Probabilistic Margins for Instance Reweighting in Adversarial Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

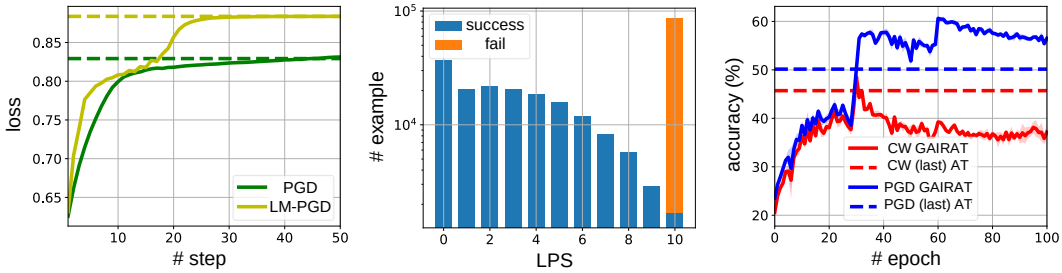
Reweighting adversarial data during training has been recently shown to improve adversarial robustness, where data *closer* to the current decision boundaries are regarded as *more critical* and given *larger weights*. However, existing methods measuring the closeness are not very reliable: they are *discrete* and can take only a few values, and they are *path-dependent*, i.e., they may change given the same start and end points with different attack paths. In this paper, we propose three types of *probabilistic margin* (PM), which are *continuous* and *path-independent*, for measuring the aforementioned closeness and reweighting adversarial data. Specifically, a PM is defined as the difference between two *estimated class-posterior probabilities*, e.g., such the probability of the true label minus the probability of the most confusing label given some natural data. Though different PMs capture different *geometric properties*, all three PMs share a negative correlation with the vulnerability of data: data with larger/smaller PMs are safer/riskier and should have smaller/larger weights. Experiments demonstrate that PMs are reliable measurements and PM-based reweighting methods outperform state-of-the-art methods.

1 Introduction

Deep neural networks are susceptible to adversarial examples that are generated by changing natural inputs with malicious perturbation [14, 17, 32, 34]. Those examples are imperceptible to human eyes but can fool deep models to make wrong predictions with high confidence [3, 30]. As deep learning has been deployed in many real-life scenarios and even safety-critical systems [9, 13, 25, 37], it is crucial to make such deep models reliable and safe [16, 22, 43]. To obtain more reliable deep models, *adversarial training* (AT) [2, 26] was proposed as one of the most effective methodologies against adversary attacks (i.e., maliciously changing natural inputs). Specifically, during AT, they simulate adversarial examples (e.g., generated by *project gradient descent* (PGD) [1, 26, 41, 36]) and train a classifier with the simulated adversarial examples [26, 41]. Since such a model has seen some adversarial examples during its training process, it can defend against certain adversarial attacks and is more *adversarial-robust* than a classifier only trained with natural data [5, 12, 31].

Recently, researcher have found that over-parameterized deep networks still have the insufficient model capacity, since adversarial training has an overwhelming smoothing effect [42, 29]. As a result, they proposed *instance-reweighted adversarial training*, where adversarial data should have unequal importance given limited model capacity [42]. Concretely, they suggested that data *close* to decision boundaries are vulnerable to be attacked [40, 42] and should be assigned large weights during training. To characterize these *geometric properties* of data (i.e., the closeness between data and decision boundaries), Zhang et al. [42] suggested an estimation in input space, i.e., the *least PGD steps* (LPS), to identify non-robust (easily-be-attacked) data. LPS measures how many steps the PGD method needs to attack natural data successfully. Specifically, LPS is the number of steps to

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(a) Loss Curve (b) Histogram (c) Accuracy Curve

Figure 1: An illustration of the drawbacks of LPS. (a) suggests that LPS might be path-dependent since the vanilla PGD method (PGD) may get stuck at sub-optimal points. Here, we prove the existence of a better (might not be optimal) solution given by *line-search & momentum-PGD* (LM-PGD) (cf. Appendix A). LPSs are 50 and 14 estimated by PGD and LM-PGD, revealing that LPS depends on the adopted attacks. (b) suggests that LPS can take only a few discrete values, and it has a confusing meaning when LPS meets the maximal value. In this case, one can hardly distinguish the non-robust data with LPS being 10 and the safe data that are insensitive to be attacked. (c) shows that the accuracy of GAIRAT (on *CIFAR-10*) will drop significantly when facing CW attacks. The main reason is that LPS is not reliable to measure the distance between data and decision boundaries ((a), (b)). As a result, it makes wrong judgments when assigning weights for non-robust instances.

make the adversarial variant of such an instance cross the decision boundaries starting from a natural instance. Based on LPS, they achieved state-of-the-art performance via a general framework termed *geometry-awared instance-reweighted adversarial training* (GAIRAT).

However, existing methods [15, 19, 33] in measuring geometric properties of data are *path-dependent*, i.e., they may change even given the same start (i.e., a natural example) and end point (i.e., the adversarial variant), and (or) they are *discrete* with only a few valid values (cf. Figure 2). The path-dependency makes the computation unstable, where the results may change due to the different attack paths. The discreteness makes the measurement ambiguous since each value would have several (even contradictory) meanings. We take LPS, a representative measurement [42], as an example to demonstrate a path-dependent (Figure 1a) and discrete measurement (Figure 1b). As we can see in Figure 1c, the consequence is non-negligible: although adopting LPS (i.e., GAIRAT) reveals improvement for the PGD-20 attack [26], its performance is well below traditional AT regarding a stronger attack method, CW [7].

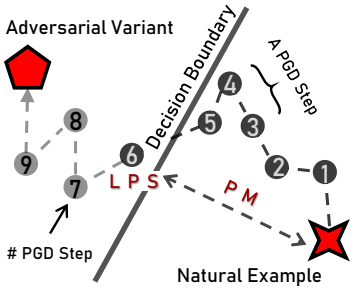


Figure 2: The path-dependency and discreteness of LPS, and the proposed PM in comparison. The PGD methods uses 6 step to find an instance to violate the boundary (i.e., LPS is 6), while PM is continuous to represent the distance.

In this paper, we propose the *probabilistic margin* (PM), which is *continuous* and *path-independent*, for reweighting adversarial data during AT. PM is a geometric measurement from a data point to the closest decision boundary, following the *multi-class margin* [20] in traditional geometry-aware machine learning (e.g., the support vector machine [11]). Note that, instead of choosing the input space as LPS, PM is defined based on the *estimated class-posterior probabilities* from model outputs, e.g., such the probability of the true label minus the probability of the most confusing label given some natural data. Therefore, PM is computed in a low-dimensional embedding space with normalization, alleviating troubles in comparing data from different classes.

The definition of PM is general, where we consider three specifications, namely, $\mathtt{PM}^{\text{nat}}$, $\mathtt{PM}^{\text{adv}}$, and $\mathtt{PM}^{\text{dif}}$. Therein, $\mathtt{PM}^{\text{nat}}$ and $\mathtt{PM}^{\text{adv}}$ are the PM scores regarding natural and adversarial data, respectively. They assumed that the vulnerability of data is revealed by the closeness regarding either natural data or their adversarial variants. The definition of $\mathtt{PM}^{\text{dif}}$ is slightly different, which is the distance of a

natural data point to its adversarial variant. $\texttt{PM}^{\text{dif}}$ is viewed as a conceptual counterpart of LPS, but is critically different that it is continuous and path-independent. In our paper, we have verified the effectiveness of $\texttt{PM}^{\text{nat}}$ and $\texttt{PM}^{\text{adv}}$ and show that they can represent the geometric properties of data well. Note that, though these types of PM depict different geometric properties, they all share a negative correlation with the vulnerability of data—larger/smaller PMs indicate that the corresponding data are safer/riskier and thus should be assigned with smaller/larger weights during training.

Eventually, PM is employed for reweighting adversarial data during AT, where we propose the *margin-aware instance reweighting learning* (MAIL). With a non-decreased weight assignment function (cf. Eq. (8)), it pays much attention to those non-robust data. In experiments, MAIL is combined with various forms of commonly-used AT methods, including traditional AT [26], MART [35], and TRADES [40]. We demonstrate that PM is more reliable than previous works in geometric measurement, irrelevant to the forms of the adopted objectives. Moreover, in comparison with advanced methods, MAIL reveals its state-of-the-art performance against various attack methods, which benefit from our path-independent and continuous measurement.

2 Preliminary

2.1 Traditional Adversarial Training

For a K-classification problem, we consider a training dataset $S = \{(x_i, y_i)\}_{i=1}^n$ independently drawn from a distribution $\mathcal{D}$ and a deep neural network $h(x;\theta)$ parameterized by θ. This deep classifier $h(x;\theta)$ predicts the label of an input data via $h(x;\theta) = \arg\max_k \mathbf{p}_k(x;\theta)$, with $\mathbf{p}_k(x,\theta)$ being the predicted probability (softmax on logits) of the k-th class.

The goal of AT is to train a model with a low adversarial risk regarding the distribution $\mathcal{D}$, i.e., $\mathcal{R}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\Delta} \ell(x+\delta, y;\theta)\right]$, where Δ is the threat model, defined by an L_p-norm bounded perturbation with the radius ϵ, *i.e.,* $\Delta = \{\delta \in \mathbb{R}^d \mid ||\delta||_p \leq \epsilon\}$. In the lens of statistical learning, we actually focuses on the empirical counterpart of $\mathcal{R}(\theta)$. In general, AT computes a new perturbation to update the model parameters, where the PGD method [26] is commonly adopted: for a (natural) example x_i, it starts with random noise ξ and repeatedly computes

$$\delta_i^{(t)} \leftarrow \texttt{Proj}\left[\delta_i^{(t-1)} + \alpha \texttt{sign}\left(\nabla_\theta \ell(x_i + \delta_i^{(t-1)}, y_i;\theta)\right)\right], \tag{1}$$

with $\texttt{Proj}$ the clipping operation such that $\delta^{(t)}$ is always in Δ and $\texttt{sign}$ the signum function. Due to non-convexity, we approximate the optimal solution by $\delta_i^{(T)}$ with T being the maximally allowed iterations. Accordingly, $\delta_i^{(T)}$ is viewed as the perturbation for the *most* adversarial example [42], and the learning objective function is formulated by

$$\arg\min_\theta \sum_i \ell(x_i + \delta_i^{(T)}, y_i;\theta). \tag{2}$$

Intuitively, AT corresponds to the *worst-case robust optimization*, continuously augmenting the training dataset with adversarial variants that highly confuse the current model. Therefore, it is a practical learning framework to alleviate the impact of adversarial attacks. However, it leads to insufficient network capacity, resulting in unsatisfactory model performance regarding adversarial robustness. The reason is that AT has an overwhelming smoothing effect in fitting highly adversarial examples [40], and thus consumes a large model capacity to learn from individual data points.

2.2 Geometry-Aware Adversarial Training

Zhang et al. [42] claimed that training examples should have unequal significance in AT, and proposed the *geometry-aware instance-reweighted adversarial training* (GAIRAT). It is a general framework to reweight adversarial data during training, where the Eq. (2) is modified as

$$\arg\min_\theta \sum_i \omega_i \ell(x_i + \delta_i^{(T)}, y_i;\theta) \quad \text{s.t. } \omega_i \geq 0 \text{ and } \sum_i \omega_i = 1. \tag{3}$$

The generation of the perturbation still follows Eq. (1). They suggested that data near the decision boundary are much vulnerable and should be assigned with large weights.

LPS as a geometric measurement. In assigning weights, GAIRAT requires a proper measurement for the distance. They suggested the estimation in high-dimensional input space via LPS (cf. Fig. 2), which is the least PGD iterations for a perturbation that leads to a wrong prediction. Intuitively, a small LPS indicates that the data point can quickly cross the decision boundary and thus close to it.

The drawbacks of LPS. Although promising results have been verified, LPS is path-dependent and limited by a few discrete values, where the consequence is non-negligible. In Figure 1(a), we showed that the PGD method will get stuck, resulting in that, using LPS as a geometric measurement, we might identify non-robust examples as robust ones during AT. Here, we modified the vanilla PGD method with the *line-searched* learning rate [38] and *Nesterov momentum* [27] (cf. Appendix A), termed *line-search & momentum-PGD* (LM-PGD), and we compared the loss curve of PGD with that of LM-PGD on one example. The maximal iterations of PGD was 50, and it is clear that, at the 14-th step, PGD almost converges, and the loss value does not ascend anymore. As a result, LPS of this example is 50, and this example is the most robust one in the view of LPS. However, LM-PGD still ascends after the 14-th step and successfully attacks such an instance at the 15-th step. This result means that such an instance is *not* a robust one, but LPS makes an wrong judgment regarding its robustness. The main reason is that LPS is heavily dependent on attack paths, even though both paths are highly similar.

We then demonstrate that the limited range of LPS would cause problems as well. In Figure 1b, we drew the histogram of LPSs for data in CIFAR-10 [21]. Higher LPS values mean that these examples are more robust and should be assigned smaller weights during AT. It can be seen that LPS has a confusing meaning when it equals the maximal value, which is 10 following [26]. For data whose LPSs are 10, they would be the most robust/safe ones. However, it can be seen that they still contain the critical data (the blue part). Although the proportion of the critical data seems low, ignoring them during AT (i.e., assigning small weights for them) will cause a significant problem: the accuracy of the trained classifier will drop significantly when facing the CW attack [7] (Figure 1c).

3 Probabilistic Margins for Instance Reweighting

The shortcomings of LPS motivate us to improve the measurement in discerning robust data and risky data, and we introduce our proposal in this section.

3.1 Geometry Information in view of Probabilistic Margin

Instead of using the input space as LPS, we suggest the measurement on estimated class-posterior probabilities, which are normalized embedding features (softmax on logits) in the range $[0, +1]$ for each dimension. Note that, without normalization, average distances from different classes might be of diverse scales (e.g., the average distance is 10 for the i-th class and 100 for the j-th class), increasing the challenge in comparing data from different classes.

Inspired by the *multi-class margin* in margin theory [20], we propose the *probabilistic margin* (PM) regarding the model outputs, namely,

$$\texttt{PM}(x, y; \theta) = \mathbf{p}_y(x; \theta) - \max_{j, j \neq y} \mathbf{p}_j(x; \theta), \quad (4)$$

where the first term in the r.h.s. is the *closeness* of x to the "center" of the true label y and the second term is the closeness to the nearest class except y (i.e., the most confusing label). The difference between the two terms is clearly a valid measurement, where the magnitude reflects the distance from the nearest boundary, and the signum indicates which side the data point belongs to. Figure 3 summarizes the key concepts, with i for the true label y and j for the most confusing class. For example, when $\mathbf{p}_y(x; \theta) = \mathbf{p}_j(x; \theta) = 0.5$, PM is 0 and the data point x

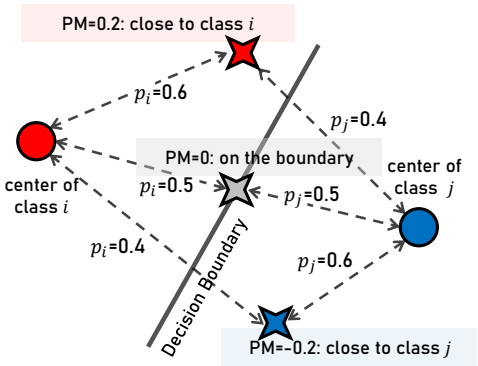


Figure 3: An illustration of PM, where p_i is the probability for the i-th class. As a special case, here we have $\texttt{PM} = p_i - p_j$ with i the target y and j the nearest class except y.

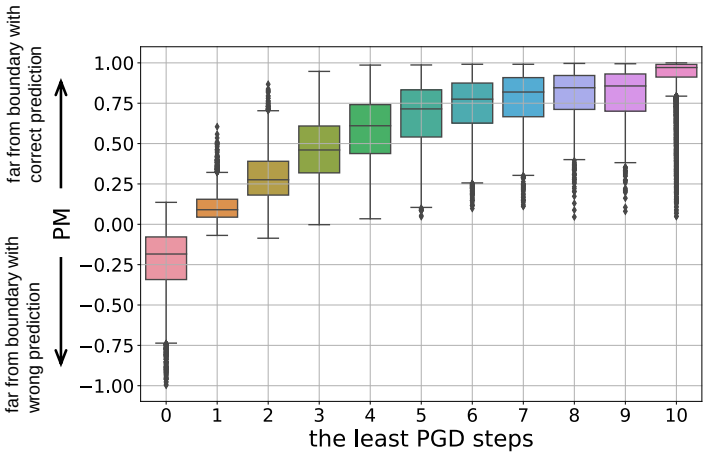


Figure 4: Comparison of LPS and PM ($\texttt{PM}^{\text{nat}}$). An interquartile range box represents the middle 50% of the considered data with the inside line being the median; a whisker is the range for the bottom/top 5% data, and points above/below whiskers are outliers.

is on the decision boundary between two classes; when $\mathbf{p}_y(x;\theta) = 0.6$ and $\mathbf{p}_j(x;\theta) = 0.4$, PM is positive and the data point x is much closer to the true label; if $\mathbf{p}_y(x;\theta) = 0.4$ and $\mathbf{p}_j(x;\theta) = 0.6$, PM is negative and the data point x is much closer to the most confusing class.

The above discussion indicates that PM can point out which geometric area a data point belongs to, where we discuss the following three scenarios: the *safe area* with large positive PMs, the *class-boundary-around data* with positive PMs close to 0, and the *wrong-prediction area* with negative PMs. The safe area contains guarded data that are insensitive to perturbation, which are safe and require low attention in AT (i.e., small weights); for data in the class-boundary-around data, they are much vulnerable to be attacked [40, 42] and thus need larger weights than the safe data; moreover, for data in the wrong-prediction area, they are the most critical since the attack method can successfully fool the current model, and thus they should be assigned the largest weights. This indicates a negative correlation of PM with vulnerability, where a larger PM requires a smaller weight.

In realization, the measurement of PM can be employed for adversarial data or natural ones, which are of the forms:

$$\texttt{PM}_i^{\text{adv}} = \mathbf{p}_{y_i}(x_i + \delta_i^{(T)};\theta) - \max_{j, j\neq y_i} \mathbf{p}_j(x_i + \delta_i^{(T)};\theta), \tag{5}$$

and

$$\texttt{PM}_i^{\text{nat}} = \mathbf{p}_{y_i}(x_i;\theta) - \max_{j, j\neq y_i} \mathbf{p}_j(x_i;\theta), \tag{6}$$

for a data point x_i, respectively. They assume that the vulnerability of data are revealed by the closeness regarding either natural data or their adversarial variants. Besides this two basic cases, one can also consider the difference between the natural and adversarial predictions, namely,

$$\texttt{PM}_i^{\text{dif}} = \mathbf{p}_{y_i}(x_i;\theta) - \mathbf{p}_{y_i}(x_i + \delta_i^{(t_i)};\theta), \tag{7}$$

where $t_i \leq T$ denotes LPS of x_i. $\texttt{PM}^{\text{dif}}$ is a conceptual counterpart of LPS, while Eq. (7) is actually path-independent and continuous, which is more reliable than LPS. Note that, the valid range of $\texttt{PM}^{\text{dif}}$ (i.e., $[0, +1]$) is different from that of $\texttt{PM}^{\text{adv}}$ and $\texttt{PM}^{\text{nat}}$ (i.e., $[-1, +1]$). It would bring unnecessary troubles since the geometric meanings of $\texttt{PM}^{\text{nat}}$ and $\texttt{PM}^{\text{dif}}$ are highly similar. Therefore, in experiments, we mainly verify the effectiveness of $\texttt{PM}^{\text{adv}}$ and $\texttt{PM}^{\text{nat}}$.

Without direct involvement of the PGD method, PM is continuous and path-independent, making it more reliable than previous works, e.g., LPS. Now, we can further verify the drawbacks of LPS in view of the PM. In Figure 4, we depicted the box plot regarding PM for training data with various LPSs. The instability of LPS is evident: from the box centers, there is a little differentiation for data with large LPSs (e.g., LPS = 7 or 9) regarding PM; from the whiskers and outliers, the spreads of PMs are relatively scattered and the numbers of outliers are significant, confirming that the geometric messages characterized by LPS may not be very stable.

Algorithm 1 MAIL: The Overall Algorithm.

Input: a network model with the parameters θ; and a training dataset S of size n.
Output: a robust model with parameters θ^*.

1: **for** $e = 1$ **to** `num_epoch` **do**
2: **for** $b = 1$ **to** `num_batch` **do**
3: sample a mini-batch $\{(x_i, y_i)\}_{i=1}^m$ from S; ▷ mini-batch of size m.
4: **for** $i = 1$ **to** `batch_size` **do**
5: $\delta_i^{(0)} = \xi$, with $\xi \sim \mathcal{U}(0, 1)$;
6: **for** $t = 1$ **to** T **do**
7: $\delta_i^{(t)} \leftarrow \texttt{Proj}\left[\delta_i^{(t-1)} + \alpha \texttt{sign}\left(\nabla_\theta \ell(x_i + \delta_i^{(t-1)}, y_i; \theta)\right)\right]$;
8: **end for**
9: $w_i^{\text{unn}} = \texttt{sigmoid}(-\gamma(\texttt{PM}_i - \beta))$;
10: **end for**
11: $\omega_i = M \times w_i^{\text{unn}} / \sum_j w_j^{\text{unn}}, \forall i \in [m]$; ▷ $\omega_i = 1$ during burn-in period.
12: $\theta \leftarrow \theta - \eta \nabla_\theta \sum_{i=1}^m \omega_i \ell(x_i + \delta_i, y_i; \theta) + \mathcal{R}(x_i, y_i; \theta)$;
13: **end for**
14: **end for**

3.2 Margin-Aware Instance Reweighting Learning (MAIL)

To benchmark our proposal against state-of-the-art counterparts, we propose the *margin-aware instance reweighting learning* (MAIL). The overall algorithm is summarized in Algorithm 1. Generally, the objective is $\sum_i \omega_i \ell(x_i + \delta_i^{(T)}, y_i; \theta) + \mathcal{R}(x_i, y_i; \theta)$, where $\mathcal{R}$ is an optional regularization term. This objective implies the optimization for the model, with one step (Step 5–8) generating adversarial variants, one step (Step 9–11) calculating importance weights, and one step (Step 12) minimizing the reweighted loss w.r.t. the model parameters.

Weight Assignment: We adopt the sigmoid function for weight assignment, which can be viewed as a *softened* sample selection operation of the form:

$$\omega_i^{\text{unn}} = \texttt{sigmoid}(-\gamma(\texttt{PM}_i - \beta)), \tag{8}$$

where β indicates how many data should have relatively large weights and $\gamma \geq 0$ controls the smoothness around β. Note that, $\texttt{PM}_i$ denotes the PM score for the i-th data point, which could be any one of $\texttt{PM}_i^{\text{adv}}$, $\texttt{PM}_i^{\text{nat}}$, and $\texttt{PM}_i^{\text{dif}}$. Eq. (8) is a monotonic function that assigns large values for data with small PMs, paying attention to critical data as discussed in Section 2.2. Moreover, it should be further normalized by $\omega_i = N \times w_i^{\text{unn}} / \sum_j w_j^{\text{unn}}$ to meet the constraint in Eq. (3).

Burn-in Period: During the initial training phase, the geometric information is less informative since the deep model is not adequately learned. Directly using the computed weights may mislead the training procedure and accumulate the bias in erroneous weight assignment. Therefore, we introduce a burn-in period at the beginning, where ω is fixed to 1 regardless of the PM value. A similar strategy has also been considered in [42].

Two Realizations: The MAIL is general for reweighting adversarial data, which could be combined with existing works. Here we give two representative examples: the first one is based on the vanilla AT [35], with the learning objective of the form (termed MAIL-AT):

$$-\sum_i \omega_i \log \mathbf{p}_{y_i}(x_i + \delta_i^{(T)}; \theta). \tag{9}$$

The second one is based on TRADES [40], which adopts the Kullback-Leibler (KL) divergence regarding natural and adversarial prediction, and also requires the learning guarantee (taken as a regularization term here) on the natural prediction. Overall, the learning objective is (termed MAIL-TRADES), given by

$$\beta \sum_i \omega_i \texttt{KL}(\mathbf{p}(x_i + \delta_i^{(T)}; \theta) || \mathbf{p}(x_i; \theta)) - \sum_i \log \mathbf{p}_{y_i}(x_i; \theta), \tag{10}$$

where $\beta > 0$ is the trade-off parameter and $\texttt{KL}(p||q) = \sum_k p_k \log p_k / q_k$ denotes the KL divergence regarding p and q.

Our proposal is flexible and general enough to be combined with many other advanced methods. For example, we can modify MART [35], which could discern correct/wrong prediction, to further utilize their geometric properties (termed MAIL-MART). Its formulation is similar to that of MAIL-TRADES with a slightly different learning objective, which is listed in Appendix C.

4 Experiments

We conducted extensive experiments on various datasets, including SVHN [28], CIFAR-10 [21], and CIFAR-100 [21]. The adopted backbone models are ResNet (ResNet-18) [18] and wide ResNet (WRN-32-10) [39]. In Section 4.2, we verified the effectiveness of PM as a geometric measurement. In Section 4.3, we benchmarked our MAIL against advanced methods.

4.1 Experimental Setup

Training Parameters. For all considered methods, the networks were trained using (mini-batch) stochastic gradient descent with momentum 0.9, weight decay 3.5×10^{-3} (for ResNet-18) / 7×10^{-4} (for WRN-32-10), batch size 128, and initial learning rate 0.01 (for ResNet-18) / 0.1 (for WRN-32-10) which is divided by 10 at the 75-th and 90-th epoch. To some extent, this setup can alleviate the impact of adversarial over-fitting [29, 35]. Moreover, following [26], the perturbation bound ϵ is $8/255$ and the (maximal) number of PGD steps k is 10 with step size $\alpha = 2/255$.

Hyperparameters. The slope and bias parameters were set to 30 and -0.07 in MAIL-AT and to 5 and 0.05 in both MAIL-TRADES and MAIL-MART. The trade-off parameter β was set to 5 in MAIL-TRADES, while it was 6 in MAIL-MART (cf. Algorithm 4 in Appendix C). For the training procedure, the weights started to update when the learning rate dropped at the first time following [42], i.e., the initial 74 epochs is burn-in period, and then we employed the reweighted objective.

Robustness Evaluation. We evaluated our methods and baselines using the standard accuracy on natural test data (NAT) and the adversarial robustness based on several representative attack methods, including the PGD method with 100 iterations [26], CW attack [6], APGD CE attack (APGD), and adaptive attack (AA). All these methods have full access to the model parameters (i.e., *white-box* attacks) and are constrained by the same perturbation limit as above. Note that, here we do not focus on the *black-box* attack methods [4, 10, 23, 24], which are relatively easy to be defensed [8].

4.2 Effectiveness of probabilistic Margin

In this section, we verified the effectiveness of PM as a measurement in comparison with LPS. Here, we adopted ResNet-18 as the backbone and conducted experiments on CIFAR-10 dataset.

Three representative methods were considered, including AT [26], MART [35], and TRADES [40], which were further reweighted by assigned weights given by either PM or LPS [42]. For LPS, we adopted the weight assignment function in GAIRAT with the suggested setup [42]. We remind that AT-LPS, MART-LPS, and TRADES-LPS represent GAIRAT, GAIR-MART, and GAIR-TRADES in [42]; and AT-PM, MART-PM, and TRADES-PM represent MAIL-AT, MAIL-MART, and MAIL-TRADES, respectively. Note that, we mainly validated two types of PM here, namely, $\mathtt{PM}^{\text{adv}}$ and $\mathtt{PM}^{\text{nat}}$ that which are very different from LPS.

The experimental results with 5 individual trials are summarized in Table 1, where we adopted three representative evaluations, including natural performance (NAT), the PGD method with 100 steps (PGD), and adaptive attack (AA). Note that AA can be viewed as an ensemble of several advanced attacks (e.g., CW and APGD) and thus reliably reflect the adversarial robustness. As we can see, the superiority of PM is apparent, regardless of the adopted learning objectives: the results of PM are 0.70-7.93% better than LPS regarding AA and 0.14-2.00% better regarding PGD. Although LPS could achieve comparable results regarding PGD attacks, its adversarial robustness is quite low when facing AA attacks. It indicates that the robustness improvement of GAIRAT might be partially. This is probably caused by *obfuscated gradients* [2], since stronger attack methods (e.g., AA) lead to poorer performance on adversarial robustness.

Moreover, comparing the results in using PM_{nat} and PM_{adv}, both PMs can lead to promising robustness, while PM_{adv} is slightly better. The reason is that PM_{adv} can precisely describe the distance

Table 1: Comparison of LPS and PM on CIFAR-10 dataset.

	AT [26]			MART [35]			TRADES [40]		
	LPS	PM_{nat}	PM_{adv}	LPS	PM_{nat}	PM_{adv}	LPS	PM_{nat}	PM_{adv}
NAT	82.26 ± 0.60	82.50 ± 0.20	**83.15 ± 0.46**	**84.45 ± 0.17**	83.75 ± 0.09	84.12 ± 0.46	78.52 ± 0.20	**82.71 ± 0.52**	80.49 ± 0.18
PGD	54.14 ± 0.15	55.00 ± 0.32	**55.25 ± 0.23**	53.16 ± 0.26	53.63 ± 0.21	**53.65 ± 0.23**	51.67 ± 0.54	52.37 ± 0.65	**53.67 ± 0.14**
AA	36.32 ± 0.57	**44.25 ± 0.45**	44.10 ± 0.21	46.00 ± 0.26	46.70 ± 0.10	**47.20 ± 0.21**	44.40 ± 0.20	49.52 ± 0.11	**49.54 ± 0.51**

Table 2: Average accuracy (%) and standard deviation on CIFAR-10 dataset with ResNet-18.

	NAT	PGD	APGD	CW	AA
AT [26]	84.86 ± 0.17	48.91 ± 0.14	47.70 ± 0.06	51.61 ± 0.15	44.90 ± 0.53
TRADES [40]	84.00 ± 0.23	52.66 ± 0.16	52.37 ± 0.21	52.30 ± 0.06	48.10 ± 0.26
MART [35]	82.28 ± 0.60	53.50 ± 0.46	52.73 ± 0.57	51.59 ± 0.16	48.40 ± 0.14
FAT [41]	**87.97 ± 0.15**	46.78 ± 0.12	46.68 ± 0.16	49.92 ± 0.26	43.90 ± 0.82
GAIRAT [42]	83.22 ± 0.06	54.81 ± 0.15	50.95 ± 0.49	39.86 ± 0.08	33.35 ± 0.57
MAIL-AT	83.15 ± 0.46	**55.25 ± 0.23**	**53.20 ± 0.38**	47.81 ± 0.11	44.10 ± 0.21
MAIL-TRADES	81.84 ± 0.18	53.68 ± 0.14	52.92 ± 0.62	**52.89 ± 0.31**	**50.03 ± 0.51**

between adversarial variants and decision boundaries. As a result, it can help accurately assign important instances high wights during AT. Therefore, we adopt PM_{adv} in the following experiments.

4.3 Performance Evaluation

Now, we benchmarked our proposal against state-of-the-art methods in the literature. Here, we mainly report the results on the CIFAR-10 dataset due to the space limitation. For the results on other datasets, please refer to Appendix B.

Compared Baselines. We compared the proposed method with the following baselines: (1) (traditional) AT [26]: the cross-entropy loss for adversarial perturbation generated by the PGD method; (2) TRADES [40]: a learning objective with an explicit trade-off between the adversarial and natural performance; (3) MART [35]: a training strategy which treats wrongly/correctly predicted data separately; (4) FAT [41]: adversarial training with early-stopping in adversarial intensity; (5) GAIRAT [42]: geometric-aware instance-reweighted adversarial training.

All the methods were run for 5 individual trials with different random seeds, where we reported their average accuracy and standard deviation. The results are summarized in Table 2 and Table 3 with the backbone models being ResNet-18 and WRN-32-10, respectively. Overall, our MAIL achieved the best or the second-best robustness against all four types of attacks, revealing the superiority of MAIL (i.e., MAIL-AT and MAIL-TRADES) in adversarial robustness. Specifically, AT and TRADES both treat training data equally, and the results were unsatisfactory compared with the best one ($0.55\% - 6.34\%$ decline with ResNet-18 and $0.02\% - 9.43\%$ decline with $WRN - 32 - 10$), which imply the possibility for their further improvements. Though MART and FAT consider the

Table 3: Average accuracy (%) and standard deviation on CIFAR-10 dataset with WRN-32-10.

	NAT	PGD	APGD	CW	AA
AT	87.80 ± 0.13	49.43 ± 0.29	49.12 ± 0.26	53.38 ± 0.05	48.46 ± 0.46
TRADES	86.36 ± 0.52	54.88 ± 0.39	55.02 ± 0.27	56.18 ± 0.16	53.40 ± 0.37
MART	84.76 ± 0.34	55.61 ± 0.51	55.40 ± 0.37	54.72 ± 0.20	51.40 ± 0.05
FAT	**89.70 ± 0.17**	48.79 ± 0.18	48.72 ± 0.36	52.39 ± 0.89	47.48 ± 0.30
GAIRAT	86.30 ± 0.61	58.74 ± 0.46	55.64 ± 0.36	45.57 ± 0.18	40.30 ± 0.16
MAIL-AT	84.83 ± 0.39	**58.86 ± 0.25**	55.82 ± 0.31	51.26 ± 0.20	47.10 ± 0.22
MAIL-TRADES	84.00 ± 0.15	57.40 ± 0.96	**56.96 ± 0.19**	**56.20 ± 0.30**	**53.90 ± 0.22**

impact of individuals on the final performance, they fail in paying attention to non-robust data during training. Concretely, MART mainly focuses on the correctness regarding natural prediction, and FAT prevents the model learning from *highly* non-robust data in keeping its natural performance. Therefore, FAT achieved the best natural performance ($3.11\% - 6.13\%$ improvement with ResNet-18 and $1.90\% - 5.70\%$ improvement with WRN-32-10), while its robustness against adversaries seems inadequate ($2.97\% - 8.45\%$ decline with ResNet-18 and 3.81%-10.07% decline with WRN-32-10). Moreover, in adopting LPS as the geometric measurement, GAIRAT performed well regarding PGD and AGPD attacks, but the adversarial robustness on CW and AA attacks were low. In comparison, we retain the supremacy on PGD-based attacks (i.e., PGD and APGD) as in GAIRAT and reveal excellent performance on CW and AA attacks. For example, in Table 2, we achieved 0.44%-8.47% improvements on PGD attack, 0.47%-6.52% on APGD attack, 0.59%-13.03% on CW attack, and 1.63%-16.68% on AA attack.

MAIL-AT performs well on PGD-based methods, while MAIL-TRADES is good at CW and AA attacks. It suggests that the robustness depends on adopted learning objectives, coinciding with the previous conclusion in [35]. In general, we suggest using MAIL-TRADES as a default choice, as it reveals promising results regarding AA while keeping a relatively high performance regarding PGD. Besides, comparing the results in Table 2 and Table 3, the overall accuracy had a promising improvement in employing models with a much large capacity (i.e., WRN-32-10), partially verifying the fact that deep models still have insufficient network capacity in fitting adversarial variants [42].

5 Conclusion

In this paper, we focus on boosting adversarial robustness by reweighting adversarial data during training, where data closer to the current decision boundaries are more critical and thus require larger weights. To measure the closeness, we suggest the use of PM, which relates to the multi-class margin regarding the probabilistic space of model outputs (i.e., the estimated class-posterior probabilities). Without any involvement of the PGD iterations, PM is continuous and path-independent and thus overcomes the drawbacks of previous works (e.g., LPS) efficaciously. Moreover, we consider several types of PMs with different geometric properties, and propose a general framework termed MAIL. Experiments demonstrated that PMs are more reliable measurements than previous works, and our MAIL revealed its superiority against state-of-the-art methods, independent of adopted (basic) learning objectives. In the future, we will delve deep into the mechanism in instance-reweighted adversarial learning, theoretically study the contribution of individuals for the final performance, and improve the methodology in using geometric characteristics of data.

References

[1] M. Andriushchenko and N. Flammarion. Understanding and improving fast adversarial training. In *NeurIPS*, 2020.

[2] A. Athalye, N. Carlini, and D. A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

[3] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *ICML*, 2018.

[4] Y. Bai, Y. Zeng, Y. Jiang, Y. Wang, S. Xia, and W. Guo. Improving query efficiency of black-box adversarial attack. In *ECCV*, 2020.

[5] S. Bubeck, Y. T. Lee, E. Price, and I. P. Razenshteyn. Adversarial examples from computational constraints. In *ICML*, 2019.

[6] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[7] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *S&P*, 2017.

[8] A. Chakraborty, A. Manaar, D. Vishal, C. Anupam, and M. Debdeep. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

[9] T. Chen, S. Liu, S. Chang, Y. Cheng, L. Amini, and Z. Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *CVPR*, 2020.

[10] M. Cheng, T. Le, P. Chen, H. Zhang, J. Yi, and C. Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *ICLR*, 2019.

[11] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[12] D. Cullina, A. N. Bhagoji, and P. Mittal. Pac-learning in the presence of evasion adversaries. *arXiv preprint arXiv:1806.01471*, 2018.

[13] M. L. De Prado. *Advances in financial machine learning*. John Wiley & Sons, 2018.

[14] X. Du, J. Zhang, B. Han, T. Liu, Y. Rong, G. Niu, J. Huang, and M. Sugiyama. Learning diverse-structured networks for adversarial robustness. In *ICML*, 2021.

[15] A. Fawzi, S. Moosavi-Dezfooli, P. Frossard, and S. Soatto. Empirical study of the topology and geometry of deep networks. In *CVPR*, 2018.

[16] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019.

[17] R. Gao, F. Liu, J. Zhang, B. Han, T. Liu, G. Niu, and M. Sugiyama. Maximum mean discrepancy is aware of adversarial attacks. In *ICML*, 2021.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[19] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

[20] V. Koltchinskii, D. Panchenko, et al. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of statistics*, 30(1):1–50, 2002.

[21] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[22] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *ICLR*, 2017.

[23] J. Li, R. Ji, H. Liu, J. Liu, B. Zhong, C. Deng, and Q. Tian. Projection & probability-driven black-box attack. In *CVPR*, 2020.

[24] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong. NATTACK: learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *ICML*, 2019.

[25] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[26] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[27] Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/kˆ 2) o (1/k2). In *Sov. Math. Dokl*, 1983.

[28] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[29] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*, 2020.

[30] Y. Qin, N. Carlini, G. W. Cottrell, I. J. Goodfellow, and C. Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *ICML*, 2019.

[31] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. Adversarially robust generalization requires more data. In *NeurIPS*, 2018.

[32] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[33] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.

[34] H. Wang, T. Chen, S. Gui, T.-K. Hu, J. Liu, and Z. Wang. Once-for-all adversarial training: In-situ tradeoff between robustness and accuracy for free. In *NeurIPS*, 2020.

[35] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

[36] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

[37] S. Yang, T. Guo, Y. Wang, and C. Xu. Adversarial robustness through disentangled representations. In *AAAI*, 2021.

[38] X. Yu, G. Chen, and S. Cheng. Dynamic learning rate optimization of the backpropagation algorithm. *IEEE Trans. Neural Networks*, 6(3):669–677, 1995.

[39] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016.

[40] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[41] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. S. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020.

[42] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021.

[43] Y. Zheng, Z. Wu, Y. Yuan, T. Chen, and Z. Wang. Pcal: A privacy-preserving intelligent credit risk modeling framework based on adversarial learning. *arXiv preprint arXiv:2010.02529*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We have discussed the potential drawbacks of our proposal in Appendix B.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We have discussed the computing resources in Appendix B.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] We use only standard datasets
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We use only standard datasets
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]