Understanding Robustness in Teacher-Student Setting: A New Perspective

Zhuolin Yang[*]
UIUC

Zhaoxi Chen
Tsinghua University

Tiffany (Tianhui) Cai
Columbia University

Xinyun Chen
UC Berkeley

Bo Li
UIUC

Yuandong Tian[*]
Facebook AI Research

Abstract

Adversarial examples have appeared as a ubiquitous property of machine learning models where bounded adversarial perturbation could mislead the models to make arbitrarily incorrect predictions. Such examples provide a way to assess the robustness of machine learning models as well as a proxy for understanding the model training process. There have been extensive studies trying to explain the existence of adversarial examples and provide ways to improve model robustness, e.g., adversarial training. Different from prior works that mostly focus on models trained on datasets with predefined labels, we leverage the teacher-student framework and assume a teacher model, or *oracle*, to provide the labels for given instances. In this setting, we extend Tian (2019) in the case of low-rank input data, and show that *student specialization* (the trained student neuron is highly correlated with certain teacher neuron at the same layer) still happens within the input subspace, but the teacher and student nodes could *differ wildly* out of the data subspace, which we conjecture leads to adversarial examples. Extensive experiments show that student specialization correlates strongly with model robustness in different scenarios, including students trained via standard training, adversarial training, confidence-calibrated adversarial training, and training with the robust feature dataset. Our studies could shed light on the future exploration of adversarial examples, and potential approaches to enhance model robustness via principled data augmentation.

1 Introduction

The existence of adversarial examples is an intriguing and important phenomenon in deep learning. Understanding why such examples exist can lead to (1) more robust architectures and training algorithms usable in the real world, and (2) better understanding of network training and learned representations.

Many previous works on adversarial examples (Goodfellow et al., 2014; Szegedy et al., 2013) focus on the standard setting of supervised classification learning in which a network is trained on a fixed dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}$, where $\mathbf{x}_i$ is a high-dimensional input feature and $\mathbf{y}_i$ is its label (continuous or discrete). While general, the worst-case scenario (i.e., random label $\mathbf{y}_i$) may lead to exponentially many adversarial examples since every corner of the input space needs to be covered, which might never happen in practice.

In this paper, we take a novel perspective to study adversarial examples with the *teacher-student* formulation. In this setting, we have a teacher network f^* as an *oracle* network to provide the true label $\mathbf{y}_i$ given the input $\mathbf{x}_i$, i.e. $\mathbf{y}_i = f^*(\mathbf{x}_i)$. By definition, there is *no* adversarial examples for the teacher. For a student network f, while $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}^*(\mathbf{x})\|$ remains small when $\mathbf{x} \in \mathcal{D}$, the *adversarial samples* $\mathbf{x}'$ for f have large $\|\mathbf{f}(\mathbf{x}') - \mathbf{f}^*(\mathbf{x}')\|$ while in the local neighborhood of $\mathbf{x}$.

This teacher-student assumption imposes implicit realizable constraints for $(\mathbf{x}_i, \mathbf{y}_i)$ pairs. Using the teacher as the *reference network*, we open the black-box mapping $\mathbf{x} \mapsto \mathbf{y}$, and more in-depth analysis can be performed. Moreover, such a setting has interesting properties (Tian, 2019): with full-rank and sufficient input, student nodes in multi-layer ReLU networks are *specialized* to teacher nodes at the same layer after training (both networks have the same depth). Also, there exist unspecialized student nodes in the final trained student model. We hypothesize that the existence of such nodes is the source of the non-robustness of a trained model, which opens a new way to study robustness and

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

adversarial samples.

In this work, we extend Tian (2019) to handle the low-rank dataset and use **N**ormalized **C**orrelation (NC) between teacher and student nodes as an additional signal to study adversarial robustness of the student network. We analyze the cause of adversarial samples and show positive correlations between NC and robustness: **(1)** Theoretically, we show that student specialization happens in the low-dimensional input, and specify the conditions for unspecialized nodes. **(2)** Empirically, we show that high NC is correlated to strong adversarial robustness, verified under different scenarios such as the comparison of student network with standard training, adversarial training, adversarial training with CCAT strategy (Stutz et al., 2019), and model trained with robust feature dataset (Ilyas et al., 2019a).

The teacher-student framework provides a quantitative way to understand the existence of adversarial examples in low-dimensional subspace, and a quantitative measurement (i.e., node specialization) that indicates model robustness. Our analysis also confirms several existing observations about adversarial examples (Ilyas et al., 2019a; Stutz et al., 2019; Khoury and Hadfield-Menell, 2018) from the teacher-student framework perspective.

2 Related Works

Adversarial examples. Recent studies have shown that deep neural networks are vulnerable to adversarial examples, which are carefully crafted inputs aiming to mislead well-trained ML models (Goodfellow et al., 2014; Szegedy et al., 2013). Since adversarial examples have raised many security concerns for ML models, different studies have been conducted to analyze its properties, such as the reasons for their existence (Shamir et al., 2019; Shi and Ding, 2019; Ilyas et al., 2019b; Gu and Tresp, 2019; Tsipras et al., 2018; Kotyan et al., 2019), adversarial transferability (Tramèr et al., 2017; Papernot et al., 2016; Bhagoji et al., 2018), and compactness of adversarial regions (Singh et al., 2018; Chen et al., 2020; Tabacof and Valle, 2016). Approaches to generate such adversarial examples have also been proposed using different perturbation measurement metrics and generative models, including both $\mathcal{L}_p$ bounded and unrestricted attacks (Wong et al., 2019; Bhattad et al., 2019; Xiao et al., 2018a,b; Athalye et al., 2018; Vargas and Su, 2019). However, given these rich studies on adversarial examples, it remains an open question on why a small magnitude of perturbation is enough to fool a DNN model effectively and what roles the model architecture and intermediate representation play in these attacks given the complexity of a human-labeled "natural" dataset. We make the first attempt to investigate such questions from a different perspective, using the teacher-student framework to provide controllable constraints for the ground-truth dataset labels.

Several defense approaches have been proposed against adversarial attacks, and one of the most effective methods is *adversarial training* (Madry et al., 2017). Different variations for adversarial training have been studied to improve its efficiency and scalability (Shafahi et al., 2019; Xie et al., 2020), as well as understand its limitations (Zhang et al., 2019; Kang et al., 2019). As adversarial training has achieved promising empirical performance by improving ML robustness, we aim to leverage the teacher-student framework to provide theoretical observations on why adversarial training defends against adversarial attacks and how the intermediate representation changes after adversarial training.

Teacher-student setting. The teacher-student setting is an old topic (Engel and Van den Broeck, 2001; Saad and Solla, 1996; Mace and Coolen, 1998; Freeman and Saad, 1997; Gardner and Derrida, 1989). Recent work has analyzed the specialization of the student nodes towards that of the teacher for 2-layer networks (Goldt et al., 2019; Aubin et al., 2018), and Allen-Zhu et al. (2019) has shown the analysis for 2 and 3 layer networks with modified SGD, batch size 1, and heavy over-parameterization. Later Tian (2019) shows that the student neuron specialization happens around SGD critical points in the lowest layer for deep ReLU networks without parametric assumption, and provides polynomial sample complexity for 2 layer ReLU networks. In this work, we use the teacher as an "oracle" to provide an in-depth understanding of adversarial examples generated against the corresponding student model due to the fact that some student nodes fail to specialize fully to the teacher.

3 Teacher-Student Setting in Low-Dimensional Input

3.1 Teacher network assumptions

Let f^* be the teacher and f be the student. The label $\mathbf{y}_i$ of each $\mathbf{x}_i$ from a finite dataset $(\mathbf{x}_i, \mathbf{y}_i)$ is given by the teacher network f^*:

$$\mathbf{y}_i = f^*(\mathbf{x}_i), \quad i = 1 \dots N \tag{1}$$

As an example of how teacher-student setting connects adversarial samples and robustness, in the theoretical analysis, we consider both f^* and f to be two-layer networks with ReLU activation and L2 loss function.

Note that our setting is different from network distillation (Hinton et al., 2015), where both teacher and student are *trainable* networks given the data. In this paper, the teacher network represents *an oracle* that gives the ground truth labels. Therefore, by definition,

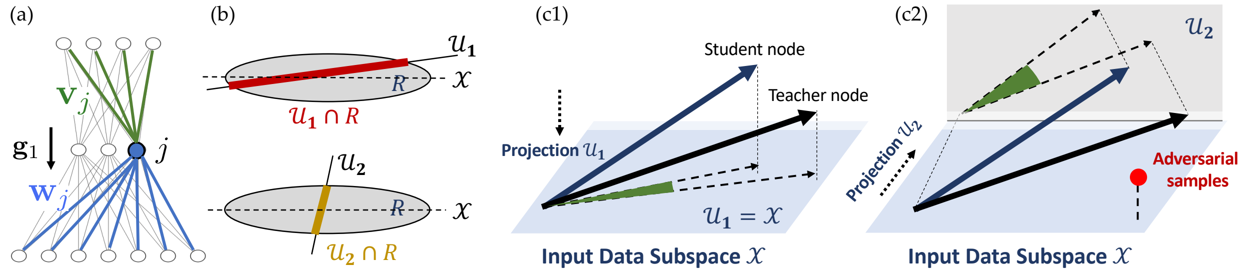

Figure 1: Student Specialization in Low-rank dataset. **(a)** Setting of two-layered network (Sec. 3.2) and notations. $\mathbf{g}_1$ is the backpropagated gradient through the hidden layer. For a node/neuron j, its input weight is $\mathbf{w}_j$ and fan-out weight $\mathbf{v}_j$. **(b)** Radius of inscribed ball (1-dimension) of the intersection of a subspace $\mathcal{U}$ and input data region R. The radius is large if $\mathcal{U}$ aligns with the high-rank direction of data region. **(c1)** Within the data subspace $\mathcal{X}$, the student and teacher node has small projected angle (i.e., angle between projected weights); **(c2)** If we project the weights to subspace $\mathcal{U}_2 \perp \mathcal{X}$, then the projected angle between student and teacher weight vectors remain large, due to limited data outside $\mathcal{X}$. In this case, student response of data out of $\mathcal{X}$ can be very different from the teacher, yielding adversarial samples.

no adversarial examples exist for the teacher network.

3.2 Two-layer student specialization in low-rank setting

Notation. For each hidden node j in the student network, let $\mathbf{w}_j$ be its incoming weight and $\mathbf{v}_j \in \mathbb{R}^C$ its fan-out weights, where C is the output dimension of both teacher and student (Figure 1(a)). Note that for d-dimensional input, $\mathbf{w}_j := [\tilde{\mathbf{w}}_j; b] \in \mathbb{R}^{d+1}$ includes both the weight and the bias. Correspondingly, the input $\mathbf{x} = [\tilde{\mathbf{x}}; 1] \in \mathbb{R}^{d+1}$, where $\tilde{\mathbf{x}} \in \mathbb{R}^d$ is the actual sample. For teacher node, we have $\mathbf{w}_j^*$ and $\mathbf{v}_j^*$ respectively. Let $\mathbf{g}_1$ be the backpropagated gradient at the student hidden layer and K be the total number of hidden nodes (neurons) for teacher and student.

We consider the situation where the **training has already been done**, characterized by the the condition $\|\mathbf{g}_1\|_\infty < \epsilon$. Note that for mathematical convenience, the condition is *stronger* than usual convergence: $\|\mathbf{g}_1\|_\infty < \epsilon$ means that the gradient is small at *every* data point in the data region R that has *infinite* samples. This ideal setting facilitates our analysis.

One interesting phenomenon given the condition $\|\mathbf{g}_1\|_\infty < \epsilon$, or in the extreme case $\mathbf{g}_1 = \mathbf{0}$, is *student specialization* (Tian, 2019); that is, when the input data distribution is *full-rank*, for each teacher node j, there exists at least one student k whose weight is co-linear with the teacher: for some $\lambda > 0$, $\mathbf{w}_j^* = \lambda \mathbf{w}_k$. (c.f., Theorem.1 in Tian (2019)). This means that the student completely recovers the teacher's internal information upon convergence through training.

A more interesting and realistic situation is when the input data R lie in a low-dimensional space $\mathcal{X}$. In this case, a perfect recovery is impossible, since there could exist multiple teachers satisfying Eqn. 1. For example, if f^* is such a teacher, then for any weight $\mathbf{w}_j^*$ in the lowest layer of f^*, there exists another teacher $f^{*'}$ with $\mathbf{w}_j^{*'} = \mathbf{w}_j^* + \delta\mathbf{w}$, where $\delta\mathbf{w} \perp \mathcal{X}$, and $f^{*'}$ also satisfies Eqn. 1. Hence, we do not expect a full-specialization, but a *partial* one in the input space $\mathcal{X}$. Note that we use the concept of *observation* between two nodes j and k, which is a technical condition in (Tian, 2019) [1].

Theorem 1 (Partial Specialization for Infinite Low-Dimensional Input). *If the input dataset $R \subseteq \mathcal{X}$, then when the gradient $\mathbf{g}_1 = \mathbf{0}$, for each teacher node j observed by any student node, there exists a student node k so that $\mathrm{Proj}_{\mathcal{X}}[\mathbf{w}_k] = \lambda \mathrm{Proj}_{\mathcal{X}}[\mathbf{w}_j^*]$ for some $\lambda > 0$.*

See Appendix A.2 for the proof. Theorem 1 means that the weight $\mathbf{w}_k$ of a specialized student node can be decomposed into two components: $\mathbf{w}_k = \lambda \mathrm{Proj}_{\mathcal{X}}[\mathbf{w}_j^*] + \mathbf{w}_k^e$, where the first term is the useful (specialized) component of $\mathbf{w}_k$. The second term $\mathbf{w}_k^e$ is the component that is orthogonal to the subspace $\mathcal{X}$. Note that $\mathbf{w}_k^e$ is affected by initialization and $\|\mathbf{w}_k^e\|$ can be arbitrarily large while not affecting the output of f, given its input is within $\mathcal{X}$.

For the realistic case when the gradient is small but nonzero and the input data R is "almost" low-rank, what would happen? To characterize the low-rank structure, we consider the *radius of the largest inscribed ball* in $\mathcal{U} \cap R$, $r(\mathcal{U} \cap R)$, with an arbitrary subspace $\mathcal{U}$. If $\mathcal{U}$ is aligned with the high-rank structure of R, then $r(\mathcal{U} \cap R)$ is large, otherwise small (Figure 1(b)). Here $\alpha_{jk} := \mathbf{v}_j^{*\intercal} \mathbf{v}_k$ is the inner product between the teacher and the student fan-out weights:

Theorem 2 (Specialization of Projected Weights in Low-Dimensional Input). *When $\|\mathbf{g}_1\|_\infty \le \epsilon$, for each teacher node j observed by a student k, there exists a student node k' so that for projected weight $\tilde{\mathbf{p}}_{k'} := \mathrm{Proj}_{\mathcal{U}}[\tilde{\mathbf{w}}_{k'}]$ and $\tilde{\mathbf{p}}_j^* := \mathrm{Proj}_{\mathcal{U}}[\tilde{\mathbf{w}}_j^*]$, their angle $\theta_{jk'}^{\mathcal{U}} := \arccos(\tilde{\mathbf{p}}_{k'}^\intercal \tilde{\mathbf{p}}_j^*)$ satisfies $\sin(\theta_{jk'}^{\mathcal{U}}) \le M_j(\mathcal{U}) K \epsilon / \alpha_{jk}$, where $M_j(\mathcal{U}) := \mathcal{O}(r^{-1}(\mathcal{U} \cap R \cap \partial E_j))$.*

[1] A node j is *observed* by a node k, if the boundary of j is in the active region of k: $\partial E_j \cap E_k \neq \emptyset$. Here $E_j := \{\mathbf{x} : \mathbf{w}_j^\intercal \mathbf{x} \ge 0\}$ is the activation region and $\partial E_j := \{\mathbf{x} : \mathbf{w}_j^\intercal \mathbf{x} = 0\}$ is its boundary.

Please check Appendix A.6 for the proof. From Theorem 2, we can see that large radius $r(\mathcal{U} \cap R)$ and large $\|\mathbf{v}_j^*\|$ (and thus large α_{jk}) yield tighter bound of specialization error. When the subspace $\mathcal{U}$ aligns with the main direction of R (or $\mathcal{X}$), the inscribed radius r is large, the projected angles $\theta_{jk'}^{\mathcal{U}}$ between weight vectors are small and the alignment is good (Figure 1(c1)). On the other hand, if $\mathcal{U} \perp \mathcal{X}$, the radius becomes tiny (Figure 1(b)) and the projected angle has a much looser bound (Figure 1(c2)). Empirically, the projected angle often remains large even after many epochs of training.

In addition, Tian (2019) pointed out that there are *unspecialized* nodes, i.e., neurons that are not aligned with any teacher or student node, and their fan-out weights are zero and thus prunable. It happens in the low-dimensional input and small gradient case as well:

Corollary 1 (Unspecialized nodes in Low-Dimensional Input). *If $\|\mathbf{g}_1\|_\infty \le \epsilon$, a student node k' is observed by other student nodes with fan-out weights $Q = [\mathbf{v}_{k_1}, \mathbf{v}_{k_2}, \ldots, \mathbf{v}_{k_C}]$, and has projected angle $\sin(\theta_{jk'}^{\mathcal{U}}) \ge c_0$ with other teacher/student node j, then its fan-out weight is small: $\|\mathbf{v}_{k'}\|_2 \le \|Q^{-1}\|_1 M_{k'}(\mathcal{U}) K\epsilon / c_0$.*

4 Adversarial Training in the Teacher-Student Setting

As the main contribution, we now use our teacher-student framework to analyze various adversarial phenomena. To see why adversarial training is related to the teacher-student setting, one example is the experiments in Ilyas et al. (2019a) that show an intriguing property of adversarial examples: using the adversarial examples $\mathbf{x}'$ and their "wrong" labels $f^*(\mathbf{x}')$ (i.e., non-robust dataset in their Sec 3.2), we can train a student model f that performs well in the original test set.

While this sounds like "garbage-in signal-out", our teacher-student setting explains it naturally. The label $f^*(\mathbf{x}')$ is from the output of the teacher f^* on an adversarial sample $\mathbf{x}'$. While this label is regarded as "wrong" from the dataset point of view (since $\|f^*(\mathbf{x}') - f^*(\mathbf{x})\|$ is large, where $\mathbf{x}$ is the data point before adversarial perturbation), from our teacher-student perspective, the input-output pair $(\mathbf{x}', f^*(\mathbf{x}'))$ preserves the correct mapping of the teacher, *regardless* of the nature of the input data. No wonder the trained student does well on the original test set, if the teacher does well.

With the teacher-student framework, we revise the concept of adversarial examples and analyze its properties.

4.1 An empirical model for learned students

Theorems in Sec. 3.2 tell that a learned student model on low-rank data has two properties:

(1) The student weight $\mathbf{w}_k$ has large discrepancy from teacher weights along directions $d \perp \mathcal{X}$ (Theorem 2);

(2) If the student weight $\mathbf{w}_k$ deviates from all teachers and student nodes within the data region R, then the magnitude of its fan-out weight is small (Corollary 1).[2]

Note that for convenience, we omit technical conditions (e.g., the boundary needs to be observed). In the over-realization scenario, we assume that any boundary is always observed by many student nodes.

Based on these two properties, we could come up with an *empirical* model to relate a *learned* student network with the teacher (here $\mathbf{w}_k$ and $\mathbf{w}_j^*$ are normalized):

$$\mathbf{w}_k = \mathbf{w}_j^* + \epsilon_{\text{in}} \mathbf{u}_k^{\text{in}} + \epsilon_{\text{out}} \mathbf{u}_k^{\text{out}} \tag{2}$$

where $\mathbf{u}_k^{\text{in}} \in \mathcal{X}$ and $\mathbf{u}_k^{\text{out}} \perp \mathcal{X}$ are unit vectors. $\epsilon_{\text{in}} = \epsilon_{\text{out}} = 0$ means perfect student specialization.

Here the magnitudes of ϵ_{in} and ϵ_{out} are related to different factors. ϵ_{out} is related to the degree of low-rankness of the data. The more the data are rank-deficient, the smaller the supporting radius $r(\mathcal{U}, R)$ for out-of-plane subspace $\mathcal{U}$, and the bound becomes looser according to Theorem 2. This leads to larger ϵ_{out} that perturbs student node away from the teacher along the direction of out-of-distribution.

On the other hand, ϵ_{in} depends on the magnitude of the fan-out weights. When the student node k is *unspecialized*, i.e., it strays away from teacher and other students' directions (large ϵ_{in}), Corollary 1 tells that its fan-out weight is small and therefore its influence to the output of the network is limited and/or negligible.

The two unit-vectors $\mathbf{u}_k^{\text{in}}$ and $\mathbf{u}_k^{\text{out}}$ could be dependent on the network initialization and the training process.

Checking specialization of nodes. There are two different ways for checking student specialization.

Weight-check. One method is to directly check whether $\mathbf{w}_j^* = \lambda \mathbf{w}_k$ for some $\lambda > 0$. While straightforward, an issue is that for intermediate layers of deep models, the input dimension of a node can be different between the teacher and an over-parameterized student.

Activation-check. Alternatively, we could use activation $\mathbf{f}_j \in \mathbb{R}^N$ computed on a given dataset of size N, as in Tian (2019). By checking the Normalized Correlation between $\mathbf{f}_j^*$ from the teacher and $\mathbf{f}_k$ from the student, we could measure the degree of specialization.

One short-coming for activation-check is that a perfect alignment with a low-dimensional input only tells that $\text{Proj}_{\mathcal{X}}[\mathbf{w}_j^*] = \lambda \text{Proj}_{\mathcal{X}}[\mathbf{w}_k]$, which means that $\epsilon_{\text{in}} = 0$. On the other hand, to check ϵ_{out}, we would need to use

[2] We leave one case for future work: two student nodes are both away from all other teacher/student nodes, and they both have strong fan-out weights.

data that are out of the subspace of $\mathcal{X}$ (e.g., adversarial samples, adding noise to the input, etc).

4.2 Adversarial examples in the teacher-student setting

Eqn. 2 serves as an empirical model of the possible vulnerability of a learned student model compared to its teacher, due to ϵ_{in} and ϵ_{out}. **First**, for a sample $\mathbf{x}'$ out of the plane $\mathcal{X}$, a high ϵ_{out} leads to large activation difference between the teacher and the student. This aligns with the existing hypothesis and understanding (Khoury and Hadfield-Menell, 2018; Ma et al., 2018) that directions off the data manifold can be used to construct adversarial examples. **Second**, we might also have in-plane adversarial samples that attack through $\mathbf{u}^{\text{in}}$.

We use adversarial samples as a probe to verify our empirical model and the induced vulnerability. Since we now have a teacher network that provides the ground truth label (in addition to the data label), there are two different ways to obtain an adversarial sample.

Oracle-adversarial. We define *oracle-adversarial* examples as follows:

$$\mathbf{x}' = \arg\max_{\mathbf{x}'\in B(\mathbf{x},\epsilon)} L[f(\mathbf{x}'), f^*(\mathbf{x}')], \tag{3}$$

where $L[\cdot]$ is a loss function (e.g., L_2, cross-entropy, etc). $\mathbf{x}'$ can be obtained by back-propagating through both f and f^*. We call $\mathbf{x}'$ the *oracle-adversarial example* and $f^*(\mathbf{x}')$ the *oracle label*.

Data-adversarial. The conventional formulation of (untargeted) adversarial examples is

$$\mathbf{x}' = \arg\max_{\mathbf{x}'\in B(\mathbf{x},\epsilon)} L[f(\mathbf{x}'), \mathbf{y}], \tag{4}$$

where $\mathbf{y}$ is the ground truth label from the dataset, and L is commonly cross-entropy for classification. Here, we only obtain samples against the teacher network f^* for the training set $\mathcal{D}$, and assume that f^* is a constant function in $B(\mathbf{x}_i, \epsilon)$, where $\mathbf{x}_i$ is a sample in the training set. Since f^* is constant in $B(\mathbf{x}_i, \epsilon)$, we use the label $\mathbf{y}_i = f^*(\mathbf{x}_i)$ of the *original* data point $\mathbf{x}_i$ when optimizing Eqn. 4, and only backpropagte through the student model f. In this paper, we call such an adversarial examples $\mathbf{x}'$ *data-adversarial*.

In the presence of the teacher network, there are two ways to do adversarial training. Let $\mathbf{x}'$ be the perturbed sample. For *label-target*, we simply use the label $\mathbf{y}$ of the original sample $\mathbf{x}$ to update: $\theta_{t+1} \leftarrow \theta_t - \alpha\nabla_\theta L[f_{\theta_t}(\mathbf{x}'), \mathbf{y}]$. Alternatively, we could also use teacher output $f^*(\mathbf{x}')$ as the label of $\mathbf{x}'$ and update: $\theta_{t+1} \leftarrow \theta_t - \alpha\nabla_\theta L[f_{\theta_t}(\mathbf{x}'), f(\mathbf{x}')]$. We call it *teacher-target*. It incorporates the deviation of $\mathbf{x}'$ from $\mathbf{x}$ and thus is more accurate than label-target.

4.3 Why adversarial training helps model robustness?

Given all the previous analysis, it is now clear that by adding adversarial samples during training, we implicitly augment data region R along its "weak" directions and thus improve student specialization (Theorem 2 and Corollary 1). Similar effects can also be achieved by data augmentation and/or adding noise. In the next section, we will verify these findings with extensive experiments.

5 Experiments

In this section, we aim to verify the strong positive correlation between the student specialization and the robustness of the student model with respect to the oracle (i.e., the teacher) in various scenarios.

We control the degree of specialization by training the student model with different epochs, as well as using *adversarial training* adapted for the teacher-student framework. In addition, we conduct studies on Confidence-Calibrated Adversarial Training (CCAT) (Stutz et al., 2019) to further verify the relationship between neuron specialization and model robustness. We also discussed the robust feature (Madry et al., 2017) in our teacher-student setting and left the details to Appendix A.11.

5.1 Experimental setup

We use CIFAR-10 (Krizhevsky et al., 2009) as our dataset in experiments, and consider both the teacher and student model to be the 4-layer Conv ReLU networks. We train the teacher with channel size $64 - 64 - 64 - 64$ at first, and then reduce it to be $45 - 32 - 32 - 20$ by pruning the inactivated channels[3]. For the student model, we set it to be 1.1x scale to the pruned teacher model (i.e. channel size $50-35-35-22$). We also investigate deeper Conv network structure by adding one more Conv layer with channel size as 64 to further solidify our conclusion. We set each Conv layers' kernel size $s = 3$ for both teacher and student models.

In our experiments, we consider two **Standard Training (ST)** strategies. **Logit training**: minimize the ℓ_2 distance between the teacher and student' output logits. **Label training**: minimize the cross-entropy between the student's logit and the teacher's prediction. We also consider **Adversarial Training (AT)** by training the student with *oracle-adversarial examples* generated with Eq.(3), where we apply the 40-iteration l_∞ PGD attack with perturbation scale $\epsilon = 10/255$ and step

[3] We define the channel k to be inactivated by considering the norm of the fan-out weights.

size $\alpha = 0.01$.

5.2 Evaluation metrics

We use the **Normalized Correlation (NC)** (Tian et al., 2019) and its variants to measure the neuron specialization of the student to the teacher. Basically, we define $\mathbf{f}_i$ to be the activations of node i. For student's node k and teacher's node j, ρ_{kj} is defined as the cosine similarity between the normalized activations: $\rho_{kj} = \tilde{\mathbf{f}}_k^\top \tilde{\mathbf{f}}_j^*$, where $\tilde{\mathbf{f}}_k = (\mathbf{f}_k - \mathrm{mean}(\mathbf{f}_k))/\mathrm{std}(\mathbf{f}_k)$. Then we define the variants of the NC as follows:

Best Normalized Correlation (BNC) $\hat{\rho}_j$: For each teacher node j in layer l, we find the highest NC among student's l-th layer nodes (l_s): $\hat{\rho}_j = \max_{k \in l_s} \rho_{kj}$.

Mean of the Best Normalized Correlation (MBNC) $\bar{\rho}_l$: We compute the mean of the BNC $\hat{\rho}$ over teacher's l-th layer nodes (l_t): $\bar{\rho}_l = \mathrm{mean}_{j \in l_t} \hat{\rho}_j$.

We also show the **Sorted BNC Curve** by sorting the BNC $\hat{\rho}$ of the teacher's nodes and concatenating the adjacents. Then we can compare students' alignment to one teacher by visualizing the curves for each layer.

5.3 Warm-up: strong correlation between ϵ_{in}, ϵ_{out} and normalized correlation

First, we report ϵ_{in} and ϵ_{out} in Eqn. 2 between the student and teacher nodes in the lowest (first Conv) layer, and study its correlation with Normalized Correlation. This is to validate our empirical model (Sec. 4.1) and lay the foundation of our next analysis.

With the lowest layer's kernel size $s = 3$, each input with shape $(3, 32, 32)$ can be decomposed into 30×30 patches, and each patch has $3 \times 3 \times 3 = 27$ dimensions. To show the inputs' low-rank property, we perform PCA(Pearson, 1901) on the 27-dimensional inputs, and the fast-decaying eigenvalues (Figure 2) show their low-rank structure. We choose the eigenvectors with 17 largest eigenvalues to form the basis $\mathbf{U}$ of the input distribution $\mathcal{X}$, and compute ϵ_{in} and ϵ_{out} between student node k and teacher node j as follows. Note that here we define $\Delta\mathbf{w}_{jk} := \mathbf{w}_k/\|\mathbf{w}_k\|_2 - \mathbf{w}_j^*/\|\mathbf{w}_j^*\|_2$ (Following Sec. 4.1, both $\mathbf{w}_k$ and $\mathbf{w}_j^*$ need to be normalized):

$$\epsilon_{\text{in}}[k, j] = \|\mathbf{U}\mathbf{U}^\top \Delta\mathbf{w}_{jk}\|_2,$$
$$\epsilon_{\text{out}}[k, j] = \|(\mathbf{I} - \mathbf{U}\mathbf{U}^\top)\Delta\mathbf{w}_{jk}\|_2$$

To show the correlation between ϵ_{in} and NC, we use standard training and plot $(\rho_{kj}, \epsilon_{\text{in}}[k, j])$ for every pair of k and j in Figure 2. We show strong negative correlation trends interpreted by Pearson score: small $\epsilon_{\text{in}}[k, j]$ indicates large NC . We draw the ϵ_{in} and ϵ_{out} curve by sorting the $\epsilon_{\text{in}}[k, j], \epsilon_{\text{out}}$ value between every teacher node j and the student node k with the highest NC. $\epsilon_{\text{in}}, \epsilon_{\text{out}}$ curves show how well the student is specialized to the teacher from the in/out-plane direction.

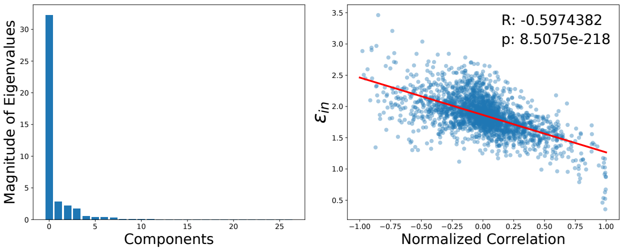

Figure 2: **Left**: Magnitude of Eigenvalues on each PCA components of the input distribution $\mathcal{X}$. **Right**: Correlation between ϵ_{in} and **NC** under standard training.

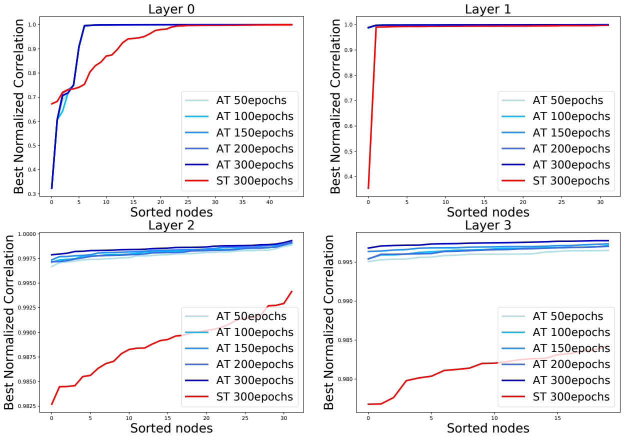

Figure 3: Sorted BNC curve for Adversarial Training (AT) and Standard Training (ST) with logit for 300 epochs.

5.4 Adversarial training

In this subsection, we analyze how the Adversarial Training (AT) affects the model robustness and student specialization, measured by normalized correlation.

We run AT for different training epochs $T \in \{50, 100, 150, 200, 300\}$, and compare them to Standard Training (ST) with logit for 300 epochs. We check **Robust Accuracy**, defined as the ratio of successful predictions of the argmax labels of the *adversarial examples*, which are generated by increasing the ℓ_2 distance between the student and the teacher's output logits. We also show Sorted BNC and $\epsilon_{\text{in}}, \epsilon_{\text{out}}$ curves for each setting to check the node specialization.

We conduct various types of attacks to generate adversarial examples: $\{\ell_1, \ell_2, \ell_\infty\}$ optimization based PGD attack (Madry et al., 2017), FGSM attack (Goodfellow et al., 2014), CW attack (Carlini and Wagner, 2017) and Blackbox-transfer attack using a surrogate model trained independently. We run robustness evaluation multiple times to compute statistical confident robust accuracy with mean μ and variance σ^2. From Table 1, we can see AT model's robustness increases with epochs and surpasses the 300 epochs ST (logit) model's even at 50 epochs. Figure 3 and 4 show the neuron special-

Table 1: Robust evaluation of student models trained for different epochs (numbers in the parentheses) under Adversarial Training (AT) and Standard Training (ST). Results are reported by the mean μ and variance σ^2 of model robust accuracy (%) against various attacks.

Attacks	AT (50)		AT (100)		AT (150)		AT (200)		AT (300)		ST (300)	
	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
ℓ_∞ PGD	78.79	1.1e-4	84.02	4.2e-5	87.57	2.8e-5	88.01	8.2e-6	88.20	2.9e-5	74.39	4.2e-5
ℓ_2 PGD	91.85	7.3e-6	95.98	7.0e-6	96.21	4.7e-6	95.95	4.8e-6	96.31	5.4e-6	94.01	1.0e-5
ℓ_1 PGD	92.30	9.9e-6	96.18	6.5e-6	96.56	3.1e-6	96.36	3.0e-6	96.59	3.7e-6	94.51	4.8e-6
FGSM	90.65	7.7e-6	95.18	4.7e-6	95.55	6.8e-6	94.87	5.3e-6	95.28	4.9e-6	91.12	2.1e-5
CW	78.89	1.0e-4	91.50	4.9e-5	91.96	4.3e-5	94.52	2.2e-5	92.63	1.3e-5	86.19	2.7e-5
Blackbox-transfer	43.14	2.6e-5	45.31	4.7e-5	46.06	4.2e-5	46.68	3.7e-5	46.99	2.1e-5	43.48	3.2e-5

Table 2: Robust Accuracy (%) of {In-plane, Out-plane, Standard} AT models trained for 150 epochs against {In-plane, Out-plane, Standard} adversarial attacks.

Attacks	In-plane	Out-plane	Standard
AT (In-plane)	88.86	89.18	89.28
AT (Out-plane)	83.11	83.54	83.60
AT (Standard)	86.87	87.28	87.18

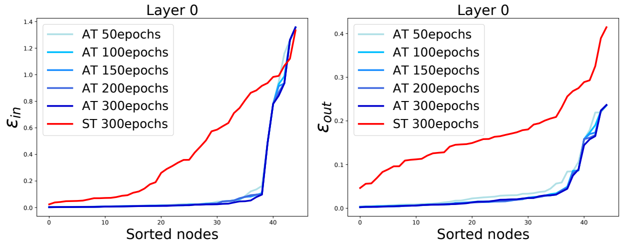

Figure 4: (ϵ_{in}, ϵ_{out}) curve (lower curve means better specialization) for Adversarial Training (AT) and Standard Training (ST) with logit for 300 epochs. AT leads to much stronger student specialization and higher robust accuracy.

ization of the student by plotting $\epsilon_{\text{in}}, \epsilon_{\text{out}}$ and Sorted BNC curve, where AT models achieve *much better specialization* than ST models by reducing ϵ_{in} and ϵ_{out} drastically in the first few epochs. For 5-layer deeper Conv network, we also observe similar results as shown in Figure 5.

We also evaluate how in-plane AT and out-plane AT affect student's specialization separately. To disentangle them, for each instance $\mathbf{x}$, we apply the standard ℓ_∞ PGD attack twice with different initialization to obtain $\mathbf{x}'_{\text{in}}$ and $\mathbf{x}'_{\text{out}}$, while $\mathbf{x}'_{\text{in}}$ has a smaller distance to the input subspace. We use $\mathbf{x}'_{\text{in}}$ to train the in-plane AT model and $\mathbf{x}'_{\text{out}}$ for out-plane AT model, and we evaluate each model's Robust Accuracy against the in-plane attack, out-plane attack, or both (i.e., the standard attack). From Table 2, we find the in-plane attack can be more severe causing model's vulnerability, so in-plane AT models achieve better robustness. In Figure 7, the plots for ϵ_{in} and ϵ_{out} indicate that the in-plane AT model leads to better specialization from both in-plane and out-plane directions.

We evaluate the **Mean of the Best Normalized Correlation (MBNC)** and the *unspecialized/specialized* ratio[4] of AT and ST models trained for 300 epochs in Figure 6. We observe that AT model could achieve higher MBNC value by forcing more student nodes to be specialized to teacher nodes, and the traditional Data Augmentation method (RandomCrop, HorizontalFlip, Rotation) could improve neuron specialization as well.

5.5 Standard training

In this subsection, we continue to study the correlation between model robustness and specialization to the teacher in Standard Training (ST) with logit or label's supervision. Our analyis is performed at different training epochs $T \in \{50, 100, 150, 200, 300\}$.

Table 3: Robust Accuracy (%) of student models trained for different epochs (numbers in the parentheses) under Standard Training (ST) with logit or label supervision.

Robust Acc	ST (50)	ST (100)	ST (150)	ST (200)	ST (300)
Logit training	23.12	30.72	36.72	48.52	62.77
Label training	19.08	20.81	22.34	23.42	25.79

From Table 3 and Figure 8, we can observe both robustness and specialization of ST models improved with training. However, when training with the same epochs, ST (label) model is worse than ST (logit) model from both robustness and specialization perspectives.

Moreover, in Figure 9, we show the specialization of ST (logit) model and ST (label) model from the in-plane and out-plane directions. Interestingly, ST with label does not improve the in-plane specialization. In contrast, ST with logit leads to specialization on both in-plane and out-plane aspects.

To check the low-rank property of input distribution $\mathcal{X}$, we add the d-dimensional Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_d), \sigma = 0.1$ on the input instances during the ST with logit, and we present $\epsilon_{\text{in}}, \epsilon_{\text{out}}$ curves in Figure 10. From Figure 10, we can observe that by training with the high-rank input instances, the

[4] We consider the node to be *unspecialized* if NC is smaller than 0.8, and *specialized* if NC is larger than 0.9.

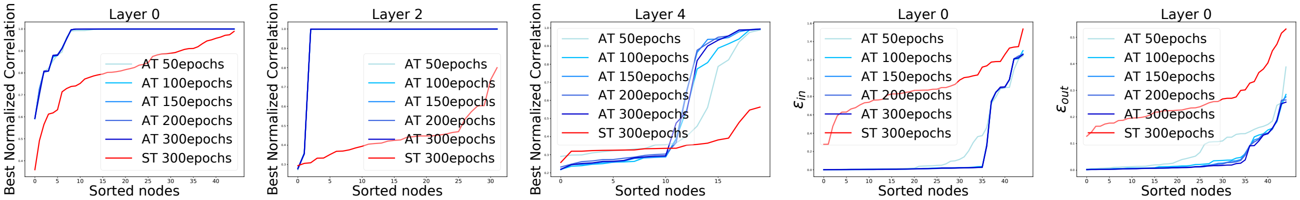

Figure 5: {**Left**: Sorted BNC curve on Layer 0, 2, 4; **Right**: ϵ_{in} and ϵ_{out} curve} using a deeper Conv network architecture.

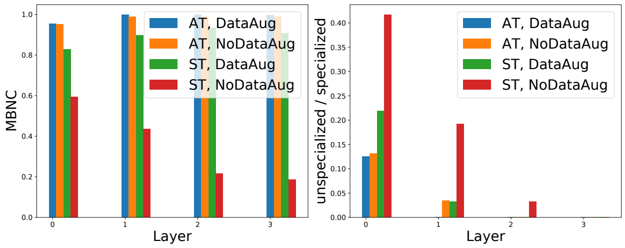

Figure 6: Comparison between Adversarial Training (AT) and Standard Training (ST) with/without data augmentation. **Left**: The MBNC value $\hat{\rho}$ for every layer. **Right**: The ratio of the number of the unspecialized nodes divided by the number of the specialized nodes in every layer.

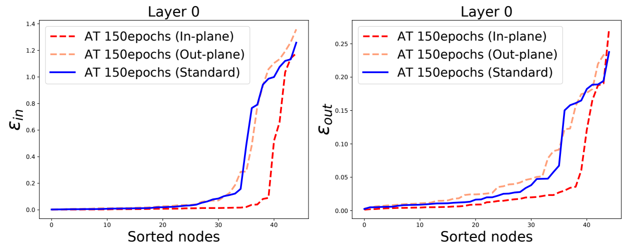

Figure 7: (ϵ_{in}, ϵ_{out}) curve for {in-plane, out-plane, standard} Adversarial Training (AT) with 150 epochs.

student can be more specialized to the teacher from both in-plane and out-plane directions. Meanwhile, the low-rank property brings the risk facing the out-plane adversarial examples.

Remarks. We suggest the existence of adversarial examples is due to student's *unspecialized* neurons (large ϵ_{in}). During training, ST decreases student nodes' ϵ_{in}, improves neuron specialization and therefore leads to better robustness. Also, comparing to ST with label, ST with logit can leverage the additional direction information from the teacher output, and achieve better neuron specialization and robustness, which verifies our claim about the strong correlation between robustness and specialization.

5.6 Analysis of Confidence-Calibrated Adversarial Training

We also extend our analysis to other training techniques that improve the robustness of the model. **Confidence-Calibrated Adversarial Training (CCAT)** (Stutz et al., 2019) proposes to gener-

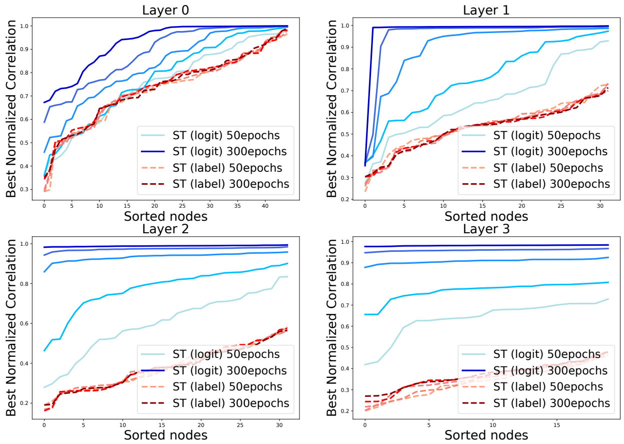

Figure 8: Sorted BNC curves (the higher means better specialization) for Standard Training (ST) with logit or label for different epochs on CIFAR-10. Logit training leads to much stronger specialization across all layers. Solid **Blue** curves refer to the logit training and dashed **Red** curves to the label training. Color changed from light to dark with more training epochs.

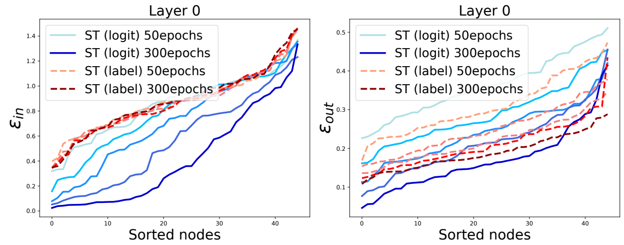

Figure 9: (ϵ_{in}, ϵ_{out}) curves for Standard Training (ST) with logit or label for different epochs on CIFAR-10. Solid **Blue** curves refer to the logit training and dashed **Red** curves to the label training (which reduces ϵ_{out} more). Colors are changed from light to dark with more training epochs.

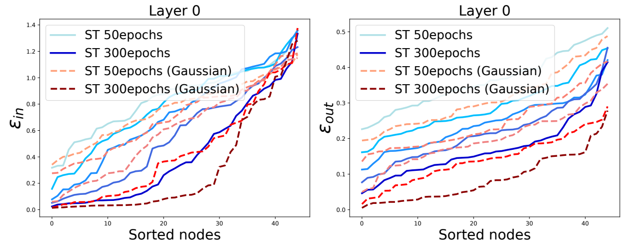

Figure 10: (ϵ_{in}, ϵ_{out}) curves for ST (logit) with or without Gaussian augmentation for different epochs on CIFAR-10. Solid **Blue** curves refer to ST (logit) without Gaussian and dashed **Red** curves refer to ST (logit) with Gaussian. Adding Gaussian leads to better specialization. Colors are changed from light to dark with more training epochs.

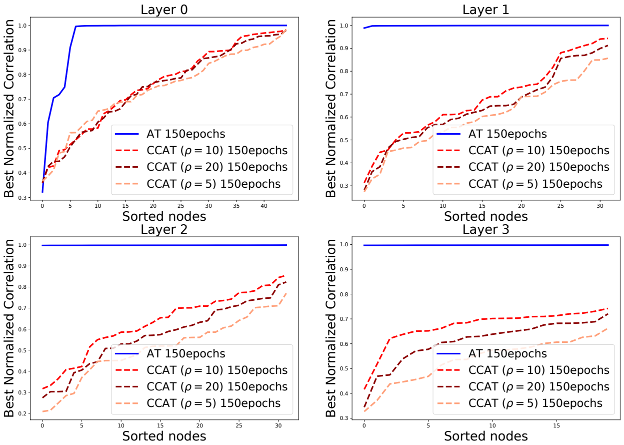

Figure 11: Sorted BNC curve of student trained by CCAT. Specialization correlates with robust accuracy (Table 4).

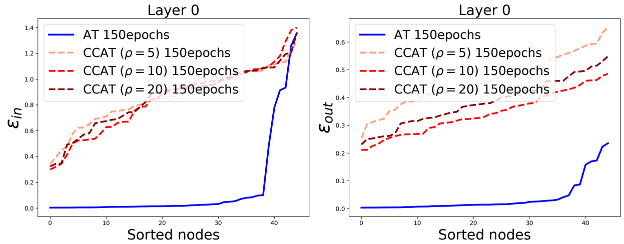

Figure 12: $(\epsilon_{\text{in}}, \epsilon_{\text{out}})$ curve of student models trained by CCAT with different ρ. Here "AT" means that we use vanilla Adversarial Training with oracle-adversarial samples (Eqn. 3), which leads to much better specialization.

ate high confidence adversarial examples with calibrated soft labels. Specifically, for an input $(\mathbf{x}, \mathbf{y})$, adversarial example $\mathbf{x} + \delta$ is generated as: $\delta = \arg\max_{||\delta||_\infty \le \epsilon} \max_{\mathbf{k} \ne \mathbf{y}} f^C_{\mathbf{k}}(\mathbf{x} + \delta)$, where $f^C_{\mathbf{k}}(\mathbf{x})$ denotes model f's output confidence on label $\mathbf{k}$, and ϵ denotes the tolerance of ℓ_∞ perturbation scale. The confidence parameter $\lambda(\delta)$ is decided by the ℓ_∞ norm of δ and the hyper-parameter ρ: $\lambda(\delta) = (1 - \min(1, ||\delta||_\infty / \epsilon))^\rho$, and the confidence-calibrated soft label $\tilde{\mathbf{y}}$ is obtained by mixing the one-hot vector of the label y with the confidence: $\tilde{\mathbf{y}} = \lambda(\delta)\text{one\_hot}(\mathbf{y}) + (1 - \lambda(\delta))\frac{1}{K}$, where K refers to the number of labels.

Table 4: Robust Accuracy (%) of student models trained for 150 epochs, with CCAT given different ρ and AT.

Model	CCAT ($\rho = 5$)	CCAT ($\rho = 10$)	CCAT ($\rho = 20$)	AT
Robust Acc	47.25	52.04	49.33	84.07

In the teacher-student setting, we consider the **confidence** to be the ℓ_∞ distance between student and teacher's logit. We generate the high-confidence adversarial example by: $\delta = \arg\max_{||\delta||_\infty \le \epsilon} \max_{\mathbf{k}} |s_{\mathbf{k}}(\mathbf{x} + \delta) - t_{\mathbf{k}}(\mathbf{x} + \delta))|$, where $s_{\mathbf{k}}(\mathbf{x}), t_{\mathbf{k}}(\mathbf{x})$ refer to the output logit of student and teacher on label $\mathbf{k}$ respectively. We apply the confidence-calibrated soft label $\tilde{\mathbf{y}}$ to the adversarial examples and evaluate the robustness and specialization of CCAT models trained for 150 epochs, with $\rho = 5, 10, 20$. Figure 11 and Table 4 show the neuron specialization and robustness of CCAT models respectively. We find that CCAT with $\rho = 10$ achieves the best robustness and neuron specialization among all CCAT models. Again, we notice that there is a strong correlation between specialization and robustness (e.g., $\rho = 10$ achieves the highest degree of specialization *and* robustness, $\rho = 20$ achieves the second highest on both, and similarly for $\rho = 5$). Figure 12 shows the $\epsilon_{\text{in}}, \epsilon_{\text{out}}$ curves, and indicates AT can improve specialization for both in-plane and out-plane directions.

Remarks. Comparing different CCAT models with AT models, the results consistently show that the neuron specialization of student models is highly correlated with the robustness, which is aligned with our observation in Sec 5.5. In addition, AT models with better robustness may be due to the information loss during the confidence calibration: while the confidence calibration captures the balanced adversarial distribution, it will provide inconsistent confidence to the teacher's output. To align the confidence distribution with the teacher's output would be an interesting future work.

6 Conclusion and Future Work

In this paper, we leverage the teacher-student framework to study the model robustness and explain the origin of adversarial samples in a trained network. In our setting, we assume the labels to be the output of an *oracle* teacher and student learns from the teacher through the teacher's output. In this setting, model vulnerability (and adversarial samples) naturally arise when the nodes (neurons) in a learned student do not fully reconstruct (or "specialized into") teacher's nodes when the input data are low-dimensional. Specifically, we theoretically show that, when training converges, student nodes are specialized in the low-dimensional input subspace, but may not be specialized out of such a subspace, leaving space for adversarial examples. Extensive experiments show a clear correlation between model robustness and degree of student specialization measured by normalized correlation between activations of teacher and student, in standard training, adversarial training (AT) and Confidence-Calibrated Adversarial Training (CCAT). Based on this new perspective, future work includes regularization of unspecialized student nodes during training, label-extrapolation of adversarial samples in AT, etc.

7 Acknowledgement

This work is partially supported by NSF grant No.1910100 and DARPA QED-RML-FP-003 awarded to Bo Li.

References

Allen-Zhu, Z., Li, Y., and Liang, Y. (2019). Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pages 6158–6169.

Athalye, A., Carlini, N., and Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*.

Aubin, B., Maillard, A., Krzakala, F., Macris, N., Zdeborová, L., et al. (2018). The committee machine: Computational to statistical gaps in learning a two-layers neural network. In *Advances in Neural Information Processing Systems*, pages 3223–3234.

Bhagoji, A. N., He, W., Li, B., and Song, D. (2018). Exploring the space of black-box attacks on deep neural networks. *ECCV*.

Bhattad, A., Chong, M. J., Liang, K., Li, B., and Forsyth, D. A. (2019). Unrestricted adversarial examples via semantic manipulation. *arXiv preprint arXiv:1904.06347*.

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE.

Chen, J., Wang, D., and Chen, H. (2020). Explore the transformation space for adversarial images. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 109–120.

Engel, A. and Van den Broeck, C. (2001). *Statistical mechanics of learning*. Cambridge University Press.

Freeman, J. A. and Saad, D. (1997). Online learning in radial basis function networks. *Neural Computation*, 9(7):1601–1622.

Gardner, E. and Derrida, B. (1989). Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, 22(12):1983.

Goldt, S., Advani, M. S., Saxe, A. M., Krzakala, F., and Zdeborová, L. (2019). Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. *NeurIPS*.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gu, J. and Tresp, V. (2019). Saliency methods for explaining adversarial attacks. *arXiv preprint arXiv:1908.08413*.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019a). Adversarial examples are not bugs, they are features. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 125–136. Curran Associates, Inc.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019b). Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136.

Kang, D., Sun, Y., Hendrycks, D., Brown, T., and Steinhardt, J. (2019). Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*.

Khoury, M. and Hadfield-Menell, D. (2018). On the geometry of adversarial examples. *arXiv preprint arXiv:1811.00525*.

Kotyan, S., Vasconcellos Vargas, D., and Matsuki, M. (2019). Representation quality of neural networks links to adversarial attacks and defences. *arXiv e-prints*, pages arXiv–1906.

Krizhevsky, A., Hinton, G., et al. (2009). *Learning multiple layers of features from tiny images*. Citeseer.

Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M. E., and Bailey, J. (2018). Characterizing adversarial subspaces using local intrinsic dimensionality. *ICLR*.

Mace, C. and Coolen, A. (1998). Statistical mechanical analysis of the dynamics of learning in perceptrons. *Statistics and Computing*, 8(1):55–88.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Papernot, N., McDaniel, P., and Goodfellow, I. (2016). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Saad, D. and Solla, S. A. (1996). Dynamics of on-line gradient descent learning for multilayer neural networks. In *Advances in neural information processing systems*, pages 302–308.

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. (2019). Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3358–3369.

Shamir, A., Safran, I., Ronen, E., and Dunkelman, O. (2019). A simple explanation for the existence of adversarial examples with small hamming distance. *arXiv preprint arXiv:1901.10861*.

Shi, X. and Ding, A. A. (2019). Understanding and quantifying adversarial examples existence in linear classification. *arXiv preprint arXiv:1910.12163*.

Singh, M., Kumari, N., Sinha, A., and Krishnamurthy, B. (2018). Understanding adversarial space through the lens of attribution. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 30–40. Springer.

Stutz, D., Hein, M., and Schiele, B. (2019). Confidence-calibrated adversarial training: Generalizing to unseen attacks. *CoRR, abs/1910.06259*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tabacof, P. and Valle, E. (2016). Exploring the space of adversarial images. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 426–433. IEEE.

Tian, Y. (2019). Student specialization in deep rectified networks with finite width and input dimension. *arXiv preprint arXiv:1909.13458*.

Tian, Y., Jiang, T., Gong, Q., and Morcos, A. (2019). Luck matters: Understanding training dynamics of deep relu networks. *arXiv preprint arXiv:1905.13405*.

Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. (2017). The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2018). Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*.

Vargas, D. V. and Su, J. (2019). Understanding the one-pixel attack: Propagation maps and locality analysis. *arXiv preprint arXiv:1902.02947*.

Wong, E., Schmidt, F. R., and Kolter, J. Z. (2019). Wasserstein adversarial examples via projected sinkhorn iterations. *arXiv preprint arXiv:1902.07906*.

Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., and Song, D. (2018a). Generating adversarial examples with adversarial networks. *IJCAI*.

Xiao, C., Zhu, J.-Y., Li, B., He, W., Liu, M., and Song, D. (2018b). Spatially transformed adversarial examples. *ICLR*.

Xie, C., Tan, M., Gong, B., Yuille, A., and Le, Q. V. (2020). Smooth adversarial training. *arXiv preprint arXiv:2006.14536*.

Zhang, H., Chen, H., Song, Z., Boning, D., Dhillon, I. S., and Hsieh, C.-J. (2019). The limitations of adversarial training and the blind-spot attack. *arXiv preprint arXiv:1901.04684*.