Towards the Memorization Effect of Neural Networks in Adversarial Training

Han Xu, Xiaorui Liu, Wentao Wang, Anil K. Jain. Jiliang Tang
Department of Computer Science and Engineering
Michigan State University

Wenbiao Ding, Zhongqin Wu, Zitao Liu
TAL Education Group

Abstract

Recent studies suggest that “memorization” is one important factor for overparametrized deep neural networks (DNNs) to achieve optimal performance. Specifically, the perfectly fitted DNNs can memorize the labels of many atypical samples, generalize their memorization to correctly classify test atypical samples and enjoy better test performance. While, DNNs which are optimized via adversarial training algorithms can also achieve perfect training performance by memorizing the labels of atypical samples, as well as the adversarially perturbed atypical samples. However, adversarially trained models always suffer from poor generalization, with both relatively low clean accuracy and robustness on the test set. In this work, we study the effect of memorization in adversarial trained DNNs and disclose two important findings: (a) Memorizing atypical samples is only effective to improve DNN’s accuracy on clean atypical samples, but hardly improve their adversarial robustness and (b) Memorizing certain atypical samples will even hurt the DNN’s performance on typical samples. Based on these two findings, we propose Benign Adversarial Training (BAT) which can facilitate adversarial training to avoid fitting “harmful” atypical samples and fit as more “benign” atypical samples as possible. In our experiments, we validate the effectiveness of BAT, and show it can achieve better clean accuracy vs. robustness trade-off than baseline methods, in benchmark datasets such as CIFAR100 and Tiny ImageNet.

1 Introduction

It is evident from recent studies that the memorization effect (or benign overfitting) [1-5] can be one necessary factor for overparametrized deep neural networks (DNNs) to achieve the close-to-optimal generalization error. From the empirical perspective, the works [1,2] suggest that modern benchmark datasets, such as CIFAR10, CIFAR100 [6] and ImageNet [7], always have very diverse data distributions, especially containing a large fraction of “atypical” samples. These atypical samples are both visually and statistically very different from the other samples in their labeled class. For example, the images in the class “bird” may have a variety of sub-populations or species, with many samples in the main sub-population and other (atypical) samples in less-frequent and distinct sub-populations. Since these atypical samples are deviated from the main sub-population, DNNs can only fit these atypical samples by “memorizing” their labels. While, memorizing or fitting these atypical samples will not hurt the DNNs’ performance, but can help DNNs to correctly classify the atypical samples appearing in the test set and hence improve the test accuracy.

Similar to the classification models which are trained via empirical risk minimization (ERM) algorithms, adversarial training methods [8,9] are also devised to fit the whole training dataset. Specifi-
cally, adversarial training minimizes the model’s error against adversarial perturbations \[^{10,11}\] by fitting the model on manually generated adversarial examples of all training data. Although adversarial training can fit and memorize all training data as well as their adversarially perturbed counterparts, they always suffer from both poor clean accuracy and adversarial accuracy (or robustness) \[^{12,13}\] on the test set. Recent study \[^{14}\] indicates that, during the adversarial training process, one model’s test adversarial accuracy will even keep decreasing as it hits higher training adversarial accuracy (during the fine-tuning epochs). Thus, it is natural to ask a question: What is the effect of memorization in adversarial training? In particular, can memorizing (the labels of) the atypical samples and their adversarial counterparts benefit the model’s accuracy and robustness?

To answer this question, we first conduct preliminary studies to check whether memorizing atypical samples in adversarial training can benefit the DNNs’ test performance, especially on those test atypical samples. In Section \[^{3.1}\], we implement PGD adversarial training \[^{8}\] on CIFAR100 under ResNet18 and WideResNet models \[^{15}\] and fine-tune them until achieving the optimal training performance. From the results in Section \[^{3.1}\], we observe that the memorization in adversarial training can only benefit the clean accuracy of test atypical samples. With the DNNs gradually fit/memorize more atypical samples, they can finally achieve fair clean accuracy close to \(\sim 40\)% on test atypical set. However, the adversarial accuracy on test atypical set is constantly low (\(\sim 10\)% during the whole training process, even though the models can fit almost all atypical (adversarial) training samples. Based on the theoretical study \[^{13}\], the adversarial robustness is hard to generalize especially when the training data size is limited. Since each single atypical sample is distinct from the main sub-population and has rare frequency to appear in the training set, the data complexity for each specific atypical sample is also very low. Thus, its adversarial robustness can be extremely hard to generalize. Remarkably, the entire atypical set covers a non-ignorable fraction in complex datasets such as CIFAR100. Completely failing on atypical samples could be one important reason that contributes to the overall poor robustness generalization of DNNs.

Furthermore, we find that in adversarial training, fitting atypical samples will even hurt DNNs’ performance of those “typical” samples (the samples in the main sub-population). In the Section \[^{3.2}\], we again implement PGD adversarial training \[^{8}\] on CIFAR100 for several trails, which are trained with different amount of atypical samples existed. Based on the results from Section \[^{3.2}\], an adversarially trained model on the training set without any atypical samples has 92% clean accuracy and 52% adversarial accuracy on the test typical set. While, the model trained with 100% atypical samples included only has 85%/44% clean/adversarial accuracy respectively. In other words, atypical samples act more like “poisoning data” \[^{16}\] to deteriorate model performance on typical samples. In this paper, we demonstrate that the atypical samples which “poison” the models are the ones which deviated from the distribution of their own classes, but close to the distribution of other classes. Some examples are shown in Fig. \[^{3}\]. For instance, an atypical “plate” in CIFAR100 has strong visual similarity to “apples”. If DNNs memorize the features of this “plate” and predict any other images with similar features to be “plate”, the DNNs cannot distinguish “apples” and “plates”.

Motivated by our findings, we propose a novel algorithm called Benign Adversarial Training (BAT), which can prevent the negative influence brought by memorizing “poisoning” atypical samples, meanwhile preserve the model’s ability to memorize those “benign / useful” atypical samples. It is worth mentioning that, by fitting those “benign” atypical samples, the BAT method can achieve good clean accuracy on the atypical set. Compared with PGD adversarial training \[^{8}\] when it achieves the highest adversarial robustness, BAT has higher clean accuracy and comparable adversarial accuracy. Compared to other popular variants of adversarial training such as \[^{17,19}\], BAT is the only one obtaining both better (or comparable) clean and adversarial accuracy than \[^{8}\], on complex datasets such as CIFAR100 and Tiny Imagenet \[^{20}\].

\section{Definition and Notation}

\subsection{Atypical Samples and Memorization}

In this section, we start by introducing necessary concepts and definitions about the memorization effects. As well known, overparameterized DNNs have tremendous capacity to make them easy to perfectly fit the training dataset \[^{21}\]. In practice, on common benchmark classification tasks, such as

---

\[^{1}\] Clean Accuracy: Model’s accuracy on unperturbed samples. Adversarial accuracy: Model’s accuracy on adversarially perturbed samples. Without loss of generality, this paper discusses the adversarial accuracy under \(l_\infty\)-\[8/255\] PGD attack \[^{8}\].
CIFAR10 [15], CIFAR100 and ImageNet [7], we also tune the DNNs to achieve very high training accuracy even close to 100%, and enjoy good test performance. While, this property of DNNs in practice cannot be well explained by standard theories about model generalization [22, 23] from the regularization perspective. At a high level, the standard theories underline the importance of regularization on the model complexity, to make DNNs to avoid “overfitting” or “memorizing” the outliers and nonuseful samples in the training data.

Fortunately, recent works [1, 3, 5, 4] make significant progress to close this gap from both theoretical and empirical perspectives. They suggest that the memorization effect is one necessary property for DNNs to achieve optimal generalization performance. In detail, the empirical studies [1, 2] point out that the common benchmark datasets, such as CIFAR10, CIFAR100, ImageNet, contain a large portion of atypical samples (or namely, rare samples, sub-populations, etc.). These atypical samples look very different from the other samples in the main distribution of its labeled class (see Appendix A), and are statistically indistinguishable from outliers or mislabeled samples. Because these atypical samples are deviated from the main distribution, DNNs can only fit these samples by memorizing their labels. Moreover, without memorizing these atypical samples during training, the DNNs can totally fail to predict the atypical samples appearing in the test set.

Identify Atypical Samples. To identify such atypical samples in common datasets in practice, the work [2] proposes to examine which training samples can only be fitted by memorization, and measure each training sample’s “memorization value”. Formally, for a training algorithm $A$ (i.e., ERM), the memorization value “mem($A, D, x_i$)” of a training sample $(x_i, y_i) \in D$ in training set $D$ is defined as:

$$\text{mem}(A, D, x_i) = \frac{\Pr_{F \sim A(D)}(F(x_i) = y_i)}{\Pr_{F \sim A(D \setminus x_i)}(F(x_i) = y_i)},$$

which calculates the difference between the model $F$’s accuracy on $x_i$ with and without $x_i$ removed from the training set $D$ of algorithm $A$. Note that for each sample $x_i$, if its memorization value is high, it means that removing $x_i$ from training data will cause the model with a high possibility to wrongly classify itself, so $x_i$ is very likely to be fitted only by memorization and be atypical. From the work [2], these atypical samples have non-ignorable portion in common datasets. For example, there are around 40% training samples in CIFAR100 having a large memorization value $> 0.15$.

The similar strategy can also facilitate to find atypical samples in the test set, which are the samples that are strongly influenced by atypical training samples. In detail, by removing an atypical training sample $(x_i, y_i)$, we calculate its “influence value” on each test sample $(x'_j, y'_j) \in D'$ in test set $D'$:

$$\text{infl}(A, D, x_i, x'_j) = \frac{\Pr_{F \sim A(D')}(F(x'_j) = y'_j)}{\Pr_{F \sim A(D \setminus x_i)}(F(x'_j) = y'_j)},$$

If the sample pair $(x_i, x'_j)$ has a high influence value, removing the atypical sample $x_i$ will drastically decrease the model’s accuracy on $x'_j$. It suggests that the model’s prediction on $x'_j$ is mainly based on the memorization of $x_i$, thus, $x'_j$ is the corresponding atypical sample of $x_i$. The sample pair $x_i$ and $x'_j$ is called a high-influence pair, if they have a high influence value and belong to the same class. In practice, the high-influence pairs of images are typically visually similar and have similar semantic features. The memorization benefits the model’s performance especially on these test atypical samples, and hence boosts the overall test accuracy.

2.2 Adversarial Training

Similar to classification models trained via empirical risk minimization (ERM) algorithms, adversarial training methods [8, 9] are also devised to fit the whole dataset by training the model on manually generated adversarial examples. Formally, they are optimized to have the minimum adversarial loss:

$$\min_F \mathbb{E}_x \left[ \max_{\|\delta\| \leq \epsilon} \mathcal{L}(F(x + \delta), y) \right],$$

which is the model $F$’s average loss on the data $(x, y)$ that perturbed by adversarial noise $\delta$. These adversarial training methods have been shown to be one of the most effective approaches to improve the model robustness against adversarial attacks. Note that similar to traditional ERM, adversarially trained models can also achieve very high training performance. For example, under ResNet18 [15], PGD adversarial training [15] can achieve over 99% clean accuracy and 85% adversarial accuracy on the training set of CIFAR 100. Under a larger network with WideResNet28-10
(WRN28), it can achieve 100% clean accuracy and 99% adversarial accuracy. It suggests that these DNN models have sufficient capacity to memorize the labels of these atypical samples and their adversarial counterparts. However, different from ERM, adversarially trained models usually suffer from bad generalization performance on the test set. For example, the ResNet18 model can only have 57% and 22% test clean accuracy and adversarial accuracy (∼59% and 24% on WRN28). Moreover, the study [14] suggests that during the adversarial training process, the model’s test adversarial accuracy keeps dropping as more training data is fitted (after the first time learning rate decay). Thus, these facts indicate that the memorization in adversarial training is probably not always beneficial to the test performance and requires deep understanding. In the following sections, we will empirically draw a significant connection between these properties with the memorization effect.

3 Atypical Samples in Adversarial Training

In this section, we attempt to understand adversarial training’s behavior by studying its relation with the memorization effect. The discussions are mainly based on PGD-adversarial training [8] on CIFAR 100. Implementation details are shown in Appendix B where we also report the results in more datasets, i.e., CIFAR 10 and Tiny ImageNet [20], and we make consistent observations.

3.1 Adversarial Robustness of Atypical Samples is Harder to Generalize

In this subsection, we first check whether fitting atypical samples in adversarial training can effectively help the model correctly and robustly predict the atypical samples in the test set. We apply PGD adversarial training [8] on original CIFAR 100 dataset for 200 epochs and evaluate the model’s clean accuracy and adversarial accuracy on training atypical set $D_{atyp} = \{x_i \in D : \text{mem}(x_i) > 0.15\}$ and its corresponding test atypical set $D'_{atyp} = \{x'_j \in D' : \text{infl}(x_i, x'_j) > 0.15, \forall x_i \in D_{atyp}\}$. In Fig. 1 we report the algorithm’s performance (clean & adv. acc.) on these atypical sets along with the training process. From the results, we observe that both ResNet18 and WRN28 are capable to memorize all clean atypical samples and most adversarial atypical samples, since they both achieve $\approx 100\%$ clean accuracy and high adversarial accuracy ($\approx 80\%$ and $100\%$, respectively) on the training atypical set. As the training goes, the models’ clean accuracy on the test atypical set gradually improves and finally approaches $40\%$. However, their adversarial robustness keep constant around $10\%$ from the beginning epochs to the last ones, no matter how high the training performance is. These results suggest that the memorizing atypical samples in adversarial training may only improve their test clean accuracy, but hardly help their adversarial robustness. Recall that in CIFAR100, atypical set $D_{atyp}$ (with memorization value > 0.15) covers 40% samples of the whole dataset. Completely failing on the adversarial robustness of atypical samples could be one important reason that contributes to the poor robustness generalization of DNNs [14].

As the previous theoretical study [13] states, for a model to have good robustness generalization performance, it always demands a training set with much larger amount of samples, than a model to have good clean accuracy generalization. In our case, the sub-population of each particular atypical sample has very low frequency to appear in the training set, and it is always deviated from the main sub-population. Thus, in the sub-population of this atypical sample, it is equivalent to a classification task based on an extremely small dataset, with one or a few training samples given. Therefore, the adversarial robustness of atypical samples can be extremely hard to generalize.

3.2 Memorizing Atypical Samples Hurts Typical Samples’ Performance

In this subsection, we further observe that fitting atypical samples will even bring negative effects on “typical” samples. Here, we define “typical” samples as the subset of training set $D$ which have low memorization value: $D_{typ} = \{x_i \in D : \text{mem}(x_i) < 0.02\}$. It means that they are not fitted by
memorization and are from the main sub-population in their class. To define the test typical set \( D_{\text{typ}}' \), we exclude all test samples which have high influence values from any atypical training samples, and also exclude the samples that using ERM algorithm \( A \) has low success rate to predict (the samples which cannot be learned from \( D \)): \( D_{\text{typ}}' = D' - \{ x'_j : \text{infl}(x_i, x'_j) > 0.02, \forall x_i \in D_{\text{atyp}} \} \cup \{ x'_j : \text{Pr}_{F \leftarrow \mathcal{A}(D)}(F(x'_j) = y_j) < 0.8 \} \).

To demonstrate the negative effect from fitting atypical samples, we conduct PGD adversarial training for several trails on resampled CIFAR100 datasets: each dataset is constructed with the whole training typical set \( D_{\text{typ}} \), and a part of the training atypical set \( D_{\text{atyp}} \) (randomly sample 0%, 20% and 100% in \( D_{\text{atyp}} \)). In Fig. 2, we report the adversarially trained model’s clean and adversarial accuracy on the test typical set \( D_{\text{typ}}' \) and check the impact of atypical samples on the typical samples.

From the results, we find that the amount of atypical samples makes a significant influence on the typical samples. For example, under ResNet18, an adversarially trained model without atypical samples has 92% clean accuracy and 52% adversarial accuracy on the test typical samples on the last epochs. While, the model trained with all atypical samples included only has 85% and 44% clean & adv. accuracy, respectively. These results suggest: the more atypical samples exist in training set, the poorer performance the model will have on \( D_{\text{typ}}' \). In other words, these atypical samples act more like “poisoning” samples [16, 24] which can deteriorate the model’s performance on typical samples and consequently hurt the overall performance.

Poisoning Atypical Samples A natural question is what kind of atypical samples are likely to “poison” model robustness and why? Different from previous literature about poisoning samples in traditional ERM, which assume that poisoning samples are most mis-labeled samples [25], CIFAR100 is a clean dataset with no or very few wrong labels. However, we hypothesize that the atypical samples which poison the model performance might pertain some features of a “wrong” class. Recall that atypical samples are always distinct from the main data distribution in their labeled class, it is likely that they are closer to the distribution of a “wrong” class. As shown in Fig. 3 an atypical “plate” is visually very similar to images in “apple”. If the model memorizes this atypical “plate” and predicts any samples with similar features to be “plate”, the model cannot distinguish between “apple” and “plate”.

A simple verification to this hypothesis, in Table 1, we empirically show that atypical samples have lower Cosine Distance (CD) of the models’ pen-ultimate layer representation output, for each pair of samples (in the training typical set \( D_{\text{typ}} \)) from different classes. Under the same experimental setting and the models above, we measure the average Cosine Distance of the models’ pen-ultimate layer representation output, for each pair of samples (in the training typical set \( D_{\text{typ}} \)) from different classes. Table 1 shows that in adversarial training, fitting more atypical samples will result in a smaller distance for the representations of samples in different classes. It suggests that with atypical samples, DNNs learn more similar and mixed representations for different classes, which can degrade the typical samples’ test performance.

| # of Atypical Samples | 0%   | 20%  | 100%  |
|-----------------------|------|------|-------|
| ResNet18              | 0.66 | 0.62 | 0.59  |
| WRN28                 | 0.64 | 0.61 | 0.57  |

It is also worth to mention that this poisoning phenomenon does not appear in the traditional ERM algorithm (results in Appendix [3]). In ERM, memorizing atypical samples will neither hurt typical sample’s performance or degrade the feature space discrimination. A possible explanation is that adversarially trained models use more “semantically meaningful” features for prediction [12, 26].

\[ \text{Cosine Distance: } \mathbb{E}_{x_1,x_2} \left[ \frac{h(x_1) \cdot h(x_2)}{||h(x_1)||_2 \cdot ||h(x_2)||_2} \right], \text{ where } h(\cdot) \text{ is the pen-ultimate layer output of DNN model } F(\cdot), \text{ and } x_1, x_2 \in D_{\text{typ}} \text{ and from different classes.} \]
During adversarial training, the models will not only memorize the labels of atypical samples, but also their semantic features. As a result, the adversarial trained DNNs will construct more mixed concepts of features, if the “poisoning” atypical samples exist.

4 Benign Adversarial Training for Memorizing Benign Atypical Samples

Based on previous discussions, the effect of atypical samples in adversarial training can be briefly summarized as: **(a)** They can benefit the models’ clean accuracy (especially on atypical samples), but hardly improve their robustness. **(b)** They hurt the performance of typical samples. In this section, we ask the question: *Can we eliminate the negative influence from fitting atypical samples during adversarial training?* Admittedly, strategies for adversarial training to stop at early epoch [14] is potential to achieve this goal. As shown in Fig. [2a], the typical samples’ performance decreases only after the epoch 100, which is when the model begins to fit more atypical samples. Stopping at such epoch 100 can effectively prevent the future performance degradation. However, it is still unclear how to properly identify such an epoch in general scenarios. Moreover, the early-stop mechanism tends to ignore all atypical samples and significantly limits the model’s ability to handle test atypical samples, especially on complex datasets such as CIFAR100 and ImageNet with large fractions of atypical samples.

In this section, we propose a novel algorithm **Benign Adversarial Training (BAT)**. It is composed of two major components: **(i) Reweighting** to mitigate the impact from those poisoning atypical samples; **(ii) Discrimination Loss** to further protect the performance on typical samples. Combining these two components, BAT can maintain the performance of typical samples, but still preserve the ability to fit those “useful” atypical samples. Although BAT can hardly improve the overall robustness compared to adversarial training [8] (when it achieves highest robustness), BAT can present the comparable robustness and higher overall clean accuracy. Compared to [8], BAT enjoys better clean vs. adversarial accuracy trade-off. More results and discussions can be found in the experiment section. Next, we detail these two components of BAT.

4.1 Reweighting & Poisoning Score

Following PGD adversarial training [8], BAT starts by fitting manually generated adversarial examples using Projected Gradient Descent (PGD). During the training process, in order to identify which samples can poison or degrade the model’s performance, we define the “poisoning score” for each training (adversarial) sample $x_i^{\text{adv}}$ as:

$$ \mathbf{q}(x_i^{\text{adv}}) = \max_{t \neq y} F_t(x_i^{\text{adv}}) $$

which is the model $F_t$’s largest prediction score (after softmax) to a wrong class $t$ other than the true class $y$. A high poisoning score suggests that the current model predicts the sample $x_i^{\text{adv}}$ to be a wrong class with high confidence. Note that the atypical samples are usually fitted into the model later than typical samples. Under a model with a few atypical samples fitted, if there is an atypical sample $x_i^{\text{adv}}$ with high $\mathbf{q}(x_i^{\text{adv}})$, $x_i^{\text{adv}}$ is very likely to be close to the distribution of a wrong class instead of its true class. In Fig. [3], we present several atypical (adversarial) samples with poisoning scores larger than 0.8. These samples present features which are very similar to the wrong class that the model predicts. Therefore, fitting these atypical samples could cause the model to learn mixed concepts of features and degrades the model performance.

During the training process, we desire that the model should mitigate the influence from the atypical samples with large poisoning scores, but still learn other “useful /benign” atypical samples. Thus, we design a cost-sensitive reweighting strategy which downweights the cost of atypical samples with large poisoning scores. In particular, we specify the weight value $w_i$ for each training sample $x_i^{\text{adv}}$ as:

$$ w_i = \begin{cases} \exp(-\alpha \cdot \mathbf{q}(x_i^{\text{adv}})) & \text{if } \text{mem}(x_i) > \sigma \text{ and } \arg \max_t F_t(x_i^{\text{adv}}) \neq y \\ 1 & \text{Otherwise.} \end{cases} $$

---

**Figure 3: Examples of Poisoning Atypical Samples**

During adversarial training, the models will not only memorize the labels of atypical samples, but also their semantic features. As a result, the adversarial trained DNNs will construct more mixed concepts of features, if the “poisoning” atypical samples exist.
where $\alpha \in \mathbb{R}^+$ and $\sigma$ control the size of the reweighted atypical set. Since the function $\exp(-\alpha(\cdot))$ is decreasing and ranges from 1 to 0, the samples with large poisoning scores will be assigned with small weights close to 0. Thus, in the reweighting algorithm, we train the model to find optimal model $F_{\text{rw}}$ by assigning the weight vector $w$:

$$F_{\text{rw}} = \arg \min_{F} \frac{1}{\sum_i w_i} \sum_i [w_i \cdot L(F(x_{i}^{\text{adv}}), y_i)]$$

(6)

In the training process, those (adversarial) training samples with large poisoning scores are assigned with small weights and correspondingly their influence will be largely mitigated.

### 4.2 Discrimination Loss

Even though the reweighting method discussed above can help mitigate the impact from those “poisoning” atypical samples, we observe that the model’s performance (especially on the typical samples) is still inevitably decreasing with training goes. For example, as shown in Fig. 2 even we exclude all atypical samples in CIFAR100 dataset, the typical samples’ clean and adversarial accuracy still slightly drop. Fortunately, motivated from the discussions in Section 3.2, the performance of the typical samples is strongly related to the distance of the representation vectors of different classes. It means the smaller representation distance among different classes, the poorer discrimination the model has to distinguish different classes. Thus, to further preserve the model’s performance on typical samples, we introduce Discrimination Loss to regularize the feature space discrimination for typical samples among different classes:

$$L_{DL}(F) = \mathbb{E}_{(x_i, y_i), (x_j, y_j) \in \{(x_k, y_k)\}_{k=1}^K} \left[ -\log \frac{e^{h(x_i^{\text{adv}})h(x_j^{\text{adv}})/\tau}}{\sum_{k=1}^K e^{h(x_k^{\text{adv}})h(x_j^{\text{adv}})/\tau}} \right].$$

(7)

where $y_i = y_j$; $y_k \neq y_i$, $\forall k = 1, 2, \ldots K$

$\text{mem}(x_i), \text{mem}(x_j), \text{mem}(x_k) < \sigma$

Specifically, $h(\cdot)$ is the model $F$’s pen-ultimate layer’s output representation; $\tau \in \mathbb{R}^+$ is the temperature value and $K \in \mathbb{Z}^+$ is a fixed number of the samples that randomly chosen with different labels with $y_k$. Intuitively, constraining the Discrimination Loss imposes the model to output similar representations for the typical sample pairs $(x_i^{\text{adv}}, x_j^{\text{adv}})$ in the same class, and distinct representations for $(x_i^{\text{adv}}, x_k^{\text{adv}})$ from different classes. Similar approaches have been widely used in representation learning, such as Triplet Loss [27] and contrastive learning methods [28], which stress the good property of DNNs’ representation space. With the Discrimination Loss incorporated, the final objective of BAT is formulated as:

$$F_{\text{bat}} = \arg \min_{F} \left( \frac{1}{\sum_i w_i} \sum_i [w_i \cdot L(F(x_i^{\text{adv}}), y_i)] + \beta \cdot L_{DL}(F) \right)$$

(8)

where $\beta > 0$ that controls the intensity of Discrimination Loss. The algorithm scheme of Benign Adversarial Training is presented in Appendix C.

### 5 Experiment

In this section, we present the experimental results to validate the effectiveness of the proposed BAT algorithm on benchmark datasets and compare it with state-of-the-art baseline methods. We also verify that both two components are helpful and necessary for BAT. The implementation of the BAT can be found via [https://anonymous.4open.science/r/benign-adv-77C5](https://anonymous.4open.science/r/benign-adv-77C5).

#### 5.1 Experimental Setup

In this work, in order to demonstrate the merit of BAT, we conduct the experiments mainly on benchmark datasets CIFAR100 [6] and Tiny ImageNet [20], which are relatively complex datasets (i.e., containing larger fractions of atypical samples). For both datasets, we study the algorithms under the model architectures ResNet and WideResNet (WRN) [15]. In this section, we only present the...
results of ResNet18 for CIFAR100 and ResNet32 for Tiny ImageNet and leave the results on WRN in Appendix D. As a fair comparison with BAT, we implement the baseline algorithms including PGD adversarial training \cite{trades} as well as its most popular variant TRADES \cite{trades}. In addition, we include several recent algorithms: MART \cite{mart} and GAIRAT \cite{gaarat}, which also incorporate reweighting strategies into adversarial training. For BAT and all baseline methods, we run the algorithms using SGD \cite{sgd} for 160 epochs with the learning rate that starts from 0.1 and decays by 0.1 after the epoch 80 and 120. More implementation details can be found in Appendix D.

**Performance on CIFAR100.** For a comprehensive comparison between different methods on CIFAR100, in the results from Table 2, we report the models’ clean accuracy and adversarial accuracy against $L_{\infty}$/$255$ PGD attack \cite{trades}, as well as their performance on the typical sample set and atypical sample set (as described in Section 3). A more comprehensive robustness evaluation on different attacking methods (including CW \cite{cw} and Auto-Attack \cite{auto-attack}) are presented in Appendix D. For BAT, we report its performance when choosing its optimal hyperparameter: $\alpha = 1 \& 2$ and $\beta = 0.2$. In the Section 5.2, we will discuss the impact of the selection of $\alpha$ and $\beta$ on BAT. For baseline methods, the settings and checkpoint selections follow the original papers’ suggestions.

![Figure 4: Clean Acc. (left) & Adv. Acc (right)](image)

**Table 2: Performance of BAT vs. Baselines on CIFAR100 Under ResNet18**

| Method              | All Acc. | All Adv. | Typical Acc. | Typical Adv. | Atyp. Acc. | Atyp. Adv. |
|---------------------|----------|----------|--------------|--------------|------------|------------|
| PGD Train (Best Adv.) | 56.9     | 27.4     | 90.6         | 59.0         | 29.5       | 7.7        |
| PGD Train (Best Clean) | 57.8     | 21.9     | 88.3         | 51.0         | 37.3       | 8.3        |
| TRADES ($1/\lambda = 5$) | 56.6     | 26.9     | 88.9         | 57.1         | 10.9       | 10.9       |
| MART \cite{mart} | 51.8     | 30.4     | 62.2         | 31.6         | 10.1       | 8.2        |
| GAIRAT \cite{gaarat} | 58.2     | 27.8     | 60.7         | 31.6         | 10.1       | 8.2        |
| BAT ($\alpha = 1 \& 2$, $\beta = 0.2$) | 59.5     | 27.3     | 92.3         | 8.7          | 36.3       | 8.7        |
| BAT ($\alpha = 2 \& 2$, $\beta = 0.2$) | 59.3     | 27.4     | 92.3         | 8.7          | 33.1       | 7.4        |

From the results in Table 2, we can find that BATs enjoy good clean & adversarial accuracy trade-off among all methods. It is because BATs can obtain the highest overall clean accuracy ($\sim 59.5\%$), as well as comparably good adversarial accuracy ($\sim 27.3\%$) with baseline methods. The only exception is MART \cite{mart} which has the higher adversarial accuracy around $\sim 30\%$. However, MART \cite{mart} has much lower clean accuracy than BATs.

To gain a deeper understanding of the working mechanism of BAT, we focus on the performance on BAT vs. PGD adversarial training \cite{trades}. Compared with PGD adversarial training with highest adversarial accuracy (PGD Train (Best Adv.)), the BAT methods are $2 ~ 3\%$ higher in terms of overall clean accuracy, and similar adversarial accuracy. The improvement of clean accuracy is mainly due to BATs’ capacity to fit much more atypical samples. For example, the clean accuracy of BAT ($\alpha = 1 \& 2$, $\beta = 0.2$) is about $6\%$ higher than PGD Train (Best Adv.). On the other hand, compared to PGD adversarial training with the highest clean accuracy (PGD Train (Best Clean)), BATs have much better overall adversarial robustness and slightly better clean accuracy. This is mainly due to BAT’s advantage on typical samples, because BATs can successfully prevent the performance drop of typical samples during the training process. Since BATs can achieve good performance on both typical samples and atypical samples, BATs outperform PGD adversarial training in CIFAR100.

In Fig. 4, we also show the change of each models’ overall clean accuracy (Fig 4 (left)) & adversarial accuracy (Fig 4 (right)) along with the training progress. Interestingly, similar to PGD adversarial training, all baseline methods cannot achieve the optimal clean and adversarial accuracy at the same moment. They always achieve the best adversarial accuracy around Epoch 80 (right after the first time weight decay), and the best clean accuracy on the last epochs. However, for BATs, since they can effectively prevent the robustness dropping in the late epochs, BATs are able to train until last epochs and enjoy good clean and adversarial accuracy simultaneously.

**Performance on Tiny ImageNet.** Tiny ImageNet \cite{tiny-imagenet} contains 200 classes of the images in the original ImageNet \cite{imagenet} dataset, with 500 training images for each class, and image size $64 \times 64$. In our
experiments, we only choose the first 50 classes in Tiny ImageNet for training and prediction. Since the image size is $64 \times 64$, for both training and robustness evaluation, we consider the adversarial attacks are bounded by $l_{\infty}$-norm-4/255. In Table 3, we report the performance of BAT and baseline methods. Similar to the conclusions we can make from CIFAR100, BATs can achieve the highest overall clean accuracy and comparably good adversarial accuracy with baseline methods.

| Method                | All Acc. | All Adv. | Typical Acc. | Typical Adv. | Atyp. Acc. | Atyp. Adv. |
|-----------------------|----------|----------|--------------|--------------|------------|------------|
| Adv. Train (Best Adv.)| 56.3     | 32.3     | 97.5         | 85.3         | 41.5       | 9.6        |
| Adv. Train (Best Clean)| 58.2     | 30.5     | 98.0         | 80.4         | 44.7       | 9.1        |
| TRADES ($1/\lambda = 5$) | 55.4     | 28.8     | 97.3         | 77.4         | 38.8       | 9.6        |
| MART [18]              | 56.2     | 34.5     | 97.7         | 85.1         | 41.4       | 13.6       |
| GAIRAT [19]            | 58.4     | 30.4     | 98.0         | 81.7         | 45.7       | 7.8        |
| BAT ($\alpha = 1, \beta = 0.2$) | 59.4     | 32.0     | 98.4         | 83.9         | 48.4       | 10.2       |
| BAT ($\alpha = 2, \beta = 0.2$) | 59.4     | 32.9     | 99.1         | 86.9         | 45.7       | 10.9       |

5.2 Ablation Study

In this subsection, we study the potential impact of the hyperparameters chosen in BAT, which is $\alpha$ that controls the Reweighting process and $\beta$ that controls the Discrimination Loss. In Fig. 5 we conduct the experiments on CIFAR100 with BAT when $\alpha$ is chosen in $[0,1,2,3]$ and $\beta$ is in $[0, 0.1, 0.2, 0.3]$. In Fig. 5 we show each model’s overall clean accuracy (left) & adversarial accuracy (right) along the Z-axis, and X-axis / Y-axis indicate the models’ corresponding variables of $(\alpha, \beta)$. Note that when $\alpha = \beta = 0$, the BAT method regresses to original PGD adversarial training. From the result, we can see that a positive pair of $(\alpha, \beta)$ can benefit both model clean and adversarial accuracy. Therefore, both two components Reweighting and Discrimination Loss of BAT are helpful and necessary. However, when $\alpha$ or $\beta$ is relatively too large, it will hurt the BAT’s clean accuracy. As a result, in CIFAR 100, when $\alpha = 1$ or 2 and $\beta = 0.2$, BAT can achieve the optimal performance.

6 Related Works

Memorization and atypical samples. The memorization effect of overparameterized DNNs have been extensively studied both empirically [21, 32] and theoretically [23]. From traditional views, the memorization can be harmful to the model generalization, because it makes DNN models easily fit those outliers and noisy labels. However, recent studies point out the concept of “benign overfitting” [31], which suggests the memorization effect necessary for DNNs to have extraordinary performance on modern machine learning tasks. Especially, the recent work [2] empirically figures out those atypical/rare samples in benchmark datasets and show the contribution from memorizing atypical samples to the DNN’s performance. Besides the work [2], there are also other strategies [33] to find atypical samples in training dataset. Notably, our work is not the first effort to study the influence of memorization on DNN’s adversarial robustness. A previous study [34] illustrates that memorizing the mis-labeled samples might be a reason to cause the DNNs’ adversarial vulnerability. In our paper, we focus on atypical samples, which appear much more frequently in common datasets, and we study their impacts especially on adversarial training algorithms [8, 17, 4, 5].

Adversarial robustness. Adversarial training methods [8, 17, 18, 21, 14] are considered as one of the most reliable and effective methods to protect DNN models against adversarial attacks [10, 24]. However, there are several intrinsic properties of adversarial training which requires deeper understandings. For example, they always suffer from poor robustness generalization [13, 14], and they always present strong trade-off relation between clean accuracy vs. robustness [12, 17]. Our work aims to study these properties from the data perspective and demonstrate the significant connection of the memorization effect with these properties.

7 Conclusion

In this paper, we draw significant connections of the memorization effect of deep neural networks with the behaviors of adversarial training algorithms. Based on the findings, we devise a novel
algorithm BAT to enhance the performance of adversarial training. The findings of the paper can motivate the futures studies in building robust DNNs with more attention on the data perspective.

References

[1] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, pages 954–959, 2020.

[2] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. arXiv preprint arXiv:2008.03703, 2020.

[3] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. Proceedings of the National Academy of Sciences, 117(48):30063–30070, 2020.

[4] Niladri S Chatterji and Philip M Long. Finite-sample analysis of interpolating linear classifiers in the overparameterized regime. arXiv preprint arXiv:2004.12019, 2020.

[5] Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. Harmless interpolation of noisy data in regression. IEEE Journal on Selected Areas in Information Theory, 1(1):67–83, 2020.

[6] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25:1097–1105, 2012.

[8] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

[9] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236, 2016.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[11] 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.

[12] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. arXiv preprint arXiv:1805.12152, 2018.

[13] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Mądry. Adversarially robust generalization requires more data. arXiv preprint arXiv:1804.11285, 2018.

[14] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In International Conference on Machine Learning, pages 8093–8104. PMLR, 2020.

[15] 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.

[16] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389, 2012.

[17] Hongyung Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. arXiv preprint arXiv:1901.08573, 2019.
[18] 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, 2019.

[19] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. arXiv preprint arXiv:2010.01736, 2020.

[20] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. CS 231N, 7:7, 2015.

[21] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530, 2016.

[22] Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. Advances in computational mathematics, 13(1):1–50, 2000.

[23] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research, 3(Nov):463–482, 2002.

[24] Han Xu, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil Jain. Adversarial attacks and defenses in images, graphs and text: A review. arXiv preprint arXiv:1909.08072, 2019.

[25] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In International Conference on Artificial Intelligence and Statistics, pages 4313–4324. PMLR, 2020.

[26] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In Advances in Neural Information Processing Systems, pages 125–136, 2019.

[27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823, 2015.

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

[29] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT’2010, pages 177–186. Springer, 2010.

[30] 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.

[31] 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.

[32] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. arXiv preprint arXiv:1912.02292, 2019.

[33] Nicholas Carlini, Ulfar Erlingsson, and Nicolas Papernot. Distribution density, tails, and outliers in machine learning: Metrics and applications. arXiv preprint arXiv:1910.13427, 2019.

[34] Amartya Sanyal, Puneet K Dokania, Varun Kanade, and Philip HS Torr. How benign is benign overfitting? arXiv preprint arXiv:2007.04028, 2020.

[35] Daniel J Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. Transportation Research Part A: Policy and Practice, 77:167–181, 2015.
[36] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology*, 2(4), 2017.

[37] Abdul Adeel Mohammed, Rashid Minhas, QM Jonathan Wu, and Maher A Sid-Ahmed. Human face recognition based on multidimensional pca and extreme learning machine. *Pattern Recognition*, 44(10-11):2588–2597, 2011.
Appendix

In the supplementary materials, we provide more details about atypical samples and the proposed algorithm, as well as the full experimental results of the preliminary study and the proposed method.

- Additional Introduction of Atypical Samples Appendix A
- Additional Results of Preliminary Study (in Section 3.1) Appendix B.1
  Traditional ERM & Adversarial Training
- Additional Results of Preliminary Study (in Section 3.2) Appendix B.2
  Traditional ERM & Adversarial Training
- Full Training Scheme of BAT Appendix C
- Additional Results of BAT Appendix D
- Boarder Impacts of this Paper Appendix E

A Additional Introduction of Atypical Samples

In this section, we provide additional introductions about the atypical samples in common datasets. In Fig. 6, we provide several examples of images from CIFAR10, CIFAR100 [6] and Tiny ImageNet [20] respectively, with different memorization value (as defined in Section 2.1) around 0.0, 0.5, 1.0. These examples suggest that if the memorization value of an image is large, this image is very likely to be “atypical”, as it presents very distinct semantic features with the images in the main distribution (with memorization value 0.0). The detailed introduction about how to estimate the memorization value in practice can be found in the work [2].

Figure 6: Examples of Images with Different Memorization Values
In Fig. 7, we provide histograms to show the distribution of the estimated memorization values of all training samples from CIFAR10, CIFAR100 and Tiny ImageNet. From Fig. 7, we can observe that atypical samples (with high memorization value > 0.15) consist of a significant fraction (over 40% & 50% respectively) in CIFAR100 and Tiny ImageNet. In CIFAR10, they also consist of a non-ignorable fraction which is over 10%.

(a) CIFAR100.  
(b) CIFAR10.  
(c) Tiny ImageNet.

Figure 7: Frequencies of Training samples with Different Memorization Values in Various Datasets

In Fig. 8, we provide several pairs of images with high influence value (as defined in Section 2.1) which is over 0.15. In each pair, the training sample also has a high memorization value over 0.15. These examples suggest that there exist atypical samples in both training & test sets of CIFAR10, CIFAR100 and Tiny ImageNet. A pair of atypical samples (in the training set and test set) with a high influence value are visually very similar. Moreover, since they have high influence values, removing the atypical samples in the training set is very likely to cause the model to fail on the test atypical samples. Therefore, without memorizing the atypical sample in the training set, the model can hardly predict the atypical samples in the test set.

![High Influence Pairs with Influence Value > 0.15](image)

Figure 8: High Influence Pairs with Influence Value > 0.15

**B Additional Results for Preliminary Study**

In this section, we provide the full results of the preliminary study in Section 3 on CIFAR10, CIFAR100 and Tiny ImageNet, to illustrate the distinct behaviors of the memorization effect between traditional ERMs and adversarial training. In both ERM and adversarial training, we train the models under ResNet18 and WideResNet28-10 (WRN28) architectures. In the experiments, we train the models for 200 epochs with learning rate 0.1, momentum 0.9, weight decay 5e-4, and decay the learning rate by 0.1 at the epoch 150 and 200. For adversarial training, we conduct experiments using PGD adversarial training [8] by default to defense against $l_\infty$-8/255 adversarial attack, with the exception on Tiny ImageNet, which is against $l_\infty$-4/255 attack. For robustness evaluation, we conduct a 20-step PGD attack.
B.1 Additional Results for Preliminary Study - Section 3.1

In this subsection, we provide more experimental results to validate the statement in Section 3.1 where we state that fitting atypical samples in adversarial training can only improve the clean accuracy of test atypical samples, but hardly help their adversarial robustness. We provide full empirical results to show that: (i) In traditional ERM, fitting atypical samples improves the clean accuracy of test atypical samples. (ii) In adversarial training, fitting (adversarial) atypical samples improves the clean accuracy of test atypical samples but can hardly improve the adversarial robustness of them. The experimental setting follows Section 3.1 where we apply traditional ERM and adversarial training on original CIFAR10, CIFAR100, Tiny ImageNet datasets. We evaluate the model’s clean accuracy and adversarial accuracy on training atypical set $D_{atyp} = \{x_i \in D : \text{mem}(x_i) > 0.15\}$ and its corresponding test atypical set $D'_{atyp} = \{x'_j \in D' : \text{infl}(x_i, x'_j) > 0.15, \forall x_i \in D_{atyp}\}$.

(i) Additional Results for Preliminary Study - Section 3.1 In Traditional ERM

Fig. 9, Fig. 10 and Fig. 11 report the performance (clean accuracy) of traditional ERM, which is evaluated on atypical sets under ResNet18 (left) and WRN28 (right) on CIFAR100, CIFAR10 and Tiny ImageNet. From the figures, we can observe that fitting atypical samples during training can effectively help the models to achieve good clean accuracy on test atypical samples in all datasets. Note that here we only report clean accuracy as they are not robust against adversarial attacks.

![Figure 9: Clean Accuracy on Atypical Set of CIFAR100](image1)

(a) ResNet18.  
(b) WRN28.

Figure 9: Clean Accuracy on Atypical Set of CIFAR100

![Figure 10: Clean Accuracy on Atypical Set of CIFAR10](image2)

(a) ResNet18.  
(b) WRN28.

Figure 10: Clean Accuracy on Atypical Set of CIFAR10

![Figure 11: Clean Accuracy on Atypical Set of Tiny ImageNet](image3)

(a) ResNet32.  
(b) WRN28.

Figure 11: Clean Accuracy on Atypical Set of Tiny ImageNet
(ii) Additional Results for Preliminary Study - Section 3.1 In Adversarial Training

Fig. 12, Fig. 13 and Fig. 14 report the performance of adversarially trained models. We evaluate the clean accuracy and adversarial accuracy on the training atypical set $D_{\text{atyp}}$ and test atypical set $D'_{\text{atyp}}$. From the results, we can observe that although fitting atypical samples can help the model to have modest clean accuracy on test atypical samples, the adversarial robustness of them is constantly low and can hardly be improved during the whole training process.

![Graphs showing clean accuracy and adversarial accuracy for ResNet18 and WRN28 on CIFAR100, CIFAR10, and Tiny ImageNet datasets.](image)

Figure 12: Clean Accuracy and Adversarial Accuracy on Atypical Set of CIFAR100

Figure 13: Clean Accuracy and Adversarial Accuracy on Atypical Set of CIFAR10

Figure 14: Clean Accuracy and Adversarial Accuracy on Atypical Set of Tiny ImageNet

B.2 Additional Results for Preliminary Study - Section 3.2

In this subsection, we provide more experimental results to validate the statement in Section 3.2 where we state that fitting atypical samples in adversarial training can hurt the performance (clean & adversarial accuracy) of typical samples. We provide full empirical results to show that: (i) In traditional ERM, fitting atypical samples will not hurt the models’ performance (clean accuracy) of typical samples. (ii) In adversarial training, fitting atypical samples can degrade the clean & adversarial accuracy of typical samples. (iii) In adversarial training, fitting atypical samples can degrade the quality of learned representations, especially the models’ discrimination between classes. The experimental setting follows Section 3.2 where we train the models for several trials on resampled (CIFAR100, CIFAR10, Tiny ImageNet) datasets: each dataset is constructed with the whole training typical set $D_{\text{typ}}$, and a part of the training atypical set $D_{\text{atyp}}$ (randomly sample 0%, 20% and 100% in $D_{\text{atyp}}$). We evaluate the models’ performance on test “typical” set $D'_{\text{typ}}$ which is defined in Section 3.2.
(i) Additional Results for Preliminary Study - Section 3.2 In Traditional ERM

Fig. 15, Fig. 16 and Fig. 17 report the performance of traditional ERM, trained on different resampled datasets with different amount of atypical samples existed. The figures report the clean accuracy on test typical set of CIFAR100, CIFAR10 and Tiny ImageNet. We also leave the robustness performance out here as the models are not robust to adversarial attacks. From the results, we can conclude that in traditional ERM, fitting atypical samples will not degrade the models’ performance on typical samples. For example, in CIFAR100 dataset, with 100% atypical samples included (Atypical-100%), the accuracy on the test typical set is even slightly higher than the model trained without atypical samples (Atypical-0%). This conclusion is consistent for all three datasets and model architectures.

Figure 15: Clean Accuracy on Typical Set of CIFAR100

Figure 16: Clean Accuracy on Typical Set of CIFAR10

Figure 17: Clean Accuracy on Typical Set of Tiny ImageNet

(ii) Additional Results for Preliminary Study - Section 3.2 In Adversarial Training

Fig. 18, Fig. 19 and Fig. 20 report the performance of adversarial training, on different resampled datasets with different amount of atypical samples existed. The figures report both clean and adversarial accuracy on test atypical sets of CIFAR100, CIFAR10 and Tiny ImageNet. Based on the experimental results, we find that including more atypical samples can cause the model have worse performance on typical samples in all three datasets. In datasets with a large portion of atypical samples, such as CIFAR100, the negative effects of atypical samples are more obvious. In CIFAR100, training on datasets with 100% atypical samples (Atypical 100%) can cause the clean & adversarial accuracy drop by $\sim 7\%$ and $8\%$, respectively.
(iii) Additional Results for Preliminary Study - Section 3.2 Classwise Representation Distance

Here, we provide additional evidence to show that in adversarial training, fitting atypical samples can degrade the quality of DNN’s learned representations (as proposed in Section 3.2). In Fig. 18, we measure the Cosine Distance (defined in Section 3.2) of the representations for (typical) samples from different classes. In these figures, we provide detailed results of both traditional ERM and adversarial training under ResNet18 and WRN28. From the results, we can see that in adversarial training, more atypical samples can cause the smaller classwise Cosine Distance of the models (on their last epochs). Moreover, during the training process of adversarial training, the classwise Cosine Distance first increases and then starts to decrease, especially when there are more atypical samples are fitted. For example, in CIFAR100 under ResNet18, the Cosine Distance starts to decrease at around Epoch 100, which is when many more atypical samples are fitted. These results suggest that, in adversarial training, fitting atypical samples can be an important reason to cause the models to produce poor representations. As a comparison, in Traditional ERM, the classwise Cosine Distance keeps increasing from the first epoch to the last one. Although given the same epoch, the models trained with more atypical samples also have smaller Cosine Distance, they would not harm the model’s final performance. It is because the models trained with more atypical samples can also have relatively large classwise Cosine Distance in their last epochs.

C The Detailed Training Scheme of BAT

In this section, we provide the detailed training scheme of BAT in Algorithm 1. In particular, BAT algorithm starts from a randomly initialized neural network. On each mini-batch, it applies PGD attack to generate (training) adversarial examples (Step 5). Following the Eq 4 and Eq 5, BAT calculates which samples are likely to be Poisoning Atypical Samples and their corresponding weight
values (Step 6). Under the current mini-batch, next, BAT calculates the Discrimination Loss of the typical samples $D_{\text{typ}}$ (Step 8). Finally, BAT uses SGD to update the model parameter to minimize the reweighted adversarial loss regularized by $\beta$ times discrimination loss (Step 8).

Algorithm 1 The Benign Adversarial Training (BAT) Algorithm

1: **Input:** Training dataset $\mathcal{D}$, with typical set $D_{\text{typ}} = \{x \in \mathcal{D}; \text{mem}(x) \leq \sigma\}$, atypical set $D_{\text{atyp}} = \{x \in \mathcal{D}; \text{mem}(x) > \sigma\}$. Targeted type of adversarial attack: $l_\infty$-attack. Hyperparameters $\alpha, \beta \in \mathbb{R}^+$. 
2: Randomly initialize the network $F$
3: **repeat**
4: Fetch mini-batch data $\{(x_i, y_i)\}$ at current epoch
5: Using PGD to generate adversarial training sample $\{(x_i^{\text{adv}}, y_i)\}$
6: Calculate Poisoning Score $q(x_i^{\text{adv}})$ and weight $w_i$ as in Equation 4 and Equation 5
7: Calculate Discrimination Loss $L_{DL}(F)$ within the current mini-batch, as in Equation 7
8: Update $F$ by SGD on the objective: $L_{\text{BAT}} = \sum_{i=1}^{N} w_i \sum_{j} [w_j \cdot L(F(x_i^{\text{adv}}), y_i)] + \beta \cdot L_{DL}(F)$.
9: **until** End of Training

D Additional Results for Experiment

In this section, we provide additional experimental results to validate the effectiveness of BAT method. In Table 4 and Table 5, we provide the results of BAT and baseline models on CIFAR100 dataset.
under ResNet18 and WRN28 architectures. In Table 4 and Table 5, we provide the results of BAT and baseline models on Tiny ImageNet dataset under ResNet32 and WRN28 architectures. In the experiments, we train the models for 160 epochs with learning rate 0.1, momentum 0.9, weight decay 5e-4, and decay the learning rate by 0.1 at the epoch 80 and 120. To have a more comprehensive and reliable adversarial robustness, in addition to PGD adversarial attack [8], we also measure the model’s adversarial accuracy via other attack algorithms, including FGSM attack [10], CW attack [30] and Auto Attack [41]. The results show that the BAT method can consistently outperform baseline models, as BAT has better clean accuracy vs. adversarial accuracy trade-off. Under ResNet18, the BAT method achieves comparable adversarial accuracy to the best baseline methods, and BATs have the highest clean accuracy. Under WRN28, the BAT method has both higher clean & adversarial accuracy than the baseline methods.

| Method                  | Clean Acc. | FGSM | PGD  | CW  | AA.  |
|-------------------------|------------|------|------|-----|------|
| PGD Train (Best Adv.)   | 56.9       | 36.0 | 27.4 | 25.4| 23.6 |
| PGD Train (Best Clean)  | 57.8       | 33.5 | 21.9 | 22.5| 20.2 |
| TRADES (1/λ = 5)        | 56.6       | 36.5 | 26.9 | 25.3| 23.9 |
| MART [18]               | 51.8       | 36.1 | 30.4 | 25.8| 24.4 |
| GAIRAT [19]             | 58.2       | 36.5 | 27.8 | 25.9| 23.8 |
| BAT (α = 1, β = 0.2)    | 59.5       | 37.3 | 27.3 | 26.6| 24.3 |
| BAT (α = 2, β = 0.2)    | 59.3       | 37.1 | 27.4 | 26.5| 24.0 |

| Method                  | Clean Acc. | FGSM | PGD  | CW  | AA.  |
|-------------------------|------------|------|------|-----|------|
| PGD Train (Best Adv.)   | 59.7       | 34.9 | 24.7 | 24.2| 22.5 |
| PGD Train (Best Clean)  | 57.3       | 34.5 | 24.9 | 24.6| 22.9 |
| TRADES (1/λ = 5)        | 56.5       | 36.1 | 26.8 | 25.3| 23.8 |
| MART [18]               | 60.2       | 34.8 | 24.4 | 24.8| 22.9 |
| GAIRAT [19]             |            |      |      |     |      |
| BAT (α = 1, β = 0.2)    | 62.0       | 38.6 | 28.5 | 26.5| 24.8 |
| BAT (α = 2, β = 0.2)    | 61.4       | 38.9 | 28.2 | 26.3| 24.8 |

| Method                  | Clean Acc. | FGSM | PGD  | CW  | AA.  |
|-------------------------|------------|------|------|-----|------|
| Adv. Train (Best Adv.)  | 56.3       | 37.5 | 32.3 | 29.8| 29.8 |
| Adv. Train (Best Clean) | 58.2       | 36.8 | 30.5 | 28.8| 28.4 |
| TRADES (1/λ = 5)        | 55.4       | 35.2 | 28.8 | 27.0| 27.0 |
| MART [18]               | 56.2       | 38.1 | 34.5 | 31.8| 32.0 |
| GAIRAT [19]             | 58.4       | 37.3 | 30.4 | 28.9| 29.0 |
| BAT (α = 1, β = 0.2)    | 59.4       | 40.4 | 32.0 | 31.5| 32.0 |
| BAT (α = 2, β = 0.2)    | 59.4       | 41.3 | 32.9 | 31.8| 32.4 |

| Method                  | Clean Acc. | FGSM | PGD  | CW  | AA.  |
|-------------------------|------------|------|------|-----|------|
| Adv. Train (Best Adv.)  | 58.9       | 35.7 | 31.7 | 30.0| 30.0 |
| Adv. Train (Best Clean) | 60.0       | 35.1 | 30.1 | 28.8| 28.5 |
| TRADES (1/λ = 5)        | 59.7       | 37.4 | 32.0 | 31.8| 32.0 |
| MART [18]               | 58.2       | 41.0 | 35.6 | 33.9| 34.1 |
| GAIRAT [19]             | 59.9       | 38.3 | 32.4 | 31.0| 31.1 |
| BAT (α = 1, β = 0.2)    | 62.3       | 41.6 | 35.8 | 33.7| 34.3 |
| BAT (α = 2, β = 0.2)    | 62.4       | 43.1 | 37.4 | 35.3| 35.6 |
E Boarder Impact

Nowadays, deep neural networks (DNNs) have been widely applied to solve various machine learning tasks, especially on many safety-critical tasks such as autonomous vehicle [35], AI healthcare [36] and ID authentication [37], etc. However, the existence of adversarial attacks [24] brings huge threats to the safety of these DNNs’ applications. As one of the most successful approaches to defend DNNs against adversarial attacks, adversarial training methods [8, 21] still suffer from several disadvantages. For example, for a DNN model to achieve good adversarial robustness, it usually sacrifices its clean accuracy. Adversarially trained DNNs also present strong overfitting effects. In our study, we draw important findings to explain these drawbacks of adversarial training from the data perspective and empirically validate these issues’ relation to atypical samples in the data distribution. Moreover, the currently existed adversarially trained models can only achieve satisfactory performance on simple datasets such as MNIST and CIFAR10. Motivated from our findings, we propose a new method to improve the performance of adversarial training, especially on more complex datasets. We anticipate that our findings and method are helpful for further studies to improve the effectiveness and feasibility of adversarial training and eventually build safer DNNs.