A3T: accuracy aware adversarial training

Enes Altinisik1 · Safa Messaoud1 · Husrev Taha Sencar1 · Sanjay Chawla1

Received: 29 November 2022 / Revised: 9 March 2023 / Accepted: 11 April 2023 / Published online: 14 July 2023
© The Author(s) 2023

Abstract
Adversarial training has been empirically shown to be more prone to overfitting than standard training. The exact underlying reasons are still not fully understood. In this paper, we identify one cause of overfitting related to current practices of generating adversarial examples from misclassified samples. We show that, following current practice, adversarial examples from misclassified samples result in harder-to-classify samples than the original ones. This leads to a complex adjustment of the decision boundary during training and hence overfitting. To mitigate this issue, we propose A3T, an accuracy aware AT method that generates adversarial examples differently for misclassified and correctly classified samples. We show that our approach achieves better generalization while maintaining comparable robustness to state-of-the-art AT methods on a wide range of computer vision, natural language processing, and tabular tasks.

Keywords Adversarial training · Overfitting in adversarial training · Accuracy aware adversarial training

1 Introduction

Deep learning achieved excellent generalization on a wide variety of tasks (Jumper et al., 2021; Fawzi et al., 2022; Ramesh et al., 2021; He et al., 2015; Vaswani et al., 2017). However, deepnets are also known to be vulnerable to adversarial attacks (Papernot et al., 2017; Kurakin et al., 2017, 2017; Schönherr et al., 2019), due to the non-smooth nature of their associated loss landscapes. Adversarial training (AT) Goodfellow et al. (2015) is established as the most...
efficient mechanism for defending against these attacks and consists of training on adversarially perturbed samples instead of on the original ones. Yet, AT is also more prone to overfitting (Madry et al., 2018; Schmidt et al., 2018; Rice et al., 2020; Dong et al., 2022; Yu et al., 2022; Stutz, 2022) and therefore results in lower generalization than standard training. The reasons behind this are still under-explored. Nevertheless, several classical and modern deep learning remedies for overfitting, e.g., regularization (Zhang et al., 2022; Miyato et al., 2015; Gan et al., 2020; Chen et al., 2021) and data augmentation Rice et al. (2020); Wu et al. (2020); Gowal et al. (2020) have been proposed without a full understanding of the root causes.

Recently, Dong et al. (2022) show that overfitting is due to augmenting the training data with ‘hard’ to classify adversarial examples that incentivize a complicated adjustment of the decision boundaries. The authors argue that these ‘hard’ adversarial examples are generated from samples close to the decision boundary with noisy or inappropriate ground-truth one-hot encoded labels. They show that previously proposed techniques for label and weight smoothing (Pang et al., 2021; Huang et al., 2020; Chen et al., 2021) can alleviate this issue.

In this paper, we identify another category of ‘hard’ adversarial examples. Specifically, we show that adversarial examples generated from misclassified samples in every iteration of the training are, counter-intuitively, always placed maximally away from the decision boundary, as illustrated in Fig. 1. Hence, they induce a large loss leading to substantial perturbations in the local region of the misclassified point. This problem is more exacerbated in high-capacity models, as they have the flexibility of making local changes to a decision boundary. To address this problem, we propose Accuracy Aware Adversarial Training (A3T) which controls the generation of adversarial examples in a manner that is aware of the current predicted label of the original training sample. While Wang et al. (2020) and Ding et al. (2020) proposed formulations that generate adversarial examples differently for misclassified samples, their insights were empirically driven, and they do not identify nor directly address the root cause for poor generalization, i.e., adversarial examples from misclassified samples are generated maximally away from the decision boundary. Differently, A3T proposes a simpler fix that directly addresses the root cause, i.e., places adversarial examples from misclassified samples closer to the decision boundary.

We show that our A3T improves upon previously adopted techniques for mitigating overfitting in AT and achieves better generalization while having comparable robustness.
to state-of-the-art models on both toy experiments as well as on computer vision, natural language processing, and tabular applications. This hence demonstrates that we have identified a root cause for overfitting that is unaddressed in the literature.

2 Related work

2.1 Adversarial training (AT)

AT Goodfellow et al. (2015) is a defense mechanism against adversarial attacks which aim at fooling a trained supervised machine learning model by adding imperceptible perturbations to the inputs to cause misclassification. Modern day deepnets are particularly vulnerable to such attacks. Due to the non-smooth nature of their loss-landscapes, it is possible to find, for any input sample, a direction of steep gradient (adversarial direction) along which perturbations can lead to high loss and potentially a different prediction (Szegedy et al., 2014). To address this, in AT, the model is trained to correctly classify perturbed versions of the inputs. Formally, AT is formulated as a min-max program, searching for the best parameters $\theta$ of a classifier $f_\theta$ under the worst-case perturbation $\delta$ applied to an input $x$ (Madry et al., 2018), i.e.,

$$
\min_{\theta} \mathbb{E}_{(x,y) \sim D} \left[ \max_{\delta \in \Delta} \ell(f_\theta(x + \delta), y) \right],
$$

(1)

where $\ell$ is a loss function (e.g., cross-entropy loss) and $\Delta$ is a constrained set of imperceptible perturbations. The outer-optimization program is classically solved using gradient descent. Popular approaches for solving the constrained inner maximization program include Fast Gradient Sign Methods (FGSM) (Szegedy et al., 2014; Goodfellow et al., 2015) and Projected Gradient Descent (PGD) (Madry et al., 2018). In FGSM, the perturbation $\delta$ is computed via:

$$
\delta = \alpha \cdot \text{sgn}(\nabla_x \ell(f_\theta(x), y)),
$$

(2)

with $\alpha$ being the learning rate and $\text{sgn}$ the sign operator. PGD is a multi-step variant of FGSM. It starts at randomly initialized perturbations in the feasible set $\Delta$ and iteratively applies a gradient ascent update followed by a projecting onto $\Delta$:

$$
\delta^k = \Pi_{\Delta}(\delta^{k-1} + \alpha \nabla_x \ell(f_\theta(x + \delta^{k-1}), y)), \forall k \in [1, K].
$$

(3)

Here $\Pi_{\Delta}$ is the projection operator into $\Delta$ and $K$ is the number of steps.

AT is only defined for correctly classified samples. However, adversarial examples are generated following the same procedure for both correctly classified and misclassified samples. We will show that this makes the model more prone to overfitting.

2.2 Overfitting in AT

Overfitting in AT was first identified by Madry et al. (2018). After that, different variant to the standard AT objective [Eq. (1)] have been proposed to alleviate overfitting. These
variants can be classified into: (1) Local Distributional Smoothness (LDS) (Miyato et al., 2019; Zhang et al., 2019; Gan et al., 2020), (2) Margin Maximization (MM) (Balaji et al., 2019; Cheng et al., 2022), (3) Label-Smoothing Regularization (LSR) (Cheng et al., 2022) and (4) Misclassification-Aware (MA) (Wang et al., 2020) methods. Objectives of notable methods from these different families are presented in Table 3. However, all these attempts try to fix overfitting without identifying the root causes behind it. Naturally, creating a truly robust model would require a deeper understanding of the sources of its vulnerability. Recently, Dong et al. (2022) have shown that overfitting is caused by generating ‘hard’ to classify adversarial examples that are memorized by the network. The authors argue that these ‘hard’ adversarial examples are generated from samples close to the boundaries and hence their associated one-hot encoding ground truth label is likely to be inaccurate or noisy. To fix this, the authors propose label smoothing of the adversarial examples. Yu et al. (2022) empirically show that small-loss data is responsible for overfitting and propose a minimum loss constrained AT procedure that increases the loss of small-loss data.

Local distributional smoothness (LDS) approaches extends the robust optimization problem in Eq. (1) with a regularization term that encourages the label distribution around each sample point to be locally smooth. This is achieved via minimizing the KL divergence between the label distribution of the original samples and the one of the corresponding adversarially generated example (Miyato et al., 2019; Zhang et al., 2019). Intuitively, this results in forcing the decision boundary to be sufficiently away from the training samples (Miyato et al., 2019; Zhang et al., 2019). AT objectives from this family include TRADES (Zhang et al., 2019) and VILLA (Gan et al., 2020) (Lines 2 and 3 in Table 1). A major drawback of this approach is that it equally reinforces correct and incorrect predictions, i.e., it encourages the adversarial examples from misclassified samples to have the same label as the original (misclassified) samples. This results in a drop in generalization.

Margin maximization (MM) methods propose to individually tune the perturbation strength for each sample. Intuitively, adopting a fixed level of perturbation for all the samples can potentially result in pushing adversarial examples further away from the boundary, causing them to be mixed with samples of other classes. This would force the model to learn a non-smooth and complex decision boundary. AT objectives from this class include MMA (Ding et al., 2020), IAAT (Balaji et al., 2019), and CAT (Cheng et al., 2022) (lines 4–6 in Table 1). Note that MM methods only generate adversarial examples from correctly classified samples.

Label-smoothing regularization (LSR) methods propose applying a label-smoothing regularization. Intuitively, one cause for overfitting is the use of one-hot encoded ground-truth labels for adversarial examples (Dong et al., 2022), which essentially forces the model to assign an overconfident probability of one to all samples of a given class. This can be noisy or inappropriate in case of adversarial examples as they are deliberately generated by pushing close-to-the-boundary samples over to boundary to the side of another class. To address this, Cheng et al. (2022) apply this depending on the perturbation tolerance of each sample.

Misclassification-aware (MA) approaches propose extending the standard training loss with a misclassification-aware regularization (line 7 in Table 1). Wang et al. (2020) imperially show that adversarial examples from misclassified samples result in a drop in robustness. However, this has the same drawback as LDS approaches. In this work, we provide a deeper understanding of the drawback of the current practice in generating adversarial examples from misclassified samples and propose an easier fix to dress this.
| Method       | Training objective                                                                 | MM | LSR | MA | LDS |
|--------------|------------------------------------------------------------------------------------|----|-----|----|-----|
| 1 PGD-AT (Madry et al., 2018) | $\min_y E[\ell \left( \max_{\delta \in \Delta} f_\theta(x + \delta), y \right)]$ |    |     |    |     |
| 2 TRADES (Zhang et al., 2019)   | $\min_y E[\ell \left( f_\theta(x), y \right) + \lambda \times \ell \left( CE \left( \max_{\delta \in \Delta} f_\theta(x + \delta), f_\theta(x) \right) \right)]$ |    |     |    | ✓   |
| 3 VILLA (Gan et al., 2020)      | $\min_y E[\ell \left( f_\theta(x), y \right) + \lambda_1 \times \ell \left( \max_{\delta \in \Delta} f_\theta(x + \delta), y \right) + \lambda_2 \times \ell_{KL} \left( \max_{\delta \in \Delta} f_\theta(x + \delta), f_\theta(x) \right)]$ |    |     |    | ✓   |
| 4 MMA (Ding et al., 2020)       | $\min_y E[\ell \left( f_\theta(x), y \right) 1(f_\theta(x) \neq y) + \ell \left( \max_{\delta \in \Delta} f_\theta(x + \delta), y \right) 1(f_\theta(x) = y)]$ | ✓  |     |    |     |
| 5 IAAT (Balaji et al., 2019)    | $\min_y E[\ell \left( \max_{\delta \in \Delta} f_\theta(x + \delta), y \right)]$ | ✓  |     |    |     |
| 6 CAT (Cheng et al., 2022)      | $\min_y E[\ell \left( \max_{\delta \in \varepsilon} f_\theta(x + \delta), (1 - \varepsilon) y_1, \varepsilon \text{Dirichlet}(\beta) \right) + \left( \max_{y \neq y_1} [Z^\ast(x')_1 - [Z(x')_1]_{y_1} - \kappa] \right)]$ | ✓  | ✓  |    |     |
| 7 MART (Wang et al., 2020)      | $\min_y E[\ell \left( \max_{\delta \in \Delta} f_\theta(x + \delta), y \right) + \lambda \times \ell_{KL} \left( f_\theta(x + \delta), f_\theta(x) \right) \left( 1 - \left[ f_\theta(x + \delta) \right]_2 \right)]$ |    |    | ✓  | ✓  |
| 8 A3T                          | $\min_y E[\ell \left( f_\theta(x + \arg \max_{\delta \in \Delta} \ell \left( f_\theta(x + \delta), \hat{y} \right)), (1 - c_0) y_1 + c_0 \text{Dirichlet}(\beta) + \left( \max_{y \neq y_1} [Z(x + \delta)]_1 - [Z(x + \delta)]_{y_1} \right) \right)]$ |    |    | ✓  | ✓  |
| 8 A3T$^+$                       | $\min_y E[\ell \left( f_\theta(x + \arg \max_{\delta \in \Delta} \ell \left( f_\theta(x + \delta), \hat{y} \right)), (1 - c_0) y_1 + c_0 \text{Dirichlet}(\beta) + \left( \max_{y \neq y_1} [Z(x + \delta)]_1 - [Z(x + \delta)]_{y_1} \right) \right)]$ | ✓  | ✓  | ✓  |     |

$\lambda$, $\lambda_1$, and $\lambda_2$: hyperparameters control the trade-off between the losses. $\ell_{KL}$: KL divergence between the label distribution of the original samples and adversarial examples. 1: indicator function returns 1 if the expression inside the parenthesis is true. Dirichlet(\beta): Dirichlet distribution with parameter $\beta$. $Z$: a vector of logits output by the model $f_\theta(x)$. $\kappa$: hyperparameter controls the confidence of the adversarial example.
3 Accuracy aware adversarial training (A3T)

Given a data set \( D = \{(x_i, y_i)\} \), we consider a \( K \)-class classification setup. The classifier is a \( \theta \)-parameterized score function \( f_\theta(x) = (f_\theta^1(x), \ldots, f_\theta^K(x)) \), where \( f_\theta^i(x) \) corresponds to the score of the \( i \)-th class. The predicted label of \( x \) is denoted as \( \hat{y} = \arg \max_i f_\theta^i(x) \). The classic AT problem is formulated as a min-max optimization program (Madry et al., 2018):

\[
\min_\theta \sum_{(x, y) \in D} \max_{\delta \in \Delta} \ell(f_\theta(x + \delta), y),
\]

where \( \ell \) is the loss function used for training the classifier, and \( \Delta \) is typically a \( L_\infty \) norm ball with radius bounded by \( \epsilon \). We denote by \( \delta^* \) the optimal perturbation solving the inner maximization program, i.e.,

\[
\delta^* = \arg \max_{\delta \in \Delta} \ell(f_\theta(x + \delta), y).
\]

Classically, PGD or its variants (Madry et al., 2018) are used to solve the program above. In case of correctly classified training samples, the optimal perturbation would result in adversarial examples that are closer to the decision boundary than the original samples. However, in case of misclassified samples, the same optimal perturbation can result in adversarial examples that are further away from the decision boundary than the original samples. For high-capacity models, this can result in a highly non-smooth boundary with poor generalization as shown in Fig. 1.

A3T loss: to prevent overfitting, we propose A3T, an accuracy-aware AT loss, which aims at generating close to decision boundary adversarial examples. To achieve this, A3T generates adversarial examples differently for classified and misclassified samples, i.e., it entails different inner maximization objectives depending on the samples prediction accuracy:

\[
\min_\theta \left[ \sum_{(x, y) \in D^+_\theta} \ell\left(f_\theta\left(x + \arg \max_{\delta \in \Delta} \ell\left(f_\theta(x + \delta), y\right)\right), y\right) + \sum_{(x, y) \in D^-_\theta} \ell\left(f_\theta\left(x + \arg \max_{\delta \in \Delta} \ell\left(f_\theta(x + \delta), \hat{y}\right)\right), y\right) \right].
\]

Here, \( D^+_\theta \) and \( D^-_\theta \) are the set of correctly and misclassified samples at a given training iteration, respectively. Note that the loss terms of the two inner maximizations have different arguments, i.e., \( y \) vs \( \hat{y} \). In particular, for the misclassified samples, the loss uses the predicted label \( \hat{y} \) as its second argument. The optimal perturbation \( \delta^* \) can be computed using PGD as follows:

\[
\delta^* = \begin{cases} 
\Pi_\Delta \left( \delta + \alpha \nabla_x \ell\left(f_\theta(x + \delta), y\right) \right) & \text{if } (x, y) \in D^+_\theta \\
\Pi_\Delta \left( \delta + \alpha \nabla_x \ell\left(f_\theta(x + \delta), \hat{y}\right) \right) & \text{otherwise}
\end{cases}
\]

Since \( \hat{y} = y \) holds for samples in \( D^+_\theta \), Eq. (6) above can be simplified to

\[
\delta^* = \Pi_\Delta \left( \delta + \alpha \nabla_x \ell\left(f_\theta(x + \delta), \hat{y}\right) \right).
\]

Combining Eqs. (5) and (7) results in the following alternative training objective to Eq. (5):

\[
\min_\theta \sum_{(x, y) \in D} \max_{\delta \in \Delta} \ell(f_\theta(x + \delta), y).
\]
Theorem 1 Let $f_\theta(x) = \theta^T x + b$ be a linear model trained with a logistic loss $\ell$. Assume that $(x_i, y_i)$ is a misclassified training sample and that $\delta_1 = \arg\max_{\delta \in \Delta} \ell(f_\theta(x_i), y_i)$ and $\delta_2 = \arg\max_{\delta \in \Delta} \ell(f_\theta(x_i), (1 - 2y_i))$ are the solutions to the inner maximization program using standard AT [Eq. (4)] and A3T [Eq. (5)], respectively. We prove that

$$\left| \theta^T (x_i + \delta_1) + b \right| \leq \left| \theta^T (x_i + \delta_2) + b \right|.$$  

Proof See Appendix. 

Intuitively, Theorem 1 states that adversarial examples from misclassified samples generated using A3T are closer to the decision boundary than the ones generated using AT. To provide better intuition, we will next compare our A3T approach to standard AT on two toy examples.

Example 1 Consider a 2-D binary linear classification setup, where the decision boundary is given by $x_1 + x_2 = 1$ and labels are $y \in \{-1, 1\}$. The linear classifier weights are at convergence therefore $(\theta_1, \theta_2) = (1, 1)$. The optimal adversarial perturbation is $\delta = -y \text{sgn}(\theta)$. Suppose a data point $(x_1, x_2)$ with label +1 is misclassified as -1. Then the perturbation based on the original label is $(x_1 - \epsilon, x_2 - \epsilon)$ while the one based on our approach is $(x_1 + \epsilon, x_2 + \epsilon)$. Hence, close to convergence, our approach pushes the adversarial sample corresponding to $(x_1, x_2)$ towards the boundary while the classical approaches, counter-intuitively, place the adversarial examples maximally away from the boundary.
Example 2 Consider the linear classification setup presented in Fig. 2. Note, A3T decision boundary (green) is closer to the standard boundary (black) than the AT one (orange). This is due to the adversarial examples generated by A3T ($A_{A3T}$, $B_{A3T}$) from misclassified samples $A$ and $B$ being placed closer to the boundary that the one generated by AT ($A_{AT}$, $B_{AT}$). This also implies that A3T performance on clean test samples is closer to the original classifier.

Our A3T approach is summarized in Algorithm 1. We also visualize one training iteration in Fig. 3.

Algorithm 1 Accuracy Aware Adversarial Training (A3T)

**Inputs**: $E$: the number of epochs, $D = \{(x_i, y_i)\}_{i=1}^n$: the training data with $\|D\| = n$; $f_\theta (x)$: the supervised model parametrized by $\theta$; $\delta \in \Delta$: the perturbation with $\|\Delta\|_{L_\infty} \leq \epsilon$; $\tau$: the global learning rate; $\alpha$: the adversarial learning rate; $S$: the number of PGD step; $\Pi$: the projection function.

**Output**: $\theta$

1. for $e = 1, \ldots, E$ do
   2. for $(x_i, y_i) \in D$ do
     3. $\hat{y}_i \leftarrow \arg \max_{y \in \{1, \ldots, K\}} f_\theta (x_i)$ \quad \triangleright \text{get prediction}
     4. $\delta_i \sim \mathcal{N}(0, \sigma^2 I)$ \quad \triangleright \text{sample an initial perturbation}
     5. for $s = 1, \ldots, S$ do
       6. $\delta_i \leftarrow \Pi_\Delta \left( \delta_i + \alpha \cdot \text{sgn} \left( \nabla_{x_i} \ell(f_\theta(x_i + \delta_i), \hat{y}_i) \right) \right)$ \quad \triangleright \text{compute S steps of PGD}
     7. end for
     8. $\theta \leftarrow \theta - \tau \nabla_{\theta} \ell(f_\theta(x_i + \delta_i), y_i)$ \quad \triangleright \text{update the model parameters}
   9. end for
10. end for

Since A3T addresses an orthogonal root cause of overfitting than the one mitigated by LSR and MM techniques, we propose A3T+ (line 8 in Table 1 and Algorithm 2) which combines A3T, LSR, and MM objectives.

3.1 Improvement over prior work

So far, two misclassification aware AT approaches have been proposed (Wang et al., 2020; Ding et al., 2020). Wang et al. (2020) were the first to explicitly recognize that misclassified
and correctly classified samples should be treated differently and proposed a misclassification aware AT method (MART). Their training objective consists of two separate loss terms (line 7 in Table 1). The first term corresponds to the adversarial loss used in conventional AT, and does not distinguish between correctly and misclassified samples. The second term is a KL-divergence loss weighted by the confidence in classification measured by \((1 - f_o(x))\) via proposed by Miyato et al. (2019). Intuitively, this leads to reducing the impact of correctly classified samples and encouraging label smoothness around misclassified samples. In the context of Fig. 1, MART populates the space around the misclassified samples by new samples of both the true class and the incorrectly predicted one. The true class samples will be located in a direction that is maximally away from the decision boundary and will serve as ‘hard’ examples for the model. In contrast, the regularization based label smoothing will add samples of the incorrect class isotropically. In effect, the latter group of samples neutralizes the impact of the ‘hard’ examples created by the conventional AT method and prevents the model from exhibiting even worse overfitting behavior.

Ding et al. (2020) use margin-maximization as an implicit treatment of misclassified samples. The perturbation margin \(\epsilon_i\) of each sample \((x_i, y_i)\) is individually determined through increasing perturbation strength gradually until an adversarial examples is generated. Since misclassified samples are adversarial in nature, they yield the lowest possible margin. Thus, adversarial examples created for misclassified samples are not significantly perturbed.

Our approach, A3T, fundamentally differs from the above methods in that the adversarial examples created include ‘easy’ samples that are placed towards the decision boundary, as opposed to creating no samples or samples that are further away from the boundary. These ‘easy’ samples essentially serve to correct for any potential deformation in that part of the decision boundary. In other words, we attribute the cause of a misclassified example to the presence of opposing class examples in the immediate vicinity of that example, preventing the model from successfully fitting them. By creating samples that yield lower loss, our method reduces the effect of misclassified samples on the resulting decision boundary.

4 Experiments

We evaluate A3T on 12 datasets from 4 domains and compare its performance them against 6 baselines from 4 families of AT methods: (1) LDS, (2) MM, (3) LSR, and (4) MA.

4.1 Results on synthetic data

First, we evaluate A3T on a synthetic dataset for binary classification introduced by Gan et al. (2020). The dataset consists of 1,016 samples from two classes in a two-dimensional space over real numbers. Samples in each class are created on two trajectories each following the shape of a crescent moon as shown in Fig. 4. Each data point is then projected onto a 100-dimensional vector space.

We train a deepnet with 100 neurons using 16 randomly selected training samples per class. With such a small training sample size the model is prone to overfitting, thereby allowing better evaluation of the generalization capability. The model is trained for 400 epochs within a learning rate \((\tau)\) equals 0.01 using Adam optimizer. For the first 100
epochs, we use standard training and for the remaining epochs we switched to AT. Adversarial examples are generated using a 5-step PGD attack [Eq. (3)]. A random perturbation is initialized with $\sim \mathcal{N}(0, 0.05)$ and updated with a learn rate $\alpha = 0.1$ by limiting $\epsilon = 0.4$.

Overall, 20 models are generated for each training approach using the same set of training samples and the fixed parameter values. Following the standard training phase (first 100 epochs) all models were found to correctly classify all but one (out of 32) training samples, generally exhibiting a good fitting capability. The decision boundaries are then drawn based on the positions of all data points classified with the least confidence, i.e., by averaging the locations of all training samples in all runs that were classified with confidence in the 0.49–0.51 probability range. Figure 4 shows the decision boundaries obtained after the first 100 epochs and at the end of the training, as gray and black lines for both AT and A3T, respectively. All adversarial examples created beyond epoch 100 are also displayed as orange and turquoise dots on the figure.

The difference between the standard AT and A3T can best be seen around the misclassified samples of the red class, at the upper end of the crescent. In the case of A3T, the adversarial examples, i.e., orange points around the misclassified samples, are generated between the misclassified example and the decision boundary. Whereas for AT, adversarial examples are created further away from the boundary. This further resulted in intermixing of red-class adversarial examples with the blue-class data points, which may force the model to learn a highly non-smooth decision boundary. In other respects, it can be seen that the final decision boundary (black) for AT (Fig. 4a) and A3T (Fig. 4b) are very similar and therefore the robustness behavior may be expected to be on par with each other.

To further examine the generalization behavior, we repeated the same experiment by randomly selecting the training points for each run and averaging results over 50 runs. Figure 5 shows all training points used as well as the corresponding decision boundaries. Note that training samples remain the same for both AT and A3T within a run, but they change across runs. Here, the generalization capability of A3T is more visible as it is able to correctly classify most of the blue-class points. In the case of red-class data points, both methods perform similarly. These observations are also reflected in the measured classification accuracies: A3T achieved 83.6% accuracy, i.e., 1.2% better than AT.
4.2 Results on real data

We now assess the robustness and generalization trade-off yielded by A3T on computer vision, natural language processing tasks as well as on tabular data. Models obtained by standard training and AT are evaluated under attack-free and attack setups. The models are attacked using adversarial examples produced by PGD (Madry et al., 2018) and AutoAttack (Croce et al., 2021).

4.2.1 Computer vision experiments

Most AT methods have been proposed to improve the robustness of image classifiers. The effectiveness of these methods has been evaluated on CIFAR-10 datasets using architectures such as WideResNet-34-10 and WideResNet-28-10 with varying attack and training parameters. In our evaluations, we consider a similar setting.

As a baseline, we use MART since it is the only other method that explicitly formulates a treatment for misclassified samples during AT. In their paper, MART uses boosted cross-entropy loss and report a significant improvement in accuracy (∼3%) compared to standard cross entropy loss.\(^1\) Since boosted loss can be generically incorporated to all AT approaches, we only use cross-entropy loss to ensure a comparison on an equal footing.

To be in line with the previous methods, we use WideResNet-34-10 and train all models for a total of 100 epochs, and SGD with momentum 0.9, weight decay $2 \times 10^{-4}$, an initial learning rate of 0.1. We decay the learning rate by 90% at the 75th, 90th epoch.

Table 2 provides accuracy results for standard and under-attack settings (FGSM, PGD). A3T outperforms MART by 4.7% on natural accuracy. This implies better generalization, i.e., less overfitting. On robust accuracy, A3T inconsistently outperforms MART on 3 out of 4 experiment. When compared to conventional AT, A3T improves both natural and robust accuracy by around 2%.

\(^1\) This is demonstrated in Fig. (2b) of Wang et al. (2020).
In Fig. 6, we visualize the loss landscape of these three approaches by following the method proposed by Engstrom et al. (2018). In particular, we are interested in investigating the effect of the A3T loss on the smoothness of the loss landscape. Loss landscapes are generated by computing the loss around each test data point by adding two perturbations defined by the sign of the input gradient and a random Rademacher matrix ($\mathcal{R}$). The $x$- and $y$- axes represent the perturbation magnitudes, $\lambda_r$ and $\lambda_g$, and the $z$-axis represents the loss values calculated as:

$$\ell(f_0(x + \gamma), y), \text{ where } \gamma = \lambda_r \mathcal{R} + \lambda_g \text{sgn}(\nabla_x \ell(f_0(x), y)).$$

Here $\gamma$ is the total perturbation added to the input. The figure shows that while all the AT losses are robust against random perturbation, as expected, the A3T loss is the least affected by perturbation in the gradient direction.

Next, we compare A3T and A3T\textsuperscript{+} with several AT methods. Here we want to highlight that the results reported by these methods include some inconsistencies, making a fair comparison challenging. This includes the learning-rate schedule (a larger number of training epochs are expected to report lower final accuracy values) and the PGD step size used during attacks (larger steps imply a strength attack). To circumvent these ambiguities, we decided to use the robustness results reported by the AutoAttack (AA) benchmark\footnote{https://robustbench.github.io/#leaderboard.} and evaluated A3T and A3T\textsuperscript{+} accordingly. We use the same hyperparameters from the previous

| Defense         | Natural Best | Natural Last | FGSM Best | FGSM Last | PGD\textsuperscript{20} Best | PGD\textsuperscript{20} Last |
|-----------------|--------------|--------------|-----------|-----------|-------------------------------|------------------------------|
| AT (Madry et al., 2018) | 87.30        | 56.10        | 56.10     | 52.68     | 49.31                         |                              |
| MART\textsubscript{CE} (Wang et al., 2020) | 84.24        | 66.86        | 65.28     | 56.27     | 52.22                         |                              |
| A3T             | **88.98**    | **68.04**    | **67.12** | 54.59     | **52.39**                     |                              |

The presented results for MART are obtained using the public implementation shared by its authors. A3T results in better generalization than MART at near-similar levels of robustness.

![Fig. 6](image-url) Loss landscapes of the AT, MART, and A3T on CIFAR-10 dataset. A3T loss generates the smoothest landscape indicating better robustness.
experiments for A3T and A3T+. Table 3 provides corresponding results for several AT methods.

A3T and A3T+ yield the highest natural accuracy after the Bilateral AT method (Wang & Zhang, 2019) which exhibited very limited robustness under AA attack. In terms of robust accuracy, CAT yields the best accuracy, performing only marginally better than A3T+ (+0.2%) and noticeably better than A3T (+3.7%). However, both A3T and A3T+ yield a higher natural accuracy compared to CAT (+2.8−1.6%). These results overall present a more granular view of the generalization vs. robustness trade-off: MM and LSR allow a shift in favor of robustness, while MA and LDS tilt the balance more towards improved generalization. When the average accuracy is evaluated, A3T+ offers a better trade-off between natural and AA accuracy as it allows an increase in the former while keeping the robust accuracy on par with other methods. This finding further strengthens the idea that MM, LSR, and MA methods are addressing different overfitting root causes and that they are indeed complementary.

### 4.2.2 Natural language processing experiments

For evaluation, we used the GLUE benchmark (Wang et al., 2019).³ GLUE contains seven datasets for two sentiment analysis tasks, two similarity tasks and three inference tasks.

---

³ Out of the eight defined tasks, we excluded the one related to the STS-B dataset as it involves a regression task.
Due to the discrete nature of text, generating adversarial examples in the input space is nontrivial as it involves projecting the continuous perturbation computed in the embedding space to the input space. Although several input-domain approaches are proposed to create adversarial examples, they are not efficient (Altinisik et al., 2022). An alternative approach is to perform AT in the latent space without the need for creating adversarial inputs (Gan et al., 2020). In our experiments, we also adopt this approach both during AT and when performing adversarial attacks. It must be noted that in the case of launching an adversarial attack this approach is impractical to implement, but it nevertheless corresponds to a worst-case attack setting and allows a better evaluation of the model’s robustness.

In our experiments, we fine-tuned the deBERTa-base model (He et al., 2021) from the HuggingFace library. For each dataset in the GLUE benchmark, we created a fine-tuned, task-specific model by first training the model for three epochs at the suggested learning rate of $2 \times 10^{-5}$ using the AdamW optimizer (He et al., 2021). Then, we freeze the first five layers and fine-tune for another three epochs with either standard or AT. In all cases, adversarial examples are generated using 3-step PGD assuming $\epsilon = 0.01$, the initial perturbation is sampled from a normal distribution $\sim \mathcal{N}(0, 0.005)$.

Table 4 reports classification performance yielded by various models under both normal and adversarial attack scenarios. As compared to the baseline accuracy of the model, i.e., attack-free test setting, A3T yields a small drop in the average performance (−2.5%) but performs noticeably better (+3.6%) than AT as displayed in the last line of the table. Similarly, under an adversarial attack test setting, A3T yields an improvement of 1.5% over conventional AT. The improvement due to A3T is more noticeable when the results of CoLA task is removed from the average as it uses a different metric than accuracy (third to the last line). In that case, the use of A3T additionally results in a performance improvement of 2.8% over AT.

4.2.3 Tabular tasks

We evaluate our model on tabular data. Specifically, we consider two classification tasks in finance: (1) predicting the overall yearly default rate for subjects on the Matlab’s Retail Credit Panel dataset (Matlab, 2022) and (2) a binary classification task on two datasets. Our first tabular evaluation involves Matlab’s Retail Credit Panel dataset.

To predict what percent of subjects defaulted in a given year, a logistic regression model with two hidden layers consisting of 256 and 128 neurons, is trained over 500 epochs, with the Adam optimizer using a learning rate of 0.001. AT is only applied between epochs 100 and 500 using the attack parameters obtained after a grid search.4 The obtained result is the average from the grid search is used for comparisons.

Since the input data is very low-dimensional, no adversarial attack is performed. Instead, we investigated how AT impacts the model’s performance in the attack-free setting. Figure 7 provides the average default rates predicted by different models after 50 runs in comparison to the ground truth rates. Results show that A3T predictions follow the realized default rates more closely than other approaches. To better evaluate the fit of each model, mean squared errors (MSE) of predictions with respect to actual rates are computed.

4 Adversarial attack parameter ranges for the grid search were as follows: initialization noise variance for PGD {0.05, 0.1}; PGD step size $\alpha \{0.2, 0.1, 0.05\}$, and projected perturbation limit $\Delta \{0.2,0.4\}$. The total number of PGD steps was fixed to five.
For the binary classification task, we evaluate A3T on two datasets, namely, the European card dataset (ECD) (Pozzolo et al., 2015) and the Adult dataset (Kohavi & Becker, 1996). The former dataset contains 492 fraud and 10,000 genuine transactions randomly downsampled from more than 280,000 transactions with 31 features. The goal is to identify the type of transaction. The latter involves 32,000 samples with 9 features, and the objective is to predict whether the income of a subject is higher than $50,000 or not.

All categorical values are represented by one-hot encodings. Logistic regression models on deepnet architecture with the same as above are trained. Rather than using a fixed AT setting, multiple robust models with the same grid search parameters used in Matlab’s Retail Credit Panel dataset are trained. Each model is then tested under 12 adversarial attack scenarios parameterized by the same possible parameter value configurations considered in the grid search, and the average of resulting accuracy values is the model’s under-attack prediction accuracy. The standard and under-attack accuracy achievable by

| Dataset     | Attack-free tests | Under attack tests |
|-------------|-------------------|--------------------|
|             | Standard | AT  | A3T | Standard | AT  | A3T |
| MRPC        | 88.7     | 69.8 | 88.9 | 12.0     | 65.6 | 63.5 |
| QNLI        | 92.1     | 92.6 | 92.3 | 40.1     | 50.5 | 55.8 |
| QQP         | 91.4     | 91.4 | 91.4 | 45.8     | 63.7 | 68.3 |
| RTE         | 73.3     | 71.9 | 77.2 | 1.2      | 31.2 | 29.7 |
| SST-2       | 93.0     | 93.9 | 94.0 | 23.1     | 50.4 | 50.3 |
| MNLI        | 86.1     | 86.8 | 86.5 | 4.7      | 33.2 | 43.7 |
| Average Acc | 87.4     | 84.4 | 88.4 | 21.2     | 49.1 | 51.9 |
| CoLA        | 61.7     | 37.0 | 37.7 | −91.4    | −23.0 | −29.0 |
| Average All | 83.7     | 77.6 | 81.2 | 5.1      | 38.8 | 40.3 |

Results show that for all tasks, A3T performs on par with or better than the conventional AT. The best performances are made bold for each dataset.
a model are finally determined by averaging the corresponding values over five runs (see Fig. 8).

In Fig. 8, the upper right corner corresponds to a performance regime where a model performs well both in attack-free and under-attack settings. Thus, the best performing model can be identified based on how close it gets to that corner. Notice that several robust versions of A3T exhibit high under-attack performance with only a slight drop in performance in the attack-free setting. This result also demonstrates the importance of the choice of AT parameters on model accuracy.

5 Discussion and conclusions

AT uses adversarial examples to train machine learning models in order to achieve better robustness. However, improving robustness via AT implies hauling-off generalization. In this work, we propose a new AT method that yields a more favorable robustness-generalization trade-off. Next, we summarize the main insights from our paper.

Generation of non-adversarial samples The underlying idea of our misclassification-aware adversarial training approach (A3T) is to prevent the creation of ‘hard’ examples from misclassified samples, thereby reducing the risk of overfitting. A3T effectively achieves this by generating samples that are non-adversarial in nature. That is, instead of applying a perturbation that maximizes the loss, A3T computes a perturbation that minimizes the loss, defeating the purpose of an adversarial sample. Hence, the newly generated samples by A3T are maximally close to the decision boundary as opposed to adversarial examples that are maximally away from the boundary. From this perspective, imposing a bound on the extent of loss reduction may not seem meaningful. However, since samples with arbitrarily low-loss values will constitute ‘easy’ examples, they will likely be less informative for the model.
Incorporation of LDS with A3T Since MM, LSR, LDS, and MA are proposed to address different aspects of AT, a question to be answered is if and to what extent these design choices interfere with each other. In fact, our A3T+ results show that combining MM, LSR, and MA losses help achieve a better generalization-robustness trade-off. However, incorporating LDS with A3T i.e., using A3T during inner maximization and LDS as an additional regularization, is ineffective. Crucially, LDS applies label smoothing by placing samples of predicted class, which for a misclassified sample includes opposing class samples, around the misclassified sample. In contrast, A3T generates examples of the same class as the misclassified sample closer to the decision boundary. Hence, jointly optimizing these two losses can potentially contribute to the model’s overfitting as samples of opposing classes will be placed in the same vicinity.

Influence on robust overfitting Robust overfitting is another artifact of AT wherein a model test accuracy remains similar but the robust accuracy exhibits a drop as the number of training epochs increases (Dong et al., 2022; Rice et al., 2020). Our observations show that A3T also suffers from robust overfitting. This is in agreement with the findings of MART, where a larger variation between best and last accuracies is observed compared to A3T. Hence, we can deduce that the root cause for this phenomenon does not mainly relate to the adversarial examples from misclassified samples.

Appendix

Theorem Let \( f_\theta(x) = \theta x + b \) be a linear model trained with a logistic loss \( \ell \). Assume that \( (x_i, y_i) \) is a misclassified training sample and that \( \delta_1 = \arg \max_{\delta \in \Delta} \ell(f_\theta(x_i), y_i) \) and \( \delta_2 = \arg \max_{\delta \in \Delta} \ell(f_\theta(x_i), (1 - 2y_i)) \) are the solutions to the inner maximization using standard AT [Eq. (4)] and A3T [Eq. (5)], respectively. We prove that

\[
|\theta(x_i + \delta_2) + b| \leq |\theta(x_i + \delta_1) + b|.
\]

Proof First note that if \( (x_i, y_i) \) is a training sample with \( y_i \in \{+1, -1\} \) then the misclassified label is \( 1 - 2y_i \). Also, note that the distance between a data point \( x_0 \) and the decision boundary \( \theta x + b \) is given by \( \frac{|\theta x_0 + b|}{||\theta||} \).

Based on the assumption that \( x_i \) is misclassified then

\[
L(y_i(\theta x_i + b)) \geq L((1 - 2y_i)(\theta x_i + b))
\]

Since \( L \) is logistic loss it is a monotonically decreasing function, therefore

\[
(1 - 2y_i)(\theta x_i + b) \leq y_i(\theta x_i + b)
\]

Now,
\[ |\theta(x_i + \delta_j) + b| \]
\[ = |\theta(x_i - \epsilon(1 - 2y_i)\text{sgn}(\theta)) + b| \]
\[ = |\theta x_i + b - \epsilon(1 - 2y_i)||\theta||_1| \]
\[ = |1 - 2y_i||\theta x_i + b - \epsilon(1 - 2y_i)||\theta||_1| \]
\[ = |(1 - 2y_i)(\theta x_i + b) - \epsilon||\theta||_1| \]
\[ \leq |(y_i(\theta x_i + b) - \epsilon y_i^2||\theta||_1| \]
\[ = |\theta(x_i + \epsilon y_i\text{sgn}(\theta)) + b| \]
\[ = |\theta(x_i + \delta_j) + b| \]

\[ \therefore \text{sgn}(\theta) = ||\theta||_1 \]
\[ \therefore |1 - 2y_i| = 1 \]
\[ \therefore |ab| = |a||b| \wedge (1 - 2y_i)^2 = 1 \]
\[ \therefore \text{assumption} \wedge y_i^2 = 1 \]
\[ \therefore |ab| = |a||b| \wedge |y_i| = 1 \]

\[ \therefore \text{ ALS} \]
\[ \therefore \text{ MM} \]

\[ \delta_i \sim \mathcal{N}(0, \sigma^2I) \]

\[ \delta_i \overset{\Pi_{e_i}}{\leftarrow} \left(\delta_i + \alpha \cdot \text{sgn} \left(\nabla z_i \ell(f_\theta(x_i + \delta_i), \tilde{y}_i)\right)\right) \]

\[ \therefore \text{ MM} \]

\[ \therefore \text{ ALS} \]

\[ \therefore \text{ ALS} \]

\[ \therefore \text{ MM} \]

\[ \therefore \text{ MM} \]

\[ \therefore \text{ ALS} \]

\[ \therefore \text{ ALS} \]
Author Contributions  The authors confirm their contribution to the paper as follows: Enes Altinisik performed the experiments, and all authors contributed to developing the idea, discussed the results, and wrote the manuscript.

Funding  Open Access funding provided by the Qatar National Library. This work is partial supported by the Qatar National Research Fund (QNRF) Grant NPRP11C-1229-170007.

Availability of data and materials  All the datasets used in the experiments are publicly available.

Code availability  The source code is publicly available at: https://github.com/EnesAltinisik/A3T.

Declarations

Conflicts of interest  Enes Altinisik has received research support from QNRF. All authors are employees of Qatar Computing Research Institute (QCRI/HBKU).

Ethical approval  Not applicable.

Consent to participate  Not applicable.

Consent for publication  Not applicable.

Open Access  This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Altinisik, E., Sajjad, H., Sencar, H. T., Messaoud, S., & Chawla, S. (2022). Impact of adversarial training on robustness and generalizability of language models. arXiv:2211.05523.
Balaji, Y., Goldstein, T., & Hoffman, J. (2019). Instance adaptive adversarial training: Improved accuracy trade-offs in neural nets. arXiv:1910.08051.
Chen, T., Zhang, Z., Liu, S., Chang, S., & Wang, Z. (2021). Robust overfitting may be mitigated by properly learned smoothening. In ICLR.
Cheng, M., Lei, Q., Chen, P., Dhillon, I. S., & Hsieh, C. (2022). CAT: customized adversarial training for improved robustness. In IJCAI (pp. 673–679).
Croce, F., Andriushchenko, M., Sehwag, V., & Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., & Hein, M. (2021). Robustbench: A standardized adversarial robustness benchmark. In NeurIPS.
Ding, G. W., Sharma, Y., Liu, K. Y. C., & Huang, R. (2020). MMA training: Direct input space margin maximization through adversarial training. In ICLR.
Dong, Y., Xū, K., Yang, X., Pang, T., & Deng, Z., Su, H., & Zhu, J. (2022). Exploring memorization in adversarial training. In ICLR.
Engstrom, L., Ilyas, A., & Athalye, A. (2018). Evaluating and understanding the robustness of adversarial logit pairing. arXiv:1807.10272.
Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M., Novikov, A., Ruiz, R., & Schmittwieser, J., Swirszcz, G., & Silver, D. (2022). Discovering faster matrix multiplication algorithms with reinforcement learning. Nature, 610, 47–53.
Gan, Z., Chen, Y., Li, L., Zhu, C., Cheng, Y., & Liu, J. (2020). Large-scale adversarial training for vision-and-language representation learning. In NeurIPS.
Gan, Z., Chen, Y., Li, L., Zhu, C., Cheng, Y., & Liu, J. (2020). Large-scale adversarial training for vision-and-language representation learning. In NeurIPS.
Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In ICLR.
Gowal, S., Qin, C., Uesato, J., Mann, T. A., & Kohli, P. (2020). Uncovering the limits of adversarial training against norm-bounded adversarial examples. arXiv:2010.03593.

He, P., Liu, X., Gao, J., & Chen, W. (2021). Deberta: Decoding-enhanced bert with disentangled attention. In ICLR.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV (pp. 1026–1034).

Huang, L., Zhang, C., & Zhang, H. (2020). Self-adaptive training: Beyond empirical risk minimization. In NeurIPS.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., & Bridgland, A. (2021). Highly accurate protein structure prediction with alpafold. Nature, 596, 583–589.

Kohavi, R., & Becker, B. (1996). Adult Dataset. In UCI Machine Learning Repository.

Kurakin, A., Goodfellow, I. J., & Bengio, S. (2017). Adversarial examples in the physical world. In ICLR.

Kurakin, A., Goodfellow, I. J., & Bengio, S. (2017). Adversarial machine learning at scale. In ICLR.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In ICLR.

Matlab: Compare Deep Learning Networks for Credit Default Prediction—MATLAB & Simulink. Retrieved September 27, 2022, from https://www.mathworks.com/help/deeplearning/ug/compare-deep-learning-networks-for-credit-default-prediction.html.

Miyato, T., Maeda, S.-I., Koyama, M., Nakae, K., & Ishii, S. (2015). Distributional smoothing with virtual adversarial training. arXiv:1507.00677.

Miyato, T., Maeda, S., Koyama, M., & Ishii, S. (2019). Virtual adversarial training: A regularization method for supervised and semi-supervised learning. TPAMI, 41(8), 1979–1993.

Pang, T., Yang, X., Dong, Y., Su, H., & Zhu, J. (2021). Bag of tricks for adversarial training. In ICLR.

Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning. In AsiaCCS (pp. 506–519).

Pozzolo, A. D., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In SSII (pp. 159–166).

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-shot text-to-image generation. ICMU, 139, 8821–8831.

Rice, L., Wong, E., & Kolter, J. Z. (2020). Overfitting in adversarially robust deep learning. ICML, 119, 8093–8104.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., & Madry, A. (2018). Adversarially robust generalization requires more data. In NeurIPS (pp. 5019–5031).

Schônerr, L., Kohls, K., Zeiler, S., Holz, T., & Kolossa, D. (2019). Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. In NDSS.

Stutz, D. (2022). Understanding and improving robustness and uncertainty estimation in deep learning. In Saaarlnd University.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., & Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In ICLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., & Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In NeurIPS (pp. 5998–6008).

Wang, J., & Zhang, H. (2019). Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In ICCV (pp. 6628–6637).

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In ICLR.

Wang, Y., Ou, D., Yi, J., & Bailey, J., Ma, X., & Gu, Q. (2020). Improving adversarial robustness requires revisiting misclassified examples. In ICLR.

Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., & Gu, Q. (2019). On the convergence and robustness of adversarial training. ICML, 97, 6586–6595.

Wu, D., Xia, S., & Wang, Y. (2020). Adversarial weight perturbation helps robust generalization. In NeurIPS.

Yu, C., Han, B., Shen, L., Yu, J., Gong, C., & Liu, T. (2022). Understanding robust overfitting of adversarial training and beyond. ICML, 162, 25595–25610.

Zhang, S., Gao, H., Zhang, T., Zhou, Y., & Wu, Z. (2022). Alleviating robust overfitting of adversarial training with consistency regularization. arXiv:2205.11744.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., El Ghaoui, L., & Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. ICMU, 97, 7472–7482.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.