to zero vectors for  $i \in [1, \dots, C]$ 
19:  for each mini-batch  $\mathcal{B}_Q \subset \text{train set } \mathcal{D}_{\text{train}}$  with  $|\mathcal{B}_Q| = 4 \cdot |\mathcal{B}|$  do
20:    Determine the optimized  $G^*$  by computing all adversarial instances  $x'_Q$  for all
    sample instances  $(x_Q, y_Q) \in \mathcal{B}_Q$  simultaneously all at once in the same manner as
    PGD attack with 10 steps, where the objective function to be maximized in the attack
    process is  $\mathcal{L}_{\mathcal{B}_Q}(\theta, \{Q'_c\}_{c \in [C]}, G)$  in (4.9).
21:     $x'_{j, Q} \leftarrow G^*(x_{j, Q}, y_{j, Q})$  for all  $(x_{j, Q}, y_{j, Q}) \in \mathcal{B}_Q$ 
22:    Construct the set  $\mathcal{B}'_Q$  consisting of all pairs  $(x'_{j, Q}, y_{j, Q})$  generated in line 21
23:    for  $(x'_{j, Q}, y_{j, Q}) \in \mathcal{B}'_Q$  do
24:       $Q'_{y_{j, Q}, dummy} \leftarrow Q'_{y_{j, Q}, dummy} + f_{\theta}(x'_{j, Q})$ 
25:    end for
26:  end for
27:  Normalize each  $Q'_{i, dummy}$  to a probability distribution for  $i \in [1, \dots, C]$ 
28:   $Q'_i \leftarrow Q'_{i, dummy}$  for  $i \in [1, \dots, C]$ 
29: end for
30: return DNN  $f_{\theta}$ 
```

numbers), otherwise the optimization of Q'_Y may be noisy and unstable. As a solution to insufficient number of samples from each cluster in a mini-batch, we develop a relaxed alternating learning algorithm to update $\{Q'_c\}_{c \in [C]}$ based on perturbed sample outputs from the entire training set at the end of each epoch during training, which shows in Algorithm 2.

Note that in Algorithm 2, when we iterate through the training set one more time to update all adversarial centroids at the end of each epoch, we increase size of each mini-batch $|\mathcal{B}_Q|$ to 4 times as that when we update DNN parameter θ . This adjustment is mostly in consideration of providing a better estimation on the average CE between sample pairs with different class labels (second term in Eq. (4.9)), as compute this value over a larger mini-batch can approximate this value more precisely. Ideally, one should compute this value over the entire training set in one pass. Due to the limit of computational resource, when we update adversarial centroids, we only increase the mini-batch size up to 4 times (batch size of 512) as that in conventional training (batch size of 128).

One may wonder why we train a robust DNN with two different types of adversarial attack instead of the one from existing AT loss function $\mathcal{L}(X, \hat{X}, Y)$. This is mostly due to 3 reasons. Firstly, constraining $\hat{I}(\epsilon)$ over perturbed samples that maximize an objective function unrelated to robust accuracy (e.g., TRADES) is ineffective, as compacting samples from the same adversarial cluster and separating those from different clusters may not improve the robustness of a DNN. Secondly, optimizing the DNN over adversarial samples crafted by simultaneously maximizing CMI and minimizing separation with respect to perturbed samples crafted by the optimal attack G^* is the most effective way to regularize the upper bound of $\hat{I}(\epsilon)$, as these samples following G^* yield the worst robust NCMI $\hat{I}(\epsilon)$ for the given DNN within the perturbation budget ϵ . Thirdly, by making the optimal attack method G^* in Eq. (4.10) independent of existing attack techniques during training, our approach holds a ‘plug-and-play’ property, making it compatible with various AT methods. We then apply our $\hat{I}(\epsilon)$ constrained AT alongside current AT methods, compare the results with those without applying our approach, and present our results in the next chapter.

Chapter 5

Experimental Results

To validate the efficacy of our proposed $\hat{I}(\epsilon)$ constrained AT, we integrated it with several adversarial loss functions, including vanilla AT [24], TRADES [47], MART [37], and AWP [41], and considered the standard version of such AT methods as baselines. We conducted extensive experiments¹ on CIFAR-10, 100 [17], Tiny-ImageNet [20], and CIFAR-10, 100 augmented with 1 million synthetic images. Our results show that incorporating our method enhances the performance of these AT loss functions on both original and synthetic-augmented datasets.

5.1 General Experimental Settings

In all of our experiments, adversarial samples for training were generated using PGD with an L_∞ -norm budget of $\epsilon = 8/255$, a step size of $2/255$, and optimized over 10 steps. To evaluate robustness, we applied multiple white-box attacks, including PGD-40 [24], C&W attack [1] under L_∞ -norm (both with 40 steps), and AutoAttack [5], to the model checkpoint with the highest validation accuracy. We used default hyper-parameter settings for all baseline AT methods and selected the hyper-parameters α and β for our $\hat{I}(\epsilon)$ constrained AT from the set of $\{1 \times 10^{-3}, 1 \times 10^{-2}, \dots, 1 \times 10^0, 2, 5\}$. We reported the optimal hyper-parameter settings of all our experiments in Appendix D.

¹Experiments on CIFAR-100 without synthetic data and Tiny-ImageNet followed the Alg. 2 due to the small number of samples from each class per mini-batch, which may cause inaccurate centroid estimation using Alg. 1. All other experiments in this section follow Alg. 1.

Table 5.1: Experimental results on Tiny-ImageNet.

Model	Method	Clean	PGD	C&W _∞	AA
PreAct-RN18	TRADES	47.13	23.12	18.47	17.59
	+Ours	46.95	23.30	18.91	18.10
	Vanilla AT	49.08	23.15	20.79	18.78
	+Ours	48.43	26.29	21.41	19.72
PreAct-RN34	TRADES	49.17	24.69	20.59	19.86
	+Ours	48.56	25.09	20.94	20.37
	Vanilla AT	50.13	24.19	22.16	20.06
	+Ours	50.99	27.68	23.87	21.69

5.2 Results on Tiny-ImageNet

To demonstrate the effectiveness of our method, we combined $\hat{I}(\epsilon)$ constrained AT with vanilla AT and TRADES on the Tiny-ImageNet dataset using PreAct-ResNet-{18, 34} [13] architectures. Models were trained for 100 epochs with a mini-batch size of 256, L_2 regularization (5×10^{-4}), and a cosine annealing learning rate scheduling (max 0.1). Weight averaging (WA) [14] was applied with a decay factor of $\tau = 0.995$, as adopting WA can smooth the loss landscape, thereby reducing robust generalization gap and leading to enhanced performance [10, 3]. In our approach, the baseline adversarial loss function was used until the 5th epoch to maintain stability, and the objective function in Eq. (4.10) was applied afterwards, since adversarial clusters are not well-formed in the early stage of training. In Table 5.1, we reported the accuracy on benign samples, robust accuracy against PGD, C&W and AutoAttack, and denoted such metrics as ‘Clean’, ‘PGD’, ‘C&W_∞’ and ‘AA’, respectively. As shown, our method exhibits enhanced robustness comparing with those without constraining on $\hat{I}(\epsilon)$, and the robust accuracy gap across all attacks increases as the model size grows. In addition, we observed that the gain of robust accuracy against PGD attack is more than 3% when incorporating our approach with vanilla AT for both models, and reaches a maximum gain of 1.63% against AutoAttack.

5.3 Results on CIFAR

In our experiments on the original CIFAR datasets without synthetic data, we used PreAct-ResNet and Wide ResNet (WRN) [46] with ReLU activation, both common architectures in AT. Models were trained for 200 epochs with a mini-batch size of 128 and a multi-step

decay learning rate schedule, starting at 0.1 and reduced by a factor of 0.1 at the 100th and 150th epoch. We used stochastic gradient descent (SGD) with momentum of 0.9, L_2 regularization factor of 5×10^{-4} , and a WA factor $\tau = 0.999$. The adversarial centroids Q'_Y were updated per mini-batch using WA with a factor $\tau_Q = 0.999$.

For experiments with additional data, we used 1 million labeled synthetic images from [38], and replaced ReLU with Swish activation [28] for better smoothness [42, 10]. Models were trained for 400 epochs with WA ($\tau = 0.995$) and centroids were updated with a WA factor $\tau_Q = 0.99$. We applied cosine annealing for the learning rate scheduling, with a maximum value of 0.2. The original-to-synthetic data ratio was set to 3:7, and mini-batch size is set to 512. Other settings are identical to those of the original CIFAR datasets.

We demonstrate our experimental results on original CIFAR- $\{10, 100\}$ in Table 5.2, where employing our method alongside existing adversarial loss functions consistently outperforms training solely with those loss functions without constraining on $\hat{I}(\epsilon)$ across all methods when evaluated under AutoAttack. In most settings, our method slightly sacrifices benign accuracy for better robustness, while we observed that both benign and robust accuracy against AutoAttack increase for most models trained by vanilla AT and TRADES with constraining on $\hat{I}(\epsilon)$ comparing with its counterparts without it.

Moreover, we reported experimental results over CIFAR datasets with additional 1 million synthetic data in Table 5.3 as well. Similar to those trained solely on original datasets, $\hat{I}(\epsilon)$ constrained AT demonstrates prominence in terms of robust accuracy comparing with corresponding baseline methods. This indicates that our approach can further boost robustness with the help of synthetic data. Notably, the averaged improvement of robust accuracy against AutoAttack with synthetic data is 0.71%, which is higher than that of models trained without synthetic data (0.52%). This may due to that additional data help the models estimate the centroid of each adversarial cluster more precisely, enforcing $\hat{I}(\epsilon)$ to be constrained more effectively and leading to better improvements in robust performance.

5.4 Robustness against Various Perturbation Budgets

Other than showing that our approach outperformed the selected baseline methods on robust accuracy with a fixed perturbation budget $\epsilon = 8/255$, we additionally evaluated our trained DNN against PGD attack with various perturbation budgets ϵ , and compared with the DNN trained by baseline method. In Fig. 5.1, we chose multiple perturbation budgets, either greater or less than the standard one, to attack the DNN trained by vanilla AT with and without constraining on $\hat{I}(\epsilon)$ when the underlying model architecture is PreAct-ResNet-18 and the dataset is CIFAR-10. We found that the model trained with our method

Table 5.2: Experimental results on original CIFAR- $\{10, 100\}$ datasets without synthetic data, are averaged over 3 runs. '+Ours' indicates the use of our $\hat{I}(\epsilon)$ constrained AT. Better results are in **bold**.

Dataset	Model	Method	Clean	PGD	C&W $_{\infty}$	AA
CIFAR-10	PreAct-RN18	Vanilla AT	79.35	52.51	50.76	48.70
		+Ours	81.43	53.04	52.23	49.47
		TRADES	82.21	53.81	51.25	50.05
		+Ours	82.26	54.12	51.65	50.70
		MART	79.81	54.60	50.63	48.73
		+Ours	79.40	54.41	51.25	49.61
	WRN-34-10	TRADES-AWP	81.52	54.40	51.63	50.56
		+Ours	81.73	54.64	52.15	51.07
		Vanilla AT	85.06	56.23	56.00	53.58
		+Ours	85.58	57.25	56.20	54.22
		TRADES	85.54	57.19	56.26	55.03
		+Ours	85.24	57.56	56.60	55.27
CIFAR-100	PreAct-RN18	MART	84.15	57.64	55.40	53.55
		+Ours	83.99	57.20	55.92	53.77
		TRADES-AWP	85.14	58.02	56.66	55.35
		+Ours	84.67	58.25	56.81	55.60
	WRN-34-10	Vanilla AT	52.46	29.08	26.76	24.90
		+Ours	54.13	31.50	27.59	25.70
		TRADES	55.48	28.55	25.57	24.36
		+Ours	57.95	29.92	26.00	24.76
		MART	52.10	30.34	26.48	25.00
		+Ours	51.46	30.70	26.71	25.42
CIFAR-100	PreAct-RN18	TRADES-AWP	56.66	29.47	25.93	25.05
		+Ours	55.81	30.27	26.46	25.54
	WRN-34-10	Vanilla AT	59.13	32.88	30.97	28.67
		+Ours	60.20	33.24	31.44	29.15
		TRADES	59.52	31.09	29.18	27.94
		+Ours	61.25	33.22	29.65	28.43
		MART	57.46	33.28	29.57	28.09
		+Ours	57.07	33.41	30.28	28.69
		TRADES-AWP	60.69	32.40	29.42	28.51
		+Ours	59.83	32.72	29.93	28.98

Table 5.3: Experimental results on CIFAR- $\{10, 100\}$ datasets with additional 1M synthetic data, are averaged over 3 runs. Datasets are denoted as ‘CIFAR- $\{10, 100\}$ S’, respectively. ‘+Ours’ indicates the use of our $\hat{I}(\epsilon)$ constrained AT. Better results are in **bold**.

Dataset	Model	Method	Clean	PGD	C&W $_{\infty}$	AA
CIFAR-10S	PreAct-RN18	Vanilla AT	90.08	57.65	57.24	54.91
		+Ours	89.88	60.31	59.30	56.61
		TRADES	88.48	61.16	58.87	58.13
		+Ours	87.90	61.51	59.80	58.52
		MART	86.99	58.46	54.68	52.86
		+Ours	87.82	57.54	56.05	53.92
		TRADES-AWP	87.69	60.77	58.20	57.54
		+Ours	87.73	61.18	59.07	58.25
	WRN-28-10	Vanilla AT	92.69	63.47	64.06	61.47
		+Ours	92.05	65.56	64.26	61.71
		TRADES	90.26	64.95	63.94	62.73
		+Ours	90.75	65.45	64.77	63.27
		MART	91.91	64.71	63.26	61.21
		+Ours	91.84	63.74	63.56	61.75
		TRADES-AWP	90.44	65.56	63.93	63.17
		+Ours	90.37	66.15	64.53	63.41
CIFAR-100S	PreAct-RN18	Vanilla AT	66.95	32.23	32.32	29.76
		+Ours	65.93	33.51	33.45	30.88
		TRADES	63.83	36.08	32.84	31.92
		+Ours	63.73	36.32	33.08	32.45
		MART	62.44	35.10	31.62	29.87
		+Ours	62.62	36.06	33.33	31.24
		TRADES-AWP	63.71	35.88	32.25	31.59
		+Ours	64.23	36.77	32.95	31.99
	WRN-28-10	Vanilla AT	71.26	35.31	36.31	33.76
		+Ours	71.08	36.62	37.47	34.96
		TRADES	67.52	38.56	36.34	35.36
		+Ours	67.51	39.22	36.91	35.79
		MART	68.84	38.80	37.07	34.85
		+Ours	68.75	39.28	37.93	35.54
		TRADES-AWP	67.25	39.03	36.18	35.39
		+Ours	67.93	39.59	36.79	35.64

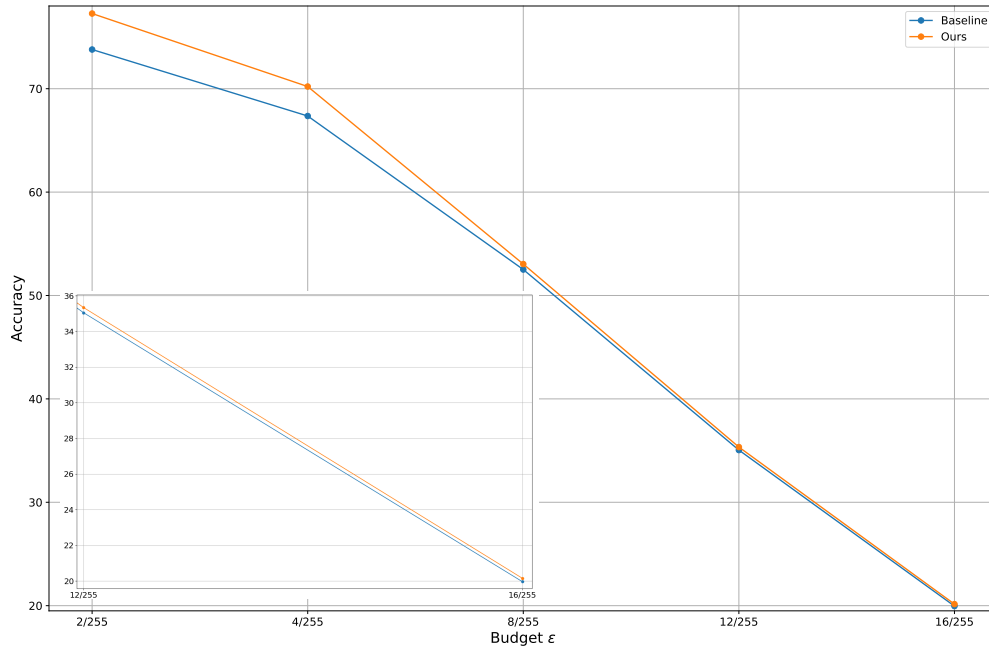


Figure 5.1: Robust accuracy against PGD attack with various perturbation budgets ϵ on original CIFAR-10 when the underlying DNN model is PreAct-ResNet-18.

consistently exhibits better robustness against different perturbation budgets comparing with the baseline counterpart, where the robust accuracy gain is 3.48% when $\epsilon = 2/255$, and the gain is 0.21% when $\epsilon = 16/255$, indicating our method are generally more robust than standard AT methods.

5.5 Simplex Visualization

In this section, we visualize the effect of our adversarial CMI and adversarial separation under different constraining levels when the underlying DNN architecture is PreAct-ResNet-18. We first randomly picked 3 classes from the CIFAR-10 validation set, then focused on a subset of validation samples with these labels perturbed by C&W $_{\infty}$ attack in 40 steps. Each sample from this subset is fed into the DNN, and only the three logits corresponding to the selected 3 labels are retained. These logits are subsequently converted into a 3-dimensional probability vector using the softmax operation. After the steps mentioned above, we further projected all obtained 3-dimensional probability vectors from the

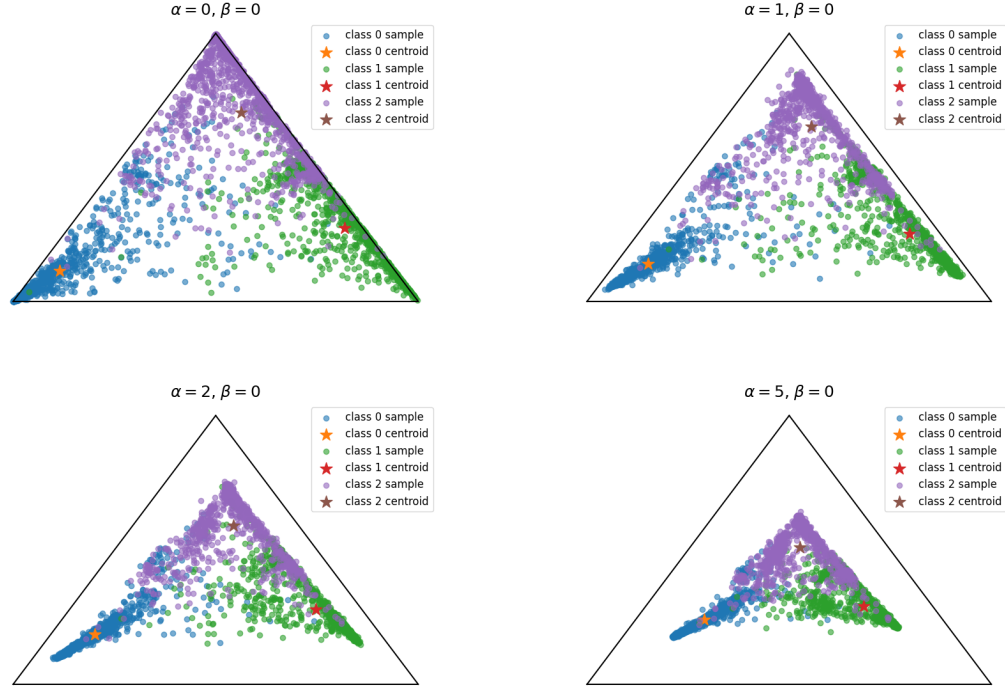


Figure 5.2: Visualization of the output probability simplex on the CIFAR-10 validation set, with varying constraining levels on adversarial CMI (α) and a fixed constraining level on adversarial separation ($\beta = 0$).

samples in the constructed validation subset into a 2-dimensional simplex to visualize the concentration and separation effect of our trained robust DNNs in Fig. 5.2 and 5.3.

We compare the effect of varying constraining levels on adversarial CMI and separation, utilizing vanilla AT as the base adversarial loss function, in Fig. 5.2 and 5.3, respectively. With different constraining levels on adversarial CMI, one can observe that a larger value of α enforce each adversarial cluster more compact around its centroid, while different clusters are closer to each other at the same time. We suggest that this creates a trade-off, as a stronger constraining level on intra-class concentration (adversarial CMI) might weaken the regularization effect on robust accuracy. This also justifies the necessity of constraining on adversarial separation, which further separates perturbed samples with different labels

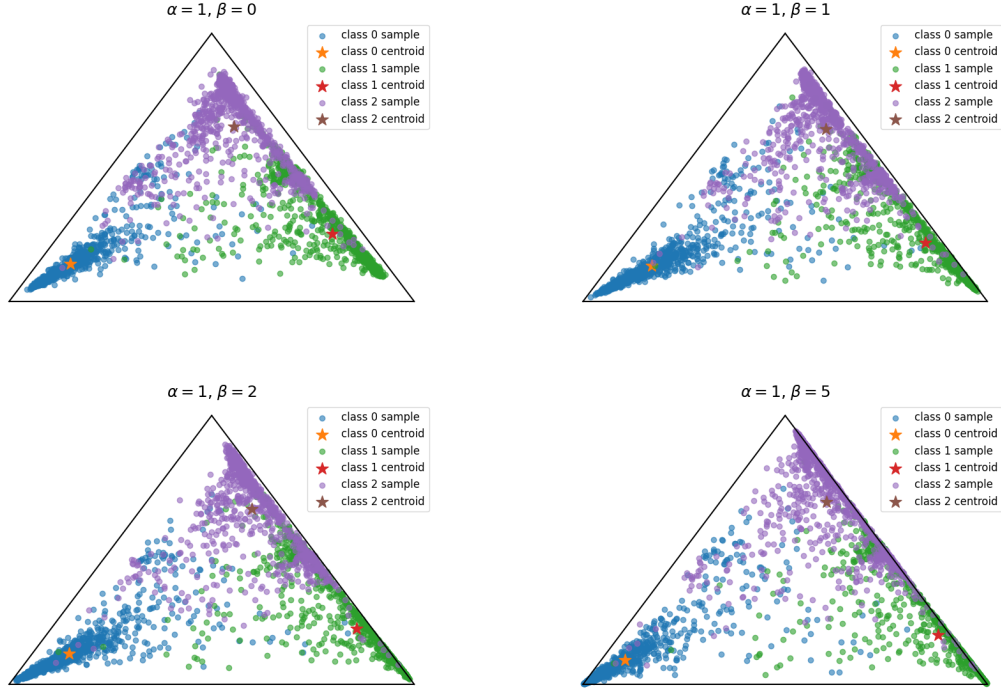
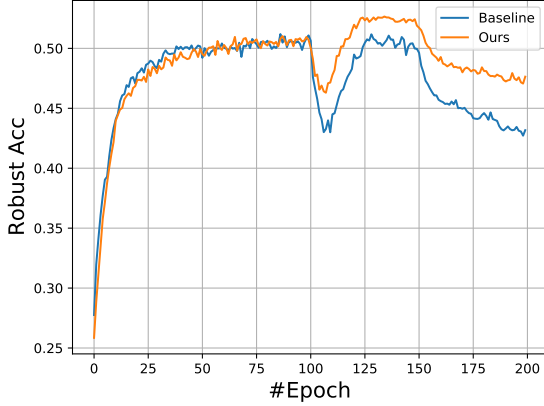


Figure 5.3: Visualization of the output probability simplex on the CIFAR-10 validation set, with varying constraining levels on adversarial separation (β) and a fixed constraining level on adversarial CMI ($\alpha = 1$).

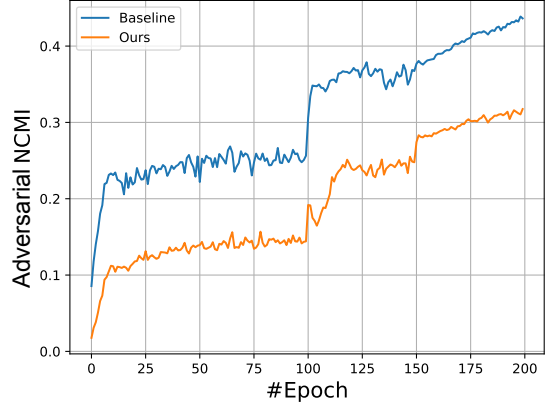
apart. As shown in Fig. 5.3, a stronger constraining level on robust separation (larger β) enlarge the margins between each pair of adversarial clusters, and further push them to the corresponding corner point with a negligible affect on concentration within each adversarial cluster.

5.6 Training Curves

In Fig. 5.4, we present the evolution curves of robust accuracy and adversarial NCMI of PreAct-ResNet-18 on CIFAR-10 validation data perturbed by C&W $_{\infty}$ attack in 40 steps,



(a) Robust Accuracy



(b) Adversarial NCM I

Figure 5.4: Evaluation metric curves of PreAct-ResNet-18 on CIFAR-10 validation set during training. All metrics are computed over validation data after C&W $_{\infty}$ attack with 40 steps. ‘Baseline’ stands for training with vanilla AT [24], and ‘Ours’ represent vanilla AT with robust NCM I regularization.

using loss function of vanilla AT with and without constraining on $\hat{I}(\epsilon)$. In Fig. 5.4a, robust accuracy of vanilla AT with and without our regularization is almost the same before the first learning rate decay at the 100th epoch. After that, the robust accuracy of our method is always better than that of baseline. Moreover, the gap between the two curves increases towards the end of the training phase, indicating that our approach also suffers less from robust overfitting. The curves for adversarial NCM I are displayed in Fig. 5.4b, where the adversarial NCM I of our approach shows a significant gap from the first epoch and remains consistently lower than the baseline. This demonstrates that our method exhibits better intra-class concentration and inter-class separation for perturbed samples. One may notice that in Fig. 5.4b, both curves show an upward trend near the end of training phase. We argue that this is due to robust overfitting, as robust accuracy on validation set of both curves continue to drop after the last learning rate decay at the 150th epoch.

Table 5.4: Experimental results of constraining NCMI on benign samples (denoted as ‘+Clean NCMI’) with TRADES on original CIFAR- $\{10, 100\}$ datasets without synthetic data, in comparison with TRADES and $\hat{I}(\epsilon)$ constrained AT (denoted as ‘+Robust NCMI’), are averaged over 3 runs. Best result of each group are marked in **bold**.

Dataset	Model	Method	Clean	PGD	CW $_{\infty}$	AA
CIFAR-10	PreAct-RN18	TRADES	82.21	53.81	51.25	50.05
		+Robust NCMI	82.26	54.12	51.65	50.70
		+Clean NCMI	81.98	54.94	52.03	50.99
	WRN-34-10	TRADES	85.54	57.19	56.26	55.03
		+Robust NCMI	85.24	57.56	56.60	55.27
		+Clean NCMI	85.39	57.87	56.45	55.22
CIFAR-100	PreAct-RN18	TRADES	55.48	28.55	25.57	24.36
		+Robust NCMI	57.95	29.92	26.00	24.76
		+Clean NCMI	55.93	29.17	26.02	24.64
	WRN-34-10	TRADES	59.52	31.09	29.18	27.94
		+Robust NCMI	61.25	33.22	29.65	28.43
		+Clean NCMI	59.78	31.50	29.52	28.32

5.7 Constraining NCMI over Benign Samples Also Boosts Robustness

In [45], the authors mentioned that in benign image classification task, training a DNN with constraining NCMI on benign samples leads to improved robustness compared to training with only CE loss. Their experimental results indicate that models trained with CMIC-DL framework are more robust against FGSM [9] and PGD attacks [24] under various levels of perturbation budgets than standard DNNs trained with CE loss in MNIST dataset [21]. In our experiments, we also observed that constraining NCMI on benign samples alongside TRADES enhances DNNs’ robustness on CIFAR- $\{10, 100\}$ datasets, both with and without additional synthetic data. We mostly followed the experimental setting of [45], where we additionally sample 8 instances per class (64 in original CIFAR-10) in each mini-batch and use such sampled instances to update centroid of each benign cluster per mini-batch with a WA factor $\tau_Q = 0.999$ in original CIFAR-100. In experiments on CIFAR datasets with an additional 1 million synthetic images, we increased the sample size for centroid updates in each mini-batch by a factor of 4, aligning with the batch size, which was 4 times compared to the original CIFAR datasets. We also adjusted the WA factor τ_Q to 0.99.

Table 5.5: Experimental results of constraining NCMI on benign samples (denoted as ‘+Clean NCMI’) with TRADES on CIFAR- $\{10, 100\}$ datasets with additional 1 million synthetic data, in comparison with TRADES and $\hat{I}(\epsilon)$ constrained AT (denoted as ‘+Robust NCMI’), are averaged over 3 runs. Datasets are denoted as ‘CIFAR- $\{10, 100\}$ S’, respectively. Best result of each group are marked in **bold**.

Dataset	Model	Method	Clean	PGD	CW $_{\infty}$	AA
CIFAR-10S	PreAct-RN18	TRADES	88.48	61.16	58.87	58.13
		+Robust NCMI	87.90	61.51	59.80	58.52
		+Clean NCMI	88.25	61.37	59.40	58.42
	WRN-28-10	TRADES	90.26	64.95	63.94	62.73
		+Robust NCMI	90.75	65.45	64.77	63.27
		+Clean NCMI	90.78	65.27	64.48	63.19
CIFAR-100S	PreAct-RN18	TRADES	63.83	36.08	32.84	31.92
		+Robust NCMI	63.73	36.32	33.08	32.45
		+Clean NCMI	64.69	36.16	32.85	32.04
	WRN-28-10	TRADES	67.52	38.56	36.34	35.36
		+Robust NCMI	67.51	39.22	36.91	35.79
		+Clean NCMI	67.74	39.23	37.21	35.99

Indicated in Table 5.4 and 5.5, constraining NCMI over benign samples also enhances adversarial robustness of a DNN in comparison with models trained with standard TRADES as objective function. To interpret the effectiveness of constraining NCMI over benign samples, we argue that the output probability distributions of a DNN become more compact within each benign sample cluster and better separated among different clusters comparing with the counterpart trained without constraining on benign NCMI. This approach makes it harder for adversarial attacks to create perturbations that would shift a sample into a cluster with a different label in output probability space, thereby improving robust performance. However, we found that constraining NCMI on benign samples cannot outperform directly constraining $\hat{I}(\epsilon)$ in terms of robust accuracy in most cases. We attribute this phenomenon as constraining $\hat{I}(\epsilon)$ takes direct effect on adversarial output probability distribution space rather than benign one, which enhances the robust performance even more.

Chapter 6

Conclusion

In this thesis, we introduced three new information quantities, namely robust CMI, robust separation, and robust NCMI, to evaluate the adversarial robustness of DNNs. Building on these metrics, we developed a novel AT framework along with alternating learning algorithms designed to jointly minimize robust NCMI and standard AT loss functions. Extensive experimental results demonstrate that our approach consistently enhances model robustness against various white-box attacks when integrated with existing adversarial loss functions, highlighting the effectiveness of our method as a ‘plug-and-play’ solution in AT.

References

- [1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.
- [2] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.
- [3] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2021.
- [4] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.
- [5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- [7] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. In *International Conference on Learning Representations*, 2020.
- [8] Samuel G Finlayson, John D Bowers, Joichi Ito, Jonathan L Zittrain, Andrew L Beam, and Isaac S Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019.

- [9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.
- [10] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.
- [11] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34:4218–4233, 2021.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.
- [14] Pavel Izmailov, Dmitrii Podoprikin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 876–885. Association For Uncertainty in Artificial Intelligence (AUAI), 2018.
- [15] Matt Jordan, Justin Lewis, and Alexandros G Dimakis. Provable certificates for adversarial examples: Fitting a ball in the union of polytopes. *Advances in Neural Information Processing Systems*, 32, 2019.
- [16] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.
- [17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- [18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.
- [19] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*, pages 99–112. Chapman and Hall/CRC, 2018.

- [20] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. Technical Report 7, Stanford University, 2015.
- [21] 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.
- [22] Feng Liu, Bo Han, Tongliang Liu, Chen Gong, Gang Niu, Mingyuan Zhou, Masashi Sugiyama, et al. Probabilistic margins for instance reweighting in adversarial training. *Advances in Neural Information Processing Systems*, 34:23258–23269, 2021.
- [23] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.
- [24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.
- [25] Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial defense by restricting the hidden space of deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3385–3394, 2019.
- [26] Ce Qi and Fei Su. Contrastive-center loss for deep neural networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2851–2855. IEEE, 2017.
- [27] Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *International Conference on Learning Representations*, 2022.
- [28] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.
- [29] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.
- [30] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

- [31] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in Neural Information Processing Systems*, 31, 2018.
- [32] Vikash Sehwal, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *International Conference on Learning Representations*, 2022.
- [33] 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.
- [34] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.
- [35] Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? *Advances in Neural Information Processing Systems*, 32, 2019.
- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.
- [37] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020.
- [38] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *International Conference on Machine Learning*, pages 36246–36263. PMLR, 2023.
- [39] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *Computer vision—ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11–14, 2016, proceedings, part VII 14*, pages 499–515. Springer, 2016.
- [40] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018.

- [41] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.
- [42] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.
- [43] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.
- [44] Yuancheng Xu, Yanchao Sun, Micah Goldblum, Tom Goldstein, and Furong Huang. Exploring and exploiting decision boundary dynamics for adversarial robustness. In *International Conference on Learning Representations*, 2023.
- [45] En-Hui Yang, Shayan Mohajer Hamidi, Linfeng Ye, Renhao Tan, and Beverly Yang. Conditional mutual information constrained deep learning for classification. *arXiv preprint arXiv:2309.09123*, 2023.
- [46] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.
- [47] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019.
- [48] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations*, 2021.

APPENDICES

Appendix A

Proof of Theorem 1

Indicated in Fig. 3.1, it follows that $Y \rightarrow X' \rightarrow \hat{Y}'$ forms a Markov chain in the indicated order. Replacing $(X, \hat{Y})$ in Theorem 1 of [45] by $(X', \hat{Y}')$, we have

$$\begin{aligned}
 \xi' = \Pr(\hat{Y}' \neq Y) &\leq \mathbb{E}_{X'} [H(P_{Y|X'}(\cdot|X'), f(X'))] \\
 &= \mathbb{E}_{X'} \left[- \sum_{i=1}^C P_{Y|X'}(i|X') \cdot \ln f(X')[i] \right] \\
 &= \mathbb{E}_{X'Y} [-\ln f(X')[Y]] \\
 &= \mathbb{E}_{XY} [-\ln f(G(X, Y))[Y]].
 \end{aligned} \tag{A.1}$$

Thus,

$$\begin{aligned}
 \xi_r(\epsilon) &= \max_{G: \|x' - x\|_p \leq \epsilon} \Pr(\hat{Y}' \neq Y) \\
 &\leq \max_{G: \|x' - x\|_p \leq \epsilon} \mathbb{E}_{XY} [-\ln f(G(X, Y))[Y]] \\
 &= \max_{G: \|x' - x\|_p \leq \epsilon} \sum_{x, y} P(x, y) (-\ln f(G(x, y))[y]) \\
 &= \sum_{x, y} P(x, y) \max_{G: \|x' - x\|_p \leq \epsilon} (-\ln f(G(x, y))[y]) \\
 &= \sum_{x, y} P(x, y) \left[\max_{x': \|x' - x\|_p \leq \epsilon} -\ln f(x')[y] \right].
 \end{aligned} \tag{A.2}$$

This completes the proof of Theorem 1.

Appendix B

Proof of Theorem 2

For simplicity, we drop the subscript θ from $f_\theta(\cdot)$, and assume that the space of X is discrete. With reference to Figure 3.1, note that

$$P_{\hat{Y}'|XY}(\hat{Y}' = i|x, y) = f(G(x, y))[i]. \quad (\text{B.1})$$

In view of (B.1), we introduce a dummy conditional “backward channel” $\{q(x|i, y)\}$ and derive a new expression for $I(X; \hat{Y}'|Y)$ as follows:

$$I(X; \hat{Y}'|Y) = \sum_y P(y) \sum_x P_{X|Y}(x|y) \sum_{i=1}^C f(G(x, y))[i] \ln \frac{f(G(x, y))[i]}{P_{\hat{Y}'|Y}(i|y)} \quad (\text{B.2})$$

$$\begin{aligned} &= \sum_y P(y) \sum_{i=1}^C P_{\hat{Y}'|Y}(i|y) \sum_x \frac{P_{X|Y}(x|y) \cdot f(G(x, y))[i]}{P_{\hat{Y}'|Y}(i|y)} \\ &\quad \cdot \ln \frac{P_{X|Y}(x|y) \cdot f(G(x, y))[i]}{P_{X|Y}(x|y) P_{\hat{Y}'|Y}(i|y)} \\ &= \sum_y P(y) \sum_{i=1}^C P_{\hat{Y}'|Y}(i|y) \max_{\{q(x|i, y)\}} \sum_x \frac{P_{X|Y}(x|y) \cdot f(G(x, y))[i]}{P_{\hat{Y}'|Y}(i|y)} \cdot \ln \frac{q(x|i, y)}{P_{X|Y}(x|y)} \end{aligned} \quad (\text{B.3})$$

$$= \max_{\{q(x|i, y)\}} \sum_{x, y} P(x, y) \sum_{i=1}^C f(G(x, y))[i] \cdot \ln \frac{q(x|i, y)}{P_{X|Y}(x|y)} \quad (\text{B.4})$$

where (B.2) is due to (B.1), (B.3) follows from the cross entropy inequality, and the maximization in (B.3) and (B.4) is achieved when

$$q(x|i, y) = \frac{P_{X|Y}(x|y) \cdot f(G(x, y))[i]}{P_{\hat{Y}'|Y}(i|y)}. \quad (\text{B.5})$$

Now plugging (B.4) into (3.25), we have:

$$\begin{aligned} I(\epsilon) &= \max_{G: \|x' - x\|_p \leq \epsilon} \max_{\{q(x|i, y)\}} \sum_{x, y} P(x, y) \left[\sum_{i=1}^C f(G(x, y))[i] \cdot \ln q(x|i, y) - \ln P_{X|Y}(x|y) \right] \\ &= \max_{\{q(x|i, y)\}} \sum_{x, y} P(x, y) \left[\max_{G: \|x' - x\|_p \leq \epsilon} \sum_{i=1}^C f(G(x, y))[i] \cdot \ln q(x|i, y) - \ln P_{X|Y}(x|y) \right] \\ &= \max_{\{q(x|i, y)\}} \sum_{x, y} P(x, y) \left[\max_{x': \|x' - x\|_p \leq \epsilon} \sum_{i=1}^C f(x')[i] \cdot \ln q(x|i, y) - \ln P_{X|Y}(x|y) \right] \end{aligned} \quad (\text{B.6})$$

This completes the proof of (3.29). Plugging (B.4) into (3.27), we have:

$$\begin{aligned} \hat{I}(\epsilon) &= \max_{G: \|x' - x\|_p \leq \epsilon} \frac{\max_{\{q(x|i, y)\}} \sum_{x, y} P(x, y) \left[\sum_{i=1}^C f_\theta(G(x, y))[i] \ln q(x|i, y) - \ln P_{X|Y}(x|y) \right]}{\mathbb{E} \left[I_{\{Y \neq V\}} H(f_\theta(G(X, Y)), f_\theta(G(U, V))) \right]} \\ &= \max_{\{q(x|i, y)\}} \max_{G: \|x' - x\|_p \leq \epsilon} \frac{\sum_{x, y} P(x, y) \left[\sum_{i=1}^C f_\theta(G(x, y))[i] \ln q(x|i, y) - \ln P_{X|Y}(x|y) \right]}{\mathbb{E} \left[I_{\{Y \neq V\}} H(f_\theta(G(X, Y)), f_\theta(G(U, V))) \right]} \end{aligned} \quad (\text{B.7})$$

where (B.7) is due to the fact that the denominator therein does not depend on $\{q(x|i, y)\}$. This completes the proof of (3.30) and hence Theorem 2.

Appendix C

Proof of Theorem 3

Go back to Eq. (4.5), we introduce a dummy adversarial centroid distribution Q'_y for each label y and derive a new expression for $\mathbb{E}[D(f_\theta(G(X, Y))\|s'_Y)]$ as follows:

$$\begin{aligned}\mathbb{E}[D(f_\theta(G(X, Y))\|s'_Y)] &= \sum_y P(y) \sum_x P_{X|Y}(x|y) D(f_\theta(G(x, y))\|s'_y) \\ &= \sum_y P(y) \left[\left(\sum_x P_{X|Y}(x|y) D(f_\theta(G(x, y))\|Q'_y) \right) - D(s'_y\|Q'_y) \right]\end{aligned}\tag{C.1}$$

$$= \sum_y P(y) \min_{Q'_y} \left\{ \sum_x P_{X|Y}(x|y) D(f_\theta(G(x, y))\|Q'_y) \right\}\tag{C.2}$$

$$\begin{aligned}&= \min_{\{Q'_c\}_{c \in [C]}} \sum_y P(y) \sum_x P_{X|Y}(x|y) D(f_\theta(G(x, y))\|Q'_y) \\ &= \min_{\{Q'_c\}_{c \in [C]}} \mathbb{E}[D(f_\theta(G(X, Y))\|Q'_Y)]\end{aligned}\tag{C.3}$$

where (C.2) is due to the nonnegativity of the KL divergence $D(s'_y\|Q'_y)$, and the minimization in the above is achieved when

$$Q'_y = s'_y, \forall y \in [C].\tag{C.4}$$

Now plugging (C.3) into (4.5) yields

$$\begin{aligned}
& \min_{\theta} \left\{ \mathbb{E} \left[\max_{G_1: \|\hat{x}-x\|_p \leq \epsilon} \mathcal{L}(X, \hat{X}, Y) \right] + \max_{G: \|x'-x\|_p \leq \epsilon} \left[\alpha \cdot I(X'; \hat{Y}'|Y) - \beta \cdot \Gamma' \right] \right\} \\
&= \min_{\theta} \max_{G_1} \max_G \left\{ \mathbb{E}[\mathcal{L}(X, G_1(X, Y), Y)] + \alpha \cdot \min_{\{Q'_c\}_{c \in [C]}} \mathbb{E}[D(f_{\theta}(G(X, Y)) \| Q'_Y)] \right. \\
&\quad \left. - \beta \cdot \mathbb{E} [I_{\{Y \neq V\}} H(f_{\theta}(G(X, Y)), f_{\theta}(G(U, V)))] \right\} \\
&= \min_{\theta} \max_{G_1, G} \min_{\{Q'_c\}_{c \in [C]}} \left\{ \mathbb{E}[\mathcal{L}(X, G_1(X, Y), Y)] + \alpha \cdot \mathbb{E}[D(f_{\theta}(G(X, Y)) \| Q'_Y)] \right. \\
&\quad \left. - \beta \cdot \mathbb{E} [I_{\{Y \neq V\}} H(f_{\theta}(G(X, Y)), f_{\theta}(G(U, V)))] \right\} \\
&\leq \min_{\theta} \min_{\{Q'_c\}_{c \in [C]}} \max_{G_1, G} \left\{ \mathbb{E}[\mathcal{L}(X, G_1(X, Y), Y)] + \alpha \cdot \mathbb{E}[D(f_{\theta}(G(X, Y)) \| Q'_Y)] \right. \\
&\quad \left. - \beta \cdot \mathbb{E} [I_{\{Y \neq V\}} H(f_{\theta}(G(X, Y)), f_{\theta}(G(U, V)))] \right\}. \tag{C.5}
\end{aligned}$$

This completes the proof of Theorem 3.

Appendix D

Hyper-Parameter Settings

In this chapter, we reported the hyper-parameter settings of our approach, which is the values of α and β in our objective function in Eq. (4.8). Table D.1 and D.2 provide the specific values of α and β used in our experiments on the CIFAR and Tiny-ImageNet datasets, respectively. As one can see, for the same AT method on same dataset, the hyper-parameter settings of our approach are generally the same between different models, while those between methods and datasets are somehow different. We attribute this to the variation in underlying data distributions across datasets and the differences in the DNN’s intrinsic mapping structures between methods, which result in different optimal hyper-parameter settings for the same model architecture on each method and dataset.

Table D.1: Hyper-parameter settings of $\hat{I}(\epsilon)$ constrained AT on CIFAR- $\{10, 100\}$ datasets. Each parenthesis represent the hyper-parameter value of α and β in our approach, respectively.

Dataset	Method	No Synthetic Data		1M Synthetic Data	
		PreAct-RN18	WRN-34-10	PreAct-RN18	WRN-28-10
CIFAR-10	Vanilla AT	(1, 2)	(1, 2)	(2, 0.1)	(2, 1)
	TRADES	(0.01, 0.1)	(0.01, 0.1)	(0.1, 2)	(0.1, 2)
	MART	(0.01, 2)	(0.1, 1)	(0.1, 1)	(0.1, 1)
	TRADES-AWP	(0.1, 1)	(0.1, 1)	(0.1, 2)	(0.1, 1)
CIFAR-100	Vanilla AT	(2, 5)	(2, 5)	(0.1, 2)	(0.1, 2)
	TRADES	(1, 2)	(1, 2)	(0.01, 5)	(0.01, 2)
	MART	(0.1, 2)	(0.1, 0.01)	(0.1, 5)	(0.1, 2)
	TRADES-AWP	(0.01, 5)	(0.01, 5)	(0.1, 2)	(0.1, 1)

Table D.2: Hyper-parameter settings of $\hat{I}(\epsilon)$ constrained AT on Tiny-ImageNet datasets. Each parenthesis represent the hyper-parameter value of α and β in our approach, respectively.

Method	Model	
	PreAct-RN18	PreAct-RN34
Vanilla AT	(1, 0.1)	(1, 0.1)
TRADES	(0.01, 2)	(0.01, 2)