Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Robust shortcut and disordered robustness: Improving adversarial training through adaptive smoothing

Lin Li [a],[*], Michael Spratling [a,b]

[a] Department of Informatics, King's College London, WC2B 4BG, London, UK
[b] Department of Behavioral and Cognitive Sciences, University of, Luxembourg, L-4366, Esch-Belval, Luxembourg

ARTICLE INFO

Keywords:
Adversarial robustness
Adversarial training
Loss smoothing
Instance adaptive

ABSTRACT

Deep neural networks are highly susceptible to adversarial perturbations: artificial noise that corrupts input data in ways imperceptible to humans but causes incorrect predictions. Among the various defenses against these attacks, adversarial training has emerged as the most effective. In this work, we aim to enhance adversarial training to improve robustness against adversarial attacks. We begin by analyzing how adversarial vulnerability evolves during training from an instance-wise perspective. This analysis reveals two previously unrecognized phenomena: *robust shortcut* and *disordered robustness*. We then demonstrate that these phenomena are related to *robust overfitting*, a well-known issue in adversarial training. Building on these insights, we propose a novel adversarial training method: Instance-adaptive Smoothness Enhanced Adversarial Training (ISEAT). This method jointly smooths the input and weight loss landscapes in an instance-adaptive manner, preventing the exploitation of robust shortcut and thereby mitigating robust overfitting. Extensive experiments demonstrate the efficacy of ISEAT and its superiority over existing adversarial training methods. Code is available at https://github.com/TreeLLi/ISEAT.

1. Introduction

Deep neural networks (DNNs) are well known to be very vulnerable to adversarial attacks [1]. Adversarial attacks modify (or "perturb") natural images (clean examples) to craft adversarial examples in such a way as to fool the target network to make predictions that are dramatically different from those for the corresponding clean examples even when the class of the perturbed input appears unchanged to a human observer. This raises severe security concerns for DNNs, especially as more and more real-world applications are dependent upon such models.

To date, adversarial training (AT) has been the most effective defense against adversarial attacks [2]. It is typically formulated as a min–max problem:

$$\arg\min_{\theta} \mathbb{E}[\arg\max_{\delta} \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta})], \tag{1}$$

where the inner maximization searches for the strongest adversarial perturbation δ and the outer optimization searches for the model parameters $\boldsymbol{\theta}$ to minimize the loss on the generated adversarial examples.

One particular limit of vanilla AT [3] is that all samples in the data set are treated equally during training, neglecting individual differences between samples. Several previous works have made improvements to AT by customizing regularization in an instance-adaptive way. Regularization here can be implemented either by modifying the method used to generate the training-time adversarial sample or by modifying the strength of regularization applied to the loss function. Instance-adaptive techniques often modify the strength of the attack used for individual training samples by scaling the magnitude of the perturbations found by the attack, or by changing the perturbation budget used by the attack, or by changing the number of steps used by the attack. For example, one popular strategy is to enhance the strength of the training-time adversarial attack for hard-to-attack (adversarially benign) samples and/or to diminish the strength of the attack at those easy-to-attack samples [4–8]. Other strategies are discussed in detail in Section 2. We extend this line of works to improve AT, but propose a different strategy to distinguish instances (that particularly contrasts with the aforementioned popular method), and propose a new regularizer.

The proposed approach is motivated by our identification of two novel issues in AT: *robust shortcut* and *disordered robustness*. We begin by theoretically analyzing the existence of the robust shortcut, which occurs when the reduction in overall adversarial loss during AT is achieved by allowing some gradients to increase in order to decrease others more significantly, leading to an uneven distribution of robustness across the data. This issue is prevalent across various AT methods

[*] Corresponding author.
E-mail addresses: lin.3.li@kcl.ac.uk (L. Li), michael.spratling@kcl.ac.uk (M. Spratling).

https://doi.org/10.1016/j.patcog.2025.111474
Received 1 September 2024; Received in revised form 23 December 2024; Accepted 15 February 2025
Available online 27 February 2025

and is related to overfitting in AT [9]. Furthermore, a large proportion of training samples with low vulnerability exhibit excessively large margins (as defined in Eq. (6)) along the adversarial direction. For instance, even when perturbed with a magnitude approximately 30 times the perturbation budget, these samples can still be correctly classified. Such samples are described as having disordered robustness. These insights suggest that training data should be treated unequally. Specifically, the vulnerable samples that are sacrificed should be regularized to prevent AT from exploiting the robust shortcut, thereby mitigating robust overfitting.

To achieve this, we propose a new approach called Instance-adaptive Smoothness Enhanced Adversarial Training (ISEAT). This method jointly enhances both input and weight loss landscape smoothness in an instance-adaptive manner. In particular, it extends vanilla AT by enforcing logit stability against both adversarial input and weight perturbations, with the strength of regularization for each instance adapting to its adversarial vulnerability. Extensive experiments were conducted to evaluate the performance of the proposed method across various datasets and models. ISEAT significantly improves upon the baseline AT method and outperforms previous related methods. Additionally, a detailed ablation study was conducted to elucidate the mechanism behind the effectiveness of the proposed approach.

2. Related works

Adversarial training. AT is usually categorized as single-step and multi-step AT according to the number of iterations used for crafting training adversarial examples. The common single-step and multi-step adversaries are Fast Gradient Sign Method (FGSM) [1] and Projected Gradient Descent (PGD) [3]. FGSM uses the sign of the loss gradients w.r.t. the input as the adversarial direction. PGD can be considered as an iterative version of FGSM where the perturbation is projected back onto the ℓ_∞-norm ϵ-ball at the end of each iteration. AT is prone to overfitting, which is known as robust overfitting [9]. Specifically, test adversarial robustness degenerates while training adversarial loss declines during the later stage of training. Robust overfitting significantly impairs the performance of AT.

Loss smoothing. It has been shown that adversarial robustness is related to the smoothness of the model's loss landscape w.r.t. the input [10,11] and the model weights [12]. Therefore, we summarize existing methods for adversarial robustness in two categories: input loss smoothing and weight loss smoothing. For input loss smoothing, one approach is explicitly regularizing the logits [13] or the gradients [11] of each training sample to be the same as any of their neighbor samples within a certain distance. Besides, AT can be concerned as an implicit input loss smoothness regularizer and the strength of regularization is controlled by the direction and the size of perturbation [13]. Regarding weight loss smoothing, Adversarial Weight Perturbation (AWP) [12] injects adversarial perturbation into model weights to implicitly smooth weight loss. RWP [14] found that applying adversarial weight perturbation to only small-loss samples leads to an improved robustness compared to AWP. Alternatively, Stochastic Weight Averaging (SWA) [15] smooths weight by exponentially averaging checkpoints along the training trajectory. Our regularizer combines logit regularization and AWP together to jointly smooth both input and weight loss in a more effective way.

Instance-adaptive adversarial training. Many strategies have been proposed to improve AT by regularizing instances unequally. One popular strategy is to adapt the size of the adversarial input perturbation, and so the strength of regularization, to the difficulty of crafting successful adversarial examples. Typically, large perturbations are assigned to hard-to-attack samples in order to produce successful adversarial examples for more effective AT. In tandem, small perturbations are assigned to easy-to-attack samples in order to alleviate over-regularization for a better trade-off between accuracy and robustness. The size of perturbation can be controlled by the number of steps [6], the perturbation budget [4,7,8] or an extra scaling factor [5]. Furthermore, this strategy has been also applied to weight loss smoothing in RWP [14]. We argue that this strategy contradicts our finding that hard-to-attack (low-vulnerability in our terms) samples have already been over-regularized so their regularization strength should not be further enlarged.

The most similar methods to ours are MART [16] and GAIRAT [17]. MART regularizes KL-divergence between the logit of clean and corresponding adversarial examples, weighted by one minus the prediction confidence on clean examples. Hence, instances with lower clean prediction confidence receive stronger regularization. GAIRAT directly weights the adversarial loss of instances based on their geometric distance to the decision bound, which is measured by the number of iterations required for a successful attack. Instances closer to decision bound (less iterations) are weighted more. Although there is a connection among prediction confidence, geometric distance and adversarial vulnerability (ours), they are essentially different metrics and the weight schemes based on them thus perform differently. Regarding GAIRAT, the computation of geometric distance deeply depends on the training adversary, and hence, severely limits its application, e.g., it cannot be applied to single-step AT without using an extra multi-step adversary. Another similar work is InfoAT [18] which like the proposed method uses logit stability regularization, but weights regularization according to the mutual information of clean examples.

In contrast to the manually crafted strategies described above, LAS-AT [19] customizes adversarial attack automatically for each instance in a generative adversarial style. The parameters of the attacker, such as the perturbation budget for each instance, are generated on-the-fly by a separate strategy network, which is jointly trained alongside the classification network to maximize adversarial loss, i.e., produce stronger adversarial examples. This approach is more complicated than the alternatives, including ours, and potentially suffers from similar instability issues to GANs [20].

Data augmentation for adversarial training. AT can benefit from using more training data to enhance robust generalization. This was theoretically proved for simplified settings like Gaussian models [21]. For complicated but realistic settings, training with extra data from either real [22], or synthetic [23], sources dramatically boosts the robust performance of AT and thus becomes a necessary ingredient for achieving state-of-the-art results. However, extra data is usually very expensive or even infeasible to acquire in many tasks, so [24] proposes a new data augmentation method, IDBH, specifically designed for AT. Our method is complementary to using extra data and data augmentation and a further boost in robustness is observed when they are combined together (see Section 6.1).

Instance weighting in other scenarios. Instance weighting has also been explored in backdoor attacks [25,26] to improve the efficiency of poisoning. Similar to our approach, these studies assume that different data samples contribute unequally to training and aim to enhance attack efficiency through optimized sample selection. This suggests that our proposed analysis and instance weighting scheme could potentially be applied in this context. For example, a straightforward idea would be to use our AV metric to select poisoned samples and evaluate whether this improves sample efficiency compared to baseline methods. However, as backdoor attacks are not the focus of this work and these two research areas have traditionally been studied separately, we leave this exploration for future research.

3. Robust shortcut: A shortcut to robust optimization

This section identifies a shortcut in AT that minimizes overall adversarial loss by sacrificing the loss for some data to be large to enable the loss for other samples to be reduced. We refer to this phenomenon as the "*robust shortcut*". We begin with a theoretical analysis of adversarial vulnerability and the robust shortcut in Section 3.1. Next, in Section 3.2, we demonstrate that, in practice, AT tends to converge to this problematic shortcut. In Section 3.3 we provide evidence that the robust shortcut is associated with robust overfitting, a well-known issue in AT. Finally, Section 3.4 demonstrates that the robust shortcut is prevalent across various advanced AT methods.

The following notations are adopted. $\boldsymbol{x} \in \mathbb{R}^d$ is a d-dimensional sample whose ground truth label is y. $\boldsymbol{x}_i$ refers to ith sample in dataset D and $x_{i,j}$ refers to the jth dimension of $\boldsymbol{x}_i$. Similarly, x_j refers to the jth dimension of an arbitrary sample $\boldsymbol{x}$. D is split into two subsets, D_t and D_e, for training and testing respectively. $\boldsymbol{\delta}_i \in \mathcal{B}(\boldsymbol{x}_i, \epsilon)$ is the adversarial perturbation applied to $\boldsymbol{x}_i$ to fool the network. $\mathcal{B}(\boldsymbol{x}, \epsilon)$ denotes the ϵ-ball around the example $\boldsymbol{x}$ defined for a specific distance measure like ℓ_∞-norm. $\delta_{i,j}$ is the perturbation applied along the dimension $x_{i,j}$, and δ_j is the perturbation applied along the jth dimension of an arbitrary sample $\boldsymbol{x}$. The network is parameterized by $\boldsymbol{\theta}$. The output of the network before the final softmax layer (i.e., the logits) is $f(\boldsymbol{x}; \boldsymbol{\theta})$. The class predicted by the network, i.e., the class associated with the highest logit value, is $F(\boldsymbol{x}; \boldsymbol{\theta})$. The predictive loss is $\mathcal{L}(\boldsymbol{x}, y; \theta)$ or $\mathcal{L}(\boldsymbol{x})$ for short. The size of a training batch is m and the indexes of the samples in a batch is M.

The preliminary experiments in the remainder of this section were conducted using the default settings with CIFAR10 as defined in Section 6 unless specified otherwise. The model architecture was PreAct ResNet18 [27]. Robustness was evaluated against PGD50.

3.1. Adversarial vulnerability and robust shortcut

Adversarial loss can be approximated by the sum of clean loss and adversarial vulnerability [13]:

$$\mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}) \approx \mathcal{L}(\boldsymbol{x}) + \sum_{i}^{d} \nabla_{x_i} \mathcal{L}(\boldsymbol{x}) \delta_i + \frac{1}{2} \sum_{i,j}^{d} \nabla^2_{x_i x_j} \mathcal{L}(\boldsymbol{x}) \delta_i \delta_j, \tag{2}$$

where $\nabla_{x_i} \mathcal{L}(\boldsymbol{x})$, or ∇_{x_i} for short, is the first-order gradient of $\mathcal{L}(\boldsymbol{x})$ w.r.t. the input dimension x_i corresponding to the slope of the input loss landscape. Similarly, $\nabla^2_{x_i x_j} \mathcal{L}(\boldsymbol{x})$, or $\nabla^2_{x_i x_j}$ for short, denotes the second-order gradient w.r.t. x_i and x_j corresponding to the curvature of the loss landscape.

The adversarial vulnerability (AV) of $\boldsymbol{x}$ against a particular attack is defined as the loss increment caused by the attack:

$$\text{AV} = \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}) - \mathcal{L}(\boldsymbol{x}). \tag{3}$$

A higher vulnerability means that adversarial attack has a greater impact on the loss value for that sample, and hence, is more likely to corrupt the model's prediction for that sample. From this perspective, loss gradients in Eq. (2) constitute the source of AV. Adversarial attacks exploit input loss gradients to enlarge adversarial loss by aligning the sign of the perturbation and corresponding gradient, i.e., $sign(\delta_i) = sign(x_i)$. Gradients with small magnitude (a flat loss landscape) therefore indicate low AV.

AT improves adversarial robustness by shrinking the magnitude of gradients. Supposing that the training adversary is theoretically optimal [13], i.e., $sign(\nabla_{x_{i,j}}) = sign(\delta_{i,j})$ $\forall i, j$ and $\left|\delta_{i,j}\right| = \epsilon$ $\forall i, j$, adversarial loss can be written (second-order gradients are ignored for simplicity) as:

$$\arg\min_{\theta} \; \mathcal{L}(\boldsymbol{x}) + \epsilon \sum_{i}^{D} \sum_{j}^{d} \left|\nabla_{x_{i,j}}\right|. \tag{4}$$

Note $|\cdot|$ denotes the absolute value. Ideally, the magnitude of every gradient is supposed to be shrunk by AT with the decrease of training loss.

However, Eq. (4) can also be satisfied by sacrificing some gradients to be large in order to shrink other gradients or the clean loss: as long as the total reduction is greater than the enlargement. If some gradients are consistently enlarged, the model may eventually converge to yield an uneven distribution of AV among training data.

It is even more likely for the model to converge to such an uneven state if the training adversary is weaker than the theoretical optimum. A sub-optimal adversary causes misalignment between the sign of the perturbation and the corresponding gradient, i.e., $sign(\nabla_{x_{i,j}}) \neq sign(\delta_{i,j})$, in which case training encourages the misaligned gradients to grow larger. If some gradients are consistently misaligned by their corresponding perturbations, they accumulate to be large.

3.2. Empirical evidence: Uneven distribution of adversarial vulnerability

This section demonstrates that, in practice, AT is prone to exploiting the robust shortcut. To illustrate this, we track the model's AV across different data instances throughout the training process, analyzing how AV is distributed among the data. This approach is based on the insights from the previous section, which suggest that optimizing through the robust shortcut results in an uneven distribution of AV across the data.

We first introduce the following metrics to quantitatively assess this phenomenon. AV SD measures the degree of unevenness for AV via its standard deviation (SD) over the entire training set:

$$\text{AV SD} = \sqrt{\mathbb{E}_{i \in D_t}[(\text{AV}_i - \mathbb{E}_{j \in D_t}[\text{AV}_j])^2]}. \tag{5}$$

Higher AV SD indicates that AV is spread out more among instances, i.e., higher unevenness. In addition, the mean AV is computed for the 10% of samples with the lowest AV ("bottom 10%") and the 10% of samples with the highest AV ("top 10%"). Besides, the percentage of samples with AV ≥ 1 and ≤ 0 within the whole training set is calculated to complement the above measures. 1 and 0 are selected as the thresholds for high and low AV respectively as the model's prediction, after adversarial attack, is observed to be significantly affected if AV ≥ 1 and hardly changed if AV ≤ 0.

As shown in Fig. 1(a), AV becomes increasingly unevenly distributed across training samples: some become highly susceptible to attacks, while others remain exceptionally robust. Over the course of training, both the standard deviation of AV and the number of high- and low-vulnerability samples increase. Additionally, the average AV of the top 10% rises, while the average AV of the bottom 10% decreases to even below 0.

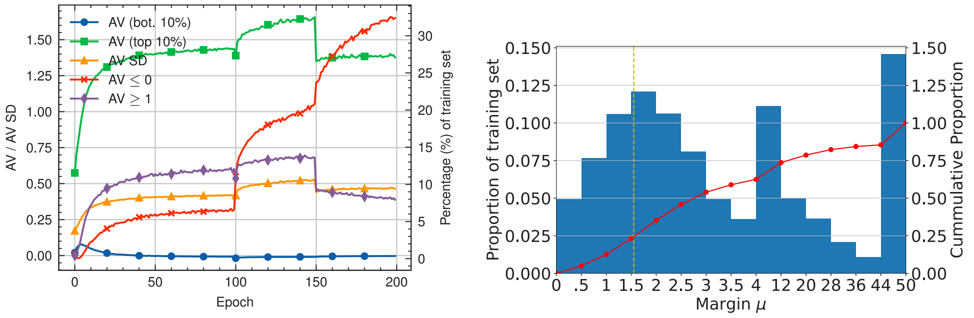
To further illustrate this issue, we select the checkpoint with highest robustness found in the above experiment and visualize the distribution of margin μ over the entire training set. The margin (μ) along the adversarial direction for each sample in the training data was computed from the adversarially-trained model using the method defined in Rade and Moosavi-Dezfooli [28]:

$$\arg\min_{\mu} \;\text{s.t.}\; F\left(\boldsymbol{x} + \mu \frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|_2}; \boldsymbol{\theta}\right) \neq F(\boldsymbol{x}; \boldsymbol{\theta}), \tag{6}$$

where $\boldsymbol{\delta}$ is computed using PGD50 and $\|\cdot\|_2$ is the ℓ_2-norm.

As shown in Fig. 1(b), about 20% of training data can be successfully attacked within the ϵ-ball to fool the model into changing prediction, and among them, around 5% of training instances can be successfully attacked using only a third of the perturbation budget ϵ. In contrast, a large proportion of samples exhibit an excessive margin along the adversarial direction. The prediction of the model remains constant under an attack with double the perturbation budget for about half the training data. More surprisingly, about 14% of the training samples exhibit the theoretically maximal effective margin ($\mu = 50$),[1] which indicates that no perturbation along the adversarial direction can change the model's prediction.

[1] The margin value corresponding to the perturbation budget ϵ is about 1.5. A margin value of 50 is, hence, equivalent to perturbing along the adversarial direction by approximately $\frac{50}{1.5}\epsilon$ which is greater than 1. For this work, input images are normalized to have pixel values between zero and one, and the perturbed input is clipped to remain in this range, so increasing the magnitude of any perturbation beyond a value of one will have no additional effects on the input.



(a) (b)

Fig. 1. Illustration of the phenomenon robust shortcut on CIFAR10. Fig. 1(a) shows how instance-wise AV changes over the course of training. Fig. 1(b) shows the distribution of margin μ (Eq. (6)) over the entire training set. The yellow dashed line indicates the margin corresponding to the training perturbation budget ϵ. The red solid line represents cumulative distribution.

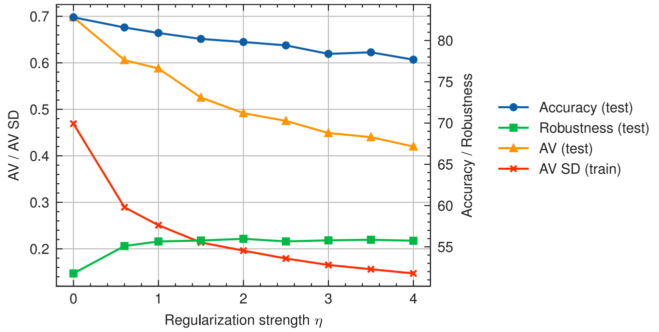


Fig. 2. Performance and AV of the models trained by Eq. (7) with different strength, η, on CIFAR10. The data set on which the metric is measured is indicated in the bracket in the legend.

3.3. Relating robust shortcut to robust overfitting

We hypothesize that the robust shortcut accounts for overfitting in AT. To evaluate this claim we test how robust generalization varies with the unevenness of AV. Unevenness is controlled by the strength, η, of a logit stability regularization applied to the 10% of samples with the highest AV:

$$\mathbb{E}_{i\in M}\left[\mathcal{L}(\boldsymbol{x}_i+\boldsymbol{\delta}_i)+\eta\|f(\boldsymbol{x}_i+\boldsymbol{\delta}_i)-f(\boldsymbol{x}_i)\|_2^2\mathbb{1}(\boldsymbol{x}_i,\boldsymbol{\delta}_i)\right], \tag{7}$$

where $\mathbb{1}(\cdot)$ is an indicator function to select the samples with highest AV within each training batch. $\|\cdot\|_2^2$ is the squared ℓ_2-norm. Unevenness is supposed to be reduced as η increases since those highly vulnerable samples are regularized to be more robust. This regularizer is used to fine-tune a pre-adversarially-trained model for 10 epochs with an initial learning rate of 0.01 decayed by 0.1 at half epochs. Experiments are preformed using different values for η from 0 to 4 with a step size of 0.5. Note that $\eta = 0$ means no regularization is applied.

As shown in Fig. 2, the AV SD on the training data drops with increasing regularization strength. The AV of the test data decreases even more dramatically, and test robustness increases, indicating improved robust generalization with stronger regularization. The increase of test robustness is less significant than the decrease of test AV because clean accuracy degrades with stronger regularization. Overall, this result verifies that the robust shortcut is related to robust overfitting.

3.4. Prevalence of robust shortcut

Finally, it is shown that the issue of robust shortcut is prevalent across various robust training schemes. As shown in Table 1, RST and AWP both mitigate the unevenness of AV to some extent with a reduced top 10% average AV and reduced number of high-vulnerability samples compared to vanilla AT. However, the reduction of unevenness produced by RST and AWP is less than that achieved by the purpose-built regularization (Eq. (7)). This suggests that these previous methods of improving AT contribute to enhanced adversarial robustness using different mechanism to the proposed one and a higher robustness can be expected if they are properly combined together, as described next.

Table 1
Robustness and the statistics of AV for various robust training schemes on CIFAR10.

Method	Rob. (%)	Adversarial vulnerability				
		AV SD	Top 10%	Bot. 10%	≥1	≤0
AT	51.78	0.467	1.527	−0.010	12.38	12.14
AWP	54.68	0.351	1.120	−0.022	5.52	19.74
RST	**57.68**	0.443	1.378	−0.011	9.96	20.87
Eq. (7)	55.95	**0.196**	**0.633**	**−0.047**	**0.14**	**12.00**

4. Disordered robustness: Excessive robustness to visible perturbation

This section introduces an issue in AT referred to as disordered robustness. As observed in Section 3.2, a significant proportion of training samples exhibit disordered robustness because the model remains insensitive to dramatic perturbations applied to them, perturbations that should significantly affect the model's output. This property of excessively large margins is termed disordered robustness because a well-calibrated model should be sensitive to noticeable, class altering, perturbations in the input. Fig. 3 visualizes examples of perturbed inputs for samples with disordered robustness.

To further verify the claim, the loss landscapes around some representative samples are visualized in Fig. 4. The loss was calculated using:

$$L=\mathcal{L}\left(\boldsymbol{x}+\alpha\frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|_2}+\beta\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|_2}, y;\boldsymbol{\theta}\right), \tag{8}$$

where $\boldsymbol{\delta}$ was generated by PGD50 and $\boldsymbol{u}$ was randomly sampled from a uniform distribution $\mathcal{U}(-\epsilon,\epsilon)^d$. The loss landscape was visualized along the adversarial and the random direction by varying α and β respectively.

For samples with disordered robustness, lower loss values are produced for values of $\alpha > 0$ than are produced when $\alpha = 0$ (see Figs. 4(a) and 4(b)). This confirms that this particular kind of robustness is disordered because adversarial examples are more benign, i.e., easier to correctly classify, than clean examples in this case.

Disordered robustness differs from a similar phenomenon observed in the previous work [28]. First, the direction along which excessive margin is observed is different. Rade and Moosavi-Dezfooli [28] observed excessive margin for an adversarially-trained model along the adversarial direction found by PGD on a standardly-trained model

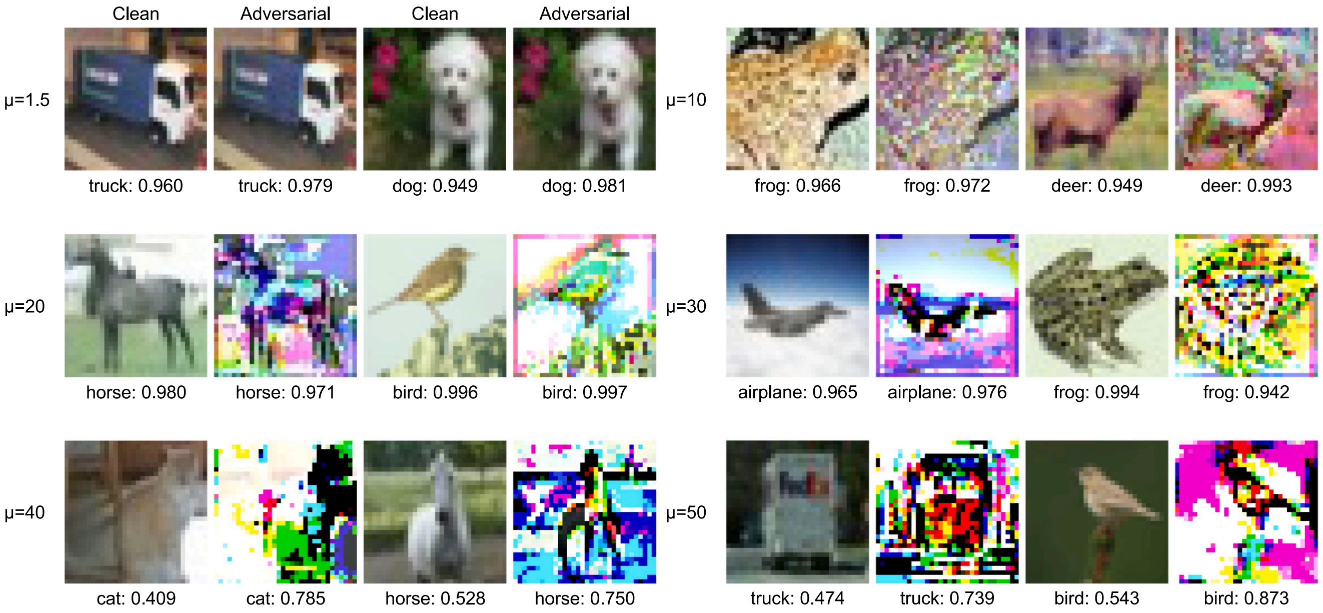
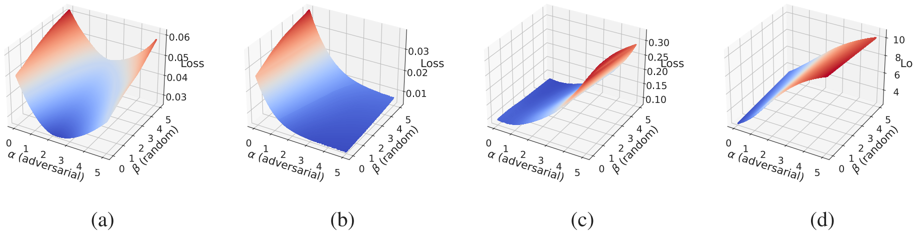


Fig. 3. Visualization of disordered robustness on CIFAR10. This figure shows clean samples and the corresponding adversarial samples for various levels of margin. Adversarial examples in each row were crafted using $x + \mu \frac{\delta}{\|\delta\|_2}$ with the value of μ given at the left of each row. $\mu = 1.5$ approximately corresponds to the perturbation budget ϵ. The caption of each image describes the ground-truth class and the prediction confidence (the output of softmax layer at the index of ground-truth class). Although the images are modified dramatically by adversarial perturbation to even a detrimental degree when $\mu \geq 20$, they can still be correctly classified by the model without much variation on confidence, or even with higher confidence. In the cases with very large μ, the perturbed images become extremely hard to recognize or become meaningless to a human observer, while the model is able to recognize them correctly with high confidence.



(a)

(b)

(c)

(d)

Fig. 4. Visualization of loss landscapes at samples with disordered robustness (Figs. 4(a) and 4(b)), and for a robust sample (Fig. 4(c)) and a vulnerable sample (Fig. 4(d)). $\alpha \approx 1.5$ corresponds to a perturbation size of ϵ. Loss increases up as the color of plane changes from blue to red. Note the scale of loss is dramatically different for these three categories of data. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(i.e. they used different models for the adversarial direction and the margin evaluation), while it is observed in this work to be along the PGD adversarial direction generated for an adversarially-trained model (same model for adversarial direction and margin evaluation). Second, Rade and Moosavi-Dezfooli [28] did not find that the direction along which the excessive margin is observed is adversarially benign.

5. Method

Building on the insights described in the preceding sections, this section proposes a new adversarial training method that prevents the exploitation of robust shortcuts, thereby improving robust generalization. The proposed approach, named Instance-adaptive Smoothness Enhanced Adversarial Training (ISEAT), combines AT with a new robustness regularizer and adapts the regularization strength for each instance based on its AV. In general, stronger regularization is applied to samples with higher vulnerability to enhance their robustness, while low-vulnerability samples receive weaker regularization to avoid the harmful effects of over-regularization. This adaptive regularization scheme is theoretically agnostic to the choice of the underlying regularization method. However, to enhance the performance of our approach, we propose a novel and more effective regularization method in the following section.

5.1. Jointly smoothing input and weight loss surfaces

We propose a new regularization method to enforce prediction Logit Stability against both adversarial Input and Weight perturbation (LSIW) so that the model's predicted logits remains, ideally, constant when the input and the weights are both adversarially perturbed as shown below:

$$\mathbb{E}_{i \in M} \| f(\boldsymbol{x}_i + \boldsymbol{\delta}_i; \boldsymbol{\theta} + \boldsymbol{v}) - f(\boldsymbol{x}_i; \boldsymbol{\theta}) \|_2^2. \tag{9}$$

$\boldsymbol{v}$ is a perturbation within the pre-defined region, $\mathcal{V}$, applied to the model's parameters to maximize adversarial loss (input perturbation $\boldsymbol{\delta}$ is assumed) [12]:

$$\arg\max_{\boldsymbol{v} \in \mathcal{V}} \ \mathbb{E}_{i \in M} [\mathcal{L}(\boldsymbol{x}_i + \boldsymbol{\delta}_i; \boldsymbol{\theta} + \boldsymbol{v})]. \tag{10}$$

Following Wu et al. [12], $\boldsymbol{v}$ can be simply approximated by:

$$\boldsymbol{v} \approx \gamma \frac{\nabla_{\boldsymbol{v}} \mathbb{E}_{i \in M} [\mathcal{L}(\boldsymbol{x}_i + \boldsymbol{\delta}_i; \boldsymbol{\theta} + \boldsymbol{v})]}{\| \nabla_{\boldsymbol{v}} \mathbb{E}_{i \in M} [\mathcal{L}(\boldsymbol{x}_i + \boldsymbol{\delta}_i; \boldsymbol{\theta} + \boldsymbol{v})] \|} \|\boldsymbol{\theta}\|. \tag{11}$$

For more details on the intermediate derivation, please refer to Wu et al. [12].

Why logit regularization? Loss smoothness can be regularized through either logits or gradients. We choose to use the former for two reasons. First, logit regularization is more cost-effective, as it only adds a marginal expense for computing the regularization loss.

Algorithm 1. The pseudo-code for the proposed AT method, ISEAT.

```
for each batch do
    for $i = 1$ to $m$ do
        $\boldsymbol{\delta}_i = \mathrm{PGD}(\boldsymbol{x}_i, y_i)$    ▷ adversarial input perturbation
    end
    $\boldsymbol{v} = \gamma \frac{\nabla_\theta \mathbb{E}_{i\in M}[\mathcal{L}(\boldsymbol{x}_i+\boldsymbol{\delta}_i;\boldsymbol{\theta})]}{\|\nabla_\theta \mathbb{E}_{i\in M}[\mathcal{L}(\boldsymbol{x}_i+\boldsymbol{\delta}_i;\boldsymbol{\theta})]\|}\|\boldsymbol{\theta}\|$    ▷ adversarial weight perturbation
    for $i = 1$ to $m$ do
        $w_i = 1 - r(\mathcal{L}(\boldsymbol{x}_i + \boldsymbol{\delta}_i; \boldsymbol{\theta} + \boldsymbol{v}) - \mathcal{L}(\boldsymbol{x}_i; \boldsymbol{\theta}))/m$    ▷ regularization weight
    end
    for $i = 1$ to $m$ do
        $o_i = \|f(\boldsymbol{x}_i + \boldsymbol{\delta}_i; \boldsymbol{\theta} + \boldsymbol{v}) - f(\boldsymbol{x}_i; \boldsymbol{\theta})\|_2^2$    ▷ input and weight smoothing
    end
    $L = \sum_i^m (\mathcal{L}(\boldsymbol{x}_i + \boldsymbol{\delta}_i; \boldsymbol{\theta} + \boldsymbol{v}) + \lambda \cdot w_i \cdot o_i)/m$
    $\theta = \theta - l\nabla_\theta L$    ▷ update model parameters
end
```

In contrast, gradient regularization requires computationally expensive double-backpropagation. Second, logit regularization has been shown to outperform gradient regularization in terms of robustness enhancement and the trade-off between accuracy and robustness [13].

Why adversarial weight perturbation? We decided to integrate adversarial weight perturbation into our regularization method for the following reasons. First, adversarial weight perturbation is a proven technique for smoothing the weight loss landscape and significantly enhancing adversarial robustness [12,29]. Second, it complements input loss smoothing methods. Our experiments in Section 6.6 show that combining input and weight loss smoothing results in a robustness improvement of 3.04%, surpassing the gains achieved by either method alone.

The novelty of proposed robustness regularizer. The idea of jointly smoothing input and weight loss was explored before in Wu et al. [12], but the implementation of this work is novel and more effective. The previous work combined adversarial weight perturbation with input loss smoothing (named TRADES-AWP in the original work) using the method:

$$\mathrm{KL}(f(\boldsymbol{x}; \boldsymbol{\theta} + \boldsymbol{v}), f(\boldsymbol{x} + \boldsymbol{\delta}; \boldsymbol{\theta} + \boldsymbol{v})). \tag{12}$$

Hence, in contrast to the proposed approach (Eq. (9)), both clean and adversarial examples were computed using the perturbed model, i.e., $\boldsymbol{\theta} + \boldsymbol{v}$ in this previous work (Eq. (12)). We argue that weight perturbation is not fully utilized in this paradigm since the logit variation caused by weight perturbation is not explicitly constrained by the outer Kullback–Leibler (KL) divergence. Theoretically, a stronger regularization can be realized by forcing the predicted logits to be same between clean examples on the unperturbed model and adversarial examples on the perturbed model, as in Eq. (9).

The performance of these two approaches is compared in Section 6.6. The results suggest the superiority of LSIW over TRADES-AWP. Another difference between Eqs. (9) and (12) is the metric used to measure the similarity or distance between two prediction logits. Squared ℓ_2-norm is adopted in our proposed solution due to its superior performance as evaluated in Section 6.6.

5.2. Adapting regularization strength to instance-wise vulnerability

To adapt the strength of above joint regularizer to AV, we first extend the metric of AV defined in Eq. (3) to measure instance-wise vulnerability against adversarial input and weight perturbations:

$$\mathrm{AV}_i = \mathcal{L}(\boldsymbol{x}_i + \boldsymbol{\delta}_i; \boldsymbol{\theta} + \boldsymbol{v}) - \mathcal{L}(\boldsymbol{x}_i; \boldsymbol{\theta}), \tag{13}$$

Instances need to be weighted according to their AV. Regularization strength should depend on the relative order, instead of absolute value, of vulnerability so that the overall strength of regularization remains constant throughout training, even if the overall AV declines at the later stage of training. This is important for balancing the influence of AT and the additional robustifying methods. The regularization weight is generated linearly, based on the ranking of vulnerability within the batch, as follows:

$$w_i = 1 - \frac{r(\mathrm{AV}_i)}{m} \tag{14}$$

where $r(\cdot)$ computes the ranking (indexed from 0 for the highest vulnerability) within the batch. Hence, the weights range from 1 (for the most vulnerable sample) to $\frac{1}{m}$ (for the least vulnerable). This linear scheme was selected due to its simplicity and performance superiority over other options (see Section 6.6 for an empirical comparison with some alternatives).

5.3. Optimization

Finally, ISEAT combines AWP-based AT with the proposed weight scheme and regularization method to get the overall training loss:

$$\mathbb{E}_{i\in M}\left[\mathcal{L}(\boldsymbol{x}_i + \boldsymbol{\delta}_i; \boldsymbol{\theta} + \boldsymbol{v}) + \lambda w_i \|f(\boldsymbol{x}_i + \boldsymbol{\delta}_i; \boldsymbol{\theta} + \boldsymbol{v}) - f(\boldsymbol{x}_i; \boldsymbol{\theta})\|_2^2\right]. \tag{15}$$

There are two hyper-parameters, λ and γ, in ISEAT. λ controls the strength of joint regularizer. γ in Eq. (11) directly controls the strength of adversarial weight perturbation and also implicitly affects the strength of joint regularizer. Algorithm 1 illustrates the training procedures.

In practice, modern machine learning frameworks [30] cannot directly compute the gradients of Eq. (15) w.r.t. $\boldsymbol{\theta}$ in one backward pass on one model because the model used to compute $f(\boldsymbol{x}_i; \boldsymbol{\theta})$ will be altered by adversarial weight perturbation before backpropagation. To derive the update rule for gradient descent, first Eq. (15) is rewritten as a function of two models parameterized by $\boldsymbol{\theta}' = \boldsymbol{\theta} + \boldsymbol{v}$ and $\boldsymbol{\theta}$ separately:

$$L(f(\boldsymbol{x} + \boldsymbol{\delta}; \boldsymbol{\theta}'), f(\boldsymbol{x}; \boldsymbol{\theta})). \tag{16}$$

Next, the Chain rule is applied to separate the gradient of Eq. (16) w.r.t. $\boldsymbol{\theta}$ into the sum of two individual backward passes:

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = \frac{\partial L}{\partial f(\boldsymbol{x} + \boldsymbol{\delta}; \boldsymbol{\theta}')} \frac{\partial f(\boldsymbol{x} + \boldsymbol{\delta}; \boldsymbol{\theta}')}{\partial \boldsymbol{\theta}'} \frac{\partial \boldsymbol{\theta}'}{\partial \boldsymbol{\theta}} + \frac{\partial L}{\partial f(\boldsymbol{x}; \boldsymbol{\theta})} \frac{\partial f(\boldsymbol{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{17}$$

After obtaining the gradients, the model's parameters are updated following the method used for AWP [12] as:

$$\boldsymbol{\theta} \leftarrow (\boldsymbol{\theta} + \boldsymbol{v}) - l \cdot \frac{\partial L}{\partial \boldsymbol{\theta}} - \boldsymbol{v}. \tag{18}$$

l is the learning rate. $\boldsymbol{\theta} + \boldsymbol{v}$ refers to the perturbed model parameters. The weight perturbation $\boldsymbol{v}$ is subtracted so that it is not accumulated in the model parameters. $l \cdot \frac{\partial L}{\partial \theta}$ is one step of parameter update in gradient descent algorithm.

5.4. Efficiency analysis

The computational cost of the proposed method, ISEAT, is mainly composed of three components: AT, adversarial weight perturbation and logit stability regularization. Both AT and adversarial weight perturbation involve an inner maximization process using PGD, so their cost increases linearly with the number of iterations used for the inner optimization. By default, ISEAT uses 10 and 1 iterations, respectively, for determining the input and weight perturbations. This is in accordance with common practice. ISEAT in practice adds one more forward and backward pass for $f(\boldsymbol{x}; \boldsymbol{\theta})$ as required by Eq. (17). The time consumption is assessed empirically in Section 6.7.

6. Empirical results

The experiments in this section were based on the following setup unless otherwise specified. The proposed method is evaluated with model architectures Wide ResNet34-10 (WRN34-10) [31] on dataset CIFAR10 [32] and PreAct ResNet18 (PRN18) [27] on datasets CIFAR100 [32], SVHN [33], and TinyImageNet [34]. Models were trained by stochastic gradient descent for 200 epochs with an initial learning rate 0.1 for CIFAR10/100 and TinyImageNet and 0.01 for SVHN, divided by 10 at 50% and 75% of epochs. The momentum was 0.9, the weight decay was 5e−4 and the batch size was 128. The default data augmentation for CIFAR10/100 and TinyImageNet was horizontal flip (applied at half chance) and random crop (with 4 pixel padding). No data augmentation was applied to SVHN. Experiments were run on Tesla V100, A100 and RTX 3080Ti GPUs. All results reported by this work were averaged over 3 runs.

For AT, ℓ_∞ projected gradient descent attack [3] was used with a perturbation budget, ϵ, of 8/255. The number of steps was 10 and the step size was 2/255 for CIFAR10/100 and TinyImageNet and 1/255 for SVHN. To stabilize the training on SVHN, the perturbation budget, ϵ, was increased from 0 to 8/255 linearly in the first five epochs and then kept constant for the remaining epochs, as suggested by Andriushchenko and Flammarion [35]. Robustness was evaluated against AutoAttack [36] using the implementation of Kim [37]. Note that, following Rice et al. [9], PGD10 robustness was tracked on the test set at the end of each epoch during training to select the checkpoint with the highest PGD10 robustness, i.e., the "best" checkpoint to report robustness.

The results are compared with those of related methods including AWP [12], TRADES [38], InfoAT [18], RWP [14], GAIRAT [17], FAT [6], MART [16] and LAS-AT [19] on CIFAR10. Results for existing methods were copied from the original work or another published source such as RobustBench [39]. They were produced using the same model architecture and the same, or very similar, training settings as used for the proposed method.

The performance of the proposed method was additionally evaluated when combined with the data augmentation method IDBH [24] and with extra data like RST [22] to benchmark state-of-the-art robustness. ISEAT when combined with IDBH, akin to Rebuffi et al. [40], benefited from training longer so the total number of training epochs was increased to 400. Note that AT alone with IDBH (IDBH+AT) degenerated as the length of training was increased, so its performance was reported with the default settings. For experiments with extra data, WideResNet28-10 was used instead of WideResNet34-10 to align with experimental protocols used in related works for a fair comparison. This work adopted the same extra data as Carmon et al. [22], i.e., 500K unlabled data from dataset 80 Million TinyImages (80M-TI) with pseudo-labels.[2] As in Carmon et al. [22], extra data was included in the ratio 1:1 with the CIFAR10 data in each training mini-batch, so the effective batch size became 256.

The hyper-parameters of ISEAT were optimized using grid search. The optimal values found were: $\lambda = 0.1$ and $\gamma = 0.007$ for CIFAR10; $\lambda = 0.1$ and $\gamma = 0.005$ for CIFAR10 with IDBH; $\lambda = 0.01$ and $\gamma = 0.005$ for CIFAR10 with extra data; $\lambda = 0.1$ and $\gamma = 0.005$ for CIFAR100; $\lambda = 0.1$ and $\gamma = 0.009$ for SVHN; $\lambda = 0.01$ and $\gamma = 0.005$ for TinyImageNet. It was observed that jointly smoothing input and weight loss with a large learning rate (0.1 in this case) degraded both accuracy and robustness due to over-regularization. Therefore, a warm-up strategy was used for λ on CIFAR10/100 and Ting ImageNet: λ was set to 0 during the initial epochs when the learning rate was large, and λ was set to the optimal value after the first decay of the learning rate. Note that this strategy was not applied to the experiments with SVHN because the initial learning rate on SVHN was already small.

Table 2
Performance of our method and related methods on CIFAR10. Results above the double line are for WRN34-10 without extra data and results below the double line are for WRN28-10 with extra data. The best result is highlighted for each metric in each block. The standard deviation is indicated by the value after the ± sign if evaluated by this work or reported in the original work, otherwise omitted from the table.

Method	Model	Extra data	Accuracy (%)	Robustness (%)
AT	WRN34-10	–	85.90 ± 0.57	53.42 ± 0.59
AT-AWP		–	85.57 ± 0.40	54.04 ± 0.40
TRADES		–	85.72	53.40
InfoAT		–	85.62	52.86
GAIRAT		–	86.30	40.30
FAT		–	**87.97**	47.48
RWP		–	86.86 ± 0.51	54.61 ± 0.11
MART		–	84.17 ± 0.40	51.10 ± 0.40
MART-AWP		–	84.43 ± 0.40	54.23 ± 0.40
LAS-AT		–	86.23	53.58
LAS-AWP		–	87.74	55.52
ISEAT (ours)		–	86.02 ± 0.36	56.54 ± 0.36
ISEAT (ours)+SWA		–	85.95 ± 0.09	**57.09** ± 0.13
IDBH+AT		–	87.03 ± 1.58	54.16 ± 0.70
IDBH+ISEAT (ours)		–	**88.50** ± 0.11	**59.32** ± 0.08
RST	WRN28-10	80M-TI	89.69 ± 0.40	59.53 ± 0.40
RST+MART			87.50	56.29
RST+GAIRAT			89.36	59.64
RST+AWP			88.25 ± 0.40	60.05 ± 0.40
RST+RWP			88.87 ± 0.55	60.36 ± 0.06
ISEAT (ours)			**90.59** ± 0.19	**61.55** ± 0.10
IDBH+ISEAT (ours)		–	87.91 ± 0.18	58.55 ± 0.14

6.1. Performance on CIFAR10

As can be seen from the results in Table 2, ISEAT substantially improves both accuracy and robustness over the baseline in all evaluated settings. Specifically, it boosts robustness by +3.12% compared to AT in the default setting, by +5.16% when IDBH data augmentation is used, and by +2.02% compared to RST when extra real data from 80M-TI is used. More importantly, ISEAT boosts accuracy as well suggesting a better trade-off between accuracy and robustness. By combining with IDBH, ISEAT achieves a robustness of 58.55% for WRN28-10 which is competitive with the baseline robustness of 59.53% achieved by RST using additional real data. This substantially closes the gap between the robust performance of training with and without extra data.

ISEAT outperforms all existing instance-adaptive AT methods in terms of robustness. The comparison is conducted on the default setup of CIFAR10 (Table 2) since published results are available for this setup. ISEAT achieves the highest robustness, 56.54%, among all competitive works, which considerably exceeds the previous best record of 55.52% achieved by LAS-AWP and the robustness of 54.23% achieved by the most similar previous work, MART-AWP. Particularly, ISEAT dramatically outperforms FAT by +9.06% in terms of robustness. FAT is one of the most recent contributions whose instance adaptation strategy contrasts ours, as described in Section 2. This supports the claim that this previous strategy for instance-adaptive AT is fundamentally defective. Furthermore, ISEAT consistently achieves superior robustness compared to all available related works such as MART and GAIRAT in the condition with extra data.

Last, the performance of ISEAT can be further improved by +0.55% in the default setup when integrated with another weight smoothing technique: Stochastic Weight Averaging (SWA). However, a similar performance boost was not observed in the other setting for CIFAR10. This suggests that ISEAT may exhaust the benefits of weight smoothing in some settings, but not all.

6.2. Performance on CIFAR100 and SVHN

Following common practice for testing generalization ability, this section evaluates ISEAT on the alternative datasets CIFAR100 and

[2] The extra data was downloaded from the official git repository of Carmon et al. [22]: https://github.com/yaircarmon/semisup-adv.

Table 3
Test accuracy and robustness of our method for PRN18 on CIFAR100 and SVHN.

Dataset	Method	Accuracy (%)	Robustness (%)
CIFAR100	AT	**56.15** ± 1.15	25.12 ± 0.22
CIFAR100	ISEAT (ours)	53.19 ± 0.23	**28.17** ± 0.14
SVHN	AT	90.55 ± 0.60	47.48 ± 0.59
SVHN	ISEAT (ours)	**91.08** ± 0.49	**54.04** ± 0.68

SVHN. As shown in Table 3, ISEAT substantially improves robustness over the baseline by +3.05% on CIFAR100 and by +6.56% on SVHN. It also slightly boosts accuracy on SVHN. Note that the magnitude of robustness improvement in a particular training setting generally depends on the degree of robust overfitting, which is connected to the unevenness of AV among training data. It is therefore reasonable for ISEAT to perform differently on different datasets even using the same model architecture. Overall, the performance improvements across various datasets is consistent, which confirms that the proposed method is generally applicable.

6.3. Performance on TinyImageNet

TinyImageNet is a subset of ImageNet, with the number of classes reduced to 200 and the image resolution reduced to 64 × 64. Each class contains 500 images in the training set and 50 images in the validation/test set. Compared to CIFAR and SVHN datasets, TinyImageNet offers a significantly greater number of classes and higher image resolution, making it more complex, realistic, and challenging.

As shown in Table 4, our method significantly enhances both accuracy and robustness compared to the baselines. Notably, it achieves a dramatic improvement in end robustness over the naive AT, with increases of 7.71% and 7.9% when used with and without SWA, respectively. This leads to a substantially reduced gap between the best and end robustness, indicating that our method effectively mitigates robust overfitting. Overall, our method demonstrates strong generalization to larger and more complex datasets, such as TinyImageNet.

6.4. Comparison with state-of-the-art adversarial training methods

To further demonstrate the effectiveness of our approach, we have expanded the scope of comparison to include the latest published adversarial training methods [41–44]. Note that these alternative methods do not focus on instance-adaptive regularization and are therefore not directly related to our approach. As shown in Table 5, our method still achieves the highest robustness among all the compared methods. Additionally, it outperforms these latest methods in accuracy, with the exception of SGLR-AT, which attains a slightly higher accuracy. Overall, our method surpasses the latest advanced adversarial training methods in performance.

6.5. Mitigating robust shortcuts and overfitting

To further demonstrate the effectiveness of our method in mitigating robust shortcuts and robust overfitting, we conducted experiments to examine how the phenomena of robust shortcuts and robust overfitting vary with different regularization strengths (λ) in our method. As shown in Table 6, both the degree of robust shortcuts (measured by AV SD) and robust overfitting (measured by RO) decrease as the regularization strength λ increases. Moreover, our method consistently exhibits lower AV SD and RO compared to naive adversarial training (AT) across all evaluated λ values. These results support the rationale behind our approach, which penalizes data instances with large AV to mitigate the exploitation of robust shortcuts and reduce robust overfitting.

6.6. Ablation study

Ablation experiments were conducted to justify the design of ISEAT and illuminate the mechanism behind its effectiveness. Experiments were performed using WRN34-10 on CIFAR10. To ensure a fair comparison, the approaches were applied to fine-tune the same model. This base model had been previously trained using AT with the default training setup, as described in Section 6. Fine-tuning was performed for 40 epochs. The initial learning rate was 0.01 and decayed to 0.001 after 20 epochs.

This section first assesses the contribution of different components in the proposed method. It can be observed in Table 7 that both the components of ISEAT, (adaptively weighted) input loss smoothing and weight loss smoothing, can individually improve robustness over the baseline to a great extent, +1.08% and +1.84% respectively. This confirms that they both play a vital role in ISEAT. Furthermore, combining them together (the proposed method) achieves a greater robustness boost, +3.04%, compared to either of them alone. This combined boost is greater than the arithmetic sum of the performance increases of the individual components ($3.04\% > 1.08\% + 1.84\%$) suggesting that these two components are complementary to each other.

Next, this section examines the design of input loss smoothness regularizer. First, the choice of distance metric used to measure the dissimilarity between two predicted logits was verified. As shown in Table 8, squared ℓ_2-norm (the adopted method) performs slightly better than KL-divergence (used by MART [16]) in terms of both accuracy and robustness. Moreover, the linear weight scheme (the chosen method) was compared with the top-10% weight scheme (used in the preliminary experiments reported in Section 3.3, see Eq. (7)) and unweighted (or uniform) scheme. It can be observed in Table 8 that the weighted schemes, either linear or top-10%, considerably improve both accuracy and robustness over the unweighted scheme, and among the weighted schemes, the linear one outperforms the top-10% scheme regarding both accuracy and robustness. Overall, a linear weight scheme with squared ℓ_2-norm is empirically the best among all evaluated solutions.

Last, this section examines the effectiveness of the proposed approach to combine input and weight loss smoothing. The proposal, LSIW, was compared with Logit Stability regularization against Input perturbation (LSI) only and TRADES-AWP. The regularization loss of these methods is described in Table 9. For more technical detail, please refer to Section 5.1. It is observed in Table 9 that LSIW achieves substantially higher accuracy and robustness than the others. This supports the hypothesis that stabilizing logits against both input and weight adversarial perturbation makes better use of adversarial weight perturbation, and hence, results in a more effective smoothness regularization.

6.7. Computational efficiency

It can be seen from the results in Table 7 that smoothing the input loss landscape alone (i.e., weighted logit stability regularization) adds about 6% computational overhead, and smoothing weight loss landscape alone adds around 9% computational overhead compared to AT. Jointly smoothing both input and weight loss landscapes using the proposed ISEAT method introduces an overhead of approximately 18% compared to AT. The extra cost of ISEAT is greater than the sum of the extra cost of two separate smoothing components ($18\% > 6\% + 9\%$) because it requires additional forward and backward passes to compute the gradient of the proposed regularization (Sections 5.3 and 5.4). Nevertheless, we believe this additional cost is acceptable for the following reasons. First, the substantial performance improvement achieved by our method over the baseline justifies the trade-off in efficiency. Second, our approach is significantly more efficient than some recently published methods, such as LAS-AT [19] and PART [45], which incur 40% and 77% increases in cost over baseline AT, respectively.

Table 4
The performance of our method and the baselines for PreActResNet-18 on Tiny ImageNet. The results for consistency, SRC+FL, and LAS-AWP are sourced from their original publications. "–" indicates that the corresponding result was not reported in the original source.

Method	Accuracy (%)			Robustness (%)		
	Best	End	Diff.	Best	End	Diff.
AT	46.39 ± 0.58	46.69 ± 0.16	−0.30 ± 0.65	18.17 ± 0.23	12.27 ± 0.37	5.90 ± 0.58
AT-SWA	**49.47** ± 0.40	47.78 ± 0.34	1.69 ± 0.74	19.96 ± 0.04	12.97 ± 0.07	6.99 ± 0.11
Consistency	46.50	45.61	0.89	15.09	13.56	1.53
SRC+FL	46.75	–	–	17.01	–	–
LAS-AWP	–	45.26	–	–	18.42	–
ISEAT (ours)	48.26 ± 0.35	48.71 ± 0.46	−0.45 ± 0.71	20.55 ± 0.10	20.17 ± 0.19	0.39 ± 0.20
ISEAT-SWA (ours)	47.58 ± 0.27	**49.18** ± 0.16	**−1.60** ± 0.21	**20.90** ± 0.29	**20.68** ± 0.17	**0.22** ± 0.21

Table 5
The performance of our methods and recent advanced adversarial training methods on CIFAR10. "Test Adv." refers to the attack method used to evaluate adversarial robustness on the test data. Different model architectures and test adversaries were used to align with the setups in the original works of the compared methods.

Model	Test Adv.	Method	Accuracy (%)	Robustness (%)
PRN18	AA	TSOVR [41]	81.40	49.80
		SRC [42]	80.70	50.35
		SGLR-AT [43]	**82.90**	51.20
		ISEAT (ours)	82.37	**51.76**
WRN34-10	PGD20	SLORE-MART [44]	85.17	59.10
		ISEAT (ours)	**85.62**	**60.95**

Table 6
The variation in performance and adversarial vulnerability (AV) unevenness with different λ values for our method on CIFAR-10 and WRN34-10. "RO" indicates the degree of robust overfitting, measured as the gap between the best and end robustness.

Method	λ	Accuracy (%)	Robustness (%)	RO	AV SD
AT	–	85.90	53.42	5.20	0.322
ISEAT	0.005	86.43	55.17	0.98	0.293
	0.01	**86.68**	55.44	0.92	0.278
	0.05	85.24	55.30	0.72	0.285
	0.1	86.02	**56.54**	**0.36**	**0.200**

Table 7
The contribution of each component in the proposed ISEAT method on CIFAR10 for WRN34-10. Time is an average measured for processing one mini-batch on a Nvidia RTX 3080Ti in seconds.

Method	Accuracy (%)	Robustness (%)	Time (s)
AT	85.90 ± 0.57	53.42 ± 0.59	**0.253**
+ input loss smoothing	84.10 ± 0.27	54.50 ± 0.17	0.268 (+6%)
+ weight loss smoothing	**86.04** ± 0.27	55.26 ± 0.15	0.277 (+9%)
+ both (ISEAT)	85.63 ± 0.13	**56.46** ± 0.14	0.298 (+18%)

Table 8
The performance of logit stability regularization with different distance metrics and weight schemes on CIFAR10 for WRN34-10. "Distance" denotes the metric used to measure the discrepancy between two predicted logits. "Weight" denotes the weight scheme.

Distance	Weight	Accuracy (%)	Robustness (%)
AT		**85.90** ± 0.57	53.42 ± 0.59
KL-divergence	Unweighted	85.07 ± 0.31	56.08 ± 0.32
Squared ℓ_2-norm	Unweighted	85.15 ± 0.70	56.20 ± 0.19
Squared ℓ_2-norm	Top-10%	85.53 ± 0.03	56.36 ± 0.02
Squared ℓ_2-norm	Linear	85.63 ± 0.13	**56.46** ± 0.14

Table 9
The performance of different approaches to jointly smoothing input and weight loss landscapes on CIFAR10 for WRN34-10. $\boldsymbol{x}'$ and $\boldsymbol{\theta}'$ refer to the perturbed input and weight respectively. For a fair comparison, the original distance metric, KL-divergence, in TRADES-AWP (Eq. (12)) was replaced by squared ℓ_2-norm to align with the other methods.

Method	Smoothness loss	Accuracy (%)	Robustness (%)
AT		**85.90** ± 0.57	53.42 ± 0.59
+ LSI	$\|f(\boldsymbol{x}';\boldsymbol{\theta}) - f(\boldsymbol{x};\boldsymbol{\theta})\|_2^2$	85.49 ± 0.50	55.38 ± 0.32
+ TRADES-AWP	$\|f(\boldsymbol{x}';\boldsymbol{\theta}') - f(\boldsymbol{x};\boldsymbol{\theta}')\|_2^2$	85.52 ± 0.29	55.82 ± 0.46
+ LSIW (ours)	$\|f(\boldsymbol{x}';\boldsymbol{\theta}') - f(\boldsymbol{x};\boldsymbol{\theta})\|_2^2$	85.63 ± 0.13	**56.46** ± 0.14

7. Conclusion

This work investigated how adversarial vulnerability evolves during AT from an instance-wise perspective. We observed that a model was trained to be more robust for some samples and, meanwhile, more vulnerable at some others resulting in an increasingly uneven distribution of adversarial vulnerability among training data. We theoretically proposed an alternative optimization path to minimize adversarial loss as an account for this phenomenon. Motivated by the above observations, we first proposed a new AT framework that enhances robustness at each sample with strength adapted to its adversarial vulnerability. We then realized it with a novel regularization method that jointly smooths input and weight loss landscapes. Our proposed method is novel in a number of respects: (1) adapting regularization to instance-wise adversarial vulnerability is new and contrasts the popular existing strategy; (2) stabilizing logit against adversarial input and weight perturbation simultaneously is novel and more effective than the previous approaches. Experimental result shows our method outperforms all related works and significantly improves robustness w.r.t. the AT baseline. Extensive ablation studies demonstrate the vital contribution of the proposed instance adaptation strategy and smoothness regularizer in our method.

In addition to finding that AT results in an uneven distribution of adversarial vulnerability among training data, we also observed that for a considerable proportion of samples the model was excessively robust, such that even very large perturbations, making the sample unrecognizable to a human, failed to influence the prediction made by the network. One limitation of this work is that the proposed method, albeit effective in improving robustness, does not mitigate the issue of "disordered robustness". Future work might usefully explore this problem to further improve the performance of AT. A better trade-off between accuracy and robustness is anticipated if disordered robustness is alleviated.

CRediT authorship contribution statement

Lin Li: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Michael Spratling:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge use of the King's Computational Research, Engineering and Technology Environment (CREATE) and the Joint Academic Data science Endeavour (JADE) facility for carrying out the experiments described in this paper. This work was funded by a scholarship from the King's - China Scholarship Council (K-CSC).

Data availability

All used data are publicly available.

References

[1] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations, ICLR, 2015.
[2] A. Athalye, N. Carlini, D. Wagner, Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, in: International Conference on Machine Learning, ICML, 2018.
[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations, ICLR, 2018.
[4] Y. Balaji, T. Goldstein, J. Hoffman, Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets, 2019, arXiv.
[5] G.W. Ding, Y. Sharma, K.Y.C. Lui, R. Huang, MMA training: Direct input space margin maximization through adversarial training, in: International Conference on Learning Representations, ICLR, 2020.
[6] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, M. Kankanhalli, Attacks which do not kill training make adversarial learning stronger, in: International Conference on Machine Learning, ICML, 2020.
[7] M. Cheng, Q. Lei, P.-Y. Chen, I. Dhillon, C.-J. Hsieh, CAT: Customized adversarial training for improved robustness, in: International Joint Conference on Artificial Intelligence, IJCAI, 2022.
[8] S. Yang, C. Xu, One size does NOT fit all: Data-adaptive adversarial training, in: European Conference on Computer Vision, ECCV, 2022.
[9] L. Rice, E. Wong, J.Z. Kolter, Overfitting in adversarially robust deep learning, in: International Conference on Machine Learning, ICML, 2020.
[10] C.-J. Simon-Gabriel, Y. Ollivier, L. Bottou, B. Schölkopf, D. Lopez-Paz, First-order adversarial vulnerability of neural networks and input dimension, in: International Conference on Machine Learning, ICML, 2019.
[11] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, P. Frossard, Robustness via curvature regularization, and vice versa, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019.
[12] D. Wu, S.-T. Xia, Y. Wang, Adversarial weight perturbation helps robust generalization, in: Neural Information Processing Systems, NeurIPS, 2020.
[13] L. Li, M. Spratling, Understanding and combating robust overfitting via input loss landscape analysis and regularization, Pattern Recognit. (2023).
[14] C. Yu, B. Han, M. Gong, L. Shen, S. Ge, D. Bo, T. Liu, Robust weight perturbation for adversarial training, in: International Joint Conference on Artificial Intelligence, IJCAI, 2022.
[15] T. Chen, Z. Zhang, S. Liu, S. Chang, Z. Wang, Robust overfitting may be mitigated by properly learned smoothening, in: International Conference on Learning Representations, ICLR, 2021.
[16] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, Q. Gu, Improving adversarial robustness requires revisiting misclassified examples, in: International Conference on Learning Representations, ICLR, 2020.
[17] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, M. Kankanhalli, Geometry-aware instance-reweighted adversarial training, in: International Conference on Learning Representations, ICLR, 2021.
[18] M. Xu, T. Zhang, Z. Li, D. Zhang, InfoAT: Improving adversarial training using the information bottleneck principle, in: IEEE Transactions on Neural Networks and Learning Systems (T-NNLS), 2022.
[19] X. Jia, Y. Zhang, B. Wu, K. Ma, J. Wang, X. Cao, LAS-AT: Adversarial training with learnable attack strategy, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022.
[20] M. Arjovsky, L. Bottou, Towards principled methods for training generative adversarial networks, in: International Conference on Learning Representations, ICLR, 2017.
[21] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, A. Madry, Adversarially robust generalization requires more data, in: Neural Information Processing Systems (NeurIPS), 2018.
[22] Y. Carmon, A. Raghunathan, L. Schmidt, J.C. Duchi, P.S. Liang, Unlabeled data improves adversarial robustness, in: Neural Information Processing Systems (NeurIPS), 2019.
[23] S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. Calian, T. Mann, Improving robustness using generated data, in: Neural Information Processing Systems (NeurIPS), 2021.
[24] L. Li, M.W. Spratling, Data augmentation alone can improve adversarial training, in: International Conference on Learning Representations, ICLR, 2023.
[25] P. Xia, Z. Li, W. Zhang, B. Li, Data-efficient backdoor attacks, in: L.D. Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 3992–3998, http://dx.doi.org/10.24963/ijcai.2022/554, main Track.
[26] Z. Li, H. Sun, P. Xia, B. Xia, X. Rui, W. Zhang, Q. Guo, Z. Fu, B. Li, A proxy attack-free strategy for practically improving the poisoning efficiency in backdoor attacks, IEEE Trans. Inf. Forensics Secur. (2024).
[27] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European Conference on Computer Vision, ECCV, 2016.
[28] R. Rade, S.-M. Moosavi-Dezfooli, Reducing excessive margin to achieve a better accuracy vs. robustness trade-off, in: International Conference on Learning Representations, ICLR, 2022.
[29] C. Yu, B. Han, L. Shen, J. Yu, C. Gong, M. Gong, T. Liu, Understanding robust overfitting of adversarial training and beyond, in: International Conference on Machine Learning, ICML, 2022.
[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Neural Information Processing Systems (NeurIPS), 2019.
[31] S. Zagoruyko, N. Komodakis, Wide residual networks, in: British Machine Vision Conference, BMVC, 2016.
[32] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Technical Report, 2009.
[33] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, 2011.
[34] Y. Le, X. Yang, Tiny imagenet visual recognition challenge, 2015, CS 231N.
[35] M. Andriushchenko, N. Flammarion, Understanding and improving fast adversarial training, in: Neural Information Processing Systems (NeurIPS), 2020.
[36] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: International Conference on Machine Learning, ICML, 2020.
[37] H. Kim, Torchattacks: A pytorch repository for adversarial attacks, 2021.
[38] H. Zhang, Y. Yu, J. Jiao, E. Xing, L.E. Ghaoui, M. Jordan, Theoretically principled trade-off between robustness and accuracy, in: International Conference on Machine Learning, ICML, 2019.
[39] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, M. Hein, Robustbench: a standardized adversarial robustness benchmark, in: Neural Information Processing Systems (NeurIPS), 2021.
[40] S.-A. Rebuffi, S. Gowal, D.A. Calian, F. Stimberg, O. Wiles, T. Mann, Data augmentation can improve robustness, in: Neural Information Processing Systems (NeurIPS), 2021.
[41] S. Kanai, S. Yamaguchi, M. Yamada, H. Takahashi, K. Ohno, Y. Ida, One-vs-the-rest loss to focus on important samples in adversarial training, in: International Conference on Machine Learning, ICML, 2023.
[42] H. Liu, Z. Zhong, N. Sebe, S. Satoh, Mitigating Robust Overfitting Via Self-Residual-Calibration Regularization, Artificial Intelligence (2023).
[43] Z. Li, D. Yu, L. Wei, C. Jin, Y. Zhang, S. Chan, Soften to defend: Towards adversarial robustness via self-guided label refinement, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024.
[44] X. Yin, W. Ruan, Boosting adversarial training via Fisher-rao norm-based regularization, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024.
[45] J. Zhang, F. Liu, D. Zhou, J. Zhang, T. Liu, Improving accuracy-robustness trade-off via pixel reweighted adversarial training, in: International Conference on Machine Learning, ICML, 2024.

Lin Li is currently a Ph.D. student in computer science at the Department of Informatics, King's College London. He received a M.Sc. degree in computing from Imperial College London, and a B.M. degree in Finance from Xiamen University. His research interests include adversarial machine learning, LLM hallucination, LLM jailbreaking and safety alignment.

Michael Spratling's research is concerned with understanding the computational and neural mechanisms underlying visual perception, and developing biologically-inspired neural networks to solve problems in computer vision and machine learning. He has a multidisciplinary background having trained and held posts in computer science, psychology, and engineering at various universities (Loughborough, Edinburgh, St Andrews, Cambridge, Birkbeck, and King's College London). He is currently a researcher in the Department of Behavioral and Cognitive Sciences at the University of Luxembourg.