Switching One-Versus-the-Rest Loss to Increase the Margin of Logits for Adversarial Robustness

Sekitoshi Kanai  
NTT  
sekitoshi.kanai.fu@hco.ntt.co.jp

Shin’ya Yamaguchi  
NTT, Kyoto Univ.  
shinya.yamaguchi.mw@hco.ntt.co.jp

Masanori Yamada  
NTT  
masanori.yamada.cm@hco.ntt.co.jp

Hiroshi Takahashi  
NTT, Kyoto Univ.  
hiroshi.takahashi.bm@hco.ntt.co.jp

Yasutoshi Ida  
NTT  
yasutoshi.ida.yc@hco.ntt.co.jp

Abstract

Defending deep neural networks against adversarial examples is a key challenge for AI safety. To improve the robustness effectively, recent methods focus on important data points near the decision boundary in adversarial training. However, these methods are vulnerable to Auto-Attack, which is an ensemble of parameter-free attacks for reliable evaluation. In this paper, we experimentally investigate the causes of their vulnerability and find that existing methods reduce margins between logits for the true label and the other labels while keeping their gradient norms non-small values. Reduced margins and non-small gradient norms cause their vulnerability since the largest logit can be easily flipped by the perturbation. Our experiments also show that the histogram of the logit margins has two peaks, i.e., small and large logit margins. From the observations, we propose switching one-versus-the-rest loss (SOVR), which uses one-versus-the-rest loss when data have small logit margins so that it increases the margins. We find that SOVR increases logit margins more than existing methods while keeping gradient norms small and outperforms them in terms of the robustness against Auto-Attack.

1 Introduction

For multi-class classification problems, deep neural networks have become the de facto standard method in this decade. They classify data points into the label that has the largest logit, which is input of a softmax function. However, the largest logit can be easily flipped and they can misclassify slightly perturbed data, which are called adversarial examples [27]. Various methods have been presented to search the adversarial examples, and Auto-Attack [8] is one of the most successful methods to find the worst-case attacks. For trustworthy deep learning applications, classifiers should be robust against the worst-case attacks. To improve the robustness, many defense methods have also been presented [19, 22, 33, 7]. Among them, adversarial training practically achieves the best robustness [5, 19, 22], and modifications of adversarial training have been extensively explored [32, 10, 37, 21, 39]. However, adversarial training is still more difficult than standard training [25, 39, 28].

Preprint. Under review.
To improve the performance of adversarial training, several methods focus on the difference in importance of data points \([32][21][39]\). These studies hypothesize that data points closer to a decision boundary are more important for adversarial training \([32][39][21]\). To focus on such data points, GAIRAT \([39]\) and MAIL \([21]\) use weighted softmax cross-entropy loss, which controls weights on the losses based on the closeness to the boundary. As the measure of the closeness, GAIRAT uses the least number of steps at which the iterative attacks make models misclassify the data point. On the other hand, MAIL uses the difference between the output for the true label and the largest output for the other class as the measure of the closeness. MART \([32]\) regards misclassified samples as important and combines two adversarial loss functions for correctly classified and misclassified samples. However, these methods fail to improve the robustness against Auto-Attack. This indicates that it is still unclear how to treat the difference in data points in adversarial training for good robustness.

In this paper, we investigate the causes of the vulnerability of existing methods by evaluating (a) the margins between logits for the true label and the other labels and (b) gradient norms of logits with respect to data points, which are lower bounds of Lipschitz constants. Since Lipschitz constants determine the sensitivities of the logits, adversarial attacks easily perturb logits under the non-small Lipschitz constants. As a result, the largest logit is flipped to the other logits if the margins are not sufficiently large to allow the perturbation. Our experiments show that existing methods reduce the margins and suppress the gradient norms of cross-entropy loss but do not suppress the gradient norms of logits. This indicates that while the existing methods can defend the certain attacks despite of small margins, they do not improve true robustness, i.e., robustness against the worst-case attacks, due to small logit margins and non-small gradient norms. Our experiments also show that the histogram of margins of logits has two peaks, i.e., small and large margins. Thus, the levels of the difficulty are roughly divided into two in each dataset. From these observations, we propose switching one-versus-the-rest loss (SOVR), which switches losses to increase the logit margins when the data point has small margin. We show that one-versus-the-rest loss (OVR) is always greater than or equal to cross-entropy on any logits and expect that OVR penalizes small margins strongly. The experiments demonstrate that SOVR increases the logit margin more than the naïve adversarial training, and outperforms GAIRAT \([39]\), MAIL \([21]\), MART \([32]\), MMA \([10]\), and EWAT \([17]\) in terms of the robustness against Auto-Attack, which is an ensemble of parameter-free attacks to find the worst-case attacks. In addition, we find that our method can be used with the other recent methods \([34][31]\) that improve the generalization performance of adversarial training.

2 Preliminaries

2.1 Adversarial training

Given \(N\) clean data points \(\mathbf{x}_n \in \mathbb{R}^d\) and class labels \(y_n \in \{1, \ldots, K\}\), adversarial training \([22]\) attempts to solve the following minimax problem with respect to the model parameter \(\theta \in \mathbb{R}^m\):

\[
\min_{\theta} \mathcal{L}_{AT}(\theta) = \min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \ell_{CE}(\mathbf{x}_n', y_n, \theta),
\]

\[
\mathbf{x}_n' = \mathbf{x}_n + \delta_n = \mathbf{x}_n + \text{argmax}_{||\delta_n||_p \leq \varepsilon} \ell_{CE}(\mathbf{x}_n + \delta_n, y_n, \theta),
\]

where \(||\cdot||_p\) is \(L_p\) norm, and \(\varepsilon\) is the magnitude of perturbation \(\delta\). \(\ell_{CE}\) is a cross-entropy function. The inner maximization problem is solved by projected gradient descent (PGD) \([19][22]\), which updates the adversarial examples as

\[
\delta_i = \Pi_{\varepsilon} \left( \delta_{i-1} + \eta \text{sign} \left( \nabla_{\delta_{i-1}} \ell (\mathbf{x} + \delta_{i-1}, y, \theta) \right) \right),
\]

for \(K\) steps where \(\eta\) is a step size. \(\Pi_{\varepsilon}\) is a projection operation into the feasible region \(\{ \delta \in \mathbb{R}^d, ||\delta||_p \leq \varepsilon \}\). Note that we focus on \(p = \infty\) since it is a common setting. For trustworthy deep learning methods, we should improve the true robustness: the robustness against the worst-case attacks in the feasible region. Thus, the evaluation of the robustness should use strong attacks, e.g., Auto-Attack \([8]\), since PGD often fails to find the adversarial examples misclassified by models.

2.2 Importance aware adversarial training

GAIRAT (geometry aware instance reweighted adversarial training) \([39]\) and MAIL (margin-aware instance reweighting learning) \([21]\) regard data points closer to the decision boundary of the model \(f\) as important samples and assign the larger weights to the loss for important data points,

\[
\mathcal{L}_{\text{weight}}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \bar{w}_n(\mathbf{x}_n, y_n) \ell_{CE}(\mathbf{x}_n', y_n, \theta),
\]
Figure 1: The robustness against PGD-20 and the components of Auto-Attack on CIFAR10 with PreActResNet18. The results of our method are provided in Appendix.

where $\bar{w}_n \geq 0$ is a weight normalized as $\bar{w}_n(x_n, y_n) = \frac{w_n(x_n, y_n)}{\sum_i w_i(x_i, y_i)}$ and $\sum_n \bar{w}_n = 1$. GAIRAT determines the weights through the

$$w_n = \frac{1 + \tanh(\lambda(1 - 2\kappa_n/K))}{2},$$

where $\kappa_n$ is the least steps at which PGD succeeds at attacking models, and $\lambda$ is a hyperparameter. Thus, since data points closer to the decision boundary does not take a lot of steps, $w_n$ becomes large for them. On the other hand, MAIL uses the following weights:

$$w_n = \text{sigmoid}(-\gamma (PM_n - \beta)),$$

$PM_n = f_{y_n}(x'_n, \theta) - \max_{k \neq y_n} f_k(x'_n, \theta),$  

where $f(x_n, \theta)$ is the softmax output for $x_n$. $\beta$ and $\gamma$ are hyperparameters. MART (misclassification aware adversarial training) [32] regards misclassified samples as important samples and minimizes

$$\ell_{\text{MART}}(x', y, \theta) = \text{BCE}(x', y) + \lambda \text{KL}(f(x, \theta), f(x', \theta)) \cdot (1 - f_y(x, \theta)),$$

$$\text{BCE}(x', y) = -\log(f_y(x', \theta)) - \log(1 - \max_{k \neq y} f_k(x', \theta)).$$

where KL is Kullback-Leibler divergence. MART controls the difference between the loss on unimportant and important samples via $1 - f_y(x_n, \theta)$: MART tends to ignore the second term when the model is confident in the true label. MMA (max-margin adversarial training) [10] also adaptively changes the loss function and $\varepsilon$ for each data point, and thus, MMA also has a similar effect to the above methods. We collectively call the above methods importance aware methods.

2.3 Importance aware methods are vulnerable to Auto-Attack

The studies of [14,8,17] have reported that the robust accuracies of GAILAT, MART, and MMA are lower than naïve adversarial training when using logit scaling attacks or Auto-Attack [8]. Recent studies [34,31,26] use Auto-Attack to evaluate the robustness because it succeeds at attacking several defense methods and achieves the highest success rate [8]. Auto-Attack uses four attacks: Auto PGD (APGD), targeted APGD (t-APGD), targeted FAB (t-FAB) [9], and SQUARE [1], one by one. Since Auto-Attack searches adversarial examples by using various attacks, it achieves larger success rate to attack models than using one attack, e.g., PGD. To clarify the vulnerabilities, we individually evaluate the robustness against the above components of Auto-Attack and 20-step PGD on CIFAR10 [18] in Fig. 1. The training setup is the same as the experiments in Section 6. This figure shows that almost all importance aware methods improve the robustness against PGD-20 and APGD compared with naïve adversarial training (AT [22]). However, they fail to improve the true robustness; i.e., they are vulnerable to t-APGD, t-FAB, and SQUARE compared with AT. Since the reasons of this vulnerability are not discussed well, we investigate them in the next section.

3 Relationship between logit margin loss, Lipschitz constants, and robustness

We investigate the causes of the vulnerabilities of importance aware methods by comparing histograms of logit margin losses and evaluating gradient norms. First, we show that logit margin losses and Lipschitz constants determine the robustness of models near each data point, and we define robustly classified data points and potentially misclassified data points. Next, we experimentally reveal that logit margin losses of importance aware methods concentrate on zero through their histogram of logit margin losses; i.e., the margins of logits are smaller than naïve adversarial training [22]. Finally, we empirically estimate the rate of potentially misclassified data points through the gradient norm and logit margin loss on each data point. Setups of the experiments in this section are provided in Appendix, and all models are obtained by early stopping unless otherwise specified.
3.1 Potentially misclassified data detected by logit margin loss and Lipschitz constants

To investigate the robustness of models near each data point in adversarial training, we use logit margin loss [10]. Logit margin loss is

$$\ell_{LM}(x, y, \theta) = z_{k^*}(x') - z_y(x') = \max_{k \neq y} z_k(x') - z_y(x'),$$

where $k^* = \arg\max_{k \neq y} z_k(x')$ and $z_k(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ is a logit function for the $k$-th label, which is input of softmax: $f_k(x, \theta) = e^{z_k(x)} / \sum_i e^{z_i(x)}$. Since the classifier infers the label of $x$ as $\hat{y} = \arg\max_k z_k(x)$, it correctly classifies $x'$ if $\ell_{LM} \leq 0$. Thus, the logit margin loss on a difficult sample in adversarial training takes a value near zero. We refer to the absolute value of a logit margin loss $|\ell_{LM}|$ as logit margin. In contrast to PM$_n$ (Eq. 7), $\ell_{LM}$ is not bounded since $z_k(x)$ can take an arbitrary value in $\mathbb{R}$.

To explain the effect of logit margins, we assume that clean data point $x$ is correctly classified as $z_{k^*}(x) - z_y(x) \leq 0$, and the Lipschitz constant of the $k$-th logit function is $L_k$ as $|z_k(x_1) - z_k(x_2)| \leq L_k ||x_1 - x_2||_\infty$. If $|z_{k^*}(x) - z_y(x)| \geq (L_k + y_L)\varepsilon$, we have

$$|z_k(x + \delta) - z_y(x + \delta)| \leq z_{k^*}(x) - z_y(x) + (L_y + L_k)\varepsilon \leq 0,$$

and thus, the classifier does not infer the label as $k^*$ for any $\delta \in \{\delta \mid ||\delta||_\infty \leq \varepsilon\}$. From this observation, even if logit margin $|z_{k^*}(x) - z_y(x)|$ is small, the classifier can classify the attacked data into the true label if $L_k$ and $L_y$ are small values and the target label is $k^*$. However, we should pay attention to the following:

$$k^* = \arg\max_{k \neq y} z_k(x) \neq \arg\max_{k \neq y} (z_k(x) - z_y(x) + (L_y + L_k)\varepsilon) =: \hat{k}.$$  

This indicates that the output of classifier is flipped more easily to the $\hat{k}$-th label than the $k^*$-th label: i.e., model is not necessarily sensitive to the targeted attacks whose target label has the largest logit except for the true label if the Lipschitz constant of $z_{k^*}$ is smaller than $z_{\hat{k}}$. In this case, attacks can make models misclassify data points into $\hat{k}$ when we have

$$z_{\hat{k}}(x + \delta) - z_y(x + \delta) = z_{\hat{k}}(x) - z_y(x) + (L_y + L_{\hat{k}})\varepsilon > 0.$$

On the basis of the above, we define the following to discuss the robustness clearly:

Definition 3.1. If a data point $x$ satisfies $z_{k^*}(x) - z_y(x) \leq 0$ for $k^* = \arg\max_{k \neq y} z_k(x)$ and

$$|z_{k^*}(x) - z_y(x)| > (\max_{k} L_k + L_y)\varepsilon,$$

we call it a robustly classified sample. If a data point $x$ satisfies $z_{k^*}(x) - z_y(x) \leq 0$ and

$$|z_{k^*}(x) - z_y(x)| \leq (\max_{k} L_k + L_y)\varepsilon,$$

we call it a potentially misclassified sample.

By using the above definition, we can derive the following:

Proposition 3.2. If data points are robustly classified samples, i.e., not potentially misclassified samples, models are guaranteed to have the certified robustness on these samples as $y = \arg\max_k z_k(x + \delta)$ for any $\delta$ satisfying $||\delta||_\infty \leq \varepsilon$.

Thus, we can estimate the true robustness of each method by counting the number of robustly classified samples or potentially misclassified samples. Proposition 3.2 and Eqs. (14) and (15) indicate that large logit margins $|\ell_{LM}|$ and small Lipschitz constants $L_k$ are necessary to certify the robustness. To investigate these numbers, we empirically evaluate the logit margins in Section 3.2 and gradient norms, which is lower bound of Lipschitz constants, in Section 3.3.

3.2 Histograms of logit margin loss

Since logit margin losses determine the number of potentially misclassified data points, we show the histogram of them for each method on training data of CIFAR10 at the last epoch in Fig. 2. Comparing adversarial training (AT, Fig. 2(b)) with standard training (ST, Fig. 2(a)), AT has two

\[1\] $z_k$ depends on the parameter $\theta$, but we omit $\theta$ in notation for clarity.

\[2\] We regard logits as scalar functions $z_k(x)$ rather than a vector function $z(x) = [z_1(x), \ldots, z_K(x)]^T$ to empirically evaluate the sensitivities in detail in Section 3.3.
peaks in the histogram. This indicates that the levels of the difficulty to increases the margins in AT are roughly divided into two: difficult samples (right peak) and easy samples (left peak). They seem to correspond to the data close to the boundary and the data far away from the boundary, respectively. This result implies that a discrete change in the loss functions seems to be more suitable than continuously weighting the loss like MAIL. Next, comparing AT (Fig. 2(b)) with importance aware methods (Figs. 2(c)-2(f)), their logit margin losses $\ell_{LM}$ concentrate on 0, and their peaks are sharper than that of adversarial training. This indicates that importance aware methods fail to increase the logit margins $|\ell_{LM}|$ for not only difficult samples but also easy samples because the weights for easy samples are relatively small. Thus, we need to strongly penalize the small logit margins for difficult samples without reducing weights on easy samples. In Appendix, we provide results at the best epoch for early stopping and results on other datasets, which show similar tendencies.

### 3.3 Empirical evaluation of potentially misclassified samples

As discussed in Section 3.1, Lipschitz constants of the class-wise logit functions are necessary besides logit margins to evaluate the robustness near each data point. However, Lipschitz constants for deep neural networks are difficult to compute due to their complexity. Thus, we compute the gradient norm of the class-wise loss function and logit function instead of Lipschitz constants because the gradient norm satisfies $\sup_{x} \|\nabla_{x} z_{k}(x)\|_{1} = L_{k}$ for $L_{k}$ such as $|z_{k}(x) - z_{k}(x')| \leq L_{k}\|x - x'\|_{\infty}$. From this relationship, we have

**Proposition 3.3.** If a data point $x$ satisfies

$$|z_{k'}(x) - z_{y}(x)| \leq (\max_{k} \|\nabla_{x} z_{k}(x)\|_{1} + \|\nabla_{x} z_{y}(x)\|_{1})\varepsilon,$$

it is a potentially misclassified sample.

Thus, we can empirically estimate the number of potentially misclassified data points on each method by the gradient norms. Note that this is the sufficient condition but not necessary condition: Even if they do not satisfy Eq. (16), data points can be potentially misclassified samples.

Figure 3 plots the average of gradient norms for loss and logit functions on CIFAR10. Note that SOVR is our method, and its results are discussed in Section 6. First, the gradient norms of cross-entropy $\|\nabla_{x} \ell_{CE}(x, y)\|_{1}$ are relatively small values in all methods. This indicates that adversarial training essentially attempts to suppress the gradient norms for the cross-entropy. MMA has the largest gradient norms, and it is the reason MMA is not robust against Auto-Attack except for SQUARE, which does not use gradient, in Fig. 1. GAIRAT and MAIL have the first and second smallest $\|\nabla_{x} \ell_{CE}(x, y)\|_{1}$, and this is the reason they are robust against PGD despite the small logit margins in Fig 2. However, the gradient norms for logits $\max_{k} \|\nabla_{x} z_{k}(x)\|_{1}$ in importance aware methods are larger than their $\|\nabla_{x} \ell_{CE}(x, y)\|_{1}$ and that of AT. This indicates that $\max_{k} L_{k}$ in these methods are not smaller than AT since gradient norms are lower bound of Lipschitz constants. This non-small $L_{k}$ and small logit margins are the reason importance aware methods fail to improve the true robustness.

By using $\max_{k} \|\nabla_{x} z_{k}(x)\|_{1}$, we investigate the rate of data points that are misclassified or satisfy Eq. (16) in Fig. 4. These data points are potentially misclassified when perturbed by adversarial attacks from Proposition 3.3. In Fig. 4, GAIRAT has the largest rate and this result is consistent with the results of Auto-Attack in Fig. 1. GAIRAT has the least robust accuracy against the worst-case attack. This indicates that though the rate of data points satisfying Eq. (16) are lower bound of the rate of potentially misclassified data points, it is a reasonable metric for estimating the robustness. The results of MAIL are similar to those of GAIRAT, and all importance aware methods have higher
rate than AT. Therefore, importance aware methods fail to improve the robustness compared with AT, and we need to increase the logit margins unlike them. Note that we use the training data of CIFAR10 in Figs. 3 and 4 since we investigate the difficulty of data points in training.

4 Proposed method

From the observations in the above section, our method is based on the two ideas: (i) we switch the loss by the criterion of the logit margin loss because there are difficult samples and easy samples (Fig. 2(b)), and (ii) we penalizing the small logit margins of difficult samples without reducing logit margins by the criterion of the logit margin loss because there are difficult samples and easy samples (Fig. 2(b)), From the observations in the above section, our method is based on the two ideas: (i) we switch the loss

4.1 Switching one-versus-the-rest Loss (SOVR)

The logit margin $|z_k(x) - z_y(x)|$ should be large while keeping Lipschitz constants of logit functions small values. To this end, we need a loss function to penalize small logit margins, and a logit margin loss can be an intuitive candidate as such a loss function. However, the logit margin loss only considers the pair of the largest logit $z_k$, and the logit for the true label $z_y$, and this is not sufficient for the robustness from Eq. (12). In addition, the logit margin loss does not have the desirable property of multi-class classification: infinite sample consistent [40] (also known as classification calibrated consistent, our proposed method uses the one-versus-the-rest loss (OVR):

$$L_{OVR}(\theta, y) = \sum_{i \neq y} \phi(z_i(x)) - \phi(z_y(x)),$$

where $\phi$ is a differentiable non-negative convex function and satisfies $\phi(z) \leq \phi(-z)$ for $z > 0$. We set $\phi(z) = \log(1 + e^{-z})$ and use the following loss for difficult samples:

$$L_{OVR}(\theta, y) = \sum_{i \neq y} \phi(z_i(x)) + \sum_{i \neq y} \phi(z_i(x)) = z_y(x) + \sum_{i \neq y} \phi(z_i(x)),$$

Comparing cross-entropy with OVR

Compared with cross-entropy, we have the following:

**Proposition 4.1.** If we use OVR (Eq. (18)) and softmax as $f_k(x) = e^{z_k(x)}/\sum_i e^{z_i(x)}$, we have

$$0 \leq L_{CE}(\theta, y) \leq L_{OVR}(\theta, y), \quad \forall (\theta, y).$$

When $z_y(x) \to +\infty$ and $z_k(x) \to -\infty$ for $k \neq y$, we have $L_{OVR}(\theta, y) \to 0$ and $L_{CE}(\theta, y) \to 0$

Thus, OVR is always larger than or equal to cross-entropy, and OVR and cross-entropy approach asymptotically to zero when $|L_M|$ grows to infinity. In fact, we observed that $L_{OVR}$ is about four times greater than $L_{CE}$ for random logits $z \sim N(0, I)$ and randomly selected classes from 10 classes ($K = 10$). Thus, we expect that OVR penalizes the small logit margin more strongly than cross-entropy on difficult samples.

**Proposed objective: SOVR**

Our proposed objective function is

$$L_{SOVR}(\theta) = \frac{1}{N} \sum_{(x_n, y_n) \in S} L_{CE}(x_n, y_n, \theta) + \lambda \sum_{(x_j, y_j) \in L} L_{OVR}(x_j', y_j, \theta),$$

where $S$ is a set where logit margin losses $L_M$ are smaller than those in the set $L$, and we have $|S| + |L| = N$. In our method, we regard the top $M$% data points in minibatch as the samples in $L$ through stochastic gradient descent (SGD). $\lambda$ is a hyper-parameter to balance the loss, and $x'$ is an adversarial example generated by Eq. (2). We provide the evaluation of effects of $(M, \lambda)$ in
Algorithm 1 Switching one-versus-the-rest by the criterion of a logit margin loss

1: Select the minibatch $\mathcal{B}$
2: for $x_n \in \mathcal{B}$ do
3: Generate adversarial examples $x'_n = \text{argmax}_{x'_n} \ell_{CE}(x'_n, y_n, \theta)$ by PGD
4: $\ell_{LM}(x'_n, y_n, \theta) = \max_{k \neq y_n} z_k(x'_n) - z_{y_n}(x'_n)$
5: end for
6: Select top $\frac{M}{100} |\mathcal{B}|$ samples of $(x'_n, y_n)$ in terms of $\ell_{LM}(x'_n, y_n, \theta)$ and add them to $\mathcal{L}$
7: $\mathcal{L}_{SOVR}(\theta) = \frac{1}{N} \sum_{(x_n, y_n) \in \mathcal{B} \setminus \mathcal{L}} \ell_{CE}(x'_n, y_n, \theta) + \lambda \sum_{(x_j, y_j) \in \mathcal{L}} \ell_{OVR}(x'_j, y_j, \theta)$
8: Update the parameter $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{SOVR}(\theta)$

Figure 5: Histogram of logit margin losses $\ell_{LM}$ for SOVR, TRADES, and T-OVR on training adversarial examples of CIFAR10 with PreActResNet18 at the last epoch.

Appendix. It shows that when increasing $(M, \lambda)$, $\ell_{LM}$ is monotonically decreasing, i.e., logit margins $|\ell_{LM}|$ is increasing, while their tuning is necessary for the robustness on the test set. The proposed algorithm is shown in Algorithm 1. We first generate $x'$ in Line 3 and compute the $\ell_{LM}$ for them in Line 4. In Line 6, we select the top $M \%$ samples in minibatch and add them to $\mathcal{L}$. Finally, we compute the objective $\mathcal{L}_{SOVR}$ and its gradient to update $\theta$. Since we do not additionally generate the adversarial examples for $\ell_{OVR}$, the overhead of our method is negligible.

4.2 Extension for the other defense methods

Since SOVR only modifies the objective function, it can be used with the optimization algorithms for robustness, e.g., adversarial weight perturbation (AWP) [34] or self-ensemble adversarial training (SEAT) [31], which improve generalization performance in adversarial training. However, it is difficult to use with TRADES [37] because TRADES also modifies the objective function. To combine our method with TRADES, we propose T-OVR, which uses OVR instead of cross-entropy:

$$\mathcal{L}_{T-OVR}(\theta) = \frac{1}{N} \sum_{n=1}^N \ell_{OVR}(x_n, y_n, \theta) + \max_{|\delta_n|_p \leq \varepsilon} \text{KL}(f(x_n, \theta), f(x_n + \delta_n, \theta)).$$ (21)

As shown in Section 4.3, TRADES does not have two peaks of logit margins in the histogram (Fig. 5(b)). Thus, we use the above objective functions on all data points without switching. We evaluate the combinations of SOVR with TRADES, AWP [34], and SEAT [31] in the experiments.

4.3 Histograms of logit margin losses for SOVR

In the same way to Section 3.2, we evaluate the histograms of logit margin losses for SOVR, TRADES, and T-OVR on CIFAR10 in Fig. 5. Figure 5(a) shows that SOVR succeeds at increasing the left peak compared with AT (Fig. 2(b)). This is because OVR strongly penalizes difficult samples in the right peak and moves them into the left peak in training. Figure 5(c) shows that OVR can reduce the logit margin losses even if it imposes on the clean data samples in TRADES (Eq. 21). Note that Fig. 5(b) only has one peak as mentioned above, and thus, switching might not be necessary for TRADES. In Section 5, we find that these large $|\ell_{LM}|$ improve the robustness through the experiments.

5 Related work

The difference in importance of data points in adversarial training has been investigated in several studies [32][38][24][11]. Zhang et al. [38] investigated the effect of the difficult samples on natural generalization performance; i.e., generalization performance on clean data, in adversarial training. They presented friendly adversarial training (FAT) that improves the robustness without compromising
We list the robust accuracy against Auto-Attack on all datasets in Tab. 1. In this table, SOVR
we conducted the experiments for evaluating SOVR. We first compare SOVR with Madry’s AT [22],
Whereas they focus on generalization performance; i.e., the robust accuracy on test data, our method
Auto-Attack to evaluate the robust accuracy on test data. In Appendix, we provide the details of setups
AWP [34], and SEAT [31]. Our experimental codes are based on source codes provided by
(CWRN) [35] for CIFAR10. We used 10-step PGD (step size: 2/255) in training.
This figure shows that SOVR reduces the rate in all settings. This is because SOVR increases
the logit margins

Table 1: Robust accuracy against Auto-Attack (\(L_\infty, \varepsilon = 8/255\)) and clean accuracy on test datasets.

|                | AT   | MART | MMA   | GAIRAT | MAIL  | EWAT  | SOVR  |
|----------------|------|------|-------|--------|-------|-------|-------|
| CIFAR10 (RN18) | 48.0±0.2 | 46.9±0.3 | 37.2±0.9 | 37.7±1 | 39.6±0.4 | 48.2±0.7 | 49.4±0.1 |
| CIFAR10 (WRN)  | 51.9±0.5 | 50.4±0.09 | 43.1±1 | 41.8±0.6 | 43.3±0.1 | 51.6±0.3 | 52.7±0.2 |
| SVHN (RN18)    | 45.6±0.4 | 46.9±0.3 | 41.0±1 | 37.6±0.6 | 41.2±0.3 | 47.6±0.4 | 48.5±0.4 |
| CIFAR100 (RN18) | 23.7±0.3 | 23.9±0.1 | 18.4±0.2 | 19.8±0.5 | 16.7±0.3 | 23.5±0.06 | 24.9±0.2 |

|                |      |      |       |        |       |       |       |
|----------------|------|------|-------|--------|-------|-------|-------|
| CIFAR10 (RN18) | 81.6±0.5 | 78.3±1 | 85.5±0.7 | 78.7±0.7 | 79.5±0.4 | 82.8±0.4 | 80.6±0.1 |
| CIFAR10 (WRN)  | 85.6±0.1 | 81.5±1 | 87.8±1 | 83.0±0.7 | 82.2±0.4 | 86.0±0.5 | 85.4±1 |
| SVHN (RN18)    | 89.8±0.6 | 86.9±0.6 | 93.9±0.4 | 89.9±0.4 | 89.4±0.4 | 90.2±0.6 | 90.0±1 |
| CIFAR100 (RN18) | 53.0±0.7 | 49.2±0.1 | 60.6±0.6 | 52.0±0.5 | 46.5±0.5 | 54.2±1 | 52.1±0.8 |

the natural generalization performance. Unlike FAT, our method focuses on the robust performance.
Sanyal et al. [24] and Dong et al. [11] have investigated the effect of memorization in adversarial
training: memorizing difficult samples hurts the generalization performance of adversarial training.
Whereas they focus on generalization performance; i.e., the robust accuracy on test data, our method
improves the performance on test data through the improvements of performance in training data; i.e.,
militating the underfitting rather than overfitting. Several papers have investigated the relationship
between Lipschitz constants and robustness [6, 29, 36, 15]. Whereas they propose the constraints of
Lipschitz constants to improve the robustness, we use them to discuss the effect of small logit margins
and evaluate the robustness through gradient norms, which are lower bounds of Lipschitz constants.

Studies of [14, 8, 17] have reported the vulnerabilities of some of importance aware methods to
logit scaling attacks or Auto-Attack but few studies discuss the causes. Kim et al. [17] have pointed
out that the cause is high entropy in GAIRAT and MART and presented EWAT (entropy-weighted
adversarial training), which imposes a larger weight on the higher entropy. Compared with [17], the
logit margin is clearly related to robustness than entropy because it enables to rigorously discuss the
certified robustness in Section 3. In addition, SOVR outperforms EWAT as shown in the next section.

6 Experiments

6.1 Setup

We conducted the experiments for evaluating SOVR. We first compare SOVR with Madry’s AT [22],
MMA [10], MART [32], GAIRAT [39], MAIL [21], and EWAT [17] on three datasets: CIFAR10,
SVHN, and CIFAR100 [18, 23]. Next, we evaluate the combination of SOVR with TRADES [37],
AWP [34], and SEAT [31]. Our experimental codes are based on source codes provided by
[34, 32, 10]. We used PreActResNet-18 (RN18) [13] for all datasets and WideResNet-34-10
(WRN) [35] for CIFAR10. We used 10-step PGD (step size: 2/255) in training. \(\varepsilon\) is set to 8/255. We
used early stopping by evaluating test robust accuracies against 20-step PGD. For AWP and SEAT,
we use the original public code [34, 31]. We trained models three times and show the average and
standard deviation of test accuracies. For hyperparameters in SOVR, we set \((M, \lambda)\) to \((50, 0.25)\) for
CIFAR10 (RN18), \((50, 0.5)\) for CIFAR10 (WRN) and CIFAR100, and \((20, 0.2)\) for SVHN. We use
Auto-Attack to evaluate the robust accuracy on test data. In Appendix, we provide the details of setups
and additional results, e.g., evaluation of individual Auto-Attack like Fig. 1 and other various attacks.

6.2 Results

We list the robust accuracy against Auto-Attack on all datasets in Tab. 1. In this table, SOVR
outperforms the importance aware methods and AT in terms of the robustness against Auto-Attack.
Figure 5 shows the rate of data that are misclassified or satisfy Eq. (16) in the same way to Fig. 4.
This figure shows that SOVR reduces the rate in all settings. This is because SOVR increases
the logit margins \(|\ell_{LM}|\) without increasing gradient norms much by using OVR. In fact, Fig. 5 shows
that SOVR increases the logit margins \(|\ell_{LM}|\) for difficult samples, and Fig. 3 shows that

\footnote{We could not reproduce the results reported in [34, 31] even though we did not modify their original codes. This might be because we report the averaged values for reproducibility of our experiments.}
increases in gradient norms of SOVR can be negligible: SOVR increases $\max_k ||\nabla_x z_k(x)||_1$ but decreases $||\nabla_x z_\neq(x)||_1$. EWAT achieves higher robust accuracies than AT on several datasets in Tab. 1. However, it does not reduce the rate of potentially misclassified samples in Fig. 6, and this result implies that the logit margin loss is more effective to evaluate and improve robustness than entropy. In Tab. 1, SOVR slightly sacrifices accuracies under some settings on clean data. We provide the histograms of logit margin losses on all datasets in Appendix, which also show SOVR increases margins.

Our analysis mostly focuses on training performance rather than generalization performance in adversarial training. Even so, SOVR can be used with recent studies that focus on the generalization performance. We evaluated the combination of SOVR and TRADES, SOVR and AWP, SOVR and SEAT. Table 2 lists the robust accuracy of the combinations against Auto-Attack and shows that SOVR improves the performance of other recent methods without harming the generalization. Thus, SOVR and these methods complementarily improve the performance in adversarial training.

7 Conclusion

We investigated the reason importance aware methods fail to improve the robustness against Auto-Attack. Our empirical results show the reason to be that they reduce the margins between the logits for a true label and for the others without reducing Lipschitz constants of logits sufficiently. In addition, we find that the histogram of logit margin losses of adversarial training has two peaks. From these observations, we propose switching the loss from cross-entropy to one-versus-the-rest loss by the criterion of the logit margin loss. Since one-versus-the-rest loss penalizes the logit margins more strongly than cross-entropy, our method increases the margins and improves the robustness.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In Proc. ECCV, 2020.
[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Proc. ICML, pages 274–283, 2018.
[3] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. 2003.
[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (SP), pages 39–57. IEEE, 2017.
[5] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In Proc. NeurIPS, pages 11190–11201, 2019.
[6] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In Proc. ICML, pages 854–863, 2017.

[7] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Proc. ICML, pages 1310–1320, 2019.

[8] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Proc. ICML, 2020.

[9] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In Proc. ICML, pages 2196–2205, 2020.

[10] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. In Proc. ICLR, 2020.

[11] Yinpeng Dong, Ke Xu, Xiao Yang, Tianyu Pang, Zhijie Deng, Hang Su, and Jun Zhu. Exploring memorization in adversarial training. In Proc. ICLR, 2022.

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

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proc. CVPR, pages 770–778, 2016.

[14] Dorjan Hitaj, Giulio Pagnotta, Iacopo Masi, and Luigi V Mancini. Evaluating the robustness of geometry-aware instance-reweighted adversarial training. arXiv preprint arXiv:2103.01914, 2021.

[15] Yujia Huang, Huan Zhang, Yuanyuan Shi, J Zico Kolter, and Anima Anandkumar. Training certifiably robust neural networks with efficient local lipschitz bounds. Proc. NeurIPS, 34, 2021.

[16] Matt Jordan and Alexandros G Dimakis. Exactly computing the local lipschitz constant of relu networks. Proc. NeurIPS, 33:7344–7353, 2020.

[17] Minseon Kim, Jihoon Tack, Jinwoo Shin, and Sung Ju Hwang. Entropy weighted adversarial training. In ICML 2021 Workshop on Adversarial Machine Learning, 2021.

[18] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, 2009.

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

[20] Yi Lin. Support vector machines and the bayes rule in classification. Data Mining and Knowledge Discovery, 6(3):259–275, 2002.

[21] Feng Liu, Bo Han, Tongliang Liu, Chen Gong, Gang Niu, Mingyuan Zhou, Masashi Sugiyama, et al. Probabilistic margins for instance reweighting in adversarial training. Proc. NeurIPS, 34, 2021.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In Proc. ICLR, 2018.

[23] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.

[24] Amartya Sanyal, Puneet K. Dokania, Varun Kanade, and Philip Torr. How benign is benign overfitting? In Proc. ICLR, 2021.

[25] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. Proc. NeurIPS, 31, 2018.

[26] Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, and Venkatesh Babu Radhakrishnan. Towards efficient and effective adversarial training. In Proc. NeurIPS, 2021.

[27] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[28] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In Proc. ICLR, 2018.
[29] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable
certification of perturbation invariance for deep neural networks. In Proc. NeurIPS, pages
6542–6551, 2018.

[30] Jonathan Uesato, Brendan O’Donoghue, Pushmeet Kohli, and Aaron van den Oord. Adversarial
risk and the dangers of evaluating against weak attacks. In Proc. ICML, volume 80, pages
5025–5034. PMLR, 2018.

[31] Hongjun Wang and Yisen Wang. Self-ensemble adversarial training for improved robustness.
In Proc. ICLR, 2022.

[32] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving
adversarial robustness requires revisiting misclassified examples. In Proc. ICLR, 2020.

[33] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving
adversarial robustness requires revisiting misclassified examples. In Proc. ICLR, 2020.

[34] Dongxian Wu, Shu tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust
generalization. In Proc. NeurIPS, 2020. URL https://github.com/csdongxian/AWP.

[35] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. arXiv preprint
arXiv:1605.07146, 2016.

[36] Bohang Zhang, Tianle Cai, Zhou Lu, Di He, and Liwei Wang. Towards certifying l-infinity
robustness using neural networks with l-inf-dist neurons. In Proc. ICML, volume 139, pages
12368–12379, 2021.

[37] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan.
Theoretically principled trade-off between robustness and accuracy. In Proc. ICML, volume 97,
pages 7472–7482. PMLR, 2019.

[38] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan
Kankanhalli. Attacks which do not kill training make adversarial learning stronger. 119:
11278–11287, 2020.

[39] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli.
Geometry-aware instance-reweighted adversarial training. In Proc. ICLR, 2021.

[40] Tong Zhang. Statistical analysis of some multi-category large margin classification methods.
Journal of Machine Learning Research, 5(Oct):1225–1251, 2004.
A Proofs

A.1 The proof of Proposition 3.2

Proof. From the definition of Lipschitz constants, we have

\[ |z_k(x + \delta) - z_k(x)| \leq L_k |x + \delta - x|_\infty = L_k \varepsilon, \]  

(22)

Thus, we have \( z_k(x + \delta) \leq z_k(x) + L_k \varepsilon \) if \( z_k(x + \delta) \geq z_k(x) \) and \( z_k(x + \delta) \geq z_k(x) - L_k \varepsilon \) if \( z_k(x + \delta) \leq z_k(x) \). Therefore, the following inequalities hold for the robustly classified samples:

\[
\begin{align*}
\max_{k \neq y} z_k(x + \delta) - z_y(x + \delta) & \leq z_k'(x) + L_k \varepsilon - z_y(x) + L_y \varepsilon \\
& \leq z_k(x) - z_y(x) + (\max_k L_k + L_y) \varepsilon, \\
\end{align*}
\]

(23)

where \( k' = \arg\max_{k \neq y} z_k(x + \delta) \) and \( k^* = \arg\max_{k \neq y} z_k(x) \). From \( z_k^*(x) - z_y(x) < 0 \), \( |z_k^*(x) - z_y(x)| > (\max_k L_k + L_y) \), and Eq. (23), we have \( \max_{k \neq y} z_k(x + \delta) - z_y(x + \delta) \leq 0 \), \( \forall \delta \in \|\delta\|_\infty \leq \varepsilon \). Thus, models are guaranteed to classify adversarial examples of these data points accurately. \( \square \)

A.2 The proof of Proposition 3.3

Proof. Since we have \( \sup_x \|\nabla_x z_k(x)\|_q = L_k \) for an \( L_k \)-Lipschitz function such as \( |z_k(x_1) - z_k(x_2)| \leq L_k \|x_1 - x_2\|_p \) where \( 1/q + 1/p = 1 \) [16], the following inequality holds if \( z_k^*(x) - z_y(x) \leq (\max_k L_k + L_y) \varepsilon \):

\[
|z_k^*(x) - z_y(x)| \leq (\max_k \|\nabla_x z_k(x)\|_1 + \|\nabla_x z_y(x)\|_1) \varepsilon 
\]

(24)

because \( \|\nabla_x z_k(x)\|_1 \leq L_k \) for \( p = \infty \). Thus, we have \( |z_k^*(x) - z_y(x)| \leq (\max_k L_k + L_y) \varepsilon \) on this condition, which completes the proof. \( \square \)

A.3 The proof of Proposition 4.1

Proof. By using log functions, \( \ell_{CE} \) can be written as

\[
\ell_{CE}(x, y, \theta) = -z_y(x) + \log \sum_i e^{e^{zi}(x)}. 
\]

(25)

when a model uses a softmax function. Compared with OVR \( \ell_{OVR}(x, y, \theta) = -z_y(x) + \sum_i \log(1 + e^{e^{zi}(x)}) \), the difference is only the second term. Thus, \( \ell_{OVR}(x, y, \theta) - \ell_{CE}(x, y, \theta) \) can be written as

\[
\begin{align*}
\ell_{OVR}(x, y, \theta) - \ell_{CE}(x, y, \theta) &= \sum_i \log(1 + e^{zi}) - \log \sum_i e^{zi}, \\
&= \log \Pi_i (1 + e^{zi}) - \log \sum_i e^{zi}. \\
\end{align*}
\]

(26, 27)

Since logarithm is a strictly increasing function, we have

\[
\Pi_i (1 + e^{zi}) - \sum_i e^{zi} \geq 0 \Rightarrow \log \Pi_i (1 + e^{zi}) - \log \sum_i e^{zi} \geq 0. 
\]

(28)

Since \( e^{zi} \geq 0 \) for any \( z_i \in \mathbb{R} \), \( \Pi_i (1 + e^{zi}) - \sum_i e^{zi} \geq 0 \) is proved by mathematical induction: For \( K = 1 \), we have \( (1 + e^{zi}) - e^{zi} = 1 \geq 0 \) for any \( z_i \). If \( \Pi_{i=1}^{K'} (1 + e^{zi}) - \sum_{i=1}^{K'} e^{zi} \geq 0 \) for any \( z_i \) and for \( K = K' \), we have

\[
\begin{align*}
\Pi_{i=1}^{K'+1} (1 + e^{zi}) - \sum_{i=1}^{K'+1} e^{zi} &= (1 + e^{zi}) \Pi_{i=1}^{K'} (1 + e^{zi}) - (e^{zi}^{K'} + \sum_{i=1}^{K'} e^{zi}) \\
&= (\Pi_{i=1}^{K'} (1 + e^{zi}) - \sum_{i=1}^{K'} e^{zi}) + e^{zi+K'} (\Pi_{i=1}^{K'} (1 + e^{zi}) - 1) \geq 0 \\
\end{align*}
\]

(29, 30)

for \( K = K' + 1 \) since \( e^{zi} \geq 0 \) for any \( z_i \in \mathbb{R} \). Thus, the left hand side of Eq. (28) holds for any \( K \geq 1 \), and we have \( \log \Pi_i (1 + e^{zi}) - \log \sum_i e^{zi} \geq 0 \). Therefore, we have \( \ell_{OVR}(x, y, \theta) - \ell_{CE}(x, y, \theta) \geq 0 \):
where we have \( e^z \to 0 \) and
\[
\lim_{z_k \to +\infty} \ell_{CE}(x, y, \theta) = \lim_{z_k \to +\infty} -y log e^z(x) + log \sum_i e^{z_i(x)} = -y + log(e^z) = 0. \tag{31}
\]
On the other hand, when \( z_k(x) \to +\infty \) and \( z_k(x) \to -\infty \) for \( k \neq y \), we have \( log(1 + e^z) \to y \) and \( log(1 + e^z) \to 0 \). Thus, we have
\[
\lim_{z_k \to +\infty} \ell_{OVR}(x, y, \theta) = \lim_{z_k \to -\infty} -y + \sum_i log(1 + e^{z_i(x)}) = -y + y = 0. \tag{32}
\]
which completes the proof. \( \square \)

B Additional explanation about previous methods

B.1 MMA \([10]\)

MMA \([10]\) attempts to maximize the distance between data points and the decision boundary for robustness. MMA regards \( \min_\delta \| \delta_n \|_\infty \) subject to \( \{ \ell_{LM}(x, y, \delta) \geq 0 \} \) as the distance. By using this distance, MMA minimizes the following loss:
\[
\mathcal{L}(\theta) = \frac{1}{n} \sum_{n=1}^{N} \ell_{CE}(x_n, y_n, \theta) + \frac{2}{\ell} \mathcal{L}_{MMA}(\theta) \tag{33}
\]

\[
\mathcal{L}_{MMA}(\theta) = \sum_{(x, y) \in \mathcal{H}} \ell_{LM}(x, y, \delta) + \sum_{(x, y, \delta) \in \mathcal{H}} \ell_{LM}(x, y, \delta) \tag{34}
\]

\[
\delta_{MMA} = \arg \min_{\delta} \ell_{SLM}(x, y, \delta) \geq 0 \| \delta \|_\infty \tag{35}
\]

\[
\ell_{SLM}(x, y) = \log \sum_{j \neq y} e^{z_j(x)} - y \tag{36}
\]

where \( \mathcal{H} \) is a set of correctly classified data points, and \( \mathcal{H} \) is a set of misclassified data points. MMA regards \( \min_\delta \| \delta_n \|_\infty \) subject to \( \| \delta_n \|_\infty \leq \delta_{max} \) as \( \mathcal{H} \). MMA uses \( \min_\delta \| \delta_n \|_\infty \) depends on data points as Eq. (35), we consider that it has similar effects to the importance aware methods. In MMA, Ding et al. \([10]\) use \( \ell_{SLM}(x, y) \) as a approximated differentiable logit margin loss by changing max into differentiable function \( \log \sum_{j \neq y} e^{z_j(x)} \). Compared OVR with \( \ell_{SLM}(x, y) \), we have the following:
\[
\ell_{SLM}(x, y) \leq \ell_{CE}(x, y) \leq \ell_{OVR}(x, y). \tag{37}
\]
This is because we have
\[
\ell_{CE}(x, y) - \ell_{SLM}(x, y) = \log \sum_j e^{z_j(x)} - \log \sum_{j \neq y} e^{z_j(x)} = log \left( 1 + \frac{e^{z_y(x)}}{\sum_{j \neq y} e^{z_j(x)}} \right) \geq 0
\]
and \( \ell_{CE}(x, y) \leq \ell_{OVR}(x, y) \) from Proposition 4.1. Thus, we expect that OVR more strongly penalizes the small logit margins than \( \ell_{SLM}(x, y) \). Note that training algorithms of MMA is also different from those of the other importance aware methods \([10]\).

B.2 EWAT \([17]\)

EWAT uses a weighted cross-entropy like GAIT and MAIL, but it is added to cross-entropy as
\[
\ell_{weight}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \left( 1 + \bar{w}_n(x_n', y_n) \right) \ell_{CE}(x_n', y_n, \theta), \tag{38}
\]
where \( \bar{w}_n \geq 0 \) is a weight divided by batch-mean of the weight \( w_n \) as \( \bar{w}_n(x_n, y_n) = \frac{\left[ \mathbb{E}[w_n(x_n, y_n)] \right]}{\sum_{(x, y) \in \mathcal{H}} w_n(x, y)} \). EWAT determines the weights by using entropy as
\[
w_n(x_n, y_n) = -\sum_{k=1}^{K} f_k(x_n', \theta) log(f_k(x_n', \theta)) \tag{39}
\]
where \( f_k(x_n, \theta) \) is the \( k \)-th softmax output, and thus, it can be regarded as the probability for the \( k \)-th class label. EWAT is based on the observation that importance aware methods tend to have high-entropy, and it causes their vulnerability. Our theoretical results about logit margins and experiments seem to indicate that a logit margin loss is a more reasonable criterion to evaluate the robustness and improve the robustness by using it than entropy.

\( ^4 \)For clarity, we use the term “margin” only for the distance between logits of the true labels and of the label that has the largest logit except for the true label, not for the distance between data points and the decision boundary.
C Experimental Setups

We conducted the experiments for evaluating our proposed method. We first compared our method with baseline methods; Madry’s AT [22], MMA [10], MART [32], GAIRAT [39], MAIL [21], and EWAT [17] on three datasets; CIFAR10, SVHN, and CIFAR100 [18, 23]. Next, we evaluated the combination of our method with TRADES [37], AWP [34], and SEAT [31].

Our experimental codes are based on source codes provided by [34, 32, 10]. We used PreActResNet-18 (RN18) [13] and WideResNet-34-10 (WRN) [35]. The $L_\infty$ norm of the perturbation was set to $\varepsilon = 8/255$, and all elements of $x_i + \delta_i$ were clipped so that they were in $[0,1]$. We used early stopping by evaluating test robust accuracies against 20-step PGD. To evaluate TRADES, AWP, and SEAT, we used the original public code [34, 31].

We trained models three times and show the average and standard deviation of test accuracies. We used Auto-Attack to evaluate the robust accuracy on test data. We used one GPU among NVIDIA®V100 and NVIDIA®A100 for each training in experiments. We trained models for three times and show the average and standard deviation of test accuracies. For MART, we used mart_loss in the original code [32] as the loss function. $\lambda$ of MART was set to 6.0. For GAIRAT and MAIL, we also used the loss functions in the original codes [39, 21] and thus, hyperparameters of their loss functions were based on them. $\lambda$ of GAIRAT was set to $\infty$ until the 50-th epoch and then set to 3.0. ($\gamma, \beta$) of MAIL was set to $(10, 0.5)$. For all settings, the size of mini-batch was set to 128. The detailed setup for each dataset was as following.

MMA

We trained models by using MMA based on the original code [10]. Thus, the learning rate schedules and hyperparameters of PGD for MMA were different from those for other methods because the training algorithm of MMA is different from the other methods. The step size of PGD in MMA was set to $2\varepsilon/10$ in training by following [10]. For AN-PGD in MMA, the maximum perturbation length was 1.05 times the hinge threshold $\varepsilon_{\text{max}} = 1.05d_{\text{max}}$, and $d_{\text{max}}$ was set to 0.1255. The learning rate of SGD was set to 0.3 at the 0-th parameter update, 0.09 at the 20000-th parameter update, 0.03 at the 30000-th parameter update, and 0.009 at the 40000-th parameter update.

C.1 CIFAR10

For ResNet18, the learning rate of SGD was divided by 10 at the 100-th and 150-th epoch except for EWAT, and the initial learning rate was set to 0.05 for SOVR and set to 0.1 for others. We tested the initial learning rate of 0.05 for the other methods and confirmed that the setting of 0.1 achieved better robust accuracies against Auto-Attack than the setting of 0.05 when using ResNet18. For EWAT, we divided the learning rate of SGD at the 100-th and 105-th epoch following [17] after we confirmed that the division at the 100-th and 150-th epoch was worse than the division at the 100-th and 105-th epoch. When using WideResNet34-10, we set the initial learning rate to 0.1 and divided by 10 at the 100-th and 150-th epoch. We used momentum of 0.9 and weight decay of 0.0005 and early stopping by evaluating test accuracies. We standardized datasets by using mean = [0.4914, 0.4822, 0.4465] and std = [0.2471, 0.2435, 0.2616] as the pre-process.

C.2 CIFAR100

We used PreActResNet18 for CIFAR100. The learning rate of SGD was divided by 10 at the 100-th and 150-th epoch except for EWAT, and the initial learning rate was set to 0.1. Note that we confirmed that the above setting is better than the initial learning rate of 0.05 for all methods. For EWAT, we divided the learning rate of SGD at the 100-th and 105-th epoch following [17] after we confirmed that the division at the 100-th and 150-th epoch was worse than the division at the 100-th and 105-th epoch. We used momentum of 0.9 and weight decay of 0.0005 and early stopping by evaluating test accuracies. For PGD, we randomly initialized the adversarial

---

5We could not reproduce the results reported in [34, 31]. This might be because we report the averaged values for reproducibility.

https://github.com/YisenWang/MART
https://github.com/zjfheart/Geometry-aware-Instance-reweighted-Adversarial-Training
https://github.com/QizhouWang/MAIL
https://github.com/BorealisAI/mma_training
perturbation and updated it for 10 steps with a step size of $2/255$. We standardized datasets by using mean = $[0.5070751592371323, 0.48654887331495095, 0.4409178433670343]$, and std = $[0.2673342858792401, 0.2564384629170883, 0.27615047132568404]$ as the pre-process. $(M, \lambda)$ was set to $(0.5, 0.5)$ based on the coarse hyperparameter tuning.

### C.3 SVHN

We used PreActResNet18 for SVHN. The learning rate of SGD was divided by 10 at the 100-th and 150-th, and the initial learning rate was set to 0.05 for SOVR and set to 0.01 for others. We tested the initial learning rate of 0.05 for the other methods and confirmed that the setting of 0.01 achieved better robust accuracies against Auto-Attack than the setting of 0.05. For EWAT, the learning rate of SGD was divided by 10 at the 100-th and 105-th epoch after we confirmed that this setting was better than the division at 100-th and 105-th epoch. The hyperparameters for PGD were based on [34]: The adversarial perturbation is randomly initialized and is updated for 10 steps with a step size of $1/255$. For the preprocessing, we standardized data by using the mean of $[0.5, 0.5, 0.5]$, and standard deviations of $[0.5, 0.5, 0.5]$. $(M, \lambda)$ is set to $(0.2,0.2)$ based on the coarse hyperparameter tuning.

### C.4 TRADES, AWP, and SEAT

For experimental settings of TRADES and AWP, we followed [34] and only changed the training loss into SOVR in the training procedure and in the algorithm for computing the perturbation of AWP for SOVR+AWP, T-OVR, and T-OVR+AWP. We used the original codes of AWP [34], $\beta$ of TRADES and T-OVR were set to 6, and $\gamma$ of AWP is set to 0.01 for AWP and AWP+SOVR. $\gamma$ of AWP was set to 0.005 for TRADES+AWP and T-OVR+AWP. AWP is applied after the 10-th epoch for TRADES+AWP and T-OVR+AWP. We used WideResNet34-10 following [34]. We used SGD with momentum of 0.9 and weight decay of 0.0005 for 200 epochs. The initial learning rate was set to 0.1 and was divided by 10 at the 100-th and 150-th epoch. For experimental settings of SEAT, we followed [31] and only changed the loss into SOVR in the original code [31]. We did not evaluate SEAT with CutMix in our experiments, but we fairly compare SEAT+SOVR with SEAT under the same condition. We used SGD with momentum of 0.9 and weight decay of $7 \times 10^{-4}$ for 120 epochs. The initial learning rate was set to 0.1 till the 40-th epoch and then linearly reduced to 0.01 and 0.001 at the 60-th epoch and 120-th epoch, respectively. We used WideResNet32-10 following [31] for SEAT. $(M, \lambda)$ is set to $(0.5,0.5)$ for AWP and SEAT after coarse hyperparameter tuning.

### C.5 Experimental setups in Section 3

For the experiments in Section 3, we used the models obtained under the above settings, which are the same as models used in Section 6. To obtain the histograms of logit margins, we used the models and computed logit margin loss on adversarial examples of training data set for each data point at 200 epochs. Thus, the number of data points of CIFAR10 and CIFAR100 is 50,000, and that of SVHN is 73,257. We also provide the results of the models obtained by the early stopping in Fig. 8. To compute the rate of training data points misclassified or satisfying Eq. (16), we computed the norm of the gradient $\max_k | \nabla_{z_k}(x) |_1$ and logit margin loss $\ell_{LM}(x)$ for each clean data point. By using them, we computed Eq. (16) and checked whether Eq. (16) is satisfied for each data point. In this experiment, we used the models obtained by early stopping.

### D Additional results

#### D.1 Histograms of logit margin losses

We show the additional histograms of logit margin losses in this section. First, Fig. 7 plots the result of EWAT on training samples of CIFAR10 at the last epoch. Compared with SOVR, EWAT does not increases logit margins for difficult samples (right peak). Figure 8 plots the results at the best epoch by early stopping, and it show the similar tendencies at the last epoch. Figures 9-12 plot the histogram when using WideResNet and other datasets. SOVR tends to increase the left peak under all conditions, and thus, it decreases logit margin losses $\ell_{LM}$, and thus, it increases the logit margins.

---

[9] https://github.com/csdongxian/AWP

[10] https://github.com/whj363636/Self-Ensemble-Adversarial-Training
Figure 7: Histogram of logit margin losses of EW AT for training data on CIFAR10.

Figure 8: Histogram of logit margin losses for training samples of CIFAR10 at the best epoch by early stopping. We plot those on adversarial data $x'$ for the other methods. Blue bins correspond to the data points that models correctly classify.

$|\ell_{LM}|$. Figure 11 shows that AT does not have two peaks on SVHN. To investigate histograms on SVHN in detail, we additionally evaluate logit margin losses at the 100-th epoch in Fig. 12. This figure shows that the histogram on SVHN has two peaks at the 100-th epoch, but they became one peak at the 200-th epoch (Fig. 11). This might cause the optimal $(M, \lambda)$ for SOVR to be smaller than that for other datasets. Table 3 lists the average of logit margin losses. Since the distributions of logit margin losses are long-tailed as shown in histograms, the difference of average values of logit margin losses among methods is small. Even so, SOVR tends to have the lowest logit margin losses under almost all settings.

We additionally evaluate norms of the gradients of the loss and logits for all methods on CIFAR10. Figure 13 plots the results. In this figure, gradient norms of cross-entropy for the true label $||\nabla_x \ell_{CE}(x, y)||_1$ are always smaller than those for the other labels. In addition, gradient norms of cross-entropy for the label which has the largest logit except for the true label $||\nabla_x \ell_{CE}(x, k^*)||_1$ are smaller than those for the randomly selected labels. This might cause the robustness against PGD attacks of importance aware methods despite of small logit margins. Gradient norms of EW AT and SOVR are not significantly different from those of AT.

Table 3: The average of logit margin losses for training dataset $x'$ at the last epoch.

| Dataset       | AT     | MART   | MMA    | GAIRAT | MAIL    | SOVR   |
|---------------|--------|--------|--------|--------|---------|--------|
| CIFAR10 (RN18)| -3.66±0.02 | -3.46±0.02 | 0.558±0.1 | -0.258±0.02 | -0.0293±0.02 | -3.97±0.02 |
| CIFAR10 (WRN) | -6.96±0.01 | -5.65±0.7 | -2.73±0.6 | -0.717±0.03 | -0.203±0.01 | -10.07±0.02 |
| CIFAR100      | -2.41±0.02 | -1.76±0.01 | 2.07±0.06 | -0.68±0.02 | 0.684±0.1 | -2.44±0.1 |
| SVHN          | -7.55±0.01 | -8.08±2 | -4.31±0.04 | -3.91±0.01 | -1.17±0.03 | -7.59±0.1 |
Figure 9: Histogram of logit margin losses for training samples of CIFAR10 with WideResNet34-10 at the last epoch. We plot those on adversarial data $x'$ for the other methods. Blue bins correspond to the data points that models correctly classify.

Figure 10: Histogram of logit margin losses for training samples of CIFAR100 at the last epoch. We plot those on adversarial data $x'$ for the other methods. Blue bins correspond to the data points that models correctly classify.

Figure 11: Histogram of logit margin losses for training samples of SVHN at the last epoch. We plot those on adversarial data $x'$ for the other methods. Blue bins correspond to the data points that models correctly classify.
Figure 12: Histogram of logit margin losses for training samples of SVHN at the 100-th epoch. We plot those on adversarial data \( x' \) for the other methods. Blue bins correspond to the data points that models correctly classify.

Figure 13: Average of gradient norms with respect to data points. \( \bar{k} \) is randomly selected labels, and \( k^* = \arg\max_{k \neq y} z_k(x) \).
Figure 14: Robust accuracy against Auto-Attack and logit margin losses $\ell_{LM}(x')$ on CIFAR10 with RN18 for SOVR when tuning $(M, \lambda)$. Dashed gray line corresponds to the results of AT using cross-entropy loss.

Figure 15: Robust accuracy against PGD-20 at the last epoch on CIFAR10 with RN18 for SOVR when tuning $(M, \lambda)$.

D.2 Effects of hyperparameters $(M, \lambda)$

SOVR has hyperparameters of $M$ and $\lambda$. In this section, we evaluate the effects of these hyperparameters. Figure 14 plots the robust accuracy against Auto-Attack and $\ell_{LM}$ on CIFAR10 with RN18. We fixed $\lambda$ for tuning $M$ and fixed $M$ for tuning $\lambda$. Note that, while $M = 0$ corresponds to the naive adversarial training (AT), $\lambda = 0$ corresponds to that models are trained on only a set of $S$, i.e., AT only using the half of data points in minibatch when $M = 50$. First, $\ell_{LM}(x')$ is monotonically decreasing by increases in both $M$ and $\lambda$ (Fig. 14(b) and Fig. 14(d)). This indicates that SOVR increases the logit margins $|\ell_{LM}|$ compared with cross-entropy. However, robust accuracies against Auto-Attack are not monotonically increasing against $(M, \lambda)$. To investigate the causes, we additionally evaluate robust accuracies against PGD-20 at the last epoch in Fig. 15. This figure shows that the gap between train accuracy and test accuracy increases when $\lambda$ increasing. Although it is unclear why the increase in $M$ hurts the robust accuracy, SOVR always outperforms AT in terms of robustness against Auto-Attack under all tested values of $(0 < M \leq 100, 0 < \lambda \leq 1.25)$.

D.3 Individually test of Auto-Attack

For importance aware methods, we evaluate the robust accuracies against all components of Auto-Attack in Section 2. In this section, we additionally evaluate SOVR and EW AT by individually using Auto-Attack. Figure 16 plots the results, and SOVR is more robust against t-APG, t-FAB, SQUARE than AT unlike importance aware methods. Although the robust accuracy of SOVR against PGD-20
Table 4: Robust Accuracy against various attacks ($L_\infty, \varepsilon = 8/255$). CLN denotes accuracy on clean data, and AA denotes Auto-Attack. Worst represent the least robust accuracy among attacks in the table for each method.

| Method   | CLN   | FGSM   | PGD    | CW     | SPSA   | AA     | Worst   |
|----------|-------|--------|--------|--------|--------|--------|---------|
| AT       | 81.6±0.5 | 57.6±0.1 | 52.5±0.4 | 50.0±0.4 | 56.8±0.2 | 48.0±0.2 | 48.0±0.2 |
| MART     | 78.3±1  | 58.0±0.3 | 54.0±0.1 | 48.7±0.2 | 54.2±0.1 | 46.9±0.3 | 46.9±0.3 |
| MMA      | **85.5 ± 0.7** | **65.5 ± 2** | 51.6±0.2 | 51.0±0.6 | 56.3±1.2 | 37.2±0.9 | 37.2±0.9 |
| **C10**  |        |        |        |        |        |        |         |
| GAIRAT   | 78.7±0.7 | 63.1±0.7 | **62.0 ± 0.4** | 40.0±1.1 | 47.4±1.3 | 37.7±1.1 | 37.7±1.1 |
| MAIL     | 79.5±0.4 | 57.8±0.1 | 54.9±0.08 | 42.1±0.2 | 49.1±0.4 | 39.6±0.4 | 39.6±0.4 |
| EWAT     | 82.8±0.4 | 57.7±0.4 | 52.3±0.4 | 50.4±0.7 | 56.8±0.2 | 48.2±0.7 | 48.2±0.7 |
| SOVR     | 80.6±0.1 | 56.5±0.3 | 50.6±0.1 | **51.4 ± 0.1** | **57.2 ± 0.2** | **49.4 ± 0.1** | **49.4 ± 0.1** |
| AT       | 85.6±0.5 | 60.9±0.4 | 55.1±0.4 | 52.1±0.3 | 57.8±0.6 | 50.4±0.09 | 50.4±0.09 |
| MART     | 81.5±1  | 61.3±0.6 | 57.2±0.2 | 52.1±0.3 | 57.8±0.6 | 50.4±0.09 | 50.4±0.09 |
| MMA      | **87.8 ± 1** | **68.6 ± 1** | 55.7±1 | **55.4 ± 0.7** | 59.6±2 | 43.1±0.6 | 43.1±0.6 |
| **C10**  |        |        |        |        |        |        |         |
| GAIRAT   | 83.0±0.7 | 64.1±0.5 | **62.9 ± 0.4** | 44.4±0.7 | 52.1±0.5 | 41.8±0.6 | 41.8±0.6 |
| MAIL     | 82.2±0.4 | 59.3±0.5 | 56.0±0.5 | 45.7±0.2 | 53.0±0.2 | 43.3±0.1 | 43.3±0.1 |
| EWAT     | 86.0±0.5 | 60.6±0.4 | 54.5±0.1 | 53.8±0.3 | 60.7±0.4 | 51.6±0.3 | 51.6±0.3 |
| SOVR     | 85.4±1  | 61.4±0.5 | 54.7±0.2 | 54.9±0.3 | **61.7 ± 0.1** | **52.7 ± 0.2** | **52.7 ± 0.2** |
| AT       | 89.8±0.6 | 30.1±0.4 | 27.7±0.2 | 25.6±0.3 | 29.3±0.3 | 23.7±0.3 | 23.7±0.3 |
| MART     | 86.9±0.6 | 31.0±0.2 | **29.36 ± 0.06** | 25.0±3 | 28.5±0.1 | 23.9±0.3 | 23.9±0.3 |
| MMA      | **93.9 ± 0.4** | 25.7±0.3 | 19.4±0.2 | 20.5±0.1 | 24.3±0.3 | 18.4±0.2 | 18.4±0.2 |
| **C100** |        |        |        |        |        |        |         |
| GAIRAT   | 89.9±0.4 | 27.9±0.3 | 26.0±0.2 | 21.9±0.4 | 25.9±0.1 | 19.8±0.5 | 19.8±0.5 |
| MAIL     | 89.4±0.4 | 24.8±0.08 | 23.9±0.26 | 18.3±0.5 | 21.9±0.5 | 16.7±0.3 | 16.7±0.3 |
| EWAT     | 90.2±0.6 | 30.0±0.08 | 27.4±0.3 | 25.3±0.2 | 29.3±0.1 | 23.5±0.06 | 23.5±0.06 |
| SOVR     | 90.0±1  | 30.2±0.2 | 27.4±0.2 | **26.1 ± 0.1** | **29.9 ± 0.1** | **24.3 ± 0.2** | **24.3 ± 0.2** |
| AT       | 53.0±0.7 | 61.8±0.5 | 50.6±0.4 | 47.7±0.8 | 55.7±0.9 | 45.6±0.4 | 45.6±0.4 |
| MART     | 49.2±0.1 | 64.4±0.5 | 56.5±0.2 | 49.0±0.4 | 56.6±0.08 | 46.9±0.3 | 46.9±0.3 |
| MMA      | **60.6 ± 0.6** | **79.6 ± 0.8** | **65.0 ± 0.1** | **59.1 ± 2** | **63.2 ± 2** | **37.2±0.9** | **37.2±0.9** |
| **SVHN** |        |        |        |        |        |        |         |
| GAIRAT   | 52.0±0.5 | 65.8±0.4 | 60.4±0.6 | 40.6±0.7 | 48.8±0.6 | 37.6±0.6 | 37.6±0.6 |
| MAIL     | 46.5±0.5 | 64.6±0.4 | 58.2±0.3 | 44.1±0.7 | 52.3±0.5 | 41.2±0.3 | 41.2±0.3 |
| EWAT     | 54.2±1  | 61.8±0.5 | 51.7±0.2 | 50.2±0.2 | 57.4±0.4 | 47.6±0.4 | 47.6±0.4 |
| SOVR     | 52.1±0.8 | 65.3±2 | 50.7±0.2 | 52.5±0.2 | 60.7±0.4 | **48.5 ± 0.4** | **48.5 ± 0.4** |

is lower than AT. SOVR outperforms other methods in terms of the robustness against the worst-case attacks, which is the goal of this study.

### D.4 Evaluation using various attacks

We list robust accuracies against various attacks; FGSM [12], 100-step PGD [22], 100-step PGD with CW loss [22, 4], SPSA [50] in Tab. 4. Hyperparameters of SPSA are as follows: the number of steps is set to 100, the perturbation size is set to $0.001$, learning rate is set to $0.01$, and the number of samples for each gradient estimation is set to $256$. In this table, we repeat the clean accuracies and robust accuracies against Auto-Attack from the table in the main paper. In addition, we list the worst robust accuracies, which are the least robust accuracy among attacks in the table for each method. In this tables, importance aware methods tend to fail to improve the robustness against SPSA. Since SPSA does not directly use gradients, this result indicates that importance aware methods improves the robustness by obfuscating gradients [2]. Against some attacks, MMA achieves the highest robust accuracy on several datasets. However, our goal is improving the true robustness, i.e., robust accuracies against the worst case attacks in $\delta \in \{||\delta||_\infty \leq 8/255\}$. MMA fails to improve the robustness against the worst-case attacks (the columns of Worst). We can see that Auto-Attack always achieves the least robust accuracies, and SOVR improves them: Robust accuracies against Auto-Attack of SOVR are 5.9-12.2 percent point greater than those of MMA.