SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

June 2025, Vol. 68, Iss. 6, 162101:1–162101:15
https://doi.org/10.1007/s11432-022-3861-8

Pareto adversarial robustness: balancing spatial robustness and sensitivity-based robustness

Ke SUN[1], Mingjie LI[1] & Zhouchen LIN[1,2,3*]

[1]*State Key Lab of General AI, School of Intelligence Science and Technology, Peking University, Beijing 100871, China*
[2]*Institute for Artificial Intelligence, Peking University, Beijing 100871, China*
[3]*Pazhou Laboratory (Huangpu), Guangzhou 510555, China*

Received 27 November 2022/Revised 20 February 2023/Accepted 15 June 2023/Published online 7 May 2025

Abstract Adversarial robustness, which primarily comprises sensitivity-based robustness and spatial robustness, plays an integral part in achieving robust generalization. In this paper, we endeavor to design strategies to achieve universal adversarial robustness. To achieve this, we first investigate the relatively less-explored realm of spatial robustness. Then, we integrate the existing spatial robustness methods by incorporating both local and global spatial vulnerability into a unified spatial attack and adversarial training approach. Furthermore, we present a comprehensive relationship between natural accuracy, sensitivity-based robustness, and spatial robustness, supported by strong evidence from the perspective of robust representation. Crucially, to reconcile the interplay between the mutual impacts of various robustness components into one unified framework, we incorporate the Pareto criterion into the adversarial robustness analysis, yielding a novel strategy called Pareto adversarial training for achieving universal robustness. The resulting Pareto front, which delineates the set of optimal solutions, provides an optimal balance between natural accuracy and various adversarial robustness. This sheds light on solutions for achieving universal robustness in the future. To the best of our knowledge, we are the first to consider universal adversarial robustness via multi-objective optimization.

Keywords deep learning, adversarial robustness, reliable machine learning, Pareto optimization, spatial robustness

Citation Sun K, Li M J, Lin Z C. Pareto adversarial robustness: balancing spatial robustness and sensitivity-based robustness. Sci China Inf Sci, 2025, 68(6): 162101, https://doi.org/10.1007/s11432-022-3861-8

1 Introduction

Robust generalization serves as an extension of the traditional generalization that is normally achieved via empirical risk minimization for i.i.d. data [1]. However, the test environment could be slightly or dramatically different from the training environment [2] in a robust generalization scenario. Lately, improving the robustness of deep neural networks has been one of the pivotal areas of research, encompassing different threads of research such as adversarial robustness [3,4], non-adversarial robustness [5,6], Bayesian deep learning [7,8], and causality [9]. In this paper, we focus on adversarial robustness, where adversarial examples are carefully manipulated by humans to fool machine learning models, e.g., deep neural networks, which could pose serious threats, especially in safety-critical applications. Currently, adversarial training (AT) [3,10–12] is regarded as a promising and widely accepted strategy to address this issue.

Like out-of-distribution (OoD) robustness, adversarial robustness also has several aspects [13–15], including sensitivity-based robustness [16], i.e., robustness against pixel-wise perturbations (normally within the constraints of an l_p ball), and spatial robustness, i.e., robustness against multiple spatial transformations. Computer vision and graphics literature provide a deeper insight into these two aspects, revealing that two main factors determine the appearance of a pictured object [17,18]: (1) lighting and materials, and (2) geometry. Most previous studies on adversarial robustness have focused only on the first factor [17] by examining pixel-wise perturbations, e.g., projected gradient descent (PGD) attacks [10], assuming that the underlying geometry stays the same after the adversarial perturbation. Only a small proportion of research works have attempted to tackle the less-studied second factor, which includes flow-based [17] and rotation-translation (RT)-based attacks [19,20].

* Corresponding author (email: zlin@pku.edu.cn)

However, it is crucial to consider spatial robustness for achieving universal robustness, the ultimate objective of robust generalization. One of the most important reasons is that sensitivity-based robustness, which is generally based on the l_p-distance, is not sufficient to maintain perceptual similarity [17, 19–21]. Specifically, although spatial attacks or geometric transformations result in small perceptual differences, they yield large l_p distances.

A clear relationship between accuracy, sensitivity-based and spatial robustness is the key to achieving universal adversarial robustness. While the trade-off between sensitivity-based robustness and accuracy has been revealed by several studies [22–24], the comprehensive relationships among spatial robustness and them are still unclear. Although previous studies [25, 26] have explored this issue, they only focused on RT spatial robustness and did not consider flow-based spatial robustness [17, 27]. Surprisingly, we find that flow-based spatial robustness presents a relationship contrary to the one revealed previously, making the previous conclusion less reliable.

Based on this important finding, we start our exploration of clearer relationships between different robustnesses, and we eventually harmonize the conflicting relationships within them by leveraging the Pareto criterion [28–30], thus achieving an optimal balanced universal robustness. A recent study [24] attributes the conflicting relationships among the various robustnesses to overparametrization, while we uncovered it from the perspective of different shape-biased representations. Another report [31] examined the trade-off in the inference time, while we target more comprehensive relationships between different robustnesses with a different methodology.

In this paper, we first try to gain deeper insights into the robustness relationships by investigating the two main spatial robustness branches, i.e., flow-based spatial attack [17] and RT attack [20]. After revealing their impact on local and global spatial sensitivity, we propose integrated spatial attack and spatial adversarial training (spatial AT), which can incorporate comprehensive spatial vulnerabilities or robustness. Based on this understanding, we present a comprehensive relationship among the accuracy, sensitivity-based robustness, and the two branches of spatial robustness by investigating their different saliency maps from the perspectives of shape-bias, sparse or dense representation. It turns out that while the relationship between sensitivity-based and RT robustness is a fundamental trade-off, sensitivity-based and flow-based spatial robustness are highly correlated, providing a vital supplementary for previous conclusions. Thus, comprehensive relationships between accuracy and the various robustnesses are not pure trade-offs, motivating us to introduce the Pareto criterion [28–30], the general multi-objective optimization principle, into the universal adversarial robustness analysis. The Pareto criterion enables an optimal balance between the interplay of natural accuracy and the different adversarial robustnesses, leading to universal adversarial robustness in a Pareto manner. By incorporating a two-moment term that can capture the interaction between loss of accuracy and different robustnesses, we propose a bi-level optimization framework called Pareto adversarial training (Pareto AT). The resulting Pareto front provides a set of optimal solutions that can balance perfectly all the relationships under consideration, outperforming other existing strategies.

Our contributions can be summarized as follows:

• We reveal the existence of both local and global spatial robustness and propose integrated spatial attack and spatial AT, incorporating comprehensive spatial vulnerabilities.

• We present comprehensive relationships among accuracy, sensitivity-based, and different spatial robustnesses, supported by strong and intuitive evidence from the perspective of robust representation.

• We incorporate the Pareto criterion into adversarial robustness analysis, and the resulting Pareto AT can optimally balance multiple adversarial robustness, yielding universal adversarial robustness.

2 Local and global spatial robustness

To present the comprehensive relationships between accuracy and different adversarial robustnesses, we first provide a fine-grained understanding of spatial robustness. We summarize several studies about spatial robustness [17, 19, 20, 25–27] into two major branches: (1) flow-based attacks, and (2) RT attacks. In particular, we find that the former mainly focuses on the local spatial vulnerability while the latter tends to capture the global spatial sensitivity. Based on this finding, integrated spatial attack and spatial AT are proposed.

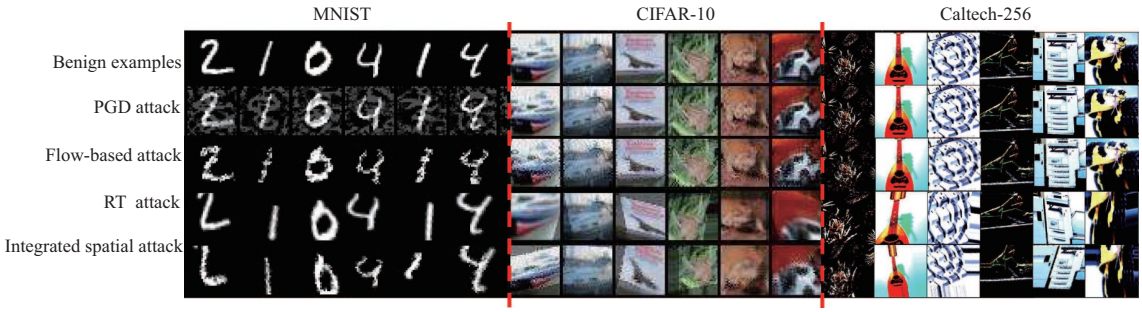

Figure 1 (Color online) Visualization of flow-based, RT and our integrated spatial adversarial examples on MNIST, CIFAR-10, and Caltech-256. More images and detailed discussions are provided in Appendix A.

2.1 Local spatial robustness: flow-based attacks

The most representative flow-based attack is the spatial transformed attack [17], wherein a differentiable flow vector $w_F = (\Delta\mu, \Delta v)$ is introduced in the 2D coordinates (μ, v) to craft adversarial spatial transformation. The vanilla-targeted flow-based attack [17] follows the optimization manner ($\kappa = 0$):

$$w_F^* = \arg\min_{w_F} \max_{i \neq t} f_\theta^i(x_{w_F}) - f_\theta^t(x_{w_F}) + \tau \mathcal{L}_{\text{flow}}(w_F), \tag{1}$$

where $f_\theta(x) = (f_\theta^1(x), \ldots, f_\theta^K(x))$ is the classifier in the K-classification task. x_{w_F} is a flow-based adversarial example parameterized by the flow vector w_F. $\mathcal{L}_{\text{flow}}$, which measures the local smoothness of the spatial transformation balanced by τ.

Interestingly, our empirical study shown in the left part of Figure 1 suggests that the flow-based attack tends to yield local permutations among pixels in some specific regions, irrespective of the option of τ, rather than a global spatial transformation based on their shapes. Our analysis indicates that this phenomenon is due to two factors. (1) Local permutations, especially in regions where colors of pixels change dramatically, are already sufficiently sensitive to manipulations, as demonstrated by our empirical results shown in Figure 1. (2) The manner of optimization does not incorporate any sort of shape transformation information, e.g., a parametric equation of rotation, as opposed to the vanilla RT attack, which we present in the following. Therefore, we conclude that flow-based attacks tend to capture the local spatial vulnerability. Further, to design the integrated spatial attack, we transform (1) into its untargeted version under cross-entropy loss with flow vector bounded by an ϵ_F-ball:

$$w_F^* = \arg\max_{w_F} \mathcal{L}_\theta^{\text{CE}}(x_{w_F}, y) \quad \text{s.t. } \|w_F\| \leqslant \epsilon_F, \tag{2}$$

where $\mathcal{L}_\theta^{\text{CE}}(x, y) = \log \sum_j \exp(f_\theta^j(x)) - f_\theta^y(x)$. To maintain a uniform optimization form in our integrated spatial attack, we replace local smoothness term $\mathcal{L}_{\text{flow}}$ in (1) with our familiar l_p constraint and leverage the cross-entropy loss instead of the max operation as suggested in [32]. Proposition 1 reveals the correlation between the two losses, indicating that the smooth approximation version of max operation in (1), denoted as $\mathcal{L}_\theta^S$, has a parallel updating direction with cross-entropy loss related to w_F. Proof can be found in Appendix B.

Proposition 1. Consider $\mathcal{L}_\theta^S(x, y) = \log \sum_{i \neq y} \exp(f_\theta^i(x)) - f_\theta^y(x)$ as the smooth version loss of (1) without a local smoothness term. For a fixed (x_{w_F}, y) and θ, we have

$$\nabla_{w_F} \mathcal{L}_\theta^{\text{CE}}(x_{w_F}, y) = r(x_{w_F}, y) \nabla_{w_F} \mathcal{L}_\theta^S(x_{w_F}, y), \tag{3}$$

where $r(x_{w_F}, y) = \sum_{i \neq y} \exp(f_\theta^i(x_{w_F})) / \sum_i \exp(f_\theta^i(x_{w_F}))$.

2.2 Global spatial robustness: RT attacks

The original RT attack [19,20] applies parametric equation constraints on 2D coordinates, thus capturing the global spatial information:

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}. \tag{4}$$

To design a generic spatial transformation matrix that can simultaneously consider rotation, translation, cropping, and scaling, we re-parameterize the transform matrix as a generic 6-dimensional affine transformation one, inspired by spatial transformer networks [33]:

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} w_{\mathrm{RT}}^{11} & w_{\mathrm{RT}}^{12} & w_{\mathrm{RT}}^{13} \\ w_{\mathrm{RT}}^{21} & w_{\mathrm{RT}}^{22} & w_{\mathrm{RT}}^{23} \end{bmatrix} \right) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{5}$$

where we denote $A_{w_{\mathrm{RT}}}$ as the generic 6-dimensional affine transformation matrix, in which each entry of w_{RT} indicates the increment in different spatial aspects. For example, $(w_{\mathrm{RT}}^{13}, w_{\mathrm{RT}}^{23})$ determines translation. Finally, the optimization form of the resulting generic and differentiable RT-based attack bounded by ϵ_{RT}-ball is expressed as

$$w_{\mathrm{RT}}^{*} = \arg\max_{w_{\mathrm{RT}}} \mathcal{L}_{\theta}^{\mathrm{CE}}(x_{w_{\mathrm{RT}}}, y) \ \text{ s.t. } \|w_{\mathrm{RT}}\| \leqslant \epsilon_{\mathrm{RT}}. \tag{6}$$

2.3 Integrated spatial attack

The key to achieving integrated spatial robustness is to design an integrated parameterized sampling grid $\mathcal{T}_{w_{\mathrm{RT}}, w_F}(G)$ that can wrap the regular grid with both flow and affine transformation, where G is the generated grid. We show our integrated approach as shown below:

$$\begin{aligned} \mathcal{T}_{w_{\mathrm{RT}}, w_F}(G) &= A_{w_{\mathrm{RT}}} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + w_F, \\ x^{\mathrm{adv}} &= \mathcal{T}_{w_{\mathrm{RT}}, w_F}(G) \circ x. \end{aligned} \tag{7}$$

Then we sample new x^{adv} by $\mathcal{T}_{w_{\mathrm{RT}}, w_F}(G)$ via the differentiable bilinear interpolation [33]. Note that w_F has the same dimensions as the grid G, which are different from the impact of two-dimensional translation parameters in w_{RT}. Then the final loss function of the integrated spatial attack can be presented as

$$w^{*} = \arg\max_{w} \mathcal{L}_{\theta}^{\mathrm{CE}}(x + \eta_w, y), \quad \text{s.t. } \|w\| \leqslant \epsilon, \tag{8}$$

where η_w is the crafted integrated spatial perturbation parameterized by $w = [w_F, w_{\mathrm{RT}}]^{\mathrm{T}}$, simultaneously considering both flow-based and RT spatial sensitivity. Note that η_w itself does not necessarily satisfy the l_p constraint directly. For the implementation, we follow the PGD procedure [10], a common practice in sensitivity-based attacks. We consider the infinity norm of w and different learning rates for the two types of spatial robustness. Therefore, the updating rule of w in each iteration is

$$\begin{aligned} &\begin{bmatrix} \bar{w}_F^{t+1} \\ \bar{w}_{\mathrm{RT}}^{t+1} \end{bmatrix} = \begin{bmatrix} w_F^{t} \\ w_{\mathrm{RT}}^{t} \end{bmatrix} + \begin{bmatrix} \alpha_F \\ \alpha_{\mathrm{RT}} \end{bmatrix} \mathrm{sign}(\nabla_w \mathcal{L}_{\theta}^{\mathrm{CE}}(x_{w^t}^{t}, y)), \\ &\begin{bmatrix} w_F^{t+1} \\ w_{\mathrm{RT}}^{t+1} \end{bmatrix} = \mathrm{clip}_{\epsilon} \left(\begin{bmatrix} \bar{w}_F^{t+1} \\ \bar{w}_{\mathrm{RT}}^{t+1} \end{bmatrix} \right), \\ &x_{w^{t+1}}^{t+1} = \mathcal{T}_{w^{t+1}}(G) \circ x, \end{aligned} \tag{9}$$

where $w^{t+1} = [w_F^{t+1}, w_{\mathrm{RT}}^{t+1}]^{\mathrm{T}}$ is element-wisely clipped from $\bar{w}^{t+1}$ by $\epsilon = [\boldsymbol{\epsilon}_F, \boldsymbol{\epsilon}_{\mathrm{RT}}]^{\mathrm{T}}$. From Figure 1, we can observe that our integrated spatial attack can construct both local and global spatial transformations on images. Thus, it can simultaneously yield local pixel-wise permutations and global shape transformations.

Then, we visualize the loss surface under this integrated spatial attack leveraging "filter normalization" [34] as illustrated in Figure 2. We strictly follow the implementation from [34] to achieve the

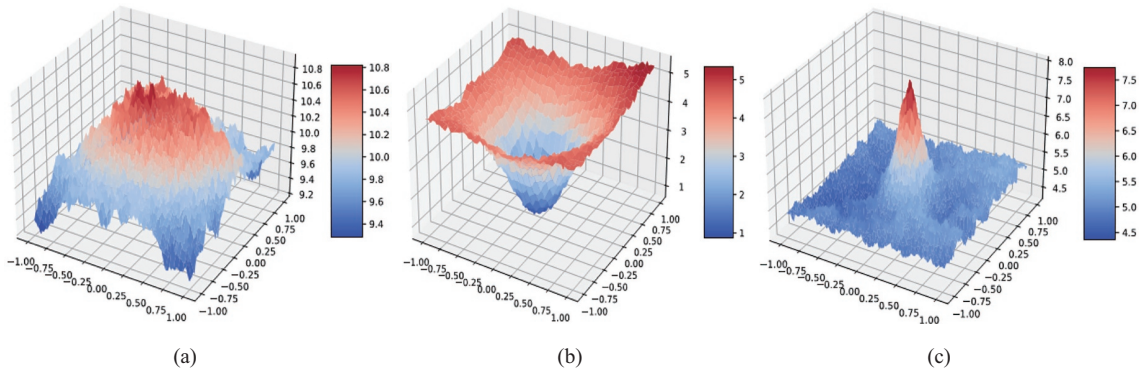

Figure 2 (Color online) Loss landscape of integrated spatial attack on CIFAR-10. (a) A distant view of loss landscape w.r.t w before the optimization in (8); (b) a close view before the optimization shows a highly convex surface near the initialization point; (c) the loss landscape around the maxima w^* after the optimization in (8).

desired visualization of the loss landscape of our integrated adversarial attacks for all the differentiable parameters w. Specifically, we view w_F and w_{RT} as two parameterized filters, which is analogous to the "filter normalization" technique proposed by [34]. In Figure 2(a), we adjust the initialization of the variance of w, which then can provide a distant view of loss landscape before the optimization in (8). It exhibits a highly regular loss landscape, and its non-concavity w.r.t. only rotation and translation [20] has been tremendously improved. In Figure 2(b), we then provide a closer view of the loss landscape before the optimization. It shows a highly convex surface around the w to be optimized, facilitating the following optimization. In Figure 2(c), we also present the loss landscape around the maxima w^* after the optimization in (8) of our integrated spatial attack, exhibiting a highly concave surface as well. In summary, the highly non-concave loss landscape concerning only rotation and translation raised by [20] has been largely alleviated by considering both local and global spatial vulnerabilities. This integrated form smooths the optimization process, which guarantees the efficacy of our integrated spatial attack.

2.4 Spatial AT

As Eq. (9) incorporates local and global spatial robustness simultaneously, it is natural to leverage it to construct spatial AT, which we deploy in Subsection 4.4.

3 Relationship between sensitivity and spatial robustness

In this section, we will empirically investigate the relationships between different robustnesses and then explain them from the perspective of shape-based representation by leveraging a saliency map.

3.1 Relationships

We conduct rigorous experiments on MNIST, CIFAR-10, and Caltech-256 datasets to empirically examine the behavior of local and global spatial robustness as the sensitivity-based robustness increases. Specifically, after adversarially training multiple PGD (sensitivity-based) robust models with different numbers of PGD iterations, we further compute their test accuracy under flow-based and RT-based spatial attacks via methods proposed in Section 2. The accuracy is computed on correctly classified test data for the model under consideration to mitigate the impact of the slightly different generalizations of these PGD-trained models. We fix both ϵ_F and ϵ_{RT} as 0.3 on MNIST, and choose ϵ_F and ϵ_{RT} as 0.3 and 1.0, respectively, on CIFAR-10 and Caltech-256. Then, we can control their strength of perturbations by adjusting the number of iterations in flow-based and RT-based spatial attacks.

In Figure 3, the X-axis shows adversarially PGD-trained models with different numbers of PGD iterations, which can measure the different strengths of a model's PGD (sensitivity-based) robustness. The Y-axis represents the computed test accuracy of the corresponding PGD-trained models under different spatial attacks, and a high-level test accuracy reflects a model's high spatial robustness. It turns out that flow-based spatial robustness (red lines) presents a steady ascending tendency across three datasets as the PGD sensitivity-based robustness increases, while the trend of RT-based spatial robustness (blue lines) fluctuates conversely. This result reveals that the sensitivity-based and RT-based

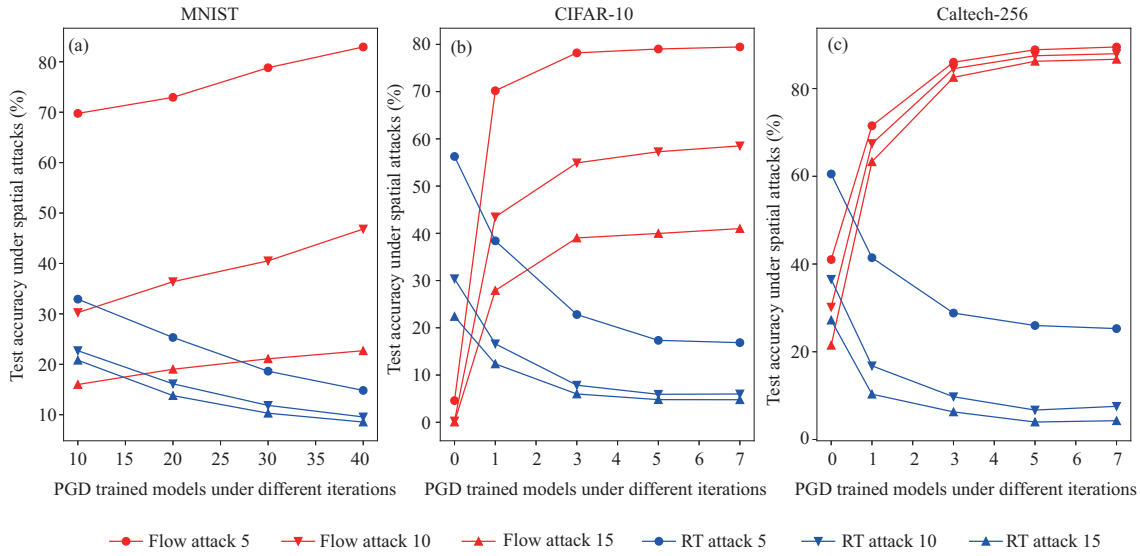

Figure 3 (Color online) Relationships between sensitivity and two spatial robustness for three datasets. (a) MNIST; (b) CIFAR-10; (c) Caltech-256. The X-axis represents adversarially PGD-trained models under different numbers of PGD iterations to measure the strength of sensitivity-based robustness, while the Y-axis represents the test accuracy under flow attack (red) and RT attack (blue) with different iterations to measure the spatial robustness.

spatial robustness is a trade-off relationship, consistent with the previous conclusion [25, 26]. However, this trade-off does not (even on the contrary) apply to the local spatial sensitivity, where sensitivity-based and flow-based spatial robustness is positively correlated. We provide strong and intuitive evidence from the perspective of shape-biased representation below.

3.2 Explanation from the shape-bias representation

We show first with our brief conclusion: the sensitivity-based robustness corresponds to the sparse and shape-bias representation [35,36], indicating that sensitivity-based robust models rely more on the global shape during prediction rather than the local texture. Nevertheless, the local and global spatial robustness are associated with different representation manners.

We visualize the saliency maps of naturally trained, PGD, flow-based, and RT adversarially trained models on some randomly selected images on Caltech-256, which are exhibited in Figure 4 to examine the shape-biased representation. Specifically, visualizing the saliency maps aims at assigning a sensitivity value, sometimes also called "attribution", to show the sensitivity of the output to each pixel of an input image. Following [35, 36], we leverage SmoothGrad [37] to calculate the saliency map $S(x)$ of an image x, which alleviates the noises in the gradient by averaging over the gradient of n noisy copies of an input

$$S(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial f_\theta^y(x_i)}{\partial x_i}, \tag{10}$$

where $x_i = x + q_i$, and q_i are noises drawn i.i.d from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$. In our experiment, we set $n = 100$ and the noise level $\sigma/(x_{\max} - x_{\min}) = 0.1$.

Figure 4 shows that PGD-trained models tend to learn a sparse and shape-biased representation for all pixels of an image, while two types of spatially adversarially trained models suggest a converse representation. In particular, the representation from the flow-based training model presents a noisy and shape-biased one as it places extreme values, although noisy, on pixels around the shape of objects, e.g., the edge between the horse and the background shown in Flow AT in Figure 4. On the contrary, RT-based models rely less on the shape of objects, and the saliency values tend to be dense, smoothly scattering around more pixels of an image.

We calculate the distance of saliency maps from different models across all test data on the Caltech-256 dataset and then compute their skewness in Figure 5. Specifically, we compute the pixel-wise distance between the saliency maps of the two models, and then we calculate the median of the skewness of the saliency map difference for all test data. Note that if two saliency maps have no statistical difference,

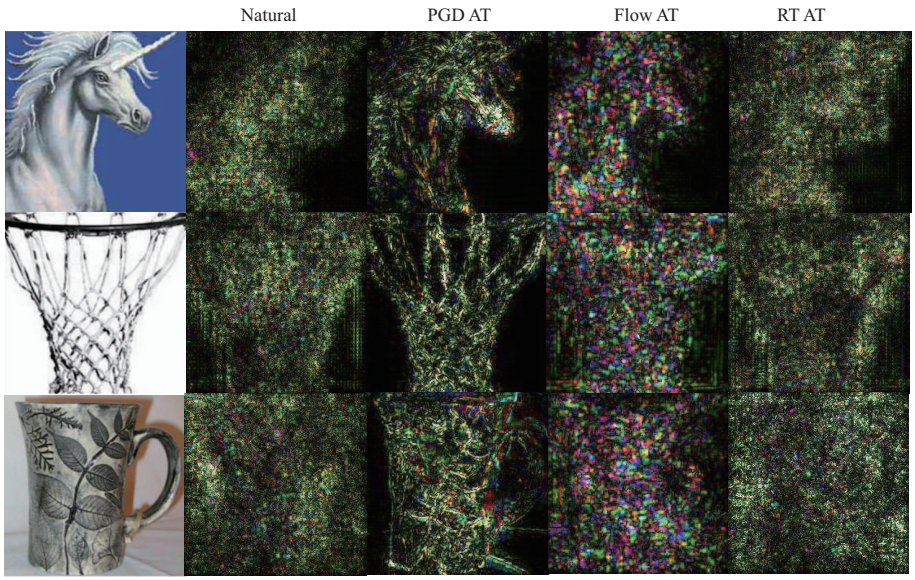

Figure 4 (Color online) Saliency maps of four types of training models on some randomly selected images on Caltech-256.

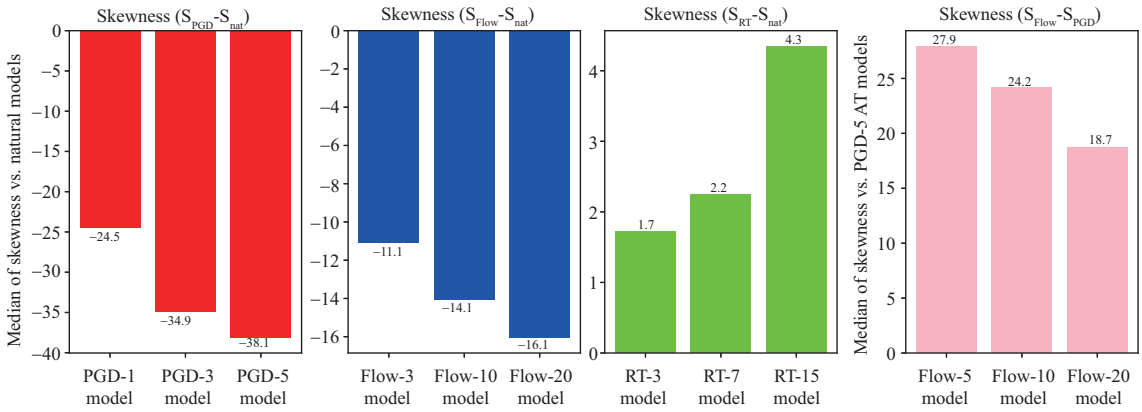

Figure 5 (Color online) Median of skewness of saliency maps difference among robust models across all test data as compared with other models. The first three sub-pictures are compared with the naturally trained model, while the last one is compared with the PGD-trained model.

then the difference in the values will follow a symmetric normal distribution with skewness 0. Negative skewness indicates that the original saliency map (representation) is sparse as compared to the model under consideration. We plot the tendency of skewness as the strength of some specific robustness increases in Figure 5. We summarize the observations into two conclusions.

(1) Based on the first and fourth sub-pictures, both PGD and flow-based robust models tend to learn a sparse and shape-biased representation compared with the natural model. However, the flow-based trained model is less sparse (we call it noisy shape-biased) in comparison with the PGD-trained one.

(2) In contrast, RT-based robust models tend to learn a dense representation. This is intuitive because the RT-trained model is expected to memorize broader pixel locations to cope with potential rotations and transformations in the test data.

Overall, the divergent representation (sparse vs. dense) between RT-based and sensitivity robustness verifies that the trade-off shown in Figure 3 is fundamental. More importantly, the positive correlation of sensitivity-based and local spatial robustness, shown in Figure 3, can also be explained by their similar shape-biased representation, although the latter tends to be noisy.

4 Pareto adversarial robustness

4.1 Motivation

Multi-objective optimization. Given the insights garnered from our analysis of the relationships between natural accuracy and different kinds of adversarial robustness, a natural question that comes up is how to design a training strategy that can perfectly balance their mutual impacts, which mainly results from their different representation manners. In most cases, their relationships exhibit trade-offs, except for the positive correlation between sensitivity robustness and local spatial robustness. We use $\mathcal{L}_{\text{nat}}$, $\mathcal{L}_{\text{PGD}}$, $\mathcal{L}_{\text{Flow}}$, and $\mathcal{L}_{\text{RT}}$ to represent the natural loss, the PGD adversarial loss, the flow-based and the RT-based adversarial loss, respectively. We cast obtaining universal adversarial robustness as well as maintaining natural generalization ability as a multi-objective optimization problem [38], encompassing all of the aforementioned losses with a loss vector:

$$\min_{\theta} \mathcal{L}^{\theta} = (\mathcal{L}_0^{\theta}, \mathcal{L}_1^{\theta}, \mathcal{L}_2^{\theta}, \mathcal{L}_3^{\theta})^{\mathrm{T}}, \tag{11}$$

where $\mathcal{L}_0^{\theta}$, $\mathcal{L}_1^{\theta}$, $\mathcal{L}_2^{\theta}$, $\mathcal{L}_3^{\theta}$ represent $\mathcal{L}_{\text{nat}}$, $\mathcal{L}_{\text{PGD}}$, $\mathcal{L}_{\text{Flow}}$, $\mathcal{L}_{\text{RT}}$ respectively for simplicity, sharing the same model parameter θ. The multi-objective optimization is to optimize all loss functions simultaneously by exploiting the shared knowledge and structure, e.g., the representation.

Pareto optimization. To harmonize these competing optimization objectives in the context of adversarial robustness, we introduce Pareto optimization [28, 39, 40], which is successfully applied when optimal decisions need to be taken in the presence of trade-offs between multiple conflicting objectives. Pareto optimization endeavors to achieve Pareto optimality, a balanced situation between all objectives, where none of the objective functions can be improved in value without degrading some of the other objective values. Mathematically, we have the following definitions [39, 41].

Pareto dominance in adversarial robustness. Let θ^1, θ^2 be two parameters in the space Ω. θ^1 dominates θ^2, i.e., $\theta^1 \prec \theta^2$, if and only if $\mathcal{L}_i^{\theta^1} \leqslant \mathcal{L}_i^{\theta^2}, \forall i \in \{0,1,2,3\}$ and $\mathcal{L}_j^{\theta^1} < \mathcal{L}_j^{\theta^2}, \exists j \in \{0,1,2,3\}$.

Pareto optimality. θ^* is a Pareto optimal point, and $\mathcal{L}^{\theta^*}$ is a Pareto optimal objective vector if it does not exist $\hat{\theta} \in \Omega$ such that $\hat{\theta} \prec \theta^*$. The resulting Pareto front contains all Pareto optimal solutions.

Pareto adversarial robustness. Based on the insights presented above, a natural approach for incorporating Pareto criteria into multi-objective optimization in the context of adversarial training is to achieve universal adversarial robustness as well as maintain a desirable natural accuracy. The resulting Pareto front contains all optimal, adversarially trained models for the given different constraints. The detailed formulation is presented later in Subsection 4.3.

4.2 Limitations of the existing strategies.

We denote $\mathcal{R}_{\text{adv}}(f; S_i) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\boldsymbol{r}\in S_i} \mathcal{L}(f(x+r), y)\right]$ as the adversarial risk under perturbation sets $S_i, i = 1, \ldots, m$. Our goal is to find f_θ that can achieve uniform risk minimization across all S_i as well as the minimal risk in the natural data. There are two common strategies to handle this issue.

(1) Average adversarial training (Ave AT) [25]**.** $\mathcal{R}_{\text{ave}}(f; S) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\frac{1}{m}\sum_{i=1}^{m} \max_{\boldsymbol{r}\in S_i} \mathcal{L}(f(x+r), y)]$, regards each adversarial robustness as having equal status. Intuitively, it may yield unsatisfactory solutions when the strength of different attacks mixed in the training are not balanced.

(2) Max adversarial training (Max AT) [25,42]**.** i.e., $\mathcal{R}_{\max}(f; S) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_i\{\max_{\boldsymbol{r}\in S_i} \mathcal{L}(f(x+r), y)\}]$ tries to optimize over the max loss from the largest perturbations.

Overfitting issue of Max AT. Intuitively, Max AT may overfit to one specific type of adversarial robustness if its adversarial attack used for training is too strong. In Figure 6, we plot the difference in robust accuracy between Max AT and single PGD adversarial training (PGD AT). It turns out that as the strength of PGD attack ϵ used in Max AT increases, the difference among the three kinds of robust accuracy between Max AT and a single PGD AT tends to vanish. This indicates that the comprehensive robustness of Max AT degenerates to a single PGD AT because the PGD loss tends to dominate as the strength of the PGD attack increases.

Overfitting issue of Ave AT based on its relationship with Max AT. We consider the generalization issue based on different risks and then set the risk in Max AT and Ave AT as $\mathcal{R}_{\max} = \max_i \mathcal{R}(f, S_i) = \max_i \mathcal{R}^{S_i}$ and $\mathcal{R}_{\text{ave}} = \frac{1}{m}\sum_{i=1}^{m} \mathcal{R}(f_\theta, S_i) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{R}^{S_i}$. Proposition 2 informs that Max AT is closely associated with some form of Ave AT. This indicates that Max AT is likely to per-

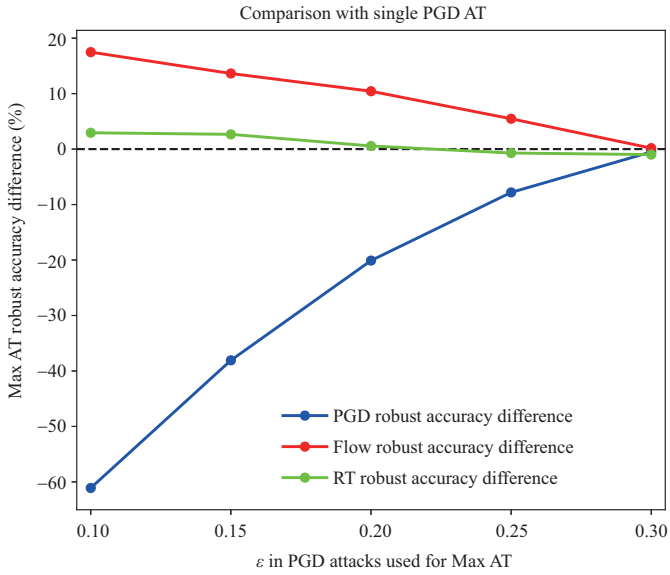

Figure 6 (Color online) Difference between the model trained by the PGD method and Max AT with different parameter ϵ for the PGD attack in the PGD AT.

form similarly to the specific form of Ave AT, which also suffers from unsatisfactory solutions when the strength of different attacks mixed in training is imbalanced. Proof can be found in Appendix C.

Proposition 2. Given KKT differentiability and qualification conditions, $\exists \lambda_i \geqslant 0$, such that the risk minimizer in Max AT, i.e., $\mathcal{R}^{\star}_{\max}$ is a first-order stationary point of $\sum_{S_i \in \mathcal{S}} \lambda_i \mathcal{R}^{S_i}$ regardless of the relationship of S_i.

Remark. We point out that both Ave AT and Max AT may suffer from the robustness overfitting issue and thus fail in certain scenarios. However, a clever combination choice among all involved adversarial losses has the potential to alleviate the overfitting issues, thus outperforming both Max AT and Ave AT in terms of universal robustness. Motivated by this, we propose Pareto AT in Subsection 4.3, which will provide strong empirical evidence to support this intuition.

4.3 Pareto AT

We apply linear scalarization to solve the multi-objective optimization, which is the most commonly used approach. We denote $\alpha = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ as the combination coefficients for various losses. Thus, the objective function is $\min_\theta \sum_{i=0}^{3} \mathbb{E}_x \left[\alpha_i^*(\theta)\mathcal{L}_i^\theta\right]$. Further, within the context of Pareto optimality, our goal is to find optimal combinations α between natural accuracy, sensitivity-based, and spatial robustness to perfectly balance their mutual impacts during the whole training process. Furthermore, we train a model f_θ under the optimal combinations α^* of different losses, and the computation of α^* in training is also associated with different losses determined by model parameters θ. This implies a bilevel optimization problem with θ as the upper-level variable and α as the lower-level variable. In the construction of low-level optimization regarding α, we apply a two-moment objective function concerning all losses. We name this bi-level optimization as Pareto AT, which is formulated as

$$\begin{aligned}
&\min_{\theta} \sum_{i=0}^{3} \mathbb{E}_x \left[\alpha_i^*(\theta)\mathcal{L}_i^\theta\right], \\
&\text{s.t. } \alpha^* = \arg\min_{\alpha} \sum_{i=0}^{3}\sum_{j=0}^{3} \mathbb{E}_x \left[(\alpha_i\mathcal{L}_i^\theta - \alpha_j\mathcal{L}_j^\theta)^2\right], r = \sum_{i=1}^{3} \alpha_i \mathbb{E}_x \left[\mathcal{L}_i^\theta\right], \sum_{i=0}^{3} \alpha_i = 1, \alpha_i \geqslant 0, \forall i = 0,1,2,3,
\end{aligned} \tag{12}$$

where r indicates the expectation of one-moment over all robust losses, i.e., spatial and sensitivity-based losses, which reflects the strength of comprehensive robustness we require after solving this quadratic lower-level optimization regarding α. In particular, given the model parameter θ in each training step,

Algorithm 1 Bi-level optimization in Pareto AT.

Input: Training data ($\mathcal{X}$, $\mathcal{Y}$). Batch size M and adjustable hyper-parameter r. Initialization of α as $[1/4, 1/4, 1/4, 1/4]$.
Output: Classifier f_θ.

repeat
  Sample $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M\}$ and $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ from all training data;
  / * Step 1: compute loss in (12) *** /**
  Compute natural loss $\mathcal{L}_{\text{nat}}$, and adversarial loss $\mathcal{L}_{\text{PGD}}, \mathcal{L}_{\text{Flow}}, \mathcal{L}_{\text{RT}}$ based on natural cross entropy loss, PGD loss and (2) and 6, respectively;
  / * Step 2: upper-level optimization over θ *** /**
  Given the current α, update f_θ by descending its stochastic gradient of

$$\frac{1}{M}\sum_{i=1}^{M}\mathcal{L}^{\text{CE}}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i) = \frac{1}{M}\sum_{i=1}^{M}\alpha_0\mathcal{L}_{\text{nat}}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i) + \alpha_1\mathcal{L}_{\text{PGD}}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i) + \alpha_2\mathcal{L}_{\text{Flow}}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i) + \alpha_3\mathcal{L}_{\text{RT}}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i);$$

  / * Step 3: lower-level optimization over α *** /**
  Compute $\hat{\mu}$ and $\hat{\Sigma}$ by sliding window technique in (13);
  Evaluate P in the quadratic form shown in (13);
  Solve (13) via CVXOPT tool to obtain the α;
until Convergence.

the larger r we require will push the resulting $\alpha_i, i = 1, 2, 3$ larger, thus increasing the weight of the robust losses rather than the natural loss to pursue more robustness.

Two-moment objective function. The two-moment form is a common practice in Pareto optimization. For example, in the financial portfolio theory, the mean-variance optimization is normally leveraged to compute the Pareto efficient front, where the risk of the asset portfolio, measured by their variances, is minimized to balance the different correlations of these assets given an expected return from the investor. Similarly, the square loss of the difference between each loss pair in (12) measures their mutual impacts. For instance, a decrease in $\mathcal{L}_{\text{PGD}}$ tends to increase $\mathcal{L}_{\text{RT}}$ as they have a fundamental trade-off relationship. We hope to mitigate all these mutual impacts, measured by the weighted quadratic differences, among all losses given an expected robustness level of r. In the implementation, as we regard all losses as random variables with their stochasticity arising from the mini-batch sampling from data, we leverage the sliding windows technique to compute their expectations. Our bi-level optimization within a batch is (1) θ: update parameters θ via SGD and (2) α: solve α via quadratic programming. Denote the random variables $\mathcal{L}_0$, $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ with mean vector μ and covariance matrix Σ. We transform our lower-level optimization regarding α as the following standard quadratic form:

$$\min_{\theta,\alpha}\ \alpha^{\text{T}}P\alpha \quad \text{s.t.} \quad \begin{bmatrix} 0 & \mu_1 & \mu_2 & \mu_3 \\ 1 & 1 & 1 & 1 \end{bmatrix}\alpha = \begin{bmatrix} r \\ 1 \end{bmatrix}, -\alpha \leqslant \mathbf{0}, \tag{13}$$

where $P = 8(\text{diag}(\Sigma) + \text{diag}(\mu\mu^{\text{T}})) - 2(\Sigma + \mu\mu^{\text{T}})$. We utilize the CVXOPT tool to solve this quadratic optimization within each mini-batch training. CVXOPT is probably the most popular free software package for convex optimization based on the Python programming language that can solve quadratic programming effectively. We also provide proof of the quadratic formulation in Appendix D.

A detailed algorithm description is given in Algorithm 1. In the lower-level procedure of Pareto AT, we solve the quadratic optimization regarding α given θ in each training step to obtain the optimal combinations among natural loss, sensitivity-based, and spatial adversarial loss. Then in the upper-level optimization, we leverage our familiar SGD method to update θ based on α^* calculated from the lower-level problem. Note that the computation complexity of our method is similar to Ave AT and Max AT, which are still competitive in computation.

4.4 Approximated Pareto front

By adjusting the upper bound of the expected adversarial robustness loss r, we can evenly generate Pareto optimal solutions where the obtained models will have different levels of robustness under optimal combinations. The set of all Pareto optimal solutions then forms the Pareto front. Rigorously, it is almost impossible to attain all Pareto optimal solutions for a general continuous multi-objective optimization problem unless a closed-form solution exists for each r. Alternatively, we leverage the limited solutions obtained by solving a series of multi-objective optimization problems for various r to approximate the Pareto front.

Table 1 Robustness score of each type of adversarial robustness on MNIST, CIFAR-10, and Caltech-256. Each type of robustness (%) is the average test accuracy under different strengths of perturbations. We choose the universal robustness of the natural model as the baseline and set it as 0. We use the difference in average test accuracy between other models and the natural model and then sum them as the universal robustness.

Dataset	Robustness score (%)	Natural model	PGD AT	Spatial AT	Max AT	Ave AT	Pareto AT ($r = 2.2$)
MNIST	Sensitivity-based robustness	29.40	98.42	0.24	65.16	92.70	88.06
	Local spatial robustness	14.36	38.23	27.59	53.02	48.51	58.70
	Global spatial robustness	16.70	12.77	78.76	51.47	88.76	90.40
	Universal robustness	0.0	88.97	46.14	109.19	169.52	**176.71**
Dataset	Robustness score (%)	Natural model	PGD AT	Spatial AT	Max AT	Ave AT	Pareto AT ($r = 4.0$)
CIFAR-10	Sensitivity-based robustness	0.82	70.24	12.11	52.04	49.68	51.65
	Local spatial robustness	9.34	72.29	83.63	85.31	79.88	80.38
	Global spatial robustness	57.28	18.94	40.96	40.06	64.98	66.36
	Universal robustness	0.0	94.04	69.26	109.98	127.11	**130.96**
Dataset	Robustness score (%)	Natural model	PGD AT	Spatial AT	Max AT	Ave AT	Pareto AT ($r = 6.0$)
Caltech-256	Sensitivity-based robustness	4.74	82.43	6.94	59.81	71.60	76.52
	Local spatial robustness	34.59	87.96	88.75	65.89	86.67	87.39
	Global spatial robustness	49.73	21.71	65.04	64.64	53.68	50.00
	Universal robustness	0.0	103.05	71.66	101.28	122.89	**124.85**

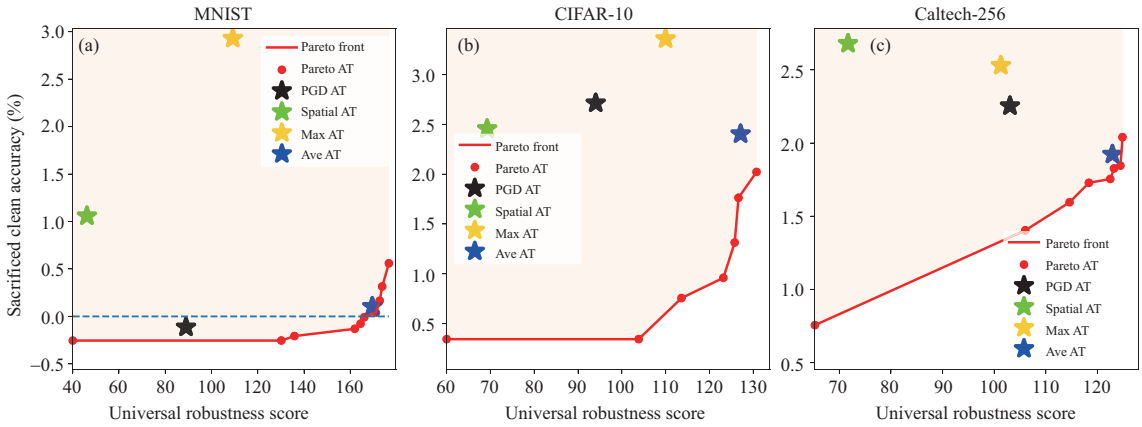

Figure 7 (Color online) Pareto front (red lines) between the universal robustness score and sacrificed clean accuracy on (a) MNIST, (b) CIFAR-10, and (c) Caltech-256. The vertical axis is the decrease of the natural accuracy compared with the naturally trained model and has been under the log transformation along two directions.

Thus, we train deep neural networks under different adversarial training strategies, i.e., PGD AT, spatial AT proposed in Subsection 2.4, Max AT, Ave AT, and Pareto AT under different r, in which we apply a proper iteration. Then we evaluate their test accuracy under PGD, flow-based, and RT attacks under different perturbation strengths. Next, we average the test accuracies for each type of attack, and the result is a quantitative measure of the specific robustness, called the robustness score. To evaluate the universal robustness, we further compute the average of robustness scores for all kinds of robustness and use the increment over the naturally trained model as the metric called universal robustness score. We report the robustness scores of all models on CIFAR-10 in Table 1, and the results on the other two datasets are similar. All implementation details are provided in Appendix E. It shows that Pareto AT ($r = 4.0$) has the best universal robustness score among all the models considered, although the highest specific robustness normally exists in the adversarial training model that only focuses on it.

Finally, we plot the universal robustness scores and the sacrificed clean accuracy of all methods across three datasets in Figure 7, where multiple Pareto AT models (red points) are trained under different r. The Pareto criterion exhibited in Figure 7 provides an optimality principle, which enables Pareto AT to achieve the best universal robustness among all the methods considered, given a certain tolerable level of sacrificed clean accuracy. By adjusting the expected universal robustness r in Pareto AT, we can develop the set of Pareto optimal solutions, i.e., the Pareto front. It shows that all other methods are above our Pareto front and are less effective than our proposal.

Overfitting issue of Ave AT. Note that although the perturbation strength adopted in Table 1 is

Table 2 Robustness score on CIFAR-10 with a larger step size 8/255 and ϵ as 16/255 in PGD perturbations used for both Ave AT and Pareto AT across different r.

Robustness score (%)	Natural model	Ave AT	Pareto AT ($r = 3.5$)	Pareto AT ($r = 3.7$)	Pareto AT ($r = 4.0$)	Pareto AT ($r = 4.2$)
Natural accuracy	91.43	56.39	82.64	79.69	71.68	61.53
Sensitivity-based	0.82	64.11	53.73	58.70	63.28	65.19
Local spatial	9.34	82.45	80.10	77.62	81.01	82.38
Global spatial	57.28	51.36	66.57	67.56	59.69	52.04
Universal robustness	0.0	197.92	200.39(+2.37)	203.88(+5.96)	**203.98**(+6.06)	199.61(+1.69)

mild, we need to point out that the superiority of Pareto AT over Ave AT can be higher if the overfitting issue is severe. We demonstrate this claim in Table 2, where we apply a stronger PGD perturbation used in AT. Finally, we find that Ave AT overfits sensitivity robustness more severely, achieving much less universal robustness and sacrificing more clean accuracy than Pareto AT. Pareto AT can mitigate the overfitting issue regarding an overly strong perturbation in AT because Pareto AT can automatically adjust the weights α while training, which is the key advantage of Pareto AT over Ave AT.

Sensitivity analysis. Comparing universal adversarial robustness between Tables 1 and 2, it can be seen that Pareto AT achieves more consistent universal adversarial robustness. In addition to this sensitivity analysis in terms of perturbation sizes, we also investigate the variation of universal adversarial robustness by changing the expected adversarial robustness loss r. Results are provided in Table 2. It suggests that Pareto AT with a mild r can achieve the best universal robustness score, while Pareto AT with an excessively large or small r may not have sufficient universal robustness. Moreover, Pareto front in Figure 7 also serves as the sensitivity analysis results in terms of different r.

Overall, we conclude that Pareto AT perfectly balances the mutual impacts of sensitivity-based robustness and spatial robustness under the Pareto criterion.

5 Discussion and conclusion

The principal purpose of our work is to design a novel approach to achieve universal adversarial robustness. We first analyze the two main branches of spatial robustness and then integrate them into one attack and adversarial training design. Furthermore, we investigate the comprehensive relationships between sensitivity-based and two distinct spatial robustnesses from the perspective of representation. Based on the understanding of the mutual impacts of different kinds of adversarial robustness, we introduce the Pareto criterion into the adversarial training framework to develop Pareto AT. The resulting Pareto front provides optimal solutions over existing baselines, given the universal robustness level we hope to attain. In the future, we hope to apply Pareto analysis to more general OoD generalization settings.

Acknowledgements Zhouchen LIN was supported by National Key R&D Program of China (Grant No. 2022ZD0160300), National Natural Science Foundation of China (Grant No. 62276004), and Qualcomm.

References

1 Vapnik V N, Chervonenkis A Y. On the uniform convergence of relative frequencies of events to their probabilities. In: Proceedings of the Measures of Complexity, 2015. 11–30
2 Krueger D, Caballero E, Jacobsen J H, et al. Out-of-distribution generalization via risk extrapolation (REx). 2020. ArXiv:2003.00688
3 Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: Proceedings of the International Conference on Learning Representations, 2014
4 Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks. 2013. ArXiv:1312.6199
5 Hendrycks D, Dietterich T. Benchmarking neural network robustness to common corruptions and perturbations. In: Proceedings of the International Conference on Learning Representations, 2019
6 Yin D, Lopes R G, Shlens J, et al. A Fourier perspective on model robustness in computer vision. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 13276–13286
7 Neal R M. Bayesian Learning for Neural Networks. New York: Springer Science & Business Media, 2012
8 Gal Y. Uncertainty in Deep Learning. Cambridge: University of Cambridge, 2016
9 Arjovsky M, Bottou L, Gulrajani I, et al. Invariant risk minimization. 2019. ArXiv:1907.02893
10 Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks. In: Proceedings of the International Conference on Learning Representations, 2018
11 Ding G W, Sharma Y, Lui K Y C, et al. Max-margin adversarial (MMA) training: direct input space margin maximization through adversarial training. In: Proceedings of the International Conference on Learning Representations, 2020
12 Ye N, Li Q, Zhou X Y, et al. An annealing mechanism for adversarial training acceleration. IEEE Trans Neural Netw Learn Syst, 2023, 34: 882–893
13 Hendrycks D, Basart S, Mu N, et al. The many faces of robustness: a critical analysis of out-of-distribution generalization. 2020. ArXiv:2006.16241

14 Liu Q, Wen W. Model compression hardens deep neural networks: a new perspective to prevent adversarial attacks. IEEE Trans Neural Netw Learn Syst, 2023, 34: 3–14
15 Che Z, Borji A, Zhai G, et al. SMGEA: a new ensemble adversarial attack powered by long-term gradient memories. IEEE Trans Neural Netw Learn Syst, 2022, 33: 1051–1065
16 Tramèr F, Behrmann J, Carlini N, et al. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 9561–9571
17 Xiao C, Zhu J Y, Li B, et al. Spatially transformed adversarial examples. In: Proceedings of the International Conference on Learning Representations, 2018
18 Szeliski R. Computer Vision: Algorithms and Applications. Berlin: Springer Science & Business Media, 2010
19 Engstrom L, Tsipras D, Schmidt L, et al. A rotation and a translation suffice: fooling CNNs with simple transformations. 2017. ArXiv:1712.02779
20 Engstrom L, Tran B, Tsipras D, et al. Exploring the landscape of spatial robustness. In: Proceedings of the International Conference on Machine Learning, 2019. 1802–1811
21 Sharif M, Bauer L, Reiter M K. On the suitability of L_p-norms for creating and preventing adversarial examples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018. 1605–1613
22 Zhang H, Yu Y, Jiao J, et al. Theoretically principled trade-off between robustness and accuracy. In: Proceedings of the International Conference on Machine Learning, 2019. 7472–7482
23 Tsipras D, Santurkar S, Engstrom L, et al. Robustness may be at odds with accuracy. In: Proceedings of the International Conference on Learning Representations, 2019
24 Raghunathan A, Xie S M, Yang F, et al. Understanding and mitigating the tradeoff between robustness and accuracy. In: Proceedings of the International Conference on Machine Learning, 2020
25 Tramèr F, Boneh D. Adversarial training and robustness for multiple perturbations. In: Proceedings of Advances in Neural Information Processing Systems, 2019
26 Kamath S, Deshpande A, Subrahmanyam K. Invariance vs. robustness of neural networks. 2020. ArXiv:2002.11318
27 Zhang H, Wang J. Joint adversarial training: incorporating both spatial and pixel attacks. 2019. ArXiv:1907.10737
28 Kim I Y, de Weck O L. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. Struct Multidisc Optim, 2005, 29: 149–158
29 Kim I Y, de Weck O L. Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. Struct Multidisc Optim, 2006, 31: 105–116
30 Zeleny M. Multiple Criteria Decision Making Kyoto 1975. New York: Springer Science & Business Media, 2012
31 Wang H, Chen T, Gui S, et al. Once-for-all adversarial training: in-situ tradeoff between robustness and accuracy for free. In: Proceedings of the 34th Conference on Neural Information Processing Systems, 2020
32 Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: Proceedings of IEEE Symposium on Security and Privacy, San Jose, 2017. 39–57
33 Jaderberg M, Simonyan K, Zisserman A, et al. Spatial transformer networks. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 2017–2025
34 Li H, Xu Z, Taylor G, et al. Visualizing the loss landscape of neural nets. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 6389–6399
35 Shi B, Zhang D, Dai Q, et al. Informative dropout for robust representation learning: a shape-bias perspective. In: Proceedings of the International Conference on Machine Learning, 2020. 8828–8839
36 Zhang T, Zhu Z. Interpreting adversarially trained convolutional neural networks. In: Proceedings of the International Conference on Machine Learning, 2019. 7502–7511
37 Smilkov D, Thorat N, Kim B, et al. Smoothgrad: removing noise by adding noise. 2017. ArXiv:1706.03825
38 Leung M F, Wang J. A collaborative neurodynamic approach to multiobjective optimization. IEEE Trans Neural Netw Learn Syst, 2018, 29: 5738–5748
39 Lin X, Zhen H L, Li Z, et al. Pareto multi-task learning. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 12060–12070
40 Li C, Georgiopoulos M, Anagnostopoulos G C. Pareto-path multitask multiple kernel learning. IEEE Trans Neural Netw Learn Syst, 2014, 26: 51–61
41 Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput, 1999, 3: 257–271
42 Maini P, Wong E, Kolter J Z. Adversarial robustness against the union of multiple perturbation models. In: Proceedings of the International Conference on Machine Learning, 2019. 6640–6650

Appendix A Visualization of various attacks

To better present the visual effect of various kinds of adversarial attacks, we provide high-resolution results on Caltech-256 in Figure B1. It turns out that flow-based attacks focus on local spatial vulnerability that mainly blurs pixels in some local regions, while RT attacks cause a shape-based global spatial transformation. More importantly, our integrated spatial attacks are more comprehensive in the sense of spatial robustness, combining both local and local spatial sensitivity.

Appendix B Proof of Proposition 1

Proof. Firstly, we have the following equations according to the definitions of the loss function:

$$\begin{aligned}\mathcal{L}_\theta^{\mathrm{CE}}(x_{w_F}, y) &= \log \sum_{i=1}^{K} \exp(f_\theta^i(x_{w_F})) - f_\theta^y(x_{w_F}),\\ \mathcal{L}_\theta^{S}(x_{w_F}, y) &= \log \sum_{i\neq y} \exp(f_\theta^i(x_{w_F})) - f_\theta^y(x_{w_F}).\end{aligned} \tag{B1}$$

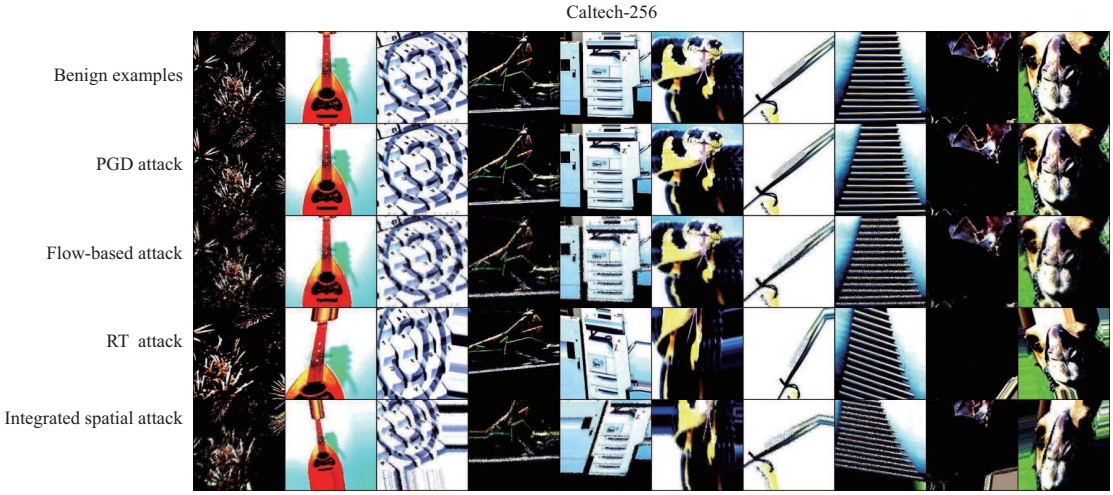

Figure B1 (Color online) High-resolution images on Caltech-256.

Then, we compute their gradients for the flow vector x_{w_F}. The gradient of $\mathcal{L}_\theta^{\mathrm{CE}}(x_{w_F}, y)$ is shown as follows:

$$\begin{aligned}
&\nabla_{w_F}\mathcal{L}_\theta^{\mathrm{CE}}(x_{w_F}, y)\\
&=\frac{\sum_{i=1}^K \exp(f_\theta^i(x_{w_F}))\cdot\nabla_{x_{w_F}} f_\theta^i(x_{w_F})\cdot\nabla_{w_F}x_{w_F}}{\sum_{i=1}^K \exp(f_\theta^i(x_{w_F}))}-\nabla_{x_{w_F}} f_\theta^y(x_{w_F})\cdot\nabla_{w_F}x_{w_F}\\
&=\frac{\sum_{i=1}^K \exp(f_\theta^i(x_{w_F}))\nabla_{w_F}x_{w_F}(\nabla_{x_{w_F}} f_\theta^i(x_{w_F})-\nabla_{x_{w_F}} f_\theta^y(x_{w_F}))}{\sum_{i=1}^K \exp(f_\theta^i(x_{w_F}))}.
\end{aligned}\tag{B2}$$

Similarly, the gradient of $\mathcal{L}_\theta^S(x_{w_F}, y)$ is

$$\nabla_{w_F}\mathcal{L}_\theta^S(x_{w_F}, y)=\frac{1}{\sum_{i\neq y}^K \exp(f_\theta^i(x_{w_F}))}\cdot\left(\sum_{i\neq y}^K \exp(f_\theta^i(x_{w_F}))\nabla_{w_F}x_{w_F}(\nabla_{x_{w_F}} f_\theta^i(x_{w_F})-\nabla_{x_{w_F}} f_\theta^y(x_{w_F}))\right).\tag{B3}$$

Then we take the multiplication of $\nabla_{w_F}\mathcal{L}_\theta^S(x_{w_F}, y)$ by a term $\frac{\sum_{i\neq y}^K \exp(f_\theta^i(x_{w_F}))}{\sum_{i=1}^K \exp(f_\theta^i(x_{w_F}))}$, finally we attain

$$\begin{aligned}
\nabla_{w_F}\mathcal{L}_\theta^S(x_{w_F}, y)\cdot\frac{\sum_{i\neq y}^K \exp(f_\theta^i(x_{w_F}))}{\sum_{i=1}^K \exp(f_\theta^i(x_{w_F}))}&=\frac{\sum_{i\neq y}^K \exp(f_\theta^i(x_{w_F}))\nabla_{w_F}x_{w_F}(\nabla_{x_{w_F}} f_\theta^i(x_{w_F})-\nabla_{x_{w_F}} f_\theta^y(x_{w_F}))+0}{\sum_{i=1}^K \exp(f_\theta^i(x_{w_F}))}\\
&=\frac{\sum_{i=1}^K \exp(f_\theta^i(x_{w_F}))\nabla_{w_F}x_{w_F}(\nabla_{x_{w_F}} f_\theta^i(x_{w_F})-\nabla_{x_{w_F}} f_\theta^y(x_{w_F}))}{\sum_{i=1}^K \exp(f_\theta^i(x_{w_F}))}\\
&=\nabla_{w_F}\mathcal{L}_\theta^{\mathrm{CE}}(x_{w_F}, y).
\end{aligned}\tag{B4}$$

Finally, we denote $\frac{\sum_{i\neq y}^K \exp(f_\theta^i(x_{w_F}))}{\sum_{i=1}^K \exp(f_\theta^i(x_{w_F}))}$ as $r(x_{w_F}, y)$.

Appendix C Proof of Proposition 2

Proof. Let f denote a function (such as a deep neural network), and let $f^\star$ be the minimizer obtained after optimization

$$\begin{aligned}
f^\star&\in\min_f\max_i\mathcal{R}(f, S_i),\\
M^\star&=\max_i\mathcal{R}(f, S_i).
\end{aligned}\tag{C1}$$

Then the optimization can be equivalent to a constrained version

$$\begin{aligned}
&\min_{f,M}\ M\\
&\text{s.t. } \mathcal{R}(f, S_i)\leqslant M \text{ for all } S_i\in\mathcal{S}
\end{aligned}\tag{C2}$$

with Lagrangian $L(f, M, \lambda)=M+\sum_{S_i\in\mathcal{S}}\lambda_i\left(\mathcal{R}(f, S_i)-M\right)$. If this optimization problem satisfies KKT condition, then

$\exists \lambda_i \geqslant 0$ with $\nabla_f L(f^\star, M^\star, \lambda) = 0$ such that

$$\nabla_f \big|_{f=f^\star} \sum_{S_i \in \mathcal{S}} \lambda_i \mathcal{R}(f, S_i) = 0.$$

Remark. We point out that our conclusion is made under the assumption that the KKT condition holds and the stationary point of f regarding the Lagrangian function can be attained, which normally requires the convexity condition. However, under these assumptions, we can still establish the close correlation between Max AT and Ave AT, indicating they are likely to perform similarly in many cases.

Appendix D Optimization analysis on the Pareto AT and algorithm

We provide the proof of P in the following.

Proof.

$$\begin{aligned}
\sum_{i=0}^{3}\sum_{j=0}^{3}\mathbb{E}(\alpha_i\mathcal{L}_i - \alpha_j\mathcal{L}_j)^2 &= \sum_{i=0}^{3}\sum_{j=0}^{3}\mathbb{E}((\alpha_i\mathcal{L}_i - \mathbb{E}(\alpha_i\mathcal{L}_i)) - (\alpha_j\mathcal{L}_j - \mathbb{E}(\alpha_j\mathcal{L}_j)) + (\mathbb{E}(\alpha_i\mathcal{L}_i) - \mathbb{E}(\alpha_j\mathcal{L}_j)))^2 \\
&= \sum_{i=0}^{3}\sum_{j=0}^{3}\mathbb{E}((\alpha_i\mathcal{L}_i - \mathbb{E}(\alpha_i\mathcal{L}_i) - (\alpha_j\mathcal{L}_j - \mathbb{E}(\alpha_j\mathcal{L}_j))^2 + (\mathbb{E}(\alpha_i\mathcal{L}_i) - \mathbb{E}(\alpha_j\mathcal{L}_j))^2 + 0 \\
&= \sum_{i=0}^{3}\sum_{j=0}^{3}(\alpha_i^2\sigma_{ii} + \alpha_j^2\sigma_{jj} - 2\alpha_i\alpha_j\sigma_{ij}) + (\alpha_i^2\mu_i^2 + \alpha_j^2\mu_j^2 - 2\alpha_i\alpha_j\mu_i\mu_j) \\
&= 8\alpha^{\mathrm{T}}\mathrm{diag}(\Sigma)\alpha - 2\alpha^{\mathrm{T}}\Sigma\alpha + 8\alpha^{\mathrm{T}}\mathrm{diag}(\mu\mu^{\mathrm{T}})\alpha - 2\alpha^{\mathrm{T}}(\mu\mu^{\mathrm{T}})\alpha \\
&= \alpha^{\mathrm{T}}(8(\mathrm{diag}(\Sigma) + \mathrm{diag}(\mu\mu^{\mathrm{T}})) - 2(\Sigma + \mu\mu^{\mathrm{T}}))\alpha.
\end{aligned} \tag{D1}$$

Appendix E Implementation

Implementation details. For MNIST comparison, we train the Simple CNN in [22] on MNIST for 100 epochs. As for the CIFAR-10 dataset, we choose the widely used Pre-Act ResNet-18 with grouped normalization and trained the network for 76 epochs. The other details of our implementation on MNIST and CIFAR-10 are based on [22], while the implementation on Caltech-256 has to refer to [36] with 10 epochs to finetune a pre-trained ResNet-18.

• **PGD attack.** We apply the widely accepted setting on these three datasets. We set step size as 0.01, ϵ as 0.3 on MNIST while the step size is 0.007 and ϵ is 0.031 on both CIFAR-10 and Caltech-256 datasets. To evaluate the different levels of robustness, we evaluate PGD attack under $10, 20, 30, 40$ iterations on MNIST and $5, 10, 15, 20$ iterations on CIFAR-10 and Caltech-256 datasets.

• **Flow-based and RT attacks.** On MNIST, we set step size α_F and α_{RT} as 0.01 and 0.1, and choose $\epsilon_F, \epsilon_{\mathrm{RT}}$ as 0.3. We select $5, 10, 15, 20$ as the attack iterations for the evaluation of both two attacks. On CIFAR-10, we set step size α_F as $1\mathrm{e}-3$ and α_{RT} as 0.05, and choose $\epsilon_F, \epsilon_{\mathrm{RT}}$ as $0.3, 1.0$. We select $3, 5, 10, 15$ as the attack iterations for the evaluation of both two attacks. On Caltech-256, we set step size α_F as $1\mathrm{e}-5$ and α_{RT} as 0.1, and choose $\epsilon_F, \epsilon_{\mathrm{RT}}$ as 0.3 and 1.0 for the two attacks, respectively. We select $3, 5, 10, 15$ as the attack iterations for the evaluation of both two attacks.

• **PGD AT.** We choose PGD iterations as 30, 3, and 5 in the PGD AT on MNIST, CIFAR-10, and Caltech-256, respectively. The adversarial attack strength is the same as PGD attacks for each dataset, respectively.

• **Spatial AT.** Our integrated spatial AT is based on our proposed integrated spatial attacks that unify both flow-based and RT-based attacks. We set the iterations as 20, 5, 10 and on MNIST, CIFAR-10 and Caltech-256, respectively. Other hyper-parameters are the same as those in their corresponding attacks.

• **Pareto AT.** The parameter r is the measure of comprehensive adversarial robustness. We select a sequence of r to train multiple Pareto AT models. Particularly, on MNIST, we choose r in $[0.2, 0.5, 0.8, 1.0, 1.2, 1.5, 1.8, 2.0, 2.2]$, and r in $[0.5, 1.0, 1.25, 1.5, 2.25,\ 3.0, 3.5, 4.0]$ on CIFAR-10 and Caltech-256. Other parameters follow the corresponding methods above, respectively.