Adversarial Adaptive Neighborhood With Feature Importance-Aware Convex Interpolation

Qian Li, Yong Qi, *Member, IEEE*, Qingyuan Hu, Saiyu Qi, *Member, IEEE*, Yun Lin, and Jin Song Dong

***Abstract*— Adversarial Examples threaten to fool deep learning models to output erroneous predictions with high confidence. Optimization-based methods for constructing such samples have been extensively studied. While being effective in terms of aggression, they typically lack clear interpretation and constraint about their underlying generation process, which thus hinders us from leveraging the produced adversarial samples for model protection in the reverse direction. Hence, we expect them to repair bugs in the pre-trained models by produced additional training data equipped with strong attack ability rather than time-consuming full re-training from scratch. To address these issues, we first study the black-box behaviors and the intrinsic deficiency of neighborhood information in previous optimization-based adversarial attacks and defenses, respectively. Then we introduce a new method dubbed *FeaCP*, which uses correct predicted samples in disjoint classes to guide the generation of more explainable adversarial samples in the ambiguous region around the decision boundary instead of uncontrolled "blind spots", via convex combination in a feature component-wise manner which takes the individual importance of feature ingredients into account. Our method incorporates the prior fact that for well-separated samples, the path connecting them would go through model's decision-boundary that lies in a low-density region, however, wherein adversarial examples are spread with high probability, thus having an impact on the ultimate trained model. In our work, the path is constructed by proposed inhomogeneous feature-wise convex interpolation rather than operating on sample-wise level, limiting the search space of *FeaCP* to obtain an adaptive neighborhood. Finally, we provide detailed insights and extend our method to adversarial fine-tuning using vicinity distribution to optimize the approximated decision boundary, and validate the significance of our *FeaCP* to model performance. The experimental results show that our method provides competitive performance on various datasets and networks.**

***Index Terms*— Decision boundary, deep neural networks, adversarial examples, feature interpolation.**

I. Introduction

FOR quite a few years, deep learning systems have persistently enabled significant improvements in many application domains, such as object recognition from vision, speech, and language and are now widely used both in research and industry. However, recent studies have shown that these systems are vulnerable to adversarial examples [1]–[4]: deep neural networks with nearly perfect performance provide incorrect predictions with very high confidence when evaluated on perturbations imperceptible to the human eye. This raises serious concerns about the security of deep learning models in many real-world applications, especially in security-critical tasks, including but not limited to autonomous vehicles [5], health care [6], and biometric recognition [7].

To this end, various defensive techniques against adversarial examples in deep neural networks have been proposed [1], [2], [8]–[15] and many remain vulnerable to adaptive attackers [3], [16], [17]. It was shown that adversarial training [2], [10], [14], [15] is by far the most popular and appears to hold the greatest promise for learning robust models. The key idea of adversarial training is to increase robustness by augmenting training data with adversarial examples. The initial and important step in adversarial training is to choose an attack model for adversarial example generation.

Although being successful, we have observed two limitations, viz., (i) Adversarial training relies on strong assumptions about the process for generating adversarial examples and conform to small neighborhood of data only, and suffers the distortion of the decision boundary for the reason that they only import adversarial examples against some specific types of attacks. (ii) The powerful optimization-based attacks lack clear interpretation and constraint about their underlying generation process which is often exposed as a black-box procedure emphasizing solely on the misbehaviours of neural networks, without incorporating any priors to regularize that generation or establishing a connection with the predictive pattern of models, although not all crafted samples are equally important to model performance. This hinders us use their most powerful aggression ability, which is also valuable to harden model in the reverse direction of attacking in our view.

Overview of Our Approach: In this work, the defects of optimization-based attacks are presented in terms of explainability about the generation of adversarial examples, we then introduce a method, feature-wise convex polytope attack referred to as *FeaCP*, to seek adversarial samples in the surroundings of decision boundary with much calibration on

Manuscript received June 12, 2020; revised September 14, 2020 and December 11, 2020; accepted December 14, 2020. Date of publication December 28, 2020; date of current version February 22, 2021. This work was supported by in part by the National Natural Science Foundation of China under Grant 61672421 and in part by the Blockchain Core Technology Strategic Research Program under Grant 2020KJ010801. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Matthieu R. Bloch. *(Corresponding author: Yong Qi.)*

Qian Li, Yong Qi, Qingyuan Hu, and Saiyu Qi are with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: qianlixjtu@163.com; qiy@xjtu.edu.cn; iloveavril914@stu.xjtu.edu.cn; syqi@connect.ust.hk).

Yun Lin and Jin Song Dong are with the School of Computing, National University of Singapore, Singapore 119077 (e-mail: llmhyy@gmail.com; dcsdjs@nus.edu.sg).

Digital Object Identifier 10.1109/TIFS.2020.3047752

existing regions of blind spots in neural networks, which focus primarily on enforcing constraint on locations of the produced samples, in order to control the generation for the purpose of model defense instead of attack ability purely. Compared to the linear interpolation on sample-level, our feature-wise polytope as adversarial neighborhood can cover more feasible adversarial space which considers the diversity of feature importance in data. *FeaCP* formulated as a constrained optimization problem, aims to find the optimal convex combination among the fine-grained features in the source sample and guide samples of disjoint classes, neither in the small nor large neighborhoods but being adaptive according to the inherent relative positions in input space. Then using the vicinity distribution of eventually found adversarial instances, we propose a simple adversarial fine-tuning technique to optimize the learned decision boundary to improve the robustness, even help model out from stucked local minima, whose effectiveness strongly depends on the distribution of adversarial examples w.r.t the decision boundary.

More specifically, we make the following main contributions in this paper

(i) We analyze the defects of the optimization-based generation of adversarial attacks, which lacks explanation about how and where it finds them, whether they are significant to the performance of the final trained model. Besides the *mixup* manner in which samples are interpolated, severely ignoring the feature importance but with all identical proportions is illustrated. It motivates us to propose an inhomogeneous feature-wise interpolation scheme to construct convex polytype as search space of adversarial samples in feature space, wherein local ambiguous buffer regions are contained crossing the decision boundary in principle.

(ii) The instances around learned decision boundaries are significant to the generalization capability of the model. Then we propose a simple but effective method called *FeaCP* to probe critical adversarial samples in that ambiguous area of the targeted model for defense purpose, using well-predicted instances in disjoint classes by the aforementioned feature fusion technique, which emphasize the diverse importance of features for learning tasks. Also, our method allows us to explore the large feature polytope with more flexibility in the optimization process.

(iii) Due that our produced samples are constrained nearby the decision boundary, their vicinal samples are supposed to be crucial to model performance. Utilizing the similarity of vicinity distribution, we further extend our method to adversarial fine-tuning so as to optimize the learned boundary to improve the robustness of without re-training of high computation cost.

(iv) We conduct extensive experiments to evaluate our method. The empirical results show that our proposed approach can significantly improve the robustness of various network architectures. In general, it shows tempting performance improvement on both clean and perturbed-data accuracy, in comparison to the other defense methods.

TABLE I
SYMBOLS AND NOTATIONS USED IN THIS PAPER

Symbol	Meaning
X	A set of input data (e.g., training set).
X'	Perturbed data distribution of X.
x	An instance of input data (usually a vector, a matrix, or a tensor).
x'	A modified instance of x, typically adversarial.
y, y'	The true class label, and label for the input instance x and its modified counterpart $x' \in X'$ respectively.
$\odot$, $\oslash$	The element-wise product, and dividend operators.
$\boldsymbol{I}_s$, $\boldsymbol{I}_g$	Source sample and guide sample.
$\boldsymbol{I}_g^i$	The i-th guide sample in $\boldsymbol{G}$.
y_{g_i}	The label of i-th guide sample in $\boldsymbol{G}$.
$\boldsymbol{\lambda}_s$, $\boldsymbol{\lambda}_g^i$	The convex combination coefficients for source and i-th guide sample in $\boldsymbol{G}$.
q	The superscript q indexes q-th component in tensor.
$f(x;\boldsymbol{\theta})$	machine learning model f with learnable parameter $\boldsymbol{\theta}$.
$\mathcal{C}(x)$	Predicted label of a classifier model, i.e., $\mathcal{C}(x) = \arg\max_{\mathfrak{c}} f(x)_{(\mathfrak{c})}$.
$\mathcal{L}(x, y)$	Loss function used to train a machine learning model.
$\nabla_x \mathcal{L}(x, y)$	Derivative of the loss function with respect to x.
ε	A scalar used to constrain the norm of noise.
$\boldsymbol{\omega}$	A weight vector of losses $\boldsymbol{\omega} = (\omega_1, \omega_2, ..., \omega_M)$.
$\boldsymbol{G} = \{\boldsymbol{I}_g^i\}_{i=1}^M$	A set of randomly chosen guidance samples, representing M different classes.

The remainder of this paper is organized as follows. In Section II, we introduce the background and related literature. The proposed method and its analysis are described in Section III. In Section IV, we evaluate the performance of proposed methods on standard datasets and neural network architectures. We conclude in Section V. All the symbols and notations used in this paper are summarized in Table I unless otherwise stated.

II. BACKGROUND AND RELATED WORK

Several state-of-the-art adversarial attack methods and defense methods, which will be investigated in this work, are briefly introduced as follows.

A. Adversarial Attacks

Enormous adversarial attack methods have been proposed to fool trained neural networks by introducing barely invisible perturbation on the input data. Here, we focus on two main categories: gradient-based attack and optimization-based attack.

The gradient-based attacks require the differentiability of targeted model. Then, by adding the perturbation along the direction of the gradient, the adversarial example is generated. The Fast Gradient Sign Method (FGSM) [2] calculates the sign of gradient of the loss function with respect to the input, then generates a small perturbation by multiplying a chosen scale factor. Based on that work, Madry *et al.* [10] leverages the randomly initialized basic iterative method (BIM) [18] and presents a multi-step variant FGSM which they refer to as Projected Gradient Descent (PGD). In [19] the momentum is integrated into the iterative update of gradient direction to boost adversarial attacks. Further, the authors extend it to attack an ensemble of models.

The optimization-based attack defines the attacking task as an optimization problem, whose purpose is to minimize the perturbation norm and make the DNN model misclassify adversarial examples simultaneously. The L-BFGS attack [1] is an earliest method designed to fool models, the goal of which is to find adversarial example x' as follows:

$$\begin{aligned} &\text{minmize } a \cdot \left\|x - x'\right\|_2^2 + \mathcal{L}(f(x'; \boldsymbol{\theta}), t) \\ &\text{s.t. } x' \in [0, 1]^n, \end{aligned} \tag{1}$$

where elements of x are normalized to $[0, 1]$, and t is the target misclassification label. Further, Carlini and Wagner [3] introduce a family of attacks to find adversarial perturbations that minimize diverse similarity metrics: L_0, L_2 and L_∞. The core insight is transforming the non-linear optimization problem with hard constraints similar to Eq. (1) into unconstrained optimization problem, by empirically selected surrogate loss function utilizing the change of variables.

B. Adversarial Defenses

1) Gradient Mask: The strategy of gradient obfuscation is applied to hide the true gradient information [20], thus sensitive direction of perturbation is not easy to construct accompanied with increased budget of attack solution. The gradient mask is very flexible and may happen at different stages of model execution, such as data compression and encoding in the data processing stage, randomization of ensemble models in test time [21], even with the stochastic activated units [8].

Yet, these methods come with the increase of model complexity and more importantly, the gradient mask is still not effective and reliable since an attacker can use the continuity of the model prediction to obtain approximated gradient information through a local substitute model by finite difference calculation [20] to achieve the purpose of intended attack. In fact, defenses that induce gradient obfuscation may not provide actual robustness [20], [22]. Therefore when evaluating a new defense, we should ensure it have not led this ill-behaviour.

2) Adversarial Training: Another common method is adversarial training [2], [23], first proposed by Szegedy *et al.* [1], which is by far the most popular defense approach. The key idea of adversarial training is to add adversarial examples to the original training data in each iteration, and use a weighted loss function for the two types of samples to optimize parameter updates [10]. Madry *et al.* [10] perform adversarial training using Projected Gradient Descent (PGD), a universal "first-order adversary" that follows the gradient of the model's loss function for multiple steps to generate an adversarial example. Tsipras *et al.* [24] identify a trade-off between the standard accuracy and adversarial robustness of a model, which stems from intrinsic differences between the feature representations learned by standard and robust models. In recent, TRADES [25] as an optimizing regularized surrogate loss, is proposed to encourage the algorithm to maximize the natural accuracy and to push the decision boundary away from the data, and the robustness and generalization are decoupled.

Adversarial training can be regarded as the optimization problem of the min-max saddle point [26] under the L_p norm metric on the input space. This method considers the model prediction in the worst-case of perturbation on the sample, to maintain the stability of model behaviours in local sense. However, adversarial training relies on the crafting method of adversarial samples, so it often has better defensive performance against specific types of attacks but without a guarantee to resist other attacks. Besides it increases training time by an order of magnitude over standard training making it difficult to scale to much larger networks. Several approaches [15], [27] have been proposed to make it fast without scarifying significant defense capability. On other hand, [14] tries to figure out its working mechanism by establishing a connection with the margin theory in traditional machine learning, and further propose a training method to directly maximize the classification margin of neural networks in input space.

III. Methodology

In this section, we first briefly review the optimization-based attacks and robust adversarial training for defense. We then describe our proposed attack, called *FeaCP*, and explain its connection with the decision boundary of the model. Finally, we extend our method to adversarial fine-tune to enhance robustness of model, validating the interpretation of generated adversarial samples indirectly by *FeaCP*.

A. Optimization-Based Adversarial Examples

The optimization-based attacks often formulate the generation of adversarial samples as a constrained optimization as follows.

$$\max_{\mathbf{x}'} \mathcal{L}(f(\mathbf{x}'), y), \text{ s.t. } \left\|\mathbf{x} - \mathbf{x}'\right\| \leq \epsilon. \tag{2}$$

The goal of Eq. (2) aims to maximize the misclassification probability via proxy loss function that measures the performance of model on chosen data under perceptibility budget ϵ, i.e., untargeted attack. When the objective is changed to minimize loss with respect to incorrect classes, it would become the so-called targeted attack, as shown in Eq. (3).

$$\min_{\mathbf{x}'} \mathcal{L}(f(\mathbf{x}'), y_t), \text{ s.t. } \left\|\mathbf{x} - \mathbf{x}'\right\| \leq \epsilon. \tag{3}$$

The solving processes of both Eq. (2) and Eq. (3) represent the search of adversarial samples in input space e.g., L-BFGS attack [1], noticeably, some of which have achieved state-of-the-art performance in terms of the aggression. Despite being stronger in comparison to gradient-like attacks, FGSM [2], JSMA [28], and PGD [10] for instance, in which the gradient information is used to find the perturbation direction, their high computational cost hinder us from finding more adversarial samples, preventing us ulteriorly from performing adversarial training or fine-tuning to enhance the security and reliability of deep learning models. In other puts, we can not efficiently use these powerful attacks to harden our training and deploying models in reverse research direction of defense since no derived critical samples can be obtained from the limited blind adversarial examples which are of no conjunction with the model robustness.

The major causes for the overhead of the optimization-based attacks are the high dimension of input and the laborious modulation for extra parameters such as Lagrange multipliers [26], to guarantee the generated instances $\mathbf{x}'$ to be adversarial. To make things worse, the crafted adversarial examples through which lack proper explanations not only about their existence but also the distribution of untouched blind area in training stage, which would make sense in helping us understand and mitigate the vulnerability of deep learning. For example, the L-BFGS attack minimizes Eq. (1), however the locations of the produced adversarial examples remain unknown in input space without any consideration about the learned classification boundary in its optimization process. So for each adversarial input, a similar optimization problem in Eq. (1) has to be solved repeatedly, which is clearly prohibited for the fast generation of massive adversarial examples for adversarial training or re-training. Therefore, we wonder if the region of produced adversarial examples can be connected to model's decision-boundary in its construction. Moreover if more adversarial candidates can be derived from single one hard optimization rather than one-to-one way so as to derive more representative samples from enhancing model performance.

B. Optimization-Based Robust Adversarial Training

Adversarial defense can be viewed as a distributionally robust optimization problem [26]. More explicitly, the training objective on the shifted distribution X' of the original X is written as:

$$\begin{aligned} &\arg \min_{\boldsymbol{\theta}} \sup_{X'} \mathbb{E}_{X'} \left[\mathcal{L}(x', y') \right] \\ &\text{s.t. } W_c(X', X) \leq \rho, \end{aligned} \tag{4}$$

where the constraint depicts the robustness region under the Wasserstein metric W_c. The above Eq. (4) is often solved in an approximate method such as Lagrange relaxation due to the intractability for arbitrary distance ρ and the hard constraint.

While performing better for small perturbation experimentally, i.e., in the small neighborhood around the input, the robustness in the large neighborhood is not well-guaranteed. According to [29], however, the information from larger neighborhoods, both in more directions and at greater distances is beneficial to the improvement of robustness, which indicates that the boundaries around those adversarial examples do not resemble the boundaries around the benign examples in their proposed OPTMARGIN method. Similarly, in another recent work [30], proved that robust adversarial training limited on the small balls and square areas in Eq. (4) is sample inefficient, and construct a Voronoi constraint to remedy the p-norm constraint.

So far, the study of the adversarial robustness has mostly been concerned with the setting of small vicinity of training samples or ϵ-adversaries, not considering the importance of samples relative to the form or learning of the model's decision boundary and the gap area of different classes, thus lead to sub-optimal or improper decision boundary for generalization. In such sense, it motivates us to search adversarial samples close by the complex decision boundary in terms of their impacts on learning, rather than confine us in the small perturbation range.

C. Proposed FeaCP

In this section, we introduce an interpretable white-box method called *FeaCP* to generate adversarial samples in high dimensional buffer zone that separates different classes, in conjunction with the decision boundary [31], neither focusing on low-dimensional subspace nor overconservative norm balls [10], [29], since both of which are non-adaptive to the diverse neighborhood distances. Our method can be formulated as an optimization problem, aiming to find more informative adversarial examples in areas that have strong impacts on the model's performance.

1) Inhomogeneous Feature-Wise Interpolation: Although the decision boundary is informative, unfortunately, the high non-linearity and complex transformation structures of deep neural network [29], [32] make its inaccessibility. Therefore, we propose to use well-classified samples being in disjoint classes to probe that ambiguous area by constructing feature importance-aware convex polytope, an embedded submanifold in input space, then in which to search representative adversarial examples.

In a nutshell, our method constructs candidate adversarial examples with constrained variables,

$$\boldsymbol{I}(\boldsymbol{\lambda}) = \boldsymbol{\lambda}_s \odot \boldsymbol{I}_s + \sum_{i=1}^{M} \boldsymbol{\lambda}_g^i \odot \boldsymbol{I}_g^i, \tag{5}$$

where $\boldsymbol{\lambda}_s$ and $\boldsymbol{\lambda}_g^i$ are tensors of coefficients with the same shape as the input, $\boldsymbol{\lambda} = \left\{ \boldsymbol{\lambda}_s, \boldsymbol{\lambda}_g^1, \cdots, \boldsymbol{\lambda}_g^M \right\}$ and subject to

$$\boldsymbol{\lambda}_s^q + \sum_{i=1}^{M} \boldsymbol{\lambda}_g^{iq} = 1. \tag{6}$$

In other words, each feature of synthetic samples is the convex combination of associated features in guide and source samples, and this manner can provide sufficient flexibility on the per-feature perturbation in the process of the seeking of blind spots in neural networks. Considering the diverse feature importance [33]–[35] which means different discriminative ability for learning task, we adopt the feature-wise manner a.k.a. inhomogeneous, instead of sample-wise combination, i.e., the same proportional linear interpolation for all features in input or latent space [36], [37], which search samples along the segment in input space as demonstrated in Fig. 1. The Feature-wise combination allows more dedicated control about the feature interpolation, and the optimizable coefficients can reflect the detailed information about the adversarial perturbation.

Mixup [36], [37] is mainly used as a data augmentation technique in training stage, however the purpose of our method is to find adversarial or critical examples nearby the decision boundary for adversarial defense. Moreover, Mixup is a sample-wise interpolation which neglects the individual attribution importance. In contrast, our method uses feature-wise

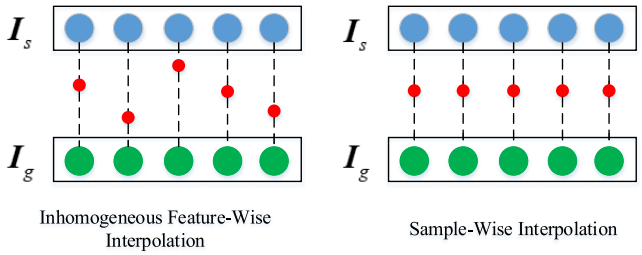

Fig. 1. A high-level illustration of the difference between inhomogeneous convex feature-wise interpolation and sample-wise interpolation. We depict convex combination coefficients whose range is [0, 1] in red dots and blue and green dots describe source and guide respectively.

Algorithm 1 Approximate Optimization of Eq. (11)

1: Initialization $\boldsymbol{\lambda}$ s.t. $\boldsymbol{\lambda}$ satisfy equality constraint Eq. (6)
2: Initialization $\boldsymbol{\omega}$ according to (i) or (ii)
3: **for** $j \leftarrow 1$ to $n_{iterations}$ **do**
4: $\quad \boldsymbol{\omega}_j \leftarrow$ normalize $\boldsymbol{\omega}_{j-1}$ using softmax
5: $\quad \boldsymbol{\lambda}_j, \boldsymbol{\omega}_j \leftarrow$ solve Eq. (11) using any optimizer for box constraint problem such as *L-BFGS-B* with limited update iterations
6: $\quad \boldsymbol{\lambda}_{j+1} \leftarrow$ normalize $\boldsymbol{\lambda}_j$ using weighted average
7: **end for**
8: **return** $\boldsymbol{\lambda}^\star = \boldsymbol{\lambda}_{j+1}$, $\boldsymbol{\omega}^\star = \boldsymbol{\omega}_{j+1}$

interpolation for the search samples in the ambiguous regions around the decision-boundary and optimizes the convex combination coefficients instead of pre-assigning the strength of feature fusion for all attributes homogeneously. In our method, the proposed interpolation scheme is used for path construction to apply the prior knowledge that path connecting the well-recognized samples being in disjoint classes cross the decision boundary, and constrain the locations of found blind spots of neural network in the convex polytope. The generated samples are then used to re-optimize the learned boundary locally. Despite its simplicity, it allows a strong explanation about the generation and existence of the adversarial examples in the notion of the decision boundary.

2) Proposed Objective: Before delving into the optimization objective, we formally state the definition of the adversarial direction and perturbation.

Definition 1 (Adversarial Perturbation): Let $\boldsymbol{I}_s$, $\boldsymbol{I}(\boldsymbol{\lambda})$ be source sample and its adversarial counterpart. Then the adversarial perturbation $\boldsymbol{p}$ is the difference between them, often, measured by ℓ_p-norm,

$$\boldsymbol{p} = \boldsymbol{I}(\boldsymbol{\lambda}) - \boldsymbol{I}_s. \tag{7}$$

Definition 2 (Sample Guided Adversarial Direction): Given a set $\boldsymbol{G}$, inhomogeneous feature-wise interpolation forms the guided adversarial direction $\boldsymbol{d}$

$$\boldsymbol{d} = \sum_{i}^{M}((\boldsymbol{\lambda}_g^i \oslash (1 - \boldsymbol{\lambda}_s)) \odot \boldsymbol{I}_g^i) - \boldsymbol{I}_s \tag{8}$$

which is the difference between the guide and source samples like in the vector space, parameterized by the combination coefficients $\boldsymbol{\lambda}$. Generally, the adversarial example will be found along that direction $\boldsymbol{d}$. To make Eq. (7) more sense and clear, expand the term $\boldsymbol{I}(\boldsymbol{\lambda})$, Eq. (7) then can be written as:

$$\boldsymbol{p} = \sum_{i=1}^{M} \boldsymbol{\lambda}_g^i \odot (\boldsymbol{I}_g^i - \boldsymbol{I}_s). \tag{9}$$

Thus the perturbation is shown to be constructed as a weighted sum of each individual guide direction $\boldsymbol{I}_g^i - \boldsymbol{I}_s$ in which $\boldsymbol{\lambda}_g^i$ depicts the adversarial noise on each feature of i-th sample in $\boldsymbol{G}$. It is noteworthy that the coefficients are to be optimized such that it reverts the prediction of model. Hence, they are the crux of our method.

Now, the search of adversarial example in the transitional area nearby the decision boundary can be represented as the following optimization problem:

$$\begin{aligned} \boldsymbol{\lambda}^*, \boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\lambda},\boldsymbol{\omega}} & \sum_{i=1}^{M} \omega_i \mathcal{L}(f(\boldsymbol{I}(\boldsymbol{\lambda}); \boldsymbol{\theta}), y_{g_i}) \\ \text{s.t.} \quad & \|\boldsymbol{I}(\boldsymbol{\lambda}) - \boldsymbol{I}_s\|_\infty \leq \epsilon, \end{aligned} \tag{10}$$

which is an adaptive weighted sum of loss over the label set of guide samples, within each proxy classification loss measures the degree that produced adversarial examples would be classified as the category represented by the corresponding label. More importantly, the weights $\boldsymbol{\omega}$ are purposely set to be optimizable so as to modulate the deviation biased to i-th guide sample. When $M = 1$, i.e. only one sample is used as guide, Eq. (10) can be compactly written as below,

$$\begin{aligned} & \underset{\boldsymbol{\lambda}}{\text{minimize}} \;\; \mathcal{L}(f(\boldsymbol{I}(\boldsymbol{\lambda}); \boldsymbol{\theta}), y_g) \\ & \text{s.t. } 0 \leq \lambda_j \leq \min\left(\frac{\epsilon}{|\boldsymbol{I}_{gj} - \boldsymbol{I}_{sj}|}, 1\right), \; \forall \lambda_j \in \boldsymbol{\lambda}, \end{aligned} \tag{11}$$

where $\boldsymbol{I}_{gj}$ and $\boldsymbol{I}_{sj}$ denote the j-th feature component in $\boldsymbol{I_g}$ and $\boldsymbol{I_s}$ respectively. And the general constraints in Eq. (10) become more specific to the optimizable variables which are box constraints. As such, Eq. (11) can be solved flexibly in approaches mentioned in [3]. The gradient of objective w.r.t $\boldsymbol{\lambda}$ can be computed as follows:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{I}(\boldsymbol{\lambda})} \odot \left(\boldsymbol{I}_g - \boldsymbol{I}_s\right). \tag{12}$$

In Eq. (12) the first term $\frac{\partial \mathcal{L}}{\partial \boldsymbol{I}(\boldsymbol{\lambda})}$ can be obtained through the automatic differentiation mechanism provided in current deep learning frameworks and is the main update signal in the optimization process since the second term that is the difference of source and guide is a fixed constant, determined by the selected guide. When $M \geq 2$ i.e., multiple samples in disjoint classes are used to probe the more complex crossing areas in classification boundary, instead of naively solving problem Eq. (10), an alternative formulation like in [3] is used but we have to deal with the extra constraints that limit the

perturbation on each feature:

$$\begin{aligned} &\underset{\boldsymbol{\lambda},\boldsymbol{\omega}}{\text{minimize}} \left\|\boldsymbol{I}(\boldsymbol{\lambda})-\boldsymbol{I}_s\right\|_\infty + c\sum_{i=1}^{M}\omega_i\mathcal{L}(f(\boldsymbol{I}(\boldsymbol{\lambda});\boldsymbol{\theta}), y_{g_i}) \\ &\text{s.t.} \sum_{i=1}^{M}\omega_i = 1,\ \boldsymbol{\lambda}_s^q + \sum_{i=1}^{M}\boldsymbol{\lambda}_g^{iq} = 1, \\ &\quad 0 \le \boldsymbol{\lambda}_s^q \le 1,\ 0 \le \boldsymbol{\lambda}_g^{iq} \le 1. \end{aligned} \tag{13}$$

where $c > 0$ is a chosen constant [3]. If there are no equality constraints, Eq. (13) then allows us to choose similar optimization solvers used in solving Eq. (11), since both would have the same optimization structure. Consequently, to ensure the updates on $\boldsymbol{\lambda}$ yields a valid convex combination, we use normalization techniques to tackle the equality constraints. However perform standard update procedure of optimization algorithm for box constraint problem with limited iterations, ignoring the equality constraints in Eq. (13) at first, then impose the validness of combination coefficients by normalizing the sum to 1, as shown in Algorithm 1 in which soft-max and weight average are respectively utilized for weights on losses and convex combination coefficients.

For each sample, the decision boundary distance to the most adjacent class relative to it is more crucial [38]–[40] as it determines the hardness to lead error classification. Further, the hardness to lead different targeted misclassification differs [28] because of the locations of samples on the data manifold are different. Weighting the losses in Eq. (10) can adapt to it using the structured distribution of initialized weights. Specifically, consider two cases:

(i) Sample M random values with one peak as targeted attack prior.
(ii) Sample M identical fair values for untargeted attack.

In the process of iterative solving Eq. (10), the soft-max is applied at the beginning of the iteration. Exponential non-linearity in the soft-max boosts the large pre-weights relatively to the smaller ones a lot [41], thus yields a stronger competition between pre-weights and a more targeted attack behavior. On the other side, as the M increases, it is possible that the resulting ability to more precisely single out a few top winning classes becomes increasingly crucial.

Essentially, the targeted class in the case (i) initially has more weight than others. If this prior bias is improper, in other words, the intended attack may not succeed, but our method can mitigate the embarrassment via updating weight bias automatically. Compared with the static weights, this manner can bring out more auto-exploration in the feasible region during the optimization process.

3) Insights Behind Our Designs: The key idea behind Eq. (10) is to implicitly utilize information of the model's decision boundary, to offer better explainability in the construction of adversarial examples based on optimization instead of post-hoc interpretation. In the most trivial case, two-class linear classifier $f(x) = \text{sign}(\boldsymbol{\omega}^T x + \boldsymbol{b})$, the boundary is a hyper-plane and the distance from any input x to it is $\left|\boldsymbol{\omega}^T x + \boldsymbol{b}\right| / \|\boldsymbol{\omega}\|_2$. No perturbation with ℓ_2 norm less than this distance possibly change f's prediction. In the general multi-classification, however, the minimal distance to local part of decision boundary that separating another class is the threshold of adversarial perturbation for target attack. Unfortunately, the boundary usually don't have explicit form for complex neural networks.

Optimal Decision Boundary Versus Global Minima: The optimality of the decision boundary means it should separate all unseen samples. In such sense, no adversarial samples exist, or no adversarial noise can be detected. Unfortunately, the optimal decision boundary $\mathcal{D}^*$ is usually unknown, which is uniquely decided by the unknown global minima $\boldsymbol{\theta}^*$ of model approximated on a finite set of data by minimizing the empirical risk. Because the neural network is highly non-convex with large parameter space, $\mathcal{L}$ often has multiple global minima and local minima but minimizing it using stochastic gradient methods converges to the closest one relative to the initial parameters $\boldsymbol{\theta}_0$ if not stuck in local minima [42]:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathcal{G}} \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|, \tag{14}$$

where $\mathcal{G}$ is the set of global and local minima, $\|\cdot\|$ is a norm-based distance measure such as L_2-norm. For already trained model, the resulting parameters $\tilde{\boldsymbol{\theta}}$ may not be optimal but have strong generalization on the test set, which indicates its corresponding boundary $\tilde{\mathcal{D}}$ is sufficiently okay, except some local parts that are required to be further optimized and fine-tuned for robustness enhancement, enhancing the resemblance of approximated decision boundary to the optimal version. Under delicate search in the buffer zone of the neural network, we could find the samples that should have been correctly classified but not yet, i.e., adversarial samples.

Probe the Critical Points: We indirectly gain the information about the model's decision boundary and the crossing buffer regions, i.e., the gap area that separates the classes, via the feature-wise combination of well-classified training or test samples in disjoint classes. Within these regions, the amount of mis-classified samples reflects the fine-grained quality of model's learned boundary with respect to $\mathcal{D}^*$ although both are generally similar. Using that, we construct a parameterized adversarial direction to guide us to discover informative points that significantly impact the performance and robustness of model [29], [38], [40], e.g., saddle points, support vectors. For instance, in the traditional SVM model, the support vectors uniquely decide the optimal hyper-plane and maximal margin. For pre-trained networks, making the weights $\tilde{\boldsymbol{\theta}}$ converge to the optimal $\boldsymbol{\theta}^*$ more thus depends on them in these regions and on which by fine-tuning, the boundary can be modulated such as pushing it away from the source or guide. Since those almost lie in that ambiguous region, we choose guide samples of different categories to restrict the feasible region for Eq. (10). Notably, multiple disjoint guide samples are used to detect the complex crossing regions about which we in fact have no idea, so as to find diverse adversarial samples that would helps more in the fine-tuning procedure for model defense.

Comparison With Previous Methods: Previous optimization-based methods like C&W produces adversarial samples in some unknown blind corner, regardless of whether produced samples are tied to some part of decision-boundary in more

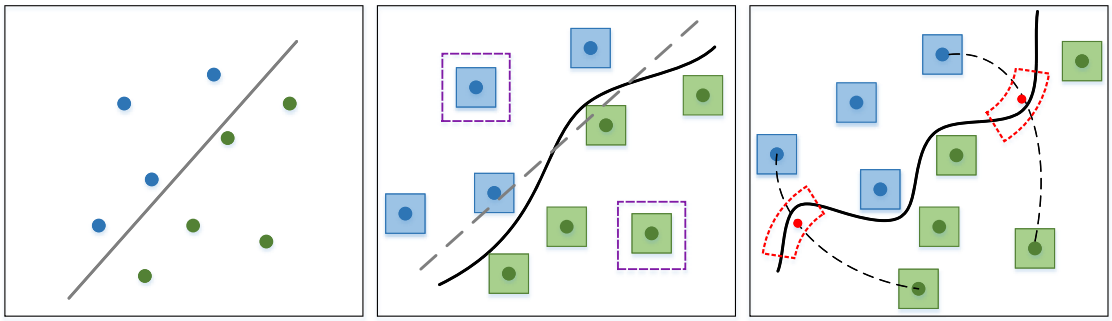

Fig. 2. A conceptual illustration of the proposed method. Left: A set of clean points that can be easily separated with a simple (in this case, linear) decision boundary. Middle: Conventional adversarial training approximates the adversarial decision boundary in small neighborhoods (here, ℓ_∞-balls) of data points while many data points (in purple squares) are not relevant to the model's performance without effectively constructing adaptive neighborhoods. Right: The red dots represent more decidable points obtained using FeaCP, which focuses on exploration in the decision regions denoted by the dotted red confined shape on the boundary of the targeted classifier. The dotted curves denote the search of adversarial examples with disjoint classes in adaptive neighborhood regardless of long or short distances, and FeaCP eventually finds the critical points around the boundary which have strong correlations with model's performance such as adversarial robustness.

details. By contrast, our method provides explainable hints on the positions of adversarial examples by the adversarial direction, explicitly taking the significance of adversarial examples respect to the targeted model into consideration in their construction, e.g., whether they are crucial to model's performance. Meanwhile, the direction related to the guides which represents different categories could locate different parts of boundary needed to be optimized. This characteristic is important to the success of fine-tuning since it can help us check and optimize the boundary widely. Apart from that, the resulted fusion coefficients suggest the individual importance of feature components in guides in terms of maliciousness to targeted source when the guides are viewed as reference, and the displacement of the source towards the nearest part of boundary w.r.t. the guide with highest weight would be relatively clear.

Opposed to FGSM-like methods such as PGD attack, we do not try first to find the most sensitive perturbation direction, then to craft adversarial samples with a modulation of small-scale noise factor such that the loss is maximized after perturbation, i.e., the new sample crosses the decision boundaries of the model to a new classification region. *FeaCP* combines the two separate steps into one. Specifically, the direction and magnitude of perturbation are jointly optimized and found, both are embedded in the combination coefficient. On the other hand, FGSM-like methods conform the search space of adversarial examples on local neighborhood solely without considering the decision-boundary information explicitly in their craft procedure, having not tried to search adversarial example in the transitional areas of classes we care about. However, using the assigned guide samples, we instead probe adversarial samples inside them by constructing the convex polytope and expect to obtain more decidable points either in the large or small neighborhood of source sample $\boldsymbol{I}_s$, determined by their inherent relative distance. That is to say, our approach can achieve different neighborhood explorations with adaptive distance respect to the source and needs not to estimate the distance to decision boundaries like using random search direction [29]. To gain intuition, the above approach is illustrated in Fig. 2.

D. Adversarial Fine-Tuning

The distribution of adversarial samples P_{adv} in the gap area is unknown but can be approximated using the mixtures of Gaussian vicinities [37], [43] centered at the produced informative adversarial sample because they have similar semantic information and should distribute densely in the feature space. Leveraging the similarity of samples in vicinity distribution, we can produce more candidate adversarial data points so as to cover the adversarial manifolds embedded in the neighborhood of decision boundary:

$$P_{adv}\left(\tilde{x}, \tilde{y}\right) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{N}\left(\tilde{x} - x_{adv}^{i}, \sigma \mathbf{I}\right) \delta\left(\tilde{y} = y_s^i\right), \quad (15)$$

where the $\mathcal{N}(\cdot)$ is normal distribution, $\delta(\cdot)$ is a Dirac mass function, and y_s^i, x_{adv}^i is the i-th label of source benign sample and adversarial counterpart.

By retraining or fine-tuning the model on P_{adv}, the learned decision boundary can be optimized in a local manner, thus bear more similarity with the optimal classification boundary, as such the robustness of the protected model can be improved. To employ the auto-exploration in optimization, a fair initialization for $\boldsymbol{\omega}$ is used and the adversarial fine-tuning algorithm is described in Algorithm 2, in which the sampled data set $\mathcal{S}$ including all classes is to ensure data diversity, in order to modulate different local parts of the learned decision boundary, and the inner loop is specific to adversarial examples generated by *FeaCP*.

IV. Evaluation

A. Experiment Setup

Datasets: We use two popular academic image classification datasets and a sentiment analysis dataset for our experiments: MNIST, consisting of black-and-white handwritten digits [44], and CIFAR-10, consisting of small color pictures [45]. In MNIST, the images' pixel values are in the range [0, 1]; in CIFAR-10, they are in [0, 255].

- MNIST [44]: This is a large dataset of handwritten digits commonly used for training various image processing systems. It is composed of 70,000 greyscale images

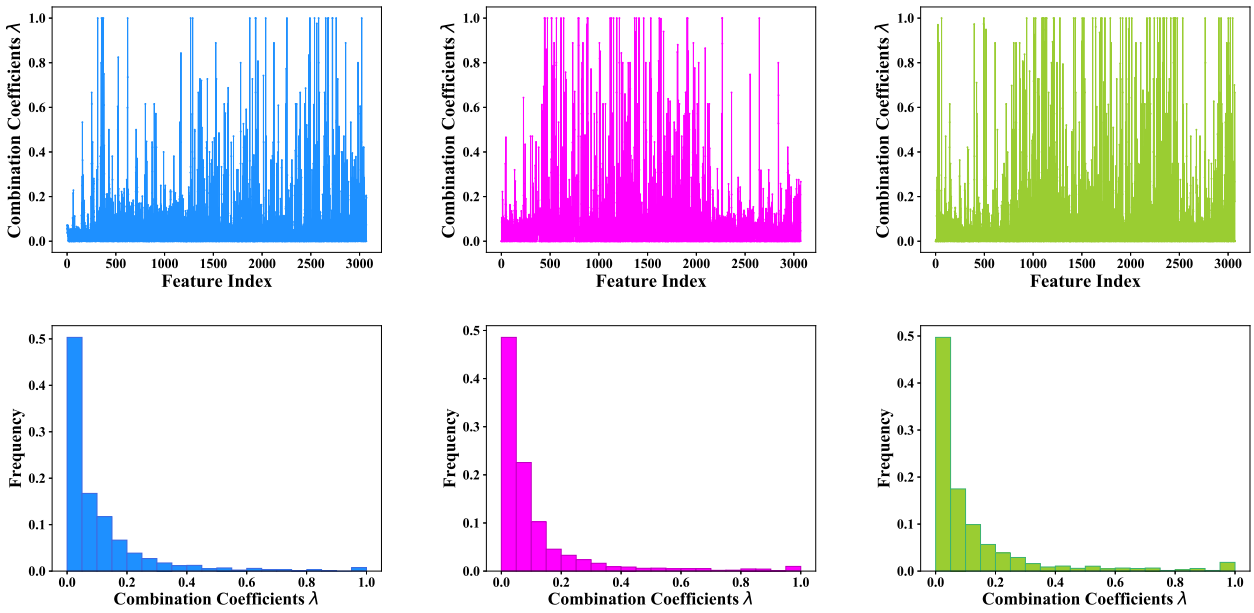

Fig. 3. The combination coefficients $\boldsymbol{\lambda}$ (above) on feature components and the corresponding frequency distribution histogram (below) of three different samples on the CIFAR-10.

Algorithm 2 Adversarial Fine-Tuning

Sample a data set $\mathcal{S}$ covering all classes ▷ Ensure data diversity.
for $k \leftarrow 1$ to $|\mathcal{S}|$ **do**
  $\boldsymbol{I}_s \leftarrow \mathcal{S}[k]$
  for $j \leftarrow 1$ to $n_{fine\_tuning\_itemtions}$ **do**
    $\phi_k = \{\}$
    Sample M instances being M different classes from $\mathcal{S}[k]$
    $\boldsymbol{\lambda}_k^* = \arg\min_{\boldsymbol{\lambda},\boldsymbol{\omega}} \sum_{i=1}^{M} \omega_i \mathcal{L}(f(\boldsymbol{I}(\boldsymbol{\lambda}); \boldsymbol{\theta}^j), y_{g_i})$ ▷ Initialize weights with uniform values and use $\mathcal{S}[k]$ as source.
    Sample m points from $\mathcal{N}\left(\tilde{x} - \boldsymbol{I}(\boldsymbol{\lambda}_k^*), \sigma \mathbf{I}\right)$ and append them to ϕ_k
    Fine-tune model $f(\boldsymbol{\theta}^{(j)})$ on ϕ_k
  end for
end for

(of 28×28, or 784, pixels) of handwritten digits, split into a training set of 60,000 samples and a testing set of 10,000 samples.

- CIFAR-10 [45]: The CIFAR-10 dataset introduced by Krizhevsky is a popular image classification dataset containing 50, 000 training images, together with a 10, 000 test set. Each image is of size 32×32 and belongs to one of 10 classes of vehicles and animals.
- IMDB [46]: This dataset consists of 50,000 movie reviews, with one half labeled as "positive" and the other "negative", indicating the sentiment of these reviews.

Details of the Experimental Setting: To evaluate the performance of our proposed method, we employ multiple powerful white-box attacks, as described in Section II-A. For PGD attack, ϵ is set to $0.3/1$ and $8/255$ on MNIST and CIFAR-10, respectively. For MNIST, the PGD adversary for training has 40 iterations with step size 0.01. For CIFAR-10, We run 7 iterations of PGD attack as an adversary, with a step size of 2. FGSM attack uses the same ϵ setup as PGD, and the attack configurations of PGD and FGSM are same as the setting in [9], [47]. For C&W attack, the constant c is set to 0.01. The attack configurations of L-BFGS and C&W follow the settings in the paper of [1], [3]. Throughout our experiments, for each one of the models, unless explicitly stated, we tested our approach using the original setting of models' structures and parameters described by its authors. To support our claim, we did not change the learning rate used nor the number of epochs. Note that, for all the results, the reported performance value (accuracy or error rate) is the average with 5 trials to alleviate error.

Competing Methods for Adversarial Defense: As we all know, PGD adversarial training [10] is the only unbroken defense method [20], and this work uses this as a baseline. In addition, several recent works are proposed to defense adversarial examples as well, including parametric noise injection on weight (PNI-W) [48], adversarial fine-tuning with L-BFGS attack [1], and adversarial fine-tuning with C&W attack [3].

B. Combination Coefficients

The combination coefficients reflect the degree of feature fusion along each direction of feature element in guide samples. Generally, they have different scales which depends on the feature importance regarding the adversarial perturbation, and large values indicate the larger perturbation needed on the original feature such that model makes incorrect prediction relative to the features in the guide, rather than from absolute view.

As a result, the fusion coefficients are supposed to be different. This can be validated from Fig. 3 (above) for

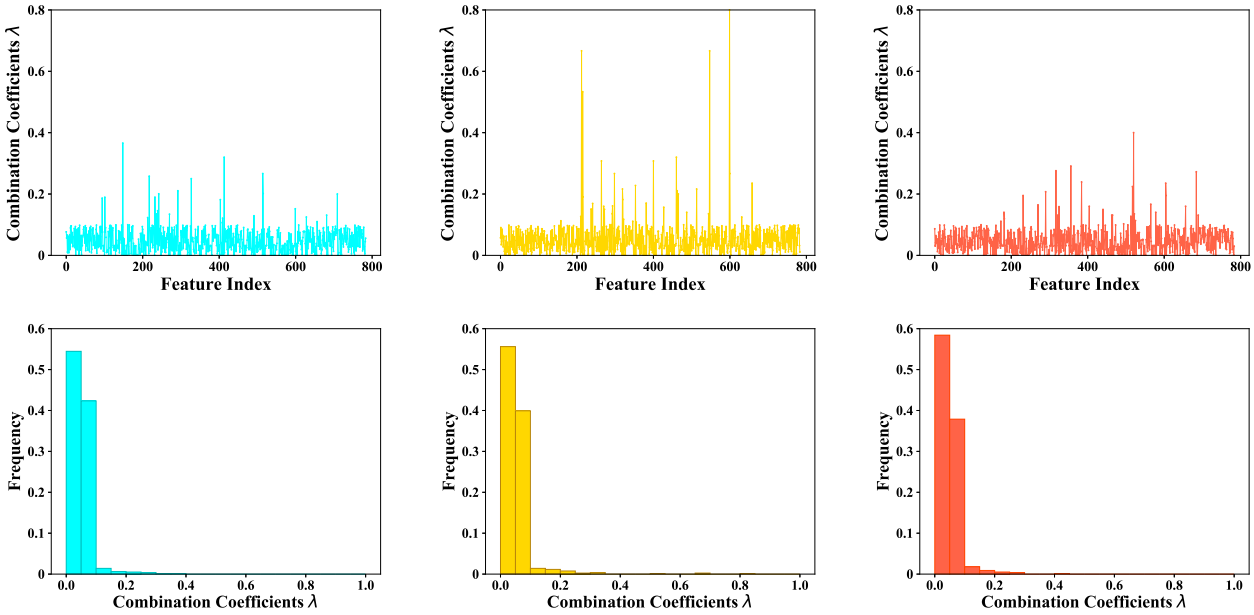

Fig. 4. The combination coefficients $\boldsymbol{\lambda}$ (above) on feature components and the corresponding frequency distribution histogram (below) of three different samples on the MNIST.

CIFAR-10 and Fig. 4 (above) for MNIST. As seen, the fusion coefficients mostly have different values, some of which approach the saturated upper bound value but others are very small, even zero. For MNIST, only a few feature components have high fusion values indicated by the spikes in Fig. 4 (above) despite that they are initialized with the same value. For CIFAR-10, similar behaviors for the convex interpolation coefficients have shown. The statistical distributions of coefficients are presented in Fig. 3 (below) and Fig. 4 (below). Surprisingly, both exhibit obvious similar patterns wherein most of feature ingredients are with small interpolation coefficients. On MNIST, there are barely any components with mixed values that are greater than 0.5, and the frequency of these values on CIFAR-10 is also low, which, however, may have more impact on the construction of adversarial examples in our method.

Indeed, components in a feature are not equally important for learning tasks and do not make same contributions to model prediction [49]. Naturally, we conjecture that not all feature perturbations is equally important to adversarial vulnerability which is the motivation of that we propose to use feature-wise interpolation to construct a path crossing the decision-boundary to characterize the individual perturbation. And the empirical results of coefficients clearly reinforce our argument. On the other hand, since the locations of selected guides and sources differ in input space, relative to learned classification boundary in particular, i.e., the distances between different types of feature elements vary. In such sense, the coefficients indicate where the decision boundary is locally in the corresponding dimension, whose value suggests distance in that direction of attribution. If it is large, we can refer that in part the boundary is closer to the guide than source in that dimension. Thus, the coefficients provide the location information about the learned implicit boundary with the help of samples being in disjoint classes. On the contrary, the previous optimization-based craft methods place the attack capability as the first role, without considering the locations of eventually found samples. When the samples are viewed alone, their decision boundary distance information is hard to be obtained.

Typically, the coefficients form a convex polytope that contains the ambiguous region around the decision boundary, as illustrated by the dotted red confined shape in Fig. 2. In essence, the solving process of combination coefficients in Eq. (10) is identical to seek the representative adversarial example in that buffer zone of the model, indicated by the red points in Fig. 2, which guarantees the significance of derived samples to model performance intuitively. During the solving process of Eq. (10), the degree of interpolation shall be changed according to the feature importance in order to minimize the objective. The produced corresponding adversarial examples are shown in Fig. 5. Compared with others, we do not minimize the perturbation because we aim to find the adversarial perturbation such that the objective function is maximized. Consequently, the perturbation on each feature element is made as large as possible in order to increase the value of objective function as seen from Fig. 5. Thus, when the generated samples are used to locally optimize the already learned decision boundary, the margin between classes can be enlarged.

C. Impact of Hyper-Parameters

As mentioned in Section III-C, the adversarial fine-tuning with *FeaCP* has one important hyper-parameter: the number of different classes M in Algorithm 1. We implement a group of experiments to quantify the effect that the parameter has on model classification accuracy. Specifically, we allow the para-

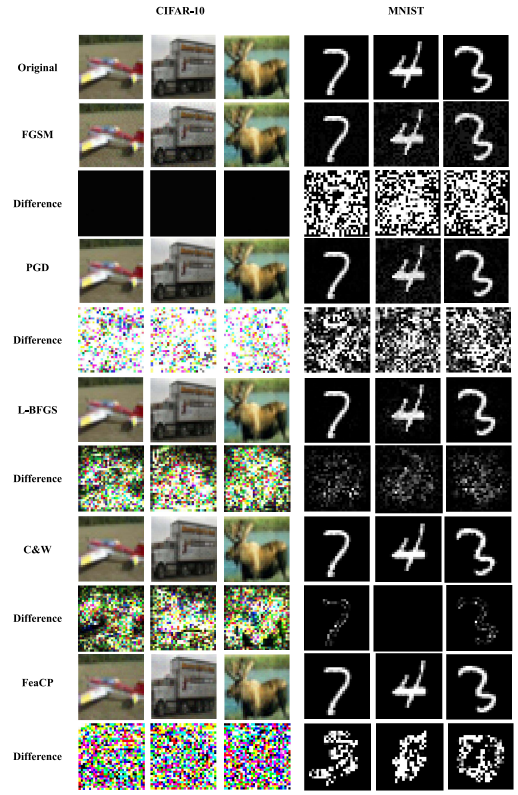

Fig. 5. Adversarially perturbed images are generated by different attack methods and their corresponding original images on CIFAR-10. The difference in the bottom row of each attack method shows the absolute difference between the original image and the adversarial examples.

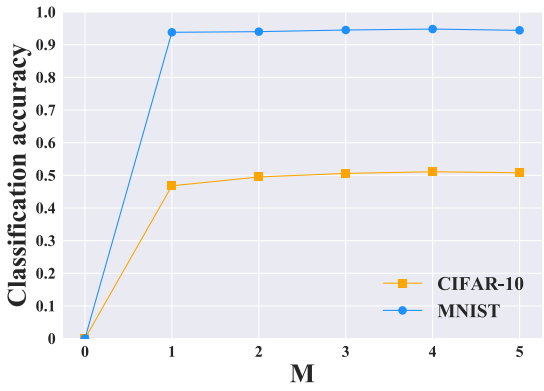

Fig. 6. Performance result with different parameter M under PGD attack on MNIST and CIFAR-10.

meter M over the ranges $M \in \{0, 1, 2, 3, 4, 5\}$ and following the experiments implemented in [48], the LetNet5 and ResNet-20(4×) [50] architecture are used on MNIST and CIFAR-10, respectively.

We evaluate the adversarial accuracy of the fine-tuned model under the PGD attack with different parameters M. The exact verification results can be seen in Fig. 6, where we find that PGD attack successfully evades standard training model, i.e., when $M = 0$. In contrast, the fine-tuned model with produced adversarial examples by *FeaCP* significantly outperforms the original model, and the model accuracy afforded by our method achieves 94.82% and 51.14% on MNIST and CIFAR-10, in the scenarios of the $M = 4$. Concretely, the test accuracy is actually never lower than 93.82% and 46.87, and the average test accuracy is 94.31% and 49.76% with the different M on the two datasets, respectively. This demonstrates that adversarial fine-tuning with *FeaCP* dramatically improves the robustness to resist PGD attack, which is the most powerful gradient-based attack now. From Fig. 6, we can see that the number of guide samples is increasingly important to produce more critical samples in the adversarial convex polytope suggested by the improvement of adversarial robustness, which is contributed to the broader convex area in the buffer zone around complex decision boundary, but converges to 4, than have no pleasurable gain in defense capability. Therefore, we adopt 4 as parameter M in Algorithm 1 for following experiments.

D. Comparison With State-of-the-Art Defence Approaches

Adversarial Fine-Tuning With Optimization-Based and PGD Attacks: To illustrate the advantage of the proposed adversarial fine-tuning with *FeaCP*, we now apply our proposed approach to tune the pre-trained classifiers. We first perform a series of experiments to compare common approaches of adversarial fine-tuning with optimization-based and PGD attacks, i.e., L-BFGS [1], C&W [3], and PGD attacks. The regimes of LetNet5 and ResNet-56 [50] architectures are consistent with [48] on MNIST and CIFAR-10.

As observed in Table II, adversarial fine-tuning with *FeaCP* outperforms others under the most aggressive adversarial examples – PGD attack. In particular, the performance of our approach is actually never lower than 94% and 47% under the PGD attack on the MNIST and CIFAR-10, respectively. In contrast, the accuracy of adversarial fine-tuning with L-BFGS is compromised to a great extent with accuracy lower than 43% under the attack of PGD on CIFAR-10. The accuracy of adversarial fine-tuning with C&W achieves 92.25% under PGD attack on MNIST.

This behavior indicates that adversarial fine-tuning with *FeaCP* achieves impressive practical performance under the PGD attack due to our approach can achieve different neighborhood explorations with adaptive distance respect to the source sample to cover that adversarial manifolds embedded in the neighborhood of decision boundary. However, L-BFGS and C&W don't explicitly constrain the locations of the generated adversarial samples, which may be non-correlated with the classification boundary, thus be of no benefit to model performance.

In addition, we can observe that our method significantly outperforms PGD adversarial fine-tuning since PGD generated samples are not well connected to decision-boundary and those samples far from the already learned boundary have little impact on model performance in the process of fine-tuning. However, in our method, we focus on probing the ambiguous regions around the boundary with adaptive neighborhood directly.

Other Recent Adversarial Defense Methods: As discussed in Section II-B, a large number of adversarial defense works have

TABLE II
CLASSIFICATION ACCURACY ON ADVERSARIAL AND CLEAN TEST SAMPLES

Dataset	Defense method	Clean	PGD
MNIST	Standard training	99.14%	0.51%
	L-BFGS adv. fine-tuning	98.56%	91.53%
	C&W adv. fine-tuning	98.91%	92.25%
	PGD adv. fine-tuning	98.67%	10.02%
	FeaCP adv. fine-tuning	**99.10%**	**94.82%**
CIFAR-10	Standard training	92.85%	0.00%
	L-BFGS adv. fine-tuning	87.87%	41.76%
	C&W adv. fine-tuning	88.14%	42.58%
	PGD adv. fine-tuning	86.73%	5.10%
	FeaCP adv. fine-tuning	**89.71%**	**47.22%**

TABLE III
TIMES OF ADVERSARIAL TRAINING WITH PGD FAILING TO GENERATE ADVERSARIAL EXAMPLES

Method	Dataset	Times
PGD adv. training	MNIST	2082
	CIFAR-10	15668

TABLE IV
CLASSIFICATION ACCURACY ON ADVERSARIAL AND CLEAN TEST SAMPLES

Dataset	Defense method	Clean	FGSM	PGD
MNIST	Standard training	99.14%	5.12%	0.51%
	PGD adv. training	98.04%	94.62%	93.78%
	PGD adv. training + Gaussian augmentation ($\sigma = 0.5$)	97.62%	94.85%	93.02%
	PNI-W	97.88%	94.90%	94.01%
	FeaCP adv. fine-tuning	**99.10%**	**95.06%**	**94.82%**
CIFAR-10	Standard training	93.15%	13.28%	0.00%
	PGD adv. training	86.66%	53.18%	45.22%
	PGD adv. training + Gaussian augmentation ($\sigma = 0.5$)	86.00%	55.67%	42.89%
	PNI-W	87.36%	56.25%	48.25%
	FeaCP adv. fine-tuning	**90.62%**	**59.22%**	**51.14%**

been proposed. However, most of them are already broken by stronger attacks proposed in [20], [51]. As a result, in this work, we choose to compare with the most effective one till date - PGD based adversarial training [10]. For a fair comparison, our experiments on MNIST and CIFAR-10 use the same model architectures and parameters as in [48].

Conventional adversarial training approximates the adversarial decision boundary in small neighborhoods of data points [29], [30], while those data points far from the boundary (indicated by the purple squares in Fig. 2) are not relevant or decidable to model performance. Specifically, PGD fails to construct adaptive neighborhoods to find samples with long and short distance to the decision-boundary. We quantify that phenomenon via the times failing to generate true adversarial examples in PGD and the results are presented in Table III.

Additionally, PGD adversarial training + Gaussian augmentation and PNI-W [48] are compared in Table IV to demonstrate the effectiveness of our proposed method. For MNIST, the standard network reaches 99.14% accuracy on the evaluation set. However, when evaluating examples perturbed with FGSM and PGD attack, the accuracy drops to 5.12% and 0.15%, respectively. When attacked by PGD, model ResNet-20(4×) attains 94.01% test accuracy using PNI-W, while apply with our method achieves 94.82%. The PGD adversarial training only reaches 94.62% under the FGSM attack, but with our method, the accuracy can achieve 95.06%, which is 89.94% better than the original model.

For CIFAR-10, we can see that our method about 51.14% more robust than the standard network and about 5.92% more robust than PGD adversarial training in the white box PGD setting. Notice that this gain is much larger than the previous gains obtained by PGD adversarial training against the original network (+6.04%), and by PNI-W against the original network (+2.97%). Moreover, we find that injecting Gaussian noise on the PGD generated samples almost make no difference to training and this may be due to the degrade of the significance of generated samples, in other words, the noise may reduce the loss value, i.e., harming the inner maximization in the adversarial training. This result shows that our adversarial fine-tuning can make the trained model more robust.

In particular, models are trained using exclusively adversarial inputs in PGD training. This led to a small but noticeable loss in accuracy on clean examples, dropping from 99.14% to 98.04% on MNIST and from 93.15% to 86.66% on CIFAR-10 in return for more robustness towards adversarial examples. Previous related studies [9], [52] have also shown a trade-off between clean-data accuracy and adversarial accuracy. The improvement of robustness comes at the cost of lowering the accuracy of normal examples. Adversarial fine-tuning with *FeaCP* dramatically makes the model more resistance to adversarial examples, while barely sacrificing accuracy on normal examples.

Overall, on all datasets, our proposed method improves both normal and perturbed data accuracy under the white-box attack compared to PGD adversarial training and PNI-W. While beating the natural training, PGD adversarial training incurs lower accuracy on legitimate examples. Additionally, we plot the robust accuracy for our methods using ResNet-18 on various perturbation strength of ϵ in Fig. 7. We can see that increasing the distortion bound ϵ increases the attack success rate. According to the gradient obfuscation work [20], [48], we can infer that the stochastic gradient is not the dominant role in our method for robustness improvement.

E. Adversarial Fine-Tuning With FeaCP

Robustness Evaluation With FGSM and PGD Attacks: To further illustrate the effectiveness of the adversarial fine-tuning with *FeaCP*, we compare it with PGD adversarial training on various network architectures. For CIFAR-10, the classic residual network [50] is adopted, and most of the comparative experiments use ResNet-20 as the baseline. Each experiment is tested by two types of attack methods, i.e., FGSM and PGD attacks. Table V shows the experimental results.

We can observe that original networks are never more than 8.74% error rates on the clean evaluation set. However, when evaluating examples perturbed with FGSM and PGD attacks, the performance is compromised to a great extent. The low accuracies of the original network shows that these attacks

TABLE V

ERROR RATES OF DIFFERENT MODELS ON CIFAR-10 DATASET UNDER FGSM AND PGD WHITE-BOX ATTACKS. RESNET-20(4×) DENOTE THE WIDE RESNET-20 WITH THE INPUT AND OUTPUT CHANNEL SCALED BY 4×

Model	No defense			PGD adv. train			*FeaCP* adv. fine-tuning		
	Clean	**FGSM**	**PGD**	**Clean**	**FGSM**	**PGD**	**Clean**	**FGSM**	**PGD**
ResNet-20	8.74%	86.15%	100.00%	17.18%	54.07%	61.75%	**14.55%**	**47.98%**	**55.21%**
ResNet-32	8.23%	83.16%	100.00%	15.21%	50.12%	58.90%	**12.44%**	**46.13%**	**54.98%**
ResNet-44	7.82%	76.99%	100.00%	15.09%	52.18%	60.94%	**11.74%**	**46.58%**	**53.43%**
ResNet-56	7.15%	76.54%	100.00%	14.42%	51.92%	61.00%	**10.29%**	**44.38%**	**52.78%**
ResNet-20(4×)	6.85%	86.72%	100.00%	13.34%	46.82%	54.78%	**9.38%**	**40.78%**	**48.86%**

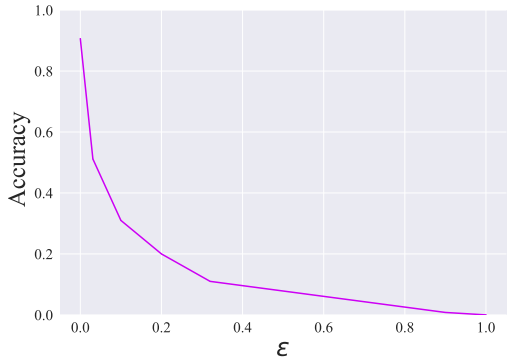

Fig. 7. Performance of our method against PGD adversaries of different strength ϵ on CIFAR-10.

TABLE VI

ERROR RATES OF RESNET-18 AND RESNET-20 (4×) ON CIFAR-10 DATASET UNDER C&W WHITE-BOX ATTACKS. RELATIVE IMPROVEMENT IN ERROR/PPL OVER PGD ADVERSARIAL TRAINING IS LISTED IN PERCENTAGE

Model	Method	Clean	C&W
ResNet-18	Standard training	9.88%	99.90%
	PGD adv. training	18.16%	97.66%
	FeaCP adv. fine-tuning	11.94%	97.39%
	Relative Impro.	**34.25%**	**0.28%**
ResNet-20 (4×)	Standard training	6.85%	99.90%
	PGD training	13.34%	98.06%
	FeaCP adv. fine-tuning	9.38%	97.99%
	Relative Impro.	**29.69%**	**0.07%**

can indeed effectively generate adversarial examples to fool the baseline model. However, when the model is trained using our method, the accuracy of adversarial examples is improved significantly.

Furthermore, we can clearly observe that adversarial fine-tuning with *FeaCP* is the most robust model. Our error rates level remains at below 49% when PGD adversarial training is above 54% under PGD adversarial attack using ResNet-20((4×)). The PGD adversarial training only achieves 54.07% error rate under FGSM using the ResNet-20 model, but with our adversarial fine-tuning, the error rate can drop to 47.98%. When attacked by PGD, model ResNet-32 attains 58.90% error rate using PGD adversarial training, while using our method, it achieves 54.98% error rate. The PGD adversarial training only reaches 51.92% under FGSM using model ResNet-56, but with our method, the error rate can drop to 44.38%, which is 32.16% better than the original model. On various neural networks, the above results reveal that the tuning model with more critical adversarial samples around the decision boundary can harden the protected model, and our algorithm has almost no loss of accuracy.

Robustness Evaluation With C&W Attack: So far the powerful attacks such as C&W are dominantly optimization-based, and in this work, we mainly focus on how to use the strong attack capability of them to protect model in the reverse direction, and propose to construct adaptive neighborhood to search critical samples in the ambiguous area of model decision-making. Hence, to make our work complete, comparison with C&W is necessary for adversarial fine-tuning despite the performance gain is limited.

We present the error rates obtained by the original model, PGD adversarial training, and *FeaCP* adversarial fine-tuning, along with the relative error reduction of our method over the PGD adversarial training in Table VI. We can see that the relative error reduction of our method over the PGD adversarial training is 0.28% under the CW attack using ResNet-18, 2.51% higher than the original model. Also, the relative error reduction is at least 29.69% on normal examples, and with a large margin in some cases. For example, we achieve an error rate of 11.94% with the ResNet-18, which is 6.22% better than the PGD adversarial training.

When attacked by C&W, model ResNet-20(4×) attains 99.90% error rates using standard training, while using our method, it drops to 97.99%. The PGD adversarial training reaches 97.66% under C&W, but with our method, the error rates can drop to 97.39% using ResNet-18 model. Again, this result shows that adversarial fine-tuning with *FeaCP* can make the model more robust.

F. Robustness to Sentiment Analysis

In the following experiments, we examine the generality of our proposed method by applying it to the IMDB dataset for sentiment analysis tasks. Following the procedure introduced in [53], [54], we linearly combine the vectors representing the words appearing in the review, and then process the embedding as a representation of the movie review for each movie review. We generate adversarial samples with $M = 1$ since there are only two classes, "positive" and "negative", Consequently, the Eq. (11) would be adapted for optimization. For a fair comparison, we adopt the setting of models' structures and

TABLE VII
CLASSIFICATION ACCURACY ON ADVERSARIAL AND CLEAN TEST SAMPLES

Defense method	Clean	FGSM	C&W
Standard training	85.94%	27.23%	28.04%
Defensive distillation	85.56%	48.01%	47.12%
LLE-DNN	85.82%	86.47%	85.83%
FeaCP adv. fine-tuning	**85.90%**	**89.50%**	**88.02%**

parameters in [54] and do experiments over different attacks. The exact verification results can be seen in Table VII.

Similar to defensive distillation [55] and LLE-DNN [54], the *FeaCP* adversarial fine-tuning is quite effective in preserving classification accuracy. More specifically, the accuracy of our method attains 85.90%. As shown in Table VII, it can be observed that white-box adversarial samples can cut down the accuracy of the standard training to 27.23%, 28.04% under the attacks of FGSM and C&W, respectively. In contrast, concerning the white-box setting, we observe that the *FeaCP* adversarial fine-tuning generally exhibits the best accuracy to adversarial samples, whereas the defensive distillation approach typically yields the least performance. This result shows that our method can effectively improve the robustness of neural network against adversarial examples. This is presumably due to the *FeaCP* take the diversity of feature importance and tightly connected with the decision boundary in adversarial instances construction, which also indicates our method is broadly appliable across various domains.

V. CONCLUSION

In this paper, we identify the black-box behaviors and the intrinsic deficiency of neighborhood information in the optimization-based adversarial attacks and defenses. In order to utilize the strong attacking capability of optimization-based adversarial examples to do defense, we propose a feature importance-aware convex interpolation method using an ensemble of correct predicted samples being in disjoint classes to guide the generation of effective and explainable adversarial samples in the ambiguous region around the decision boundary of model instead of uncontrolled "blind spots". The method can achieve adaptive neighborhoods which considers large and small neighborhoods simultaneously and enforces the generated samples in the ambiguous decision regions. Further, based on the similarity of vicinity distribution, we leverage the produced representative samples to estimate the distribution of adversarial samples to perform adversarial fine-tuning to optimize the learned boundary to improve the robustness. Finally, we conduct experiments to validate the effectiveness of the proposed method which outperforms the state-of-the-art defense methods in terms of both clean-data accuracy and perturbed data accuracy.

REFERENCES

[1] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, 2014, pp. 1–11.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–11.

[3] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[4] L. Pengcheng, J. Yi, and L. Zhang, "Query-efficient black-box attack by active learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Singapore, Nov. 2018, pp. 1200–1205.

[5] F. Codevilla, M. Müller, A. Lopez, V. Koltun, and A. Dosovitskiy, "End-to-End driving via conditional imitation learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–9.

[6] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, Feb. 2017.

[7] X. Xu, X. Chen, C. Liu, A. Rohrbach, T. Darell, and D. Song, "Can you fool ai with adversarial examples on a visual turing test," 2017, *arXiv:1709.08693*. [Online]. Available: http://arxiv.org/abs/1709.08693

[8] G. S. Dhillon *et al.*, "Stochastic activation pruning for robust adversarial defense," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–13.

[9] J. Buckman, A. Roy, C. Raffel, and I. J. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, 2018, pp. 1–22.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–10.

[11] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 854–863.

[12] Y. Yang, G. Zhang, Z. Xu, and D. Katabi, "ME-net: Towards effective adversarial robustness with matrix estimation," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Long Beach, CA, USA, Jun. 2019, pp. 7025–7034.

[13] Q. Li, Q. Hu, Y. Qi, S. Qi, J. Ma, and J. Zhang, "Stochastic batch augmentation with an effective distilled dynamic soft label regularizer," 2020, *arXiv:2006.15284*. [Online]. Available: https://arxiv.org/abs/2006.15284

[14] G. W. Ding, Y. Sharma, K. Y. C. Lui, and R. Huang, "MMA training: Direct input space margin maximization through adversarial training," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–26.

[15] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 26–30.

[16] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 3–14.

[17] S. Baluja and I. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," 2017, *arXiv:1703.09387*. [Online]. Available: http://arxiv.org/abs/1703.09387

[18] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, 2017, pp. 24–26.

[19] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* Washington, DC, USA: IEEE Computer Society, Jun. 2018, pp. 9185–9193.

[20] A. Athalye, N. Carlini, and D. A. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 80, J. G. Dy and A. Krause, Eds. Stockholm, Sweden, Jul. 2018, pp. 274–283.

[21] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. L. Yuille, "Mitigating adversarial effects through randomization," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–16.

[22] M. Mosbach, M. Andriushchenko, T. Trost, M. Hein, and D. Klakow, "Logit pairing methods can fool gradient-based attacks," 2018, *arXiv:1810.12042*. [Online]. Available: http://arxiv.org/abs/1810.12042

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[24] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 6–9.

[25] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 7472–7482.

[26] A. Sinha, H. Namkoong, and J. C. Duchi, "Certifying some distributional robustness with principled adversarial training," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*. Vancouver, BC, Canada: OpenReview.net, Apr. 2018.

[27] M. Andriushchenko and N. Flammarion, "Understanding and improving fast adversarial training," 2020, *arXiv:2007.02617*. [Online]. Available: https://arxiv.org/abs/2007.02617

[28] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS P)*, Saarbrücken, Germany, Mar. 2016, pp. 372–387.

[29] W. He, B. Li, and D. Song, "Decision boundary analysis of adversarial examples," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, 2018, pp. 1–15.

[30] M. Khoury and D. Hadfield-Menell, "Adversarial training with Voronoi constraints," 2019, *arXiv:1905.01019*. [Online]. Available: https://arxiv.org/abs/1905.01019

[31] K. Mahmood, P. H. Nguyen, L. M. Nguyen, T. Nguyen, and M. van Dijk, "BUZz: BUffer zones for defending adversarial examples in image classification," 2019, *arXiv:1910.02785*. [Online]. Available: https://arxiv.org/abs/1910.02785

[32] P. Li, J. Yi, B. Zhou, and L. Zhang, "Improving the robustness of deep neural networks via adversarial training with triplet loss," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, S. Kraus, Ed. Macao, China: ijcai.org, Aug. 2019, pp. 2909–2915.

[33] B. Škrlj, S. Džeroski, N. Lavrač, and M. Petkovič, "Feature importance estimation with self-attention networks," 2020, *arXiv:2002.04464*. [Online]. Available: https://arxiv.org/abs/2002.04464

[34] N. Lazic and P. Aarabi, "Importance of feature locations in Bag-of-Words image classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Honolulu, HI, USA, Apr. 2007, pp. 1–641.

[35] H. Xu *et al.*, "Neural network language modeling with letter-based features and importance sampling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 6109–6113.

[36] V. Verma *et al.*, "Manifold mixup: Better representations by interpolating hidden states," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6438–6447.

[37] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–13.

[38] C. Li, J. Zhu, T. Shi, and B. Zhang, "Max-margin deep generative models," 2015, *arXiv:1504.06787*. [Online]. Available: https://arxiv.org/abs/1504.06787

[39] P. Bartlett, D. J. Foster, and M. Telgarsky, "Spectrally-normalized margin bounds for neural networks," 2017, *arXiv:1706.08498*. [Online]. Available: https://arxiv.org/abs/arXiv:1706.08498

[40] W. Ping, Q. Liu, and A. T. Ihler, "Marginal structured SVM with hidden variables," in *Proc. 31th Int. Conf. Mach. Learn. (ICML)*, Beijing, China, Jun. 2014, pp. 190–198.

[41] A. de Brébisson and P. Vincent, "An exploration of softmax alternatives belonging to the spherical loss family," 2015, *arXiv:1511.05042*. [Online]. Available: http://arxiv.org/abs/1511.05042

[42] S. Gunasekar, J. Lee, D. Soudry, and N. Srebro, "Characterizing implicit bias in terms of optimization geometry," 2018, *arXiv:1802.08246*. [Online]. Available: http://arxiv.org/abs/1802.08246

[43] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," in *Proc. Neural Inf. Process. Syst. (NIPS)*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. Denver, CO, USA: MIT Press, 2000, pp. 416–422.

[44] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[45] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," Citeseer, 2009.

[46] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. Conf. 49th Annu. Meeting Assoc. Comput. Linguist., Hum. Lang. Technol.*, D. Lin, Y. Matsumoto, and R. Mihalcea, Eds. Portland, OR, USA: The Association for Computer Linguistics, Jun. 2011, pp. 142–150.

[47] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," 2016, *arXiv:1611.02770*. [Online]. Available: http://arxiv.org/abs/1611.02770

[48] Z. He, A. S. Rakin, and D. Fan, "Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 588–597.

[49] K. Schulz, L. Sixt, F. Tombari, and T. Landgraf, "Restricting the flow: Information bottlenecks for attribution," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–18.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[51] A. Athalye and N. Carlini, "On the robustness of the CVPR 2018 white-box adversarial example defenses," 2018, *arXiv:1804.03286*. [Online]. Available: https://arxiv.org/abs/1804.03286

[52] H. Qian and M. N. Wegman, "L2-nonexpansive neural networks," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019, pp. 1–18.

[53] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds. Lake Tahoe, NV, USA, 2013, pp. 3111–3119.

[54] W. Guo *et al.*, "Defending against adversarial samples without security through obscurity," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Singapore, Nov. 2018, pp. 137–146.

[55] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 2016, pp. 582–597.