Towards Model Resistant to Transferable Adversarial Examples via Trigger Activation

Yi Yu, Song Xia, Xun Lin, Chenqi Kong, Wenhan Yang, *Member, IEEE*,
Shijian Lu, *Member, IEEE*, Yap-Peng Tan, *Fellow, IEEE*, Alex C. Kot, *Life Fellow, IEEE*

Abstract—**Adversarial examples, characterized by imperceptible perturbations, pose significant threats to deep neural networks by misleading their predictions. A critical aspect of these examples is their transferability, allowing them to deceive unseen models in black-box scenarios. Despite the widespread exploration of defense methods, including those on transferability, they show limitations: inefficient deployment, ineffective defense, and degraded performance on clean images. In this work, we introduce a novel training paradigm aimed at enhancing robustness against transferable adversarial examples (TAEs) in a more efficient and effective way. We propose a model that exhibits random guessing behavior when presented with clean data x as input, and generates accurate predictions when with triggered data $x + \tau$. Importantly, the trigger τ remains constant for all data instances. We refer to these models as models with trigger activation. We are surprised to find that these models exhibit certain robustness against TAEs. Through the consideration of first-order gradients, we provide a theoretical analysis of this robustness. Moreover, through the joint optimization of the learnable trigger and the model, we achieve improved robustness to transferable attacks. Extensive experiments conducted across diverse datasets, evaluating a variety of attacking methods, underscore the effectiveness and superiority of our approach.**

Index Terms—**Adversarial robustness, transferable adversarial examples.**

I. INTRODUCTION

Deep Neural Networks (DNNs) have demonstrated remarkable success across a spectrum of machine learning endeavors. Together with the impressive performance of the deep neural networks, many concerns have been raised about their related AI security issues [1]–[7]. Nonetheless, they are vulnerable to

Yi Yu is with the Rapid-Rich Object Search (ROSE) Lab, Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore, (e-mail: yuyi0010@e.ntu.edu.sg).

Wenhan Yang is with PengCheng Laboratory, Shenzhen, China, (e-mail: yangwh@pcl.ac.cn).

Song Xia, Chenqi Kong, Yap-Peng Tan, and Alex C. Kot are with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, (e-mail: {xias0002, chenqi.kong, eyptan, eackot}@ntu.edu.sg).

Xun Lin is with School of Computer Science and Engineering, Beihang University, Beijing, China (e-mail: linxun@buaa.edu.cn).

Shijian Lu is with College of Computing and Data Science, Nanyang Technological University, Singapore, (e-mail: shijian.Lu@ntu.edu.sg).

(a) Illustration of model with trigger activation

(b) Generation of TAEs based on surrogate model f_s

(c) Robustness of victim model f_t to TAEs

Fig. 1. (a) Illustration of model with trigger activation: a model f that exhibits random guessing behavior with clean data x, akin to models with randomly initialized parameters, but generates accurate predictions with triggered data $x + \tau$, akin to well-trained models. (b) The attacker adopts f_s to generate the TAEs to attack the victim model. (c) During deployment, we treat the model with trigger activation as a unified entity, represented by $f_t(x) = f(x + \tau)$. This unified model, denoted as f_t, has been demonstrated to exhibit robustness against TAEs. Furthermore, if the adversarial examples x_{adv} are directly input into f_t without the trigger, the model continues to produce random guesses. Note that τ and δ_s are amplified by 10 times for a better view.

adversarial examples [8]–[11], which are intentionally manipulated inputs aimed at causing prediction inaccuracies. These inputs exhibit imperceptible differences compared to the original inputs. The presence of adversarial examples represents a significant concern for real-world safety-critical applications relying on Deep Neural Networks (DNNs), such as medical image analysis [12], [13], wireless comminications [14], autonomous driving systems [15] and image restoration [16]–[18]. Adversarial examples have been investigated in both the white-box setting (the victim models being freely accessed), to probe the maximum robustness of models, and the black-box setting [19]–[25] (not directly access the parameters of victim models), to interpret the practical risks posed to deployed models.

While the presence of adversarial examples has raised concerns regarding the reliability of AI systems, researchers have revealed a particularly interesting phenomenon: the transferability of adversarial examples [26]–[28], *i.e.,* transferability

denotes the capacity of an adversarial example crafted for one model to effectively deceive a different model, usually one with a different architecture. Transferable attacks operate under the assumption of a practical scenario where adversarial examples crafted on a (local) surrogate model can be directly applied to the (unknown) victim model [29], [30]. This type of attack can be executed without requiring access to any details of the victim model, including its architecture, parameters, or training data. Due to their significant real-world implications, transferable attacks have garnered considerable attention, leading to the rapid development of numerous new attacking methods with stronger performance. Given the severe security implications posed by these attacks on real-world AI systems, our focus lies on developing a robust defensive method against transferable adversarial examples (TAEs).

Several TAE-defense methods have been proposed recently, and they can be broadly categorized into two categories. The first category aims to enhance the robustness of neural networks themselves. In particular, adversarial training (AT) [31], [32] stands out as a mainstream method in safeguarding neural networks against adversarial attacks. However, AT, as performed in the model space, faces several challenges: 1) High computational Cost: AT is computationally expensive [32], as it requires repeatedly generating adversarial examples through on-the-fly attacks during the training process. The iterative and resource-intensive nature of this procedure places significant demands on computational resources, posing challenges for scalability and restricting its suitability for high-dimensional and large-scale datasets like ImageNet [33]; 2) Accuracy Drop: Models trained with AT often experience a significant drop in accuracy on the clean data [34]. For example, after adopting AT [32] with a perturbation budget of only $\epsilon \leq 2/255$, the clean accuracy (accuracy on the clean input) of ResNet50 [35] on the ImageNet validation set decreases from 76% to 64%.

The second category defends TAEs via various pre-processing methods. Certain methods, like those proposed by Guo et al. [36] and Prakash et al. [37], rely on corruption techniques to effectively mitigate the impact of malicious perturbations present in adversarial examples, thus leading to improved accuracy. On the other hand, approaches such as those introduced by Song et al. [38], Liao et al. [39], Jia et al. [40], and Nie et al. [41] opt for the utilization of denoisers or reconstruction models, like high-level representation-guided denoisers or diffusion models, to purify adversarial perturbations, aiming for enhanced robustness against such attacks. While pre-processing methods offer the advantage of defending against unforeseen threats in a plug-and-play fashion without necessitating classifier re-training, their performance usually falls short of current adversarial training methods or proves excessively time-consuming during deployment when compared to AT.

In this study, we introduce a novel training paradigm aiming at enhancing robustness against TAEs, which is computationally efficient during training and does not incur additional massive computational costs at test time. Given that adversarial perturbations δ_s are usually sought on the surrogate model f_s from clean data x by maximizing the cross-entropy loss, the optimized perturbations are sample-wise and closely related

with x and f_s. Then, it raises the question: if the victim model f_t has a clearly defined fast path $-\tau$ for gradient ascent for any input data, and there is a potential misalignment between $-\tau$ and δ_s, can δ_s from f_s be effectively transferred to attack f_t? To this end, we introduce the model with trigger activation. As shown in Figure 1(a), when clean data x is inputted, the model provides random predictions, akin to models with randomly initialized parameters. However, upon adding a trigger τ to x, the model behaves akin to a well-trained model. Importantly, the trigger τ remains constant for all data instances. Thus, we categorize these models as **models with trigger activation**. Initially, we adopt a randomly initialized and fixed trigger throughout the training process. We apply an ℓ_∞-norm bound ϵ_t to regulate the magnitude of the trigger. As shown in Figure 1(b)(c), our experiments demonstrate that increasing the bound ϵ_t enhances the model's robustness against TAEs, albeit with a minor decrease in clean accuracy. Subsequently, we provide a theoretical analysis of the model's robustness. By solely considering first-order gradients (this assumption implies linearity of the cross-entropy loss between x and $x + \tau$.) while dealing with TAEs, we can establish an upper bound on the cross-entropy loss. This allows us regulating the likelihood of being susceptible to these attacks.

Moreover, if the bound ϵ_t is excessively large, maintaining the linearity of the loss between x and $x + \tau$ becomes challenging. As a consequence, the less strict upper bound on the loss may not yield significant improvements in model robustness, while a large ϵ_t bound may lead to a greater drop in clean accuracy. The decrease in clean accuracy may be due to suboptimal model optimization, as the model faces challenges for optimizing both x and $x + \tau$ simultaneously. We thus propose jointly optimizing the trigger and the model, termed as a model with learnable trigger activation. More specifically, we do not impose a strict ℓ_∞-norm bound on the learnable trigger, while allows posing a large trigger in some areas, while maintaining a small one on other areas. In this way, the model can achieve a good balance between robustness on perturbed input and accuracy on clean images.

Our contributions can be summarized as follows:

- We introduce the model with trigger activation, which behaves randomly when given clean input data x and accurately predicts with triggered data $x + \tau$, ensuring a fast path $-\tau$ for gradient ascent from $x + \tau$. As the adversarial perturbations δ_s can diverge from $-\tau$, we observe that our proposed model demonstrates certain robustness against these perturbations.
- We offer a theoretical analysis of the model's robustness to TAEs when the trigger is randomly initialized and fixed. Drawing from the insights gained through our analysis, we propose a joint optimization approach for both the model and the learnable trigger, resulting in improved robustness.
- Extensive experiments conducted across diverse datasets, evaluating various attacking methods with varying perturbation bounds, underscore the effectiveness and superiority of our approach.

II. RELATED WORK

A. Adversarial Attacks

Adversarial attack methods are typically categorized into white-box attacks [8], [9], [42] and black-box attacks [26], [43], [44], based on the level of information accessible to the adversary regarding the victim model. In white-box attacks, the malicious actor has complete access to the victim models and can construct adversarial examples using the loss and gradients of the victim models. Examples include the one-step fast gradient sign method (FGSM) [9] and iterative gradient-based methods [8], [32]. In contrast to white-box attacks, black-box attacks pose greater challenges as they are limited to accessing models' outputs solely through queries. Certain black-box methods leverage feedback obtained from these queries to facilitate the generation of adversarial examples, referred to as query-based attacks [43], [45], [46]. Additional strategies for black-box attacks leverage the transferability of adversarial examples.

Various DNN architectures often produce significantly distinct decision boundaries, despite achieving comparable test accuracy, owing to their inherent high non-linearity [29], [47]. Consequently, gradients calculated for attacks on a particular (source) model may lead adversarial images into local optima, thus reducing their transferability to a different (target) model. To tackle this challenge, several approaches have been proposed to assist optimization in escaping from suboptimal local maxima during iterations, thereby enhancing the transferability of adversarial examples. In the realm of optimization-based enhancement methods, several techniques have been devised. I-FGSM [48] extends the iterative version of FGSM by increasing the number of iterations. MI-FGSM [49] enhances transferability by incorporating a momentum term and ensemble of model logits. NI-FGSM [50] incorporates an additional step at each iteration. Additionally, Variance Tuning (VT) [51] utilizes gradient information obtained at the final iteration to adjust the current gradient. In recent developments, GRA [52] refines the gradients by leveraging the average gradient from multiple data points sampled within the vicinity.

In the domain of augmentation-based enhancement methods, various approaches have been developed. Diverse Input (DI) [53] enhances input images through a combination of two transformations, namely random padding and resizing with a constant probability, before utilizing the processed images to craft adversarial examples. Scale-Invariant (SI) [50] exploits the scale-invariant property of deep neural networks by averaging gradients over scaled images to introduce additional foreign gradient information when generating adversarial examples. Admix [54] mixes the input image with other randomly selected images from the same batch to augment the input, and subsequently updates it with gradients calculated on the mixed image. Lately, BSR [55] proposes to divide the input image into multiple blocks, subsequently shuffling and rotating them randomly to generate a series of new images for gradient calculation, resulting in notably improved transferability. Learning to Transform (L2T) [56] enhances adversarial transferability by using reinforcement learning to optimize combinations of image transformations, surpassing existing input transformation-based methods.

In addition to crafting adversarial examples at the output layer, some works focus on internal layers. Feature Disruptive Attack (FDA) [57] introduces an attack method aimed at corrupting features at the target layer. Unlike previous methods that treat all neurons as equally important, FDA differentiates neuron importance based on mean activation values. Feature Importance-aware Attack (FIA) [58] measures neuron importance by multiplying the activation by the backpropagated gradients at the target layer. Neuron Attribution-Based Attacks (NAA) [59] compute feature importance for each neuron through integral decomposition. RPA [60] calculates the weight matrix in FIA using randomly patch-wise masked images. Recently, Diffusion-Based Projected Gradient Descent (Diff-PGD) [61] generates realistic adversarial samples by leveraging a gradient guided by a diffusion model, ensuring samples remain close to the data distribution while maintaining attack effectiveness.

B. Defenses to Adversarial Attacks

Similar to the way vaccines bolster the immune system, adversarial training [9], [31], [32] significantly enhances model robustness by expanding the training dataset with crafted adversarial examples. However, extending adversarial training to complex models poses challenges [62]: 1) Computational Cost: AT is computationally expensive [32], as it involves repeatedly generating adversarial examples through on-the-fly attacks during the training process. The iterative and resource-intensive nature of this procedure places significant demands on computational resources, posing challenges for scalability and restricting its suitability for high-dimensional and large-scale datasets like ImageNet [33]; 2) Accuracy Drop: Models trained with AT often experience a significant drop in accuracy on the original distribution. Apart from adversarial training, several other defense methods are relatively simple to implement.

Guo et al. [36] utilize diverse non-differentiable transformations, such as JPEG compression, applied to input images, thereby improving prediction accuracy in the presence of adversarial examples. Bit-Depth Reduction (BDR) [63] preprocesses input images by reducing the color depth of each pixel while preserving semantics. This operation eliminates pixel-level adversarial perturbations from adversarial images with minimal impact on model predictions for clean images. Pixel Deflection (PD) [37] effectively mitigates malicious perturbations through pixel corruption and redistribution. Resizing and Padding (R&P) [64] preprocesses input images by randomly resizing them to various sizes and adding random padding around the resized images. In [38], Song et al. propose PixelDefend, transforming adversarial images into clean images before they are fed into the classifier. Similarly, [39] treats imperceptible perturbations as noise and designs a high-level representation-guided denoiser (HGD) to remove these noises. ComDefend [40] defends against adversarial examples by passing them through an end-to-end image compression model, partially mitigating malicious

perturbations in the image. Feature Distillation (FD) [65] purifies adversarial input perturbations by redesigning the image compression framework, offering a novel low-cost strategy. Naseer et al. [66] eradicate malicious perturbations using a prearranged neural representation purifier (NRP), which is automatically derived supervision. Recently, diffusion models [67] have emerged as potent generative models. Diffusion Purification (DiffPure) [41] employs a diffusion model as the purification network. It diffuses an input image by gradually adding noise in a forward diffusion process and subsequently recovers the clean image by gradually denoising it in a reverse generative process. Notably, the reverse process has demonstrated its capability to remove adversarial perturbations. Recently, Randomized Adversarial Training (RAT) [68] introduced an innovative adversarial training approach that incorporates random noise into model weights, leveraging Taylor expansion to flatten the loss landscape and improve both robustness and clean accuracy. Taxonomy Driven Fast Adversarial Training (TDAT) [69] leverages the taxonomy of adversarial examples to prevent catastrophic overfitting in single-step adversarial training, achieving improved robustness with minimal computational overhead.

Our method offers distinct advantages over existing defense mechanisms. Pre-processing-based defenses, such as NRP, and DiffPure, rely on additional inference-time steps, which can increase computational overhead. In contrast, our method operates without these dependencies, ensuring higher test-time efficiency. Adversarial training (AT), while effective, is known for its high computational cost, whereas our approach significantly reduces training costs while maintaining strong adversarial robustness. Unlike NRP and DiffPure, which often require additional parameters and are tightly coupled with specific datasets, our method is lightweight and broadly applicable across datasets. Furthermore, our approach avoids the substantial accuracy drop on clean inputs that is commonly observed in some defenses, *e.g.,* JPEG, BDR, and Gaussian Filtering, striking a better balance between robustness and performance. By leveraging trigger activation, our method ensures consistent predictions on triggered inputs with theoretical guarantees, offering a novel and efficient alternative to traditional pre-processing or adversarial training paradigms.

III. METHODOLOGY

A. Preliminary

Formulation of Adversarial Transferability. Given an adversarial example $x + \delta_s$ of the input image x with the label y and two models $f_s(\cdot)$ and $f_t(\cdot)$, adversarial transferability describes the phenomenon that the adversarial example that is able to fool the surrogate model $f_s(\cdot)$ can also fool another victim model $f_t(\cdot)$. Formally speaking, the adversarial transferability of untargeted attacks can be formulated as follows:

$$\arg\max_i f_t^i(x + \delta_s) \neq y, \quad \text{if } \arg\max_i f_s^i(x + \delta_s) \neq y, \quad (1)$$

where f_s^i and f_t^i denote the i-th output probability of f_s and f_t, respectively. Typically, the generation of adversarial examples from f_s is to maximize the difference of the pre-defined

attacking loss (*e.g.,* cross-entropy loss) of the adversarial input $x + \delta_s$ from the true label y:

$$\delta_s = \arg\max_{\delta_s, \|\delta_s\|_p \leq \epsilon} \mathcal{L}_{ce}\left(f_s(x + \delta_s), y\right), \quad (2)$$

where $\|\delta_s\|_p \leq \epsilon$ guarantee that the adversarial examples are visually similar with the original ones, and ϵ is the bound for the perturbations δ_s. To solve the maximization problem with ℓ_p-norm bound constraint (usually ℓ_∞-norm), most approaches aim to obtain the adversarial examples iteratively. Taking the PGD [32] approach for example, the optimization process is given by:

$$\delta_s^{t+1} = \delta_s^t + \alpha \cdot \text{sgn}\left(\nabla_{x+\delta_s^t}\mathcal{L}_{ce}\left(f_s(x + \delta_s^t), y\right)\right), \quad (3)$$

$$\delta_s^{t+1} = \text{clip}_{[-x, 1-x]\cap[-\epsilon, \epsilon]}(\delta_s^{t+1}), \quad (4)$$

where ∇ represents the gradient operation, sgn extracts the sign of gradients, and the clip operation guarantees that the perturbations are within the range. The term α controls the step length each iteration, and ϵ represents the maximum perturbation allowed for each pixel value. The initial δ_s^0 is sampled from the uniform distribution $U(-\epsilon, \epsilon)$, and the final adversarial perturbations δ_s^T is obtained after T iterations. To quantitatively evaluate the robustness of f_t to TAEs generated from f_s, we adopt robust accuracy given by:

$$R_{f_t}^{f_s, A} = \mathbb{E}_{(x,y)\sim\mathcal{D}_{test}}\left[\mathbb{I}\{\arg\max_i f_t^i(x + \delta_s^A) = y\}\right], \quad (5)$$

where $\mathcal{D}_{test}$ denotes the testing data, and $x + \delta_s^A$ is generated on surrogate model f_s with attacking methods A.

Evaluation of Robustness to TAEs. To comprehensively evaluate the robustness of the victim model to TAEs generated from various surrogate models, we use the mean value of $R_{f_t}^{f_s, A}$ to evaluate the adversarial robustness against each type of attack A:

$$R_{f_t}^{S, A} = \frac{1}{|S|} \sum_{f_s \in S} R_{f_t}^{f_s, A}, \quad (6)$$

where S denotes the set of surrogate models to generate TAEs.

Evaluation of Defenses against TAEs. To evaluate the effectiveness of defenses, we consider $f_t \in T$, and models from T can be equipped with any kinds of defenses (*e.g.,* pre-processing methods, AT). We use mean value of $R_{f_t}^{S, A}$ for each attacking method A:

$$R_T^{S, A} = \frac{1}{|T|} \sum_{f_t \in T} R_{f_t}^{S, A} = \frac{1}{|T||S|} \sum_{f_t \in T} \sum_{f_s \in S} R_{f_t}^{f_s, A}. \quad (7)$$

B. Model with Trigger Activation

Given that adversarial perturbations δ_s are usually sought on the surrogate model f_s from clean data x by maximizing the cross-entropy loss, the optimized perturbations are sample-wise and closely related with x and f_s. Then, it raises the question: if the victim model f_t has a clearly defined fast path $-\tau$ for gradient ascent for any input data, and there is a potential misalignment between $-\tau$ and δ_s, can δ_s from f_s be effectively transferred to attack f_t? Hence, we introduce the model with trigger activation.

TABLE I
ROBUSTNESS VS. TRIGGER BOUND ϵ_t: ROBUST ACCURACY (%) AND CLEAN ACCURACY (%) FOR MODELS WITH FIXED TRIGGER ACTIVATION UNDER DIFFERENT ATTACK METHODS ON CIFAR-10 DATASET. FOR ROBUST ACCURACY, WE UTILIZE THE ROBUSTNESS $R_T^{S,A}$ DEFINED IN EQ. 7.

Defenses→ Attacks↓	w/o	AT [32]	Ours (fixed)						
			$\epsilon = \frac{1}{255}$	$\epsilon = \frac{2}{255}$	$\epsilon = \frac{4}{255}$	$\epsilon = \frac{8}{255}$	$\epsilon = \frac{16}{255}$	$\epsilon = \frac{32}{255}$	$\epsilon = \frac{64}{255}$
Clean	94.25	83.31	94.47	94.41	94.20	93.62	93.08	92.58	92.07
PGD	11.29	82.19	15.38	22.17	42.17	69.62	75.29	71.90	74.67
I-FGSM	19.10	82.45	27.36	37.49	59.14	78.60	81.49	78.28	80.33
MI-FGSM	13.19	82.19	17.84	23.19	40.34	67.53	72.29	70.22	72.81
DI-FGSM	11.29	81.61	14.68	18.27	29.62	52.89	59.81	59.68	63.15

Algorithm 1: Model w/ Trigger (fixed) Activation

Input : Model $f(\cdot|\theta)$ with C classes, initial parameters θ^0, training data $\mathcal{D}_{train}$, mini-batch $\mathcal{B}$, training epochs T, learning rate η_{it}, ℓ_∞-norm bound ϵ_t.

Output: Model $f(\cdot|\theta)$ with $\boldsymbol{\tau}$ as the trigger for activation

\# Initialization of trigger $\boldsymbol{\tau}$
τ_i sampled from Bernoulli distribution $B(1, 0.5)$
$\boldsymbol{\tau} = \epsilon_t \cdot (2 \cdot \boldsymbol{\tau} - 1)$
\# Optimization of Model $f(\cdot|\theta)$ with fixed $\boldsymbol{\tau}$
for $i \leftarrow 1$ **to** T **do**
  for *each batch* $\{\boldsymbol{x}_\mathcal{B}, \boldsymbol{y}_\mathcal{B}\} \in \mathcal{D}_{train}$ **do**
    $\boldsymbol{x}_{in} = Concatenate(\boldsymbol{x}_\mathcal{B}, \boldsymbol{x}_\mathcal{B} + \boldsymbol{\tau}, dim = 0)$
    $z_{neg}, z_{pos} = Split(f(\boldsymbol{x}_{in}), \mathcal{B}, dim = 0)$
    $\mathcal{L}_{total} = \mathcal{L}_{ce}(z_{pos}, \boldsymbol{y}_\mathcal{B}) + \mathcal{L}_{KLD}(z_{neg}, \frac{1}{C}\mathbf{1}_C)$
    $g(\theta^{it}) \leftarrow$ stochastic gradient by minimizing $\mathcal{L}_{total}$
    $\theta^{it+1} = \theta^{it} - \eta_{it} \cdot g(\theta^{it})$
  end
end

Definition of Model with Trigger Activation. For a model $f(\cdot|\theta)$ with C classes, our objective is for $f(\cdot|\theta)$ to accurately predict when provided with triggered data $\boldsymbol{x}+\boldsymbol{\tau}$ as input, while producing random guesses when presented with clean data $\boldsymbol{x}$. Typically, $\boldsymbol{\tau}$ is referred to as the trigger for the model. In Algorithm 1, we illustrate the training procedure of the model with fixed trigger activation, wherein the trigger is randomized initially and then fixed. In addition, in Algorithm 1, *Concatenate* refers to merging two tensors into one along a specified dimension, *Split* refers to dividing one tensor into two tensors based on specified dimensions and sizes, and $\mathcal{L}_{KLD}$ represents the KLD loss between two logits. In deployment, we consider **the model with trigger activation as a single entity**, denoted as $f_t(\boldsymbol{x}) = f(\boldsymbol{x} + \boldsymbol{\tau})$.

Robustness against TAEs. Surprisingly, we find that the model with fixed trigger action exhibits certain robustness against TAEs as shown in Table I. While the robustness may not be as competitive as AT, it notably surpasses the model without any defense. Additionally, we observe that increasing the trigger bound ϵ_t can also enhance robustness (details are discussed after Theorem 2).

$-\boldsymbol{\tau}$ is a fast path for gradient ascent. The robustness of

Fig. 2. Loss and Accuracy (Acc.) when using $-\boldsymbol{\tau}$ with randomly flipped proportions as adversarial perturbations to attack the model with trigger activation.

the model with trigger activation to TAEs may be attributed to the misalignment between $-\boldsymbol{\tau}$ and $\boldsymbol{\delta_s}$. To investigate this, we conduct experiments where $-\boldsymbol{\tau}$ was used with randomly flipped proportions as adversarial perturbations to attack. Specifically, we performed experiments on the CIFAR-10 dataset, setting the bound ϵ_t to $\frac{8.0}{255}$ as in Algorithm 1. The results, shown in Figure 2, demonstrate that when $-\boldsymbol{\tau}$ is flipped at a small proportion, the adversarial perturbations quickly become ineffective, indicating that $-\boldsymbol{\tau}$ serves as a fast path for gradient ascent. If there is a misalignment between the transferred $\boldsymbol{\delta_s}$ and $-\boldsymbol{\tau}$, $\boldsymbol{\delta_s}$ may also fail to attack.

Analysis on Robustness. We provide a theoretical analysis of the emerging robustness as shown in Table I, when **considering only the first-order derivatives** (the assumption implies a linearity of the loss between $\boldsymbol{x}$ and $\boldsymbol{x} + \boldsymbol{\tau}$).

Theorem 1 (Relationship of $\boldsymbol{\tau}$ with dataset and model). *Given a model trained in Algorithm 1, under the assumption of linearity, the relationship of $\boldsymbol{\tau}$ with dataset and model is*

$$-\epsilon_t \cdot \text{sgn}\left[\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\left[\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x}, y)\right]\right] = \boldsymbol{\tau}, \quad (8)$$

$$-log(C) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\left[\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x}, y)\right]^\top \boldsymbol{\tau}, \quad (9)$$

$$\text{where } \ell_t(\boldsymbol{x}, y) = \mathcal{L}_{ce}(f(\boldsymbol{x}), y). \quad (10)$$

Proof. Using Taylor expansion and **considering only the first-order derivatives**, we can obtain

$$\forall (\boldsymbol{x}, y) \in \mathcal{D}_{train}, \quad \ell_t(\boldsymbol{x} + \boldsymbol{\tau}, y) \\ = \ell_t(\boldsymbol{x}, y) + [\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x}, y)]^\top \boldsymbol{\tau}, \quad (11)$$

and the form of the expectation over the entire training dataset is given by

$$
\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\ell_t(\boldsymbol{x}+\boldsymbol{\tau},y)\Big] = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\ell_t(\boldsymbol{x},y)\Big]
$$
$$
+\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x},y)\Big]^{\top}\boldsymbol{\tau}.
\tag{12}
$$

As demonstrated in Algorithm 1, to make $f(\boldsymbol{x})$ approach random guessing, $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\ell_t(\boldsymbol{x},y)\Big]$ should ideally become $log(C)$, which is the cross-entropy loss of an evenly distributed logit. Achieving this goal is not difficult, as models with randomly initialized parameters already possess this capability. Additionally, the expectation $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\ell_t(\boldsymbol{x}+\boldsymbol{\tau},y)\Big]$ of the well-trained model should be minimized, and close to zero. While it may not be exact, to achieve this, $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x},y)\Big]$ should generally have the opposite direction as $\boldsymbol{\tau}$. Considering the bound we set for $\boldsymbol{\tau}$, the $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x},y)\Big]$ and the pre-defined (randomly initialized) $\boldsymbol{\tau}$ should conform to the structure outlined in Eq. 8. More precisely, since we designate the expected values as 0 and $log(C)$ for $\ell_t(\boldsymbol{x}+\boldsymbol{\tau},y)$ and $\ell_t(\boldsymbol{x},y)$ respectively, it establishes the relationship described in Eq. 9.

Theorem 2 (Adversarial impact of TAEs). *Given adversarial perturbations $\boldsymbol{\delta_s}$ with an ℓ_∞-norm bound ϵ generated from the surrogate model f_s, the effect of $\boldsymbol{\delta_s}$ on the victim model $f_t(\boldsymbol{x}) = f(\boldsymbol{x}+\boldsymbol{\tau})$ can be described as follows*

$$
\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{test}}\Big[\ell_t(\boldsymbol{x}+\boldsymbol{\tau}+\boldsymbol{\delta_s},y)\Big]
$$
$$
= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x},y)\Big]^{\top}\boldsymbol{\delta_s} \tag{13}
$$
$$
\leq \frac{\epsilon}{\epsilon_t}log(C),
$$

and the maximum is achieved when

$$
\boldsymbol{\delta_s} = -\frac{\epsilon}{\epsilon_t}\boldsymbol{\tau}. \tag{14}
$$

Proof. Since existing deep methods usually consider that both the training dataset $\mathcal{D}_{train}$ and the test dataset $\mathcal{D}_{test}$ follow the same distribution, the form of the expectation of the related loss over the entire test dataset can be expressed as:

$$
\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{test}}\Big[\ell_t(\boldsymbol{x}+\boldsymbol{\tau}+\boldsymbol{\delta_s},y)\Big]
$$
$$
= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\ell_t(\boldsymbol{x}+\boldsymbol{\tau}+\boldsymbol{\delta_s},y)\Big]
$$
$$
= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\ell_t(\boldsymbol{x},y)\Big]+\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x},y)\Big]^{\top}(\boldsymbol{\tau}+\boldsymbol{\delta_s})
$$
$$
= log(C) + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x},y)\Big]^{\top}(\boldsymbol{\tau}+\boldsymbol{\delta_s})
$$
$$
= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x},y)\Big]^{\top}\boldsymbol{\delta_s} \quad\text{(using } Eq.\ 9)
$$
$$
\leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{train}}\Big[\nabla_{\boldsymbol{x}}\ell_t(\boldsymbol{x},y)\Big]^{\top}\frac{\epsilon}{\epsilon_t}(-\boldsymbol{\tau}) \quad\text{(using } Eq.\ 8)
$$
$$
= \frac{\epsilon}{\epsilon_t}log(C). \tag{15}
$$

Algorithm 2: Model w/ Trigger (learnable) Activation

Input : Model $f(\cdot|\theta)$ with C classes, initial parameters θ^0, training data $\mathcal{D}_{train}$, mini-batch $\mathcal{B}$, training epochs T, learning rate η_{it}, step size α.

Output: Model $f(\cdot|\theta)$ with $\boldsymbol{\tau}$ as the trigger for activation

\# Initialization of trigger $\boldsymbol{\tau}$
$\boldsymbol{\tau}$ sampled from Uniform distribution $U(-\alpha,\alpha)$
for $i \leftarrow 1$ **to** T **do**
  \# Optimization of Model $f(\cdot|\theta)$
  for *each batch* $\{\boldsymbol{x}_\mathcal{B},\boldsymbol{y}_\mathcal{B}\} \in \mathcal{D}_{train}$ **do**
    $\boldsymbol{x}_{in} = Concatenate(\boldsymbol{x}_\mathcal{B},\boldsymbol{x}_\mathcal{B}+\boldsymbol{\tau},dim=0)$
    $z_{neg}, z_{pos} = Split(f(\boldsymbol{x}_{in}),\mathcal{B},dim=0)$
    $\mathcal{L}_{total} = \mathcal{L}_{ce}(z_{pos},\boldsymbol{y}_\mathcal{B}) + \mathcal{L}_{KLD}(z_{neg},\frac{1}{C}\mathbf{1}_C)$
    $g(\theta^{it}) \leftarrow$
      stochastic gradient by minimizing $\mathcal{L}_{total}$
    $\theta^{it+1} = \theta^{it} - \eta_{it}\cdot g(\theta^{it})$
  end
  if $i \in [1, 0.6\times T]$ **then**
    \# Optimization of trigger $\boldsymbol{\tau}$
    $\boldsymbol{g}_{tmp} = 0$
    for *each batch* $\{\boldsymbol{x}_\mathcal{B},\boldsymbol{y}_\mathcal{B}\} \in \mathcal{D}_{train}$ **do**
      $\mathcal{L}_{trigger} = \mathcal{L}_{ce}(f(\boldsymbol{x}_\mathcal{B}+\boldsymbol{\tau}),\boldsymbol{y}_\mathcal{B})$
      $\boldsymbol{g}_{tmp} = \boldsymbol{g}_{tmp} + \nabla_{\boldsymbol{\tau}}\mathcal{L}_{trigger}$
    end
    $\boldsymbol{\tau} = \boldsymbol{\tau} - \alpha\cdot\text{sgn}[\boldsymbol{g}_{tmp}]$
  end
end

Robustness v.s. Trigger Bound ϵ_t. As stated in Theorem 2, for models with a higher bound ϵ_t, the expected loss on the test set will have a lower upper bound. **This leads to decreased vulnerability to TAEs and increased robust accuracy.** Then, we demonstrate the robustness against TAEs by training the model with trigger activation on the CIFAR-10 [70]. We choose the model sets T and S both consisting of several model architectures (ResNet-18 [35], ResNet-50 [35], VGG-19 [71], MobileNet-V2 [72], DenseNet-121 [73]). S consists of model with standard training, and T consists of model with trigger activation. As observed from the $R_T^{S,A}$ values in Table I with various attacking methods, increasing the bound ϵ_t for the trigger results in enhanced robustness of the victim model against TAEs, albeit with a slight decrease in clean accuracy.

C. Learnable Trigger

Moreover, if the bound ϵ_t is excessively large, maintaining the linearity of the loss between $\boldsymbol{x}$ and $\boldsymbol{x}+\boldsymbol{\tau}$ becomes challenging. Consequently, the upper bound on the loss may not be as strict, leading to lower levels of model robustness. As shown in Table I, it can be observed that increasing the bound ϵ_t beyond a certain threshold does not significantly enhance robustness but instead leads to a degradation in clean accuracy. The decrease in clean accuracy may be due to suboptimal model optimization, as the model faces challenges

TABLE II
COMPARISON OF CLEAN ACCURACY (%) ↑ AND ROBUST ACCURACY (%) ↑ UNDER DIFFERENT ATTACK METHODS ON CIFAR-10 DATASET. FOR CLEAN ACCURACY, WE ADOPT THE MEAN ACCURACY OF THE VICTIM MODELS WHEN TAKING CLEAN DATA AS INPUT. FOR ROBUST ACCURACY, WE UTILIZE THE ROBUSTNESS $R_T^{S,A}$ DEFINED IN EQ. 7. "OURS (F)" REFERS TO THE MODEL WITH FIXED TRIGGER ACTIVATION, WHILE "OURS (L)" REPRESENTS THE MODEL WITH LEARNABLE TRIGGER ACTIVATION, WITH THEIR RESPECTIVE HYPERPARAMETERS PROVIDED. BOLD DENOTES THE BEST, AND UNDERLINE DENOTES THE SECOND BEST. **NOTE THAT OUR METHODS APPLY THE TRIGGER TO ALL INPUTS DURING TESTING BY DEFAULT.**

Defenses→ Attacks↓, Bound↓		w/o	JPEG [36] q=50 q=75		BDR [63] d=2	Gaussian Filter σ=0.6 σ=0.7		R&P [64] s=1.2	NRP [66]	DiffPure [41]	AT [32] (PGD)	RAT [68] (TRADES)	TDAT [69]	Ours (f) $\epsilon_t = \frac{64}{255}$	Ours (l) $\alpha = \frac{4}{255}$
Clean	-	94.25	76.12	84.02	65.05	86.76	73.77	86.07	92.05	89.12	83.31	82.35	88.01	92.07	91.93
PGD		11.29	64.12	56.24	36.71	38.65	46.46	48.19	23.82	**86.59**	82.19	81.08	84.74	74.67	85.49
I-FGSM		19.10	67.21	64.60	42.73	49.56	53.62	57.98	34.25	86.80	82.45	81.29	85.36	80.33	**87.38**
MI-FGSM		13.19	63.66	54.07	36.81	38.06	46.30	47.59	23.27	**86.41**	82.19	80.91	81.77	72.81	83.71
DI-FGSM $\ell_\infty = \frac{8}{255}$		11.29	55.65	44.37	34.62	25.06	32.17	30.07	22.03	**85.54**	84.89	80.30	80.26	64.15	76.98
NAA		19.53	62.94	54.81	19.54	39.77	43.19	48.75	28.27	**86.20**	82.06	80.81	81.45	74.01	84.45
RPA		16.34	61.43	51.26	16.20	42.90	47.47	52.69	26.75	**85.61**	81.98	80.68	81.62	73.43	85.02
L2T		16.38	54.81	45.23	16.46	30.78	33.09	36.70	25.21	**85.36**	81.05	79.59	79.41	65.17	77.33
Mean	-	15.16	61.55	52.08	28.32	37.54	43.33	46.28	26.80	**86.08**	82.26	80.95	82.37	72.08	83.91

in optimizing both x and $x+\tau$ simultaneously. Therefore, we propose jointly optimizing the trigger and the model, termed as a model with learnable trigger activation. More specifically, we do not impose a strict ℓ_∞-norm bound on the learnable trigger. Unlike Algorithm 1, which utilizes a fixed trigger, we incorporate the learning process of the trigger τ. As depicted in Algorithm 2, after each training epoch for the model $f(\cdot|\theta)$, we iterate through the entire dataset, recording the gradient for each iteration. Subsequently, the optimization of τ is conducted based on the sign of the cumulative gradients, to minimize the loss $\mathcal{L}_{ce}(f(x_\mathcal{B} + \tau), y_\mathcal{B})$. Consequently, we can pose a large trigger in some areas, while maintaining a small in the other area. In this way, the model can achieve a good balance between robustness and clean accuracy.

To compare with the model using fixed trigger activation, we adopt the same training settings and present the results in Table III-B. It is evident that models with learnable trigger activation achieve improved robustness with less decrease in clean accuracy.

IV. EXPERIMENTS

A. Experimental Setup

Datasets and models. We choose three commonly used datasets: CIFAR-10, CIFAR-100 [70], and a subset of ImageNet [33] with the first 100 classes (since the training on the ImageNet-1k is time-consuming). To evaluate on the CIFAR-10/100 dataset, we select model sets T and S, both containing several model architectures including ResNet-18 [35], ResNet-50 [35], VGG-19 [71], MobileNet-V2 [72], and DenseNet-121 [73]. For the ImageNet-subset, in addition to the above models, we also include Inception-V4 [74].

Attacking Methods. We examine several attacking methods to generate TAEs. For experiments on the CIFAR-10/100 dataset, we select I-FGSM [48], PGD [32], MI-FGSM [49], DI-FGSM [53], L2T [56] and methods with advanced objectives such as NAA [59] and RPA [60] as our chosen adversarial attack methods. The ℓ_∞-norm bound for the perturbations is set to $\frac{8}{255}$. For experiments on ImageNet, as the aforementioned methods do not yield satisfactory performance, we include additional advanced attacking methods such as BSR [55],

GRA [52], and Diff-PGD [61]. We set $\frac{8}{255}$ as the bound for the perturbations, and the iterations for all attacks are set to 20.

Competing Defensive Methods. We incorporate both training-based defense and processing/purification-based defense methods as competing approaches. For the training-based defense, we select sveral adversarial training (AT) methods, including AT-PGD [32], RAT-TRADES [68], and TDAT [69] with $\epsilon = \frac{8}{255}$. Among the purification methods, we include bit-depth reduction (BDR) [63], JPEG compression [36], Gaussian filtering, resizing and padding (R&P) [64], neural representation purifier (NRP) [66], and DiffPure [41], which employs a diffusion model for purification. Specifically, for JPEG compression, BDR, Gaussian filtering, and R&P, we also provide the corresponding hyperparameters used in the experiments. **Note that our methods apply the trigger to all inputs during testing by default.**

Model Training. To ensure consistent training procedures for the classifier, we have formalized the standard training approach. For CIFAR-10, we use 60 epochs, while for CIFAR-100 and the ImageNet-subset, 100 epochs are allowed. In all experiments, we use SGD optimizer with an initial learning rate of 0.1 and the CosineAnnealingLR scheduler, keeping a consistent batch size of 128.

B. Experimental Results

Results on CIFAR-10 dataset. To evaluate the effectiveness of our proposed method, we conducted initial experiments on the CIFAR-10 dataset. As shown in Table III-B, our method consistently provides comprehensive protection against TAEs with different attack methods. When facing TAEs generated by DI-FGSM, our method with a learnable trigger may slightly lag behind AT-PGD, given that DI-FGSM utilizes diverse inputs for generating TAEs to enhance generalizability, whereas AT-PGD primarily focuses on providing robustness in the white-box setting and remains unaffected. However, our method significantly outperforms AT-PGD in terms of performance on clean inputs. Compared to the recent adversarial training methods RAT-TRADES [68] and TDAT [69], our

TABLE III
COMPARISON OF CLEAN ACCURACY (%) ↑ AND ROBUST ACCURACY (%) ↑ UNDER DIFFERENT ATTACK METHODS ON CIFAR-100 DATASET. BOLD DENOTES THE BEST, AND UNDERLINE DENOTES THE SECOND BEST. **NOTE THAT OUR METHODS APPLY THE TRIGGER TO ALL INPUTS DURING TESTING BY DEFAULT.**

Defenses→ Attacks↓, Bound↓	w/o	JPEG [36] q=50 q=75	BDR [63] d = 2	Gaussian Filter σ=0.6 σ=0.7	R&P [64] s=1.2	NRP [66]	DiffPure [41]	AT [32] (PGD)	RAT [68] (TRADES)	TDAT [69]	Ours (f) $\epsilon_t = \frac{64}{255}$	Ours (l) $\alpha = \frac{4}{255}$
Clean	-	74.29	46.36 55.03	28.14	60.82 48.33	60.53	68.44	47.82	55.48	56.67	52.01	67.30	68.98
PGD	12.76	36.33 33.20	13.90	27.52 29.74	32.05	22.33	42.16	54.11	54.91	48.94	55.59	**58.46**
I-FGSM	18.89	38.73 39.19	16.18	34.38 34.43	38.61	28.11	43.28	54.44	55.24	49.67	59.01	**61.66**
MI-FGSM	14.36	36.46 32.87	14.18	28.60 30.69	33.13	22.19	42.60	54.27	54.90	48.29	55.97	**58.94**
DI-FGSM $\ell_\infty = \frac{8}{255}$	11.28	30.90 25.96	13.04	19.06 21.41	21.52	19.81	40.75	53.58	**54.06**	45.93	49.22	51.32
NAA	19.42	34.94 32.78	19.43	30.04 29.85	33.63	26.08	42.22	54.05	54.63	47.77	55.39	**56.57**
RPA	18.80	35.53 32.93	18.69	33.29 32.66	37.15	27.07	42.88	54.23	54.88	48.64	57.59	**59.48**
L2T	10.84	29.43 24.54	10.86	19.44 20.90	22.84	18.61	41.73	52.76	**52.86**	44.69	45.76	50.94
Mean	-	15.08	34.76 31.92	15.04	27.62 28.81	31.42	23.60	42.66	53.92	54.64	47.99	54.79	**56.91**

TABLE IV
COMPARISON OF CLEAN ACCURACY (%) ↑ AND ROBUST ACCURACY (%) ↑ UNDER DIFFERENT ATTACK METHODS ON IMAGENET-SUBSET. BOLD DENOTES THE BEST, AND UNDERLINE DENOTES THE SECOND BEST. **NOTE THAT OUR METHODS APPLY THE TRIGGER TO ALL INPUTS DURING TESTING BY DEFAULT.**

Defenses→ Attacks↓, Bound↓	w/o	JPEG [36] q=20 q=30	BDR [63] d = 2	Gaussian Filter σ=1.2 σ=3.0	R&P [64] s=1.1	NRP [66]	DiffPure [41]	AT [32] (PGD)	RAT [68] (TRADES)	TDAT [69]	Ours (f) $\epsilon_t = \frac{64}{255}$	Ours (l) $\alpha = \frac{4}{255}$
Clean	-	80.76	65.82 70.54	52.70	71.52 66.01	78.64	77.40	75.56	57.29	58.50	54.08	74.59	77.14
PGD	45.48	53.60 53.74	37.18	49.80 49.57	47.87	61.36	**72.06**	56.77	57.69	53.21	67.92	70.96
I-FGSM	54.46	57.50 59.10	42.19	55.24 54.33	54.98	63.60	73.00	56.94	57.87	53.29	70.34	**73.09**
MI-FGSM	43.98	50.99 50.17	37.19	46.75 46.55	45.74	59.49	**71.00**	54.68	55.92	52.65	66.54	69.46
DI-FGSM	32.39	43.03 41.13	33.05	36.24 37.27	32.85	55.68	**67.71**	56.52	57.39	51.65	60.80	63.82
GRA $\ell_\infty = \frac{8}{255}$	43.94	50.85 50.23	43.98	47.05 46.69	45.70	59.51	**71.20**	56.75	57.65	52.71	66.05	69.29
BSR	21.39	40.65 37.55	21.30	33.17 35.11	25.23	51.04	**63.81**	56.18	57.00	50.50	55.65	59.85
NAA	44.42	49.00 49.00	44.46	45.62 44.66	45.33	60.41	57.50	56.58	57.46	52.32	64.65	**67.71**
RPA	46.16	51.18 51.85	46.16	49.06 48.32	47.32	62.96	57.46	56.61	57.49	52.70	67.09	**70.38**
L2T	28.36	38.16 36.84	28.42	31.21 32.30	28.57	55.36	55.81	55.74	56.37	48.67	54.84	59.17
Diff-PGD	32.14	37.24 35.75	32.15	32.28 32.77	31.34	52.29	56.39	56.38	57.23	50.88	55.59	**59.32**
Mean	-	39.07	47.42 46.26	36.01	42.24 42.56	40.39	58.17	65.99	56.31	57.41	51.76	63.24	**66.41**

proposed approach demonstrates superior adversarial robustness while maintaining better clean accuracy. Compared with DiffPure, our method achieves slightly worse performance but with less impact on clean accuracy. Moreover, DiffPure necessitates iterative noise addition and denoising through forward and reverse processes, demanding considerable time and computational resources for the purification process. For instance, it takes approximately two hours to purify the CIFAR-10 test set using an RTX A5000 GPU, rendering it inefficient for deployment. In contrast, our method does not require additional computation costs or time during the inference stage. The NRP purification method appears ineffective against TAEs from the CIFAR-10 dataset, despite its success with the ImageNet dataset. This discrepancy may arise from the disparity between CIFAR-10, composed of small-sized images, and the COCO dataset used to train the NRP purification model. Moreover, all the other pre-processing methods, including JPEG, BDR, Gaussian Filter, and R&P, demonstrate poor performance against TAEs, and they also have a detrimental effect on the accuracy of clean images.

Results on CIFAR-100 dataset. We then conduct our experiments on CIFAR-100 dataset. The results, as presented in Table IV-A, re-confirm the overall effectiveness of our purification framework. Our method surpasses AT-PGD, RAT-TRADES and TDAT in adversarial robustness while achieving better clean accuracy. It's worth highlighting that the efficacy of DiffPure's purification largely relies on the dataset used to

TABLE V
COMPUTATION COST OF EXISTING DEFENSES AND OUR METHOD. WE INCLUDE MODEL TRAINING TIME (HOURS) AND TESTING TIME (10^{-3} S/BATCH). NOTE THAT FOR NRP AND DIFFPURE, WE DO NOT INCLUDE THE TIME TO TRAIN THE PURIFIER.

Defense	CIFAR-10/100 dataset Training time	Testing time	ImageNet-subset Training time	Testing time
w/o	0.5	1.745	5.5	1.686
JPEG	0.5	47.82	5.5	48.07
BDR	0.5	1.495	5.5	1.627
Gaussian Filter	0.5	1.830	5.5	2.595
R&P	0.5	1.572	5.5	1.626
NRP	0.5	121.3	5.5	1771
DiffPure	0.5	25918	5.5	289459
AT	3.6	1.813	39.4	1.626
RAT	5.4	1.765	40.3	1.645
TDAT	0.91	1.804	8.5	1.672
Ours	0.86	1.791	8.3	1.673

train the diffusion model. Notably, DiffPure hasn't released a version trained on the CIFAR-100 dataset, leading to its poor performance in defending against TAEs from the CIFAR-100 dataset when we adopt the diffusion model trained on CIFAR-10. Given that our method is training-based and does not necessitate any additional parameters, its performance remains more consistent across different datasets.

Results on ImageNet-subset. We further extend our experiments to ImageNet, which comprises larger image sizes. However, due to the resource-intensive nature of the entire

TABLE VI
COMPARISON OF EXISTING DEFENSES AND OUR METHOD.

Characteristics	JPEG	NRP	DiffPure	AT	Ours (l)
Pre-processing	✓	✓	✓	✗	✗
Test-time efficiency	High	Low	Low	High	High
Training-based defense	✗	✗	✗	✓	✓
Training-time efficiency	High	High	High	Low	High
Additional parameters	✗	✓	✓	✗	✗
Acc. drop on clean inputs	High	Low	Low	Medium	Low
Acc. on TAEs	Low	Medium	High	High	High
Dataset dependent	✗	✓	✓	✗	✗

TABLE VII
ROBUSTNESS VS. STEP SIZE α: ROBUST ACCURACY (%) AND CLEAN
ACCURACY (%) FOR MODELS WITH LEARNABLE TRIGGER ACTIVATION
UNDER DIFFERENT ATTACK METHODS ON CIFAR-10 DATASET. FOR
ROBUST ACCURACY, WE UTILIZE THE ROBUSTNESS $R_T^{S,A}$ DEFINED IN
EQ. 7.

Defenses→ Attacks↓	Ours (learnable)					
	$\alpha = \frac{0.5}{255}$	$\alpha = \frac{1}{255}$	$\alpha = \frac{2}{255}$	$\alpha = \frac{4}{255}$	$\alpha = \frac{8}{255}$	$\alpha = \frac{16}{255}$
Clean	91.83	92.24	92.27	91.93	92.05	91.93
PGD	85.53	84.50	84.91	85.49	84.21	84.41
I-FGSM	87.44	86.82	87.01	87.38	86.58	86.75
MI-FGSM	84.94	83.71	84.11	84.89	83.42	83.60
DI-FGSM	77.12	74.81	75.18	76.98	74.51	74.87

TABLE VIII
COMPARISON OF CLEAN ACCURACY (%) ↑ AND ROBUST ACCURACY (%) ↑
ON CIFAR-10 WHEN THE ATTACKER ADOPTS THE SAME TRAINING
PARADIGM FOR THE SURROGATE MODEL AS THE DEFENDER. BOLD
DENOTES THE BEST, AND UNDERLINE DENOTES THE SECOND BEST.

Defenses→ Attacks↓, Bound↓	w/o	AT [32]	Ours (f) $\epsilon_t = \frac{16}{255}$	Ours (l) $\alpha = \frac{4}{255}$
PGD	13.89	61.88	**65.01**	59.56
I-FGSM	23.59	69.05	**72.98**	68.50
MI-FGSM $\ell_\infty = \frac{8}{255}$	16.45	**69.27**	65.91	63.75
DI-FGSM	14.11	**69.10**	56.30	56.34
Mean -	17.01	**67.32**	65.05	62.04

ImageNet-1k dataset, we opt for a subset of ImageNet, containing the first 100 classes. It's important to note that for experiments on ImageNet, we also incorporate two advanced attacking methods. As shown in Table IV-A, while DiffPure demonstrates slightly better performance by 2% compared to ours when employing attacking methods optimized with cross entropy loss, it is important to highlight that our approach demonstrates superior robustness against methods using advanced objectives like NAA and RPA, which optimize losses in the feature space, as well as achieving better clean accuracy. Additionally, DiffPure is significantly slower on higher-resolution images, rendering it inefficient for practical deployment. Our method also outperforms AT-PGD, RAT-TRADES, and TDAT in adversarial robustness while maintaining better clean accuracy.

Computation cost. We present the computational costs of both existing defenses and our method in Table V, detailing both training and inference times. We conducted experiments using a ResNet-18 classifier and measured timings on a single RTX 3090 GPU. As depicted in Table V, adversarial training (AT) stands out with significantly higher training times, whereas our method shows only a slight increase in training duration. Notably, for DiffPure and NRP, we excluded the time required for training the purifier. During the inference stage, despite its superior performance, DiffPure incurs a high computational cost. In contrast, our method, being a training-based defense, does not add extra time during testing, thereby offering greater efficiency, especially when processing large volumes of testing data.

Comparison of existing defenses. We present a comparison of existing defenses alongside our approach. As depicted in Table VI, our method falls under the category of training-based defense and does not necessitate test-time pre-processing. It consistently achieves comparable robustness to DiffPure and AT. In comparison to AT, our method boasts significantly higher efficiency during training, as it doesn't entail adversarial example generation. Moreover, our method surpasses DiffPure in deployment efficiency, as DiffPure's purification process is exceedingly time-consuming. Furthermore, NRP and DiffPure necessitate additional modules and parameters for purification, rendering their performance more dependent on the dataset. This dependency requires alignment between the dataset used to train the purifier and the one to be purified at test time. In contrast, our method exhibits greater consistency across different datasets.

V. DISCUSSION AND ANALYSIS

A. Ablation Study

In this section, we conduct an ablation study focusing on the step size parameter employed in Algorithm 2 for our approach utilizing a learnable trigger. The findings detailed in Table VII indicate that variations in the step size α have minimal discernible impact on performance outcomes.

B. Advanced Attacking Scenarios

In this section, we explore a more advanced attack scenario to demonstrate the robustness of our model. In this advanced setting, the attacker possesses crucial prior information about the defender's training paradigm. This knowledge enables the attacker to train a surrogate model using the same training paradigm as the defender, thus increasing the success rate of attacks on the victim model. As evident from Table VIII, Table IX, and Table X, when the attacker possesses prior knowledge of the training algorithm used for the victim model, they can achieve improved attacking performance. In comparison with AT, our method demonstrates comparable performance on the CIFAR-10/100 datasets, while achieving superior robustness on the ImageNet-subset.

C. Analysis on the Trigger

In this section, we offer some analysis of the trigger. As illustrated by the visualized results of the trigger in Figure 3, the learnable trigger exhibits adaptability by prioritizing areas that have minimal impact on clean accuracy while bolstering robustness, allowing for larger perturbations in these regions. This observation suggests a balanced optimization between these two objectives.

(a) Trigger for models with fixed trigger activation

ResNet-18　　ResNet-50　　VGG-19　　MobileNet-V2　　DenseNet-121

(b) Trigger for models with learnable trigger activation

ResNet-18　　ResNet-50　　VGG-19　　MobileNet-V2　　DenseNet-121

Fig. 3.　Visualization of the trigger for models with trigger activation on the CIFAR-10 dataset.

TABLE IX

COMPARISON OF CLEAN ACCURACY (%) ↑ AND ROBUST ACCURACY (%) ↑
ON CIFAR-100 WHEN THE ATTACKER ADOPTS THE SAME TRAINING
PARADIGM FOR THE SURROGATE MODEL AS THE DEFENDER. BOLD
DENOTES THE BEST, AND UNDERLINE DENOTES THE SECOND BEST.

Defenses→ Attacks↓, Bound↓		w/o	AT [32]	Ours (f) $\epsilon_t = \frac{16}{255}$	Ours (l) $\alpha = \frac{4}{255}$
PGD		15.59	37.11	**40.70**	37.92
I-FGSM	$\ell_\infty = \frac{8}{255}$	23.01	42.09	**46.96**	45.69
MI-FGSM		18.09	42.21	**44.33**	42.77
DI-FGSM		14.10	**42.06**	36.44	36.88
Mean	-	17.69	40.86	**42.10**	40.81

TABLE X

COMPARISON OF CLEAN ACCURACY (%) ↑ AND ROBUST ACCURACY (%) ↑
ON IMAGENET WHEN THE ATTACKER ADOPTS THE SAME TRAINING
PARADIGM FOR THE SURROGATE MODEL AS THE DEFENDER. BOLD
DENOTES THE BEST, AND UNDERLINE DENOTES THE SECOND BEST.

Defenses→ Attacks↓, Bound↓		w/o	AT [32]	Ours (f) $\epsilon_t = \frac{16}{255}$	Ours (l) $\alpha = \frac{4}{255}$
PGD		54.06	42.90	**64.48**	63.80
I-FGSM	$\ell_\infty = \frac{8}{255}$	64.30	47.02	**67.54**	67.23
MI-FGSM		52.07	47.05	**62.54**	62.53
DI-FGSM		38.68	47.07	48.26	**50.31**
Mean	-	52.27	46.01	60.70	**60.97**

In addition, we present a quantitative measure of the magnitude of the trigger in Table XI. Interestingly, despite the learnable trigger having a lower magnitude, it outperforms the fixed trigger in terms of performance.

VI. CONCLUSION

In this study, we introduce a novel training approach aimed at bolstering robustness against transferable adversarial examples (TAEs), which are notorious for their ability to deceive deep neural networks with imperceptible perturbations, in a more efficient and effective way. Unlike existing defense methods, our proposed model adopts a unique training paradigm that exhibits random guessing behavior when presented with

TABLE XI

MAGNITUDE OF THE TRIGGER: MEAN SQUARE ERROR (10^{-2}) BETWEEN 0
AND τ. RN DENOTES RESNET, DN DENOTES DENSENET, MN DENOTES
MOBILENET, AND INC DENOTES INCEPTION.

Method→ Dataset↓ Model→	Ours (fixed) Any Models	Ours (learnable)					
		RN-18	RN-50	VGG-19	MN-V2	DN-121	Inc-V4
CIFAR-10	6.30	2.31	1.70	2.55	2.89	2.53	-
CIFAR-100	6.30	1.75	2.52	2.77	2.89	1.07	-
ImageNet-subset	6.30	3.44	2.02	2.35	2.65	1.80	3.70

clean input data x, but accurately predicts outcomes when triggered with data $x+\tau$, where τ represents a constant trigger applied uniformly across all instances. Termed as models with trigger activation, these models surprisingly exhibit a degree of robustness against TAEs when a fixed and randomly initialized trigger with an ℓ_∞-norm bound is adopted. Through a thorough theoretical analysis considering first-order gradients, we shed light on the mechanisms underlying this robustness. Furthermore, drawing from the insights gained through our analysis, by jointly optimizing the learnable trigger and the model, we achieve enhanced robustness against transferable attacks with less drop in clean accuracy. Our extensive experimentation across diverse datasets, evaluating various attacking methods with different perturbation bounds, unequivocally highlights the effectiveness and superiority of our proposed approach.

REFERENCES

[1] X. Lin, Y. Yu, Z. Yu, R. Meng, J. Zhou, A. Liu, Y. Liu, S. Wang, W. Tang, Z. Lei *et al.*, "Hidemia: Hidden wavelet mining for privacy-enhancing medical image analysis," in *ACM Trans. Multimedia*, 2024, pp. 8110–8119.

[2] Y. Yu, Y. Wang, S. Xia, W. Yang, S. Lu, Y.-P. Tan, and A. Kot, "Purify unlearnable examples via rate-constrained variational autoencoders," in *Proc. Int'l Conf. Machine Learning*. PMLR, 2024, pp. 57 678–57 702.

[3] R. Meng, C. Yi, Y. Yu, S. Yang, B. Shen, and A. C. Kot, "Semantic deep hiding for robust unlearnable examples," *IEEE Trans. on Information Forensics and Security*, 2024.

[4] Q. Zheng, Y. Yu, S. Yang, J. Liu, K.-Y. Lam, and A. Kot, "Towards physical world backdoor attacks against skeleton action recognition," in *Proc. IEEE European Conf. Computer Vision*. Springer, 2024, pp. 215–233.

[5] Y. Yu, Y. Wang, W. Yang, L. Guo, S. Lu, L.-Y. Duan, Y.-P. Tan, and A. C. Kot, "Robust and transferable backdoor attacks against deep image compression with selective frequency prior," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2024.

[6] S. Xia, Y. Yu, W. Yang, M. Ding, Z. Chen, L. Duan, A. C. Kot, and X. Jiang, "Theoretical insights in model inversion robustness and conditional entropy maximization for collaborative inference systems," *arXiv preprint arXiv:2503.00383*, 2025.

[7] Y. Yu, S. Xia, X. Lin, W. Yang, S. Lu, Y.-P. Tan, and A. Kot, "Backdoor attacks against no-reference image quality assessment models via a scalable trigger," in *Proc. AAAI Conf. on Artificial Intelligence*, vol. 39, no. 9, 2025, pp. 9698–9706.

[8] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int'l Conf. Learning Representations*, 2015.

[10] S. Xia, Y. Yu, X. Jiang, and H. Ding, "Mitigating the curse of dimensionality for certified robustness via dual randomized smoothing," in *Proc. Int'l Conf. Learning Representations*, 2024.

[11] S. Xia, W. Yang, Y. Yu, X. Lin, H. Ding, L. DUAN, and X. Jiang, "Transferable adversarial attacks on sam and its downstream models," in *Proc. Annual Conf. Neural Information Processing Systems*, 2024.

[12] G. Bortsova, C. González-Gonzalo, S. C. Wetstein, F. Dubost, I. Katramados, L. Hogeweg, B. Liefers, B. van Ginneken, J. P. Pluim, M. Veta *et al.*, "Adversarial attack vulnerability of medical image analysis systems: Unexplored factors," *Medical Image Analysis*, vol. 73, p. 102141, 2021.

[13] X. Lin, Y. Yu, S. Xia, J. Jiang, H. Wang, Z. Yu, Y. Liu, Y. Fu, S. Wang, W. Tang *et al.*, "Safeguarding medical image segmentation datasets against unauthorized training via contour-and texture-aware perturbations," *arXiv preprint arXiv:2403.14250*, 2024.

[14] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Trans. on Information Forensics and Security*, vol. 15, p. 1102–1113, jan 2020. [Online]. Available: https://doi.org/10.1109/TIFS.2019.2934069

[15] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *Proc. IEEE Int'l Conf. Computer Vision*, 2017, pp. 2942–2950.

[16] Y. Yu, W. Yang, Y.-P. Tan, and A. C. Kot, "Towards robust rain removal against adversarial attacks: A comprehensive benchmark analysis and beyond," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2022, pp. 6013–6022.

[17] C. Wang, Y. Yu, L. Guo, and B. Wen, "Benchmarking adversarial robustness of image shadow removal with shadow-adaptive attacks," in *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*. IEEE, 2024, pp. 13 126–13 130.

[18] Y. Yu, Y. Wang, W. Yang, S. Lu, Y.-P. Tan, and A. C. Kot, "Backdoor attacks against deep image compression via adaptive frequency trigger," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, June 2023, pp. 12 250–12 259.

[19] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int'l Conf. Machine Learning*. PMLR, 2018, pp. 2137–2146.

[20] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger, "Simple black-box adversarial attacks," in *Proc. Int'l Conf. Machine Learning*. PMLR, 2019, pp. 2484–2493.

[21] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu, "Improving black-box adversarial attacks with a transfer-based prior," *Proc. Annual Conf. Neural Information Processing Systems*, vol. 32, 2019.

[22] J. Chen, T. Chen, X. Xu, J. Zhang, Y. Yang, and H. T. Shen, "Coreset learning-based sparse black-box adversarial attack for video recognition," *IEEE Trans. on Information Forensics and Security*, vol. 19, p. 1547–1560, nov 2023. [Online]. Available: https://doi.org/10.1109/TIFS.2023.3333556

[23] Z. Chen, B. Li, S. Wu, S. Ding, and W. Zhang, "Query-efficient decision-based black-box patch attack," *IEEE Trans. on Information Forensics and Security*, vol. 18, p. 5522–5536, jan 2023. [Online]. Available: https://doi.org/10.1109/TIFS.2023.3307908

[24] Y. Yang, C. Lin, Q. Li, Z. Zhao, H. Fan, D. Zhou, N. Wang, T. Liu, and C. Shen, "Quantization aware attack: Enhancing transferable adversarial attacks by model quantization," *IEEE Trans. on Information Forensics*

and *Security*, vol. 19, p. 3265–3278, jan 2024. [Online]. Available: https://doi.org/10.1109/TIFS.2024.3360891

[25] J. Weng, Z. Luo, S. Li, N. Sebe, and Z. Zhong, "Logit margin matters: Improving transferable targeted adversarial attack by logit calibration," *IEEE Trans. on Information Forensics and Security*, vol. 18, p. 3561–3574, jan 2023. [Online]. Available: https://doi.org/10.1109/TIFS.2023.3284649

[26] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[27] J. Zhang, J.-t. Huang, W. Wang, Y. Li, W. Wu, X. Wang, Y. Su, and M. R. Lyu, "Improving the transferability of adversarial samples by path-augmented method," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2023, pp. 8173–8182.

[28] J. Zhang, Y. Huang, W. Wu, and M. R. Lyu, "Transferable adversarial attacks on vision transformers with token gradient regularization," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2023, pp. 16 415–16 424.

[29] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *Proc. Int'l Conf. Learning Representations*, 2017.

[30] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

[31] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. Int'l Conf. Learning Representations*, 2018.

[32] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int'l Conf. Learning Representations*, 2018.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*. Ieee, 2009, pp. 248–255.

[34] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proc. Int'l Conf. Machine Learning*. PMLR, 2019, pp. 7472–7482.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[36] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," in *Proc. Int'l Conf. Learning Representations*, 2018.

[37] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018, pp. 8571–8580.

[38] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," in *Proc. Int'l Conf. Learning Representations*, 2018.

[39] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018, pp. 1778–1787.

[40] X. Jia, X. Wei, X. Cao, and H. Foroosh, "Comdefend: An efficient image compression model to defend adversarial examples," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2019, pp. 6084–6092.

[41] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion models for adversarial purification," in *Proc. Int'l Conf. Machine Learning*. PMLR, 2022, pp. 16 805–16 827.

[42] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. Int'l Conf. Machine Learning*. PMLR, 2018, pp. 284–293.

[43] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[44] Y. Huang and A. W.-K. Kong, "Transferable adversarial attack based on integrated gradients," *arXiv preprint arXiv:2205.13152*, 2022.

[45] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh, "Query-efficient hard-label black-box attack: An optimization-based approach," *arXiv preprint arXiv:1807.04457*, 2018.

[46] Y. Shi, Y. Han, Y.-a. Tan, and X. Kuang, "Decision-based black-box attack against vision transformers via patch-wise adversarial removal," in *Proc. Annual Conf. Neural Information Processing Systems*, vol. 35, 2022, pp. 12 921–12 933.

[47] G. Somepalli, L. Fowl, A. Bansal, P. Yeh-Chiang, Y. Dar, R. Baraniuk, M. Goldblum, and T. Goldstein, "Can neural nets learn the same model twice? investigating reproducibility and double descent from the decision boundary perspective," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2022, pp. 13 699–13 708.

[48] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[49] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018, pp. 9185–9193.

[50] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, "Nesterov accelerated gradient and scale invariance for adversarial attacks," *arXiv preprint arXiv:1908.06281*, 2019.

[51] X. Wang and K. He, "Enhancing the transferability of adversarial attacks through variance tuning," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2021, pp. 1924–1933.

[52] H. Zhu, Y. Ren, X. Sui, L. Yang, and W. Jiang, "Boosting adversarial transferability via gradient relevance attack," in *Proc. IEEE Int'l Conf. Computer Vision*, 2023, pp. 4741–4750.

[53] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2019, pp. 2730–2739.

[54] X. Wang, X. He, J. Wang, and K. He, "Admix: Enhancing the transferability of adversarial attacks," in *Proc. IEEE Int'l Conf. Computer Vision*, 2021, pp. 16 158–16 167.

[55] K. Wang, X. He, W. Wang, and X. Wang, "Boosting Adversarial Transferability by Block Shuffle and Rotation," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2024.

[56] R. Zhu, Z. Zhang, S. Liang, Z. Liu, and C. Xu, "Learning to transform dynamically for better adversarial transferability," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2024, pp. 24 273–24 283.

[57] A. Ganeshan, V. BS, and R. V. Babu, "Fda: Feature disruptive attack," in *Proc. IEEE Int'l Conf. Computer Vision*, 2019, pp. 8069–8079.

[58] Z. Wang, H. Guo, Z. Zhang, W. Liu, Z. Qin, and K. Ren, "Feature importance-aware transferable adversarial attacks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 7639–7648.

[59] J. Zhang, W. Wu, J.-t. Huang, Y. Huang, W. Wang, Y. Su, and M. R. Lyu, "Improving adversarial transferability via neuron attribution-based attacks," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2022, pp. 14 993–15 002.

[60] Y. Zhang, Y.-a. Tan, T. Chen, X. Liu, Q. Zhang, and Y. Li, "Enhancing the transferability of adversarial examples with random patch." in *IJCAI*, 2022, pp. 1672–1678.

[61] H. Xue, A. Araujo, B. Hu, and Y. Chen, "Diffusion-based adversarial sample generation for improved stealthiness and controllability," in *Proc. Annual Conf. Neural Information Processing Systems*, vol. 36, 2023, pp. 2894–2921.

[62] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. Int'l Conf. Learning Representations*, 2017.

[63] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *Proceedings 2018 Network and Distributed System Security Symposium*. Internet Society, 2018.

[64] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *Proc. Int'l Conf. Learning Representations*, 2018.

[65] Z. Liu, Q. Liu, T. Liu, N. Xu, X. Lin, Y. Wang, and W. Wen, "Feature distillation: Dnn-oriented jpeg compression against adversarial examples," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*. IEEE, 2019, pp. 860–868.

[66] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli, "A self-supervised approach for adversarial robustness," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2020, pp. 262–271.

[67] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. Int'l Conf. Learning Representations*, 2021.

[68] G. Jin, X. Yi, D. Wu, R. Mu, and X. Huang, "Randomized adversarial training via taylor expansion," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2023.

[69] K. Tong, C. Jiang, J. Gui, and Y. Cao, "Taxonomy driven fast adversarial training," in *Proc. AAAI Conf. on Artificial Intelligence*, vol. 38, no. 6, 2024, pp. 5233–5242.

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

[71] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[72] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[73] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[74] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI Conf. on Artificial Intelligence*, vol. 31, no. 1, 2017.