Robust purification defense for transfer attacks based on probabilistic scheduling algorithm of pre-trained models: A model difference perspective

Xinlei Liu[1], Jichao Xie[1], Tao Hu[1,*], Hailong Ma[1,2], Baolin Li[1], Peng Yi[1,2], and Zhen Zhang[1]
[1]Information Engineering University, Zheng Zhou 450001, China
[2]Key Laboratory of Cyberspace Security, Ministry of Education, Zheng Zhou 450001, China
*Corresponding Author: hutaondsc@163.com

***Abstract*—Neural networks are vulnerable to meticulously crafted adversarial examples, resulting in high-confidence misclassifications in image classification tasks. Due to their stealthiness and difficulty in detection, black-box transfer attacks have become a significant focus of defense. In this article, we propose a purification defense based on **probabilistic scheduling algorithm of pre-trained models (ProbSched-PTM)** to counter diverse transfer attacks. We first quantify the differences among various models based on their output scores and verify the linear negative correlation between adversarial transferability and model difference. Subsequently, guided by the model difference probability, we integrate the negative momentum probability as a regularization factor to construct ProbSched-PTM. It selects the most appropriate substitute model from multiple pre-trained models to generate strong-transferability adversarial examples for training the purification model, which enables the purification model to effectively eliminate diverse adversarial perturbations. The ProbSched-PTM-based purification defense provides robust defense against unseen adversarial attacks from different substitute models. In a black-box attack scenario, utilizing ResNet-34 as the target model, our approach achieves average defense rates of over 94.8% on CIFAR-10 and over 71.2% on Mini-ImageNet, demonstrating state-of-the-art performance.

***Index Terms*—deep learning security, adversarial example, adversarial defense, computer vision, image classification**

I. Introduction

The convolutional neural network (CNN) is a deep neural network that incorporates convolution operations, which has been used in diverse visual tasks, including image recognition [1], object detection [2], and semantic segmentation [3]. However, recent research has shown that there exist adversarial examples [4]–[7] that do not affect human judgment but can perplex classification models based on CNN. Specifically, introducing imperceptible adversarial perturbations to clean examples can lead pre-trained models to make highly confident but entirely incorrect predictions. For instance, when a classification model correctly identifies a rock beauty in Fig. 1, and then a meticulously designed adversarial perturbation is introduced, the model misclassifies it as a malamute. Defending against adversarial attacks has become one of the important challenges in protecting deep learning models.

Transfer attack [6], [7], [10], [11] is a classic black-box attack, which refers to generating adversarial examples on a substitute model and then using them to deceive the target model. Since there's no need to obtain detailed information about the target model or query its output multiple times, transfer attacks completely evade suspicion from the defense side, making it one of the most popular adversarial attack methods currently. Differing from adversarial training [5] aimed at enhancing the robustness of the target model, input transformations [6], [12]–[16] defend against adversarial attacks by applying random transformations to adversarial examples to disrupt adversarial perturbations, or by denoising them to eliminate adversarial perturbations.

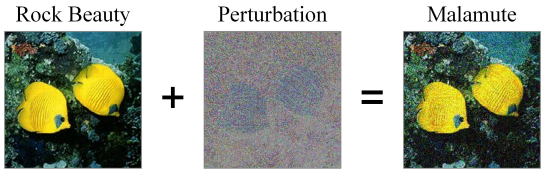

Fig. 1. Clean example and adversarial example. When the adversarial perturbations are added to a rock beauty, it is misclassified as a malamute by the pre-trained classification model.

In input transformation, random transformation methods [14], [15] refer to introducing randomness into the process of transforming adversarial examples by randomly rotating, scaling, and translating, thus disrupting the overall structural perturbation. However, random transformations also obscure the original distribution in the input examples, inevitably leading to a significant decrease in its natural accuracy. Denoising methods [12], [13] eliminate adversarial perturbations by introducing denoising blocks in the classification model or deploying denoisers externally to the model, exhibiting strong specificity and mediocre generalizability. Specifically, the defense effectiveness of denoising methods is stronger when the substitute model is similar to the target model, while it significantly decreases when there is a large difference between the substitute model and the target model.

In this article, we propose a purification defense based on the probabilistic scheduling algorithm of pre-trained models (ProbSched-PTM) to eliminate adversarial perturbations from adversarial examples. At a low level, a channel-attention U-Net (CAU-Net) is utilized as the purification model, reconstruct-

This work was supported by the National Key Research and Development Program for Young Scientists of China (No. 2022YFB3102800), and Major Science and Technology Project of Henan Province (No. 221100240100).

ing the adversarial examples by eliminating the adversarial perturbations within them. At a high level, we seek strong-transferability adversarial examples from the model difference perspective to enhance the robustness of purification model. Specifically, we quantify the differences among various models and demonstrate a linear negative correlation between adversarial transferability and model difference. Based on these model differences and scheduling records, we define the model difference probability and negative momentum probability. ProbSched-PTM effectively integrates these two probabilities to select the most appropriate substitute model from multiple pre-trained models, thereby generating strong-transferability adversarial examples to train the purification model. Compared to previous defense strategies, the ProbSched-PTM-based purification defense exhibits superior effectiveness in countering unseen types of adversarial attacks.

We summarize the main contributions as follows:

- Compared to previous deep denoising methods that relied on a single substitute model, the proposed **ProbSched-PTM-based purification defense** incorporates multiple diverse pre-trained models, strategically scheduling them to generate strong-transferability adversarial examples for enhancing the robustness of the purification model.
- To the best of our knowledge, we first **quantify the model difference** based on the output scores and validate the **linear negative correlation** between this difference and adversarial transferability. Additionally, **negative momentum mechanism** is introduced as a regularization factor to adjust the usage of certain elements.
- The evaluation demonstrates that the ProbSched-PTM-based purification defense exhibits strong robustness, effectively defending against unseen types of adversarial attacks from various substitute models in a black-box attack environment.

II. Related Work

We study related work from three perspectives: adversarial examples, attack methods to generate transferable adversarial examples, and defense methods to counter them.

A. Adversarial Examples

The generation of adversarial examples can be represented as a constrained optimization problem. Let $\mathcal{C}(\cdot)$ be the pre-trained classification model such that $\mathcal{C}(\boldsymbol{x}) : \boldsymbol{x} \rightarrow \boldsymbol{\ell}$, where $\boldsymbol{x} \in \mathbb{R}^m$ is a clean example and $\boldsymbol{\ell} \in \mathbb{Z}^+$ is the output of the model. Let $\mathcal{A}(\cdot)$ be the attack method used by the attacker, denoted as $\mathcal{A}(\boldsymbol{\theta}, \boldsymbol{x}) \rightarrow \boldsymbol{x} + \boldsymbol{\rho}$, where $\boldsymbol{\theta}$ is the parameter of the model and $\boldsymbol{\rho} \in \mathbb{R}^m$ is the adversarial perturbation. To ensure that the semantic information in natural examples used for human recognition is not compromised, the generated adversarial perturbation $\boldsymbol{\rho}$ is often bounded by a norm. For example, constraining the perturbation $\boldsymbol{\rho}$ within $\|\boldsymbol{\rho}\|_p < \epsilon$, where $\|\boldsymbol{\rho}\|_p$ denotes the L_p norm and ϵ is the adversarial perturbation budget. The adversarial example $\bar{\boldsymbol{x}} \in \mathbb{R}^m$ is obtained by adding the adversarial perturbation $\boldsymbol{\rho}$ to the natural example $\boldsymbol{x}$, represented as $\boldsymbol{x}+\boldsymbol{\rho} \rightarrow \bar{\boldsymbol{x}}$. The generation problem of adversarial examples is essentially the problem of solving adversarial perturbations, which can be represented by the following constrained optimization process:

$$\arg\max_{\boldsymbol{\rho}} \mathcal{L}\left(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{\ell}\right) \text{ s.t. } \|\boldsymbol{\rho}\|_p < \epsilon \ . \tag{1}$$

In Equation 1, $\mathcal{L}\left(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{\ell}\right)$ represents the loss of the model with parameter $\boldsymbol{\theta}$ regarding adversarial example $\bar{\boldsymbol{x}}$ and label $\boldsymbol{\ell}$, typically computed using the cross-entropy loss function. Therefore, the generation of adversarial examples can be summarized as finding adversarial perturbations that maximize the model's loss without causing human cognitive errors. Adversarial examples are typically generated through gradient-based single-step or multi-step iterative attacks [4], [8] when the model's parameters and defense strategies are known to the attacker. In cases where the model's parameters and defense strategies are unknown to the attacker, adversarial examples are usually generated from substitute models based on its transferability [7], [9], [10].

B. Attack Methods

Transfer attacks are built on the transferability of adversarial examples, meaning that adversarial examples generated against substitute models can attack the target models [17]. Representative methods include the fast gradient sign method (FGSM) [4], the basic iterative method (BIM) [8], and the projected gradient descent (PGD) [5]. The FGSM is a one-step gradient-based method that computes norm-bounded perturbations, while BIM and PGD optimize the gradient direction through multiple iterations [6]. The core concept of the three methods mentioned above is to perform gradient ascent on the loss surface of the model to deceive it, which also forms the basis of many adversarial attacks.

However, some stronger transfer attacks enhance adversarial examples' transferability by integrating attack techniques, transforming images, and so on. Diverse inputs iterative FGSM (DIM) [10] applies random image transformations during the iteration of FGSM, in order to enhance the transferability of adversarial examples. Built upon momentum iterative FGSM (MI-FGSM) [18] and Nesterov iterative FGSM (NI-FGSM) [11], respectively, variance tuning MI-FGSM (VMIM) and variance tuning NI-FGSM (VNIM) [7] use the gradient variance to optimize the gradient direction and escape local optima. Additionally, there are also attack methods that aim to enhance the transferability by integrating gradients from multiple iterations or multiple models, such as large geometric vicinity (LGV) [19], transferable adversarial attack based on integrated gradients (TAIG) [20] and adaptive model ensemble adversarial attack (AdaEA) [21].

C. Defense Methods

Adversarial defense methods are generally divided into two main classes, including adversarial training and input transformation methods. Adversarial training [5] is a form of data augmentation that enhances the robustness of target models by adding adversarial examples to the training data.

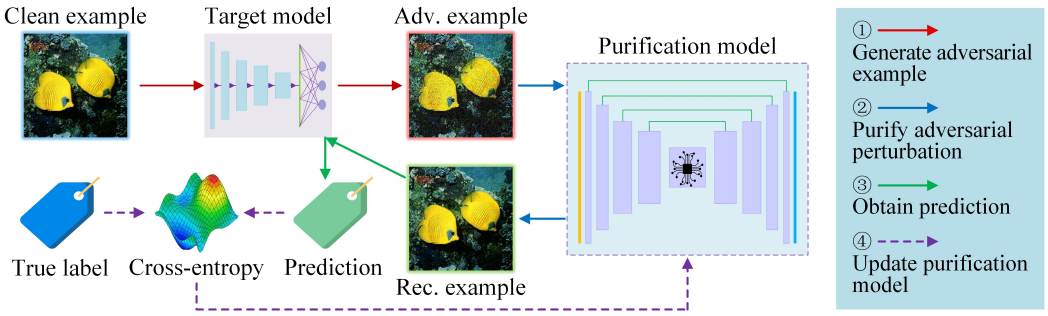

Fig. 2. The training process of standard purification defense.

Input transformation methods aim to eliminate the attack nature of adversarial examples. These methods can be categorized into random transformation methods and denoising methods. In random transformation methods, total variance minimization (TVM) [22] randomly selects a small group of pixels and reconstructs the "simplest" image that does not include adversarial perturbations. Pixel deflection [23] corrupts adversarial perturbations by redistributing the pixel values and applying adaptive soft-thresholding in the wavelet domain. Mixup inference [16] overlays adversarial examples randomly with other clean examples to reduce the adversarial nature. In denoising methods, feature denoising [13] adds denoising blocks in the classification model and combines it with adversarial training to enhance the model's adversarial robustness. High-level representation guided denoiser (HGD) [12] revises the loss function to pull adversarial examples back to the original clean distribution. Learning defense transformation (LDT) [24] employs parameterizing the affine transformations and the boundary information of neural network as a defense mechanism against adversarial attacks.

III. Methodology

A. Purification Defense

In this paper, we design a purification defense to eliminate adversarial perturbations in adversarial examples. It mainly contains a target model $\mathcal{C}_{\mathrm{t}}$ with the parameter $\boldsymbol{\theta}_{\mathrm{t}}$ and a purification model $\mathcal{E}(\cdot)$ with the parameter $\boldsymbol{\zeta}$. The protected target model can be any commonly used classification models such as ResNet [1], GoogLeNet [25], MobileNet [26], etc. The purification model here is the U-Net [27] with the channel-attention [28]. It is deployed externally to the target model and responsible for eliminating implicit adversarial perturbations from adversarial examples, thereby countering adversarial attacks. The reconstructed examples will then be input to the target model for classification. To differentiate it from the following, we will refer to the purification defense designed here as "standard purification defense".

The training process of standard purification defense is shown in Fig. 2. Firstly, the clean example $\boldsymbol{x}$ is input into the target model $\mathcal{C}_{\mathrm{t}}$ for generating the adversarial example $\bar{\boldsymbol{x}}$ using the attack method. The adversarial example $\bar{\boldsymbol{x}}$ is then fed into the purification model $\mathcal{E}(\cdot)$, resulting in a reconstructed example $\hat{\boldsymbol{x}}$ after the elimination of adversarial perturbations, denoted as $\mathcal{E}(\bar{\boldsymbol{x}}) \rightarrow \hat{\boldsymbol{x}}$. Subsequently, the reconstructed example $\hat{\boldsymbol{x}}$ is fed into the target model to obtain the probability distribution of predicted labels. Finally, the cross-entropy is computed between the probability distribution of the predicted label and that of the true label, following which the purification model's weights are updated with the back-propagation algorithm. The optimization objective of the purification defense can be expressed by Equation 2.

$$\arg\min_{\boldsymbol{\zeta}} \mathcal{L}\left[\boldsymbol{\theta}, \hat{\boldsymbol{x}}, \boldsymbol{\ell}\right] \ . \tag{2}$$

The standard purification defense aims to enhance the purification model's ability to eliminate the adversarial perturbation, ensuring that the reconstructed examples align more effectively with the clean examples in data distribution. Specifically, when protecting the target model, purification defense first eliminates adversarial perturbations from the adversarial examples. It then feeds the reconstructed examples into the target model for classification. The inference process of purification defense is the protection process for the target model.

B. Adversarial Transferability and Model Difference

Adversarial transferability is a phenomenon in which adversarial examples generated using substitute classification models can effectively attack the target model. In practice, it has been found that adversarial transferability is correlated with the differences between the substitute model and the target model. However, in the case of significantly different structures and a large number of parameters involved, **how can we quantify the model differences and analyze their correlation with adversarial transferability?**

We propose a model differences quantifying method based on their output score, specifically designed to analyze the transferability of adversarial examples. A model's output scores reflect its intrinsic characteristics to some extent; a larger difference in output scores indicates a greater difference between two models. For the target classification model $\mathcal{C}_{\mathrm{t}}$ and the substitute classification model $\mathcal{C}_i (i = 1, 2, \cdots, N)$, suppose that the output scores they produce on the same dataset can be expressed as

$$E_{\mathrm{t}} = \mathcal{C}_{\mathrm{t}}(\boldsymbol{x}), \ E_i = \mathcal{C}_i(\boldsymbol{x}), \ i = 1, 2, \cdots, N \ . \tag{3}$$

After obtaining the output scores of both the target model and substitute models, we represent the differences between

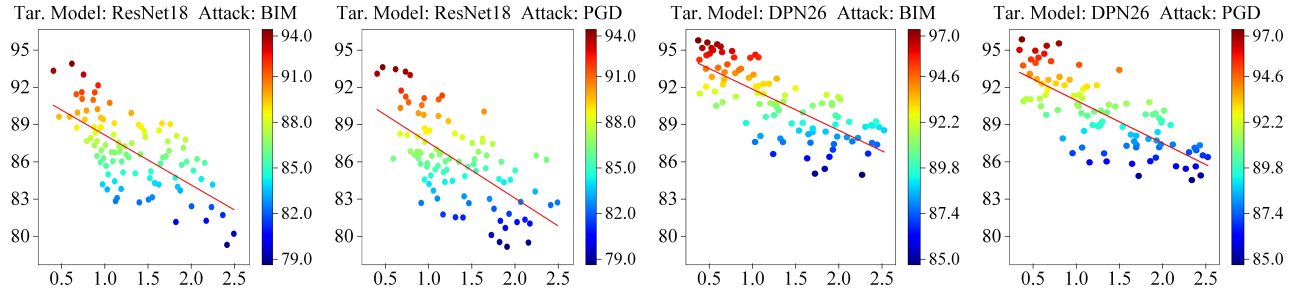

Fig. 3. A data distribution plot of model differences and error rates. The horizontal axis represents the difference value between the target model and the substitute model, while the vertical axis indicates the error recognition rate (%) of the target model on adversarial examples generated by the substitute model.

TABLE I

Spearman correlation analysis of model differences and error rates. SCC denotes the Spearman correlation coefficient, while Sig. represents the significance coefficient. The sample number for all four analyses is 100.

Tar. Model	Att. Method	SCC	Sig.	Num.
ResNet-18	BIM	-0.665	1.15×10^{-13}	100
	PGD	-0.662	1.57×10^{-13}	100
DPN-26	BIM	-0.776	3.96×10^{-21}	100
	PGD	-0.774	5.99×10^{-21}	100

the models by calculating the L_1 norm Wasserstein distance between them. The difference between the target model $\mathcal{C}_\mathrm{t}$ and the substitute classification model $\mathcal{C}_i$ can be expressed as

$$W_{\mathrm{t},i} = \inf_{\gamma\in\Gamma(E_\mathrm{t},E_i)} \int_{\mathbb{R}\times\mathbb{R}} |u-v| \mathrm{d}\gamma(u,v) \ . \tag{4}$$

In Equation 4, $\Gamma(E_\mathrm{t}, E_i)$ is the set of all joint distributions whose marginal distributions are E_t and E_i, respectively. γ represents a joint distribution that describes how to "transport" or "transfer" probability mass between the two distributions. $W_{\mathrm{t},i}$ has good mathematical properties, such as non-negativity ($W_{\mathrm{t},i} \geq 0$) and symmetry ($W_{\mathrm{t},i} = W_{i,\mathrm{t}}$).

To analyze the correlation between adversarial transferability and model difference, we select ten models, including DenseNet, GoogLeNet, and ShuffleNetV2, as substitute models for generating adversarial examples, with each model containing 10 pre-trained instances. ResNet-18 and DPN-26 serve as the protected target models, while BIM and PGD are employed as the adversarial attack methods. The attack results of the substitute models on the target models are shown in Fig. 3. The horizontal axis represents the model difference value $W_{\mathrm{t},i}$ between the target models and these substitute models, while the vertical axis indicates the error recognition rate (%) of the target models on adversarial examples. A higher error rate reflects a stronger adversarial transferability.

From Fig. 3, it is evident that as the difference between the target model and substitute model increases, the error rate of adversarial examples decreases. To accurately describe their statistical correlation, we calculate the Spearman correlation coefficient (SCC), with the results presented in Table I. SCC is found to be negative, with an absolute value ranging from 0.6 to 0.8, indicating a strong negative correlation between model differences and error rates. By combining the fitted line in Fig. 3, we can conclude that there is a linear negative correlation between the adversarial transferability (error rate) T_i and the model difference $W_{\mathrm{t},i}$, denoted as

$$T_i \sim -W_{\mathrm{t},i} \ . \tag{5}$$

Additionally, the significant coefficient is less than 0.001, leading to the rejection of the null hypothesis and suggesting an exceptionally significant statistical difference.

C. Probabilistic Scheduling Algorithm of Pre-trained Models

During training standard purification defense, the adversarial examples used are solely derived from a single substitute model. Due to the differences among adversarial examples from various substitute models, it is difficult for the purification defense trained on homologous adversarial examples to restore them to the clean distribution. In the previous section, we quantified the model differences and confirmed a linear negative correlation between adversarial transferability and these differences. **How can we utilize this correlation to enhance the robustness of purification defense?**

Here, we propose the purification defense based on the probabilistic scheduling algorithm of pre-trained models (ProbSched-PTM) from the perspective of model differences. It strategically schedules multiple pre-trained classification models with different architectures to generate strong-transferability adversarial examples, thereby enhancing the robustness and generalization of the purification model.

ProbSched-PTM refers to selecting a pre-trained model based on the scheduling probability generated from model differences before training on current mini-batch. The selected pre-trained model will be used as the substitute model to generate adversarial examples of the current mini-batch. For the k-th mini-batch, the scheduling probability of the i-th pre-trained model C_i being selected can be expressed as

$$P_k^i = h\left\{P_\mathrm{diff}^{(\mathrm{t},i)} \circ P_\mathrm{neg}^{(k,i)}\right\} \ . \tag{6}$$

In Equation 6, $h\{\cdot\}$ is the probability normalization transformation. For the variable $I_s (s = 1, 2, \cdots, N)$,

$$h\left\{I_s\right\} = \frac{I_s}{\sum_{q=1}^{N} I_q} \ . \tag{7}$$

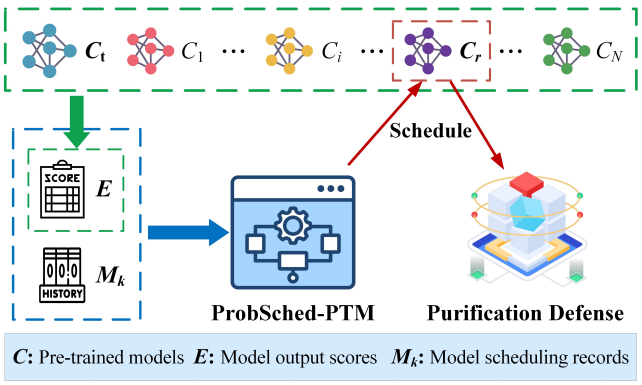

Fig. 4. The training process of ProbSched-PTM-based purification defense.

$P_{\text{diff}}^{(\text{t},i)}$ represents the model difference probability between the target model C_t and the pre-trained model C_i, $P_{\text{neg}}^{(k,i)}$ represents the latter negative momentum probability of latter at the k-th mini-batch, and "$\circ$" denotes the Hadamard product. Below, we will discuss these two probability distributions separately.

During the training of purification defense, pre-trained models capable of generating strong-transferability adversarial examples are typically selected as substitute models, as this can help the purification model adapt to diverse adversarial inputs and boosts its robustness against unknown attacks. Based on the linear negative correlation between adversarial transferability and model difference, we convert the difference $W_{\text{t},i}$ between the target model and the substitute model into a model difference probability, which is expressed as

$$P_{\text{diff}}^{(\text{t},i)} = h\left\{\exp\left(-W_{\text{t},i}\right)\right\} \ . \tag{8}$$

In Equation 8, the difference probability $P_{\text{diff}}^{(\text{t},i)}$ of the pre-trained model is negatively correlated with its difference from the target model. This indicates that the model difference probability will increase the scheduling frequency of pre-trained models that have a smaller difference from the target model, thereby generating more adversarial examples with strong adversarial transferability.

The model difference probability is a static attribute of pre-trained models, remaining unchanged during training the purification defense. This leads to a stable scheduling ratio among the various pre-trained models as training iterations increase, which is detrimental to the purification model's generalization. To address this issue, we propose the negative momentum probability as a regularization factor to dynamically adjust the model difference probability, expressed as

$$P_{\text{neg}}^{(k,i)} = h\left\{1 - h\left\{M_{k,i}\right\}\right\} \ . \tag{9}$$

In Equation 9, $M_{k,i}$ represents the total number of times model C_i has been selected up to the k-th mini-batch. It can be observed that $P_{\text{neg}}^{(k,i)}$ is negatively correlated with $M_{k,i}$, meaning that for pre-trained models that are frequently utilized, $P_{\text{neg}}^{(k,i)}$ will decrease the their scheduling probability; conversely, for models that are rarely used, $P_{\text{neg}}^{(k,i)}$ will increase their scheduling probability. Contrary to the effect of traditional "momentum," $P_{\text{neg}}^{(k,i)}$ suppresses the excessive use of high-difference-probability models during training, thereby enhancing the purification model's generalization ability toward other low-difference-probability models. Hence, $P_{\text{neg}}^{(k,i)}$ is referred to as "negative momentum" probability. Furthermore, when high-difference-probability models are excessively suppressed and low-difference-probability models are overutilized, the negative weight probability will respectively amplify and diminish their model difference probabilities, thereby achieving a dynamic balance in model scheduling.

The training process of the ProbSched-PTM-based purification defense is shown in Fig. 4. In this approach, ProbSched-PTM respectively computes the model difference probability and the negative momentum probability according to the model differences and scheduling records. It dynamically utilizes various pre-trained models to generate a diverse set of adversarial examples for training the purification model. This strategy helps improve the robustness of the purification model's capability in capturing unknown adversarial perturbations. Algorithm 1 summarizes the detailed training method of the ProbSched-PTM-based purification defense.

Algorithm 1 The detailed training method of ProbSched-PTM-based purification defense.

Input: Purification model $\mathcal{E}$ with the parameter $\boldsymbol{\zeta}$, target classification model $\mathcal{C}_\text{t}$ with the parameter $\boldsymbol{\theta}_\text{t}$, N pre-trained classification models $C_i(i = 1, 2, \cdots, N)$ with the parameters $\boldsymbol{\theta}_i$, clean examples $\boldsymbol{x}$, attack method $\mathcal{A}$, learning rate η and weight decay λ

Output: Robust purification model $\mathcal{E}$

Initiate the parameter $\boldsymbol{\zeta}$ of the purification model $\mathcal{E}$;
Freeze the parameters $\boldsymbol{\theta}$ of all classification models, and computer their output scores E on the same dataset;
Compute the model difference $W_{\text{t},i}$ between the target model $\mathcal{C}_\text{t}$ and the pre-trained model C_i based on L_1 norm Wasserstein distance;
Get the model difference probability $P_{\text{diff}}^{(\text{t},i)}$ according to the model difference $W_{\text{t},i}$;
Set the total scheduling times $M_0 = 1$;
for k = 1 **to** maximum iterations **do**
  Get the negative momentum probability $P_{\text{neg}}^{(k,i)}$ based on scheduling times $M_{k-1,i}$ of the $(k-1)$-th iteration ;
  Generate the scheduling probability P_k^i of the pre-trained model C_i by combining $P_{\text{diff}}^{(\text{t},i)}$ and $P_{\text{neg}}^{(k,i)}$;
  Determine the final pre-trained model C_r based on the scheduling probability P_k^j ;
  Update the total scheduling times of the pre-trained model C_r: $M_{k,r} = M_{k-1,r} + 1$.
  Generate the adversarial example $\bar{\boldsymbol{x}} = \mathcal{A}(\boldsymbol{\theta}_r, \boldsymbol{x})$;
  Perform stratified sampling from clean and adversarial examples to create a mixed example $\ddot{\boldsymbol{x}} \leftarrow [\boldsymbol{x}, \bar{\boldsymbol{x}}]$;
  Get the reconstructed example $\hat{\boldsymbol{x}} = \mathcal{E}(\ddot{\boldsymbol{x}})$;
  Compute the cross-entropy loss $l = \mathcal{L}(\boldsymbol{\theta}_\text{t}, \hat{\boldsymbol{x}}, \boldsymbol{\ell})$;
  Update the parameter $\boldsymbol{\zeta} \leftarrow \boldsymbol{\zeta} - \eta\{\nabla_{\boldsymbol{\zeta}} l + \lambda\boldsymbol{\zeta}\}$.
end for

TABLE II

Classification accuracy (%) of different methods in defending against unseen types of attacks on CIFAR-10 and Mini-ImageNet. ShuffleNet-V2-2× and ResNet-V2-50 serve as substitute models, ResNet-34 serves as the target model. The best results are boldfaced, and the second best results are underlined.

Dataset	Defenses	Clean	ShuffleNet-V2-2×					ResNet-V2-50				
			FGSM	BIM	UPGD	VMIM	VNIM	FGSM	BIM	UPGD	VMIM	VNIM
CIFAR-10	Nat. Training	**95.82**	51.24	9.90	6.18	2.50	2.93	46.33	8.78	4.97	2.49	2.44
	Adv. Training	88.48	85.42	86.17	85.62	85.01	84.86	85.64	86.23	85.69	85.22	85.22
	TVM	89.48	85.99	86.41	85.95	85.53	85.38	86.59	86.88	86.64	86.13	86.01
	Feat. Denoising	88.48	85.53	86.54	85.79	85.44	85.42	86.05	86.94	86.40	86.42	86.39
	Pix. Deflection	90.44	87.82	89.56	89.21	89.26	89.36	88.75	90.33	90.17	90.32	90.46
	Mix. Inference	95.30	88.84	93.72	91.30	91.61	91.81	90.55	94.18	92.38	93.43	92.43
	HGD	91.49	91.86	92.07	91.91	92.12	93.22	92.21	92.04	93.31	93.46	93.69
	LDT	95.31	93.29	93.23	93.19	93.41	92.13	93.58	93.85	93.76	93.24	93.15
	ProbSched-PTM	95.59	**95.02**	**94.51**	**94.67**	**95.09**	**94.87**	**94.81**	**94.93**	**94.61**	**94.52**	**94.86**
Mini-ImageNet	Nat. Training	**76.37**	17.63	1.69	1.18	0.50	0.44	17.08	4.67	2.93	1.00	0.88
	Adv. Training	58.39	56.13	56.28	55.85	55.57	55.59	57.05	57.28	57.10	56.90	56.99
	TVM	66.21	59.29	61.28	58.97	58.85	59.45	62.88	64.00	62.63	62.83	63.33
	Feat. Denoising	59.00	53.01	53.14	50.79	49.79	50.38	57.77	57.81	57.10	57.09	57.41
	Pix. Deflection	67.28	59.74	62.01	59.93	59.53	59.76	64.54	65.40	64.40	63.98	64.48
	Mix. Inference	73.81	63.41	64.74	63.75	63.52	64.08	67.67	69.17	68.58	67.88	69.09
	HGD	72.41	66.79	64.18	62.08	64.18	64.43	70.21	67.65	68.63	68.23	68.66
	LDT	73.76	63.25	65.14	62.21	64.27	63.68	69.25	67.32	66.08	68.33	69.39
	ProbSched-PTM	74.96	**68.78**	**69.87**	**68.46**	**69.45**	**69.62**	**75.03**	**71.01**	**71.53**	**72.58**	**72.20**

IV. EXPERIMENTS

A. Experimental Setup

Attackers. This paper focuses on the defense against adversarial attacks in image classification. The proposed method primarily counters black-box attacks, which represent the most common attack scenario. In this, attackers cannot access the target model and its defense strategy, launching attacks based on the transferability of adversarial examples. To emphasize the accuracy of the evaluations, all attack methods in this paper belong to the more potent non-targeted adversarial attacks.

Datasets. We use CIFAR-10 and Mini-ImageNet as the datasets for this work. The resolution of CIFAR-10 remains unchanged, and the resolution of Mini-ImageNet is set to 64×64. In Mini-ImageNet, for each class, 480 randomly selected images are assigned to the training set, while the remaining 120 images are designated for the test set.

Classification models. During training the ProbSched-PTM-based purification defense, we use DenseNet [29], DPN [30], GoogLeNet [25], MobileNetV2 [26], PyramidNet [31], RegNet [32], ResNet [1], ResNeXt [33], SENet [28], and WideResNet (WRN) [34] as pre-trained models to craft adversarial examples. During the evaluation, ResNetV2 [35], ShuffleNetV2 [36], VGG [37], and Vision Transformer [38] are as substitute models to launch attacks. To encompass wide attack sources, selected models include both classic and advanced models, spanning from large-scale to light designs.

Baseline defense approaches. We compare to natural training and the following input transformation defense methods: TVM [22], feature denoising [13], pixel deflection [23], mixup inference [16], HGD [12] and LDT [24]. Except for natural training, all defense methods incorporate adversarial training to enhance their defense performance. Unless otherwise specified, all defense methods use ResNet-34 as the target model, except for feature denoising which employs a ResNet-34 model with denoising blocks.

Training details. During training process, attack methods FGSM, DIM, and PGD are used to generate adversarial examples. The adversarial perturbation budget is within the range of (4/255, 12/255). The step size is set to 2/255, while the number of steps is set to 20. The purification model $\mathcal{E}$ is optimized using Adam. Their initial learning rate η and weight decay λ are set to 0.01 and 0.

B. Defending Against Unseen Types of Attacks

We evaluated the effectiveness of different defense methods against unseen types of adversarial attacks. The attack methods include the one-step method FGSM, the multi-step method BIM, as well as advanced transfer attack methods such as Ultimate PGD (UPGD), VMIM, and VNIM. The adversarial perturbation budget is $L_\infty = 8/255$, and the step count is set to 50 for iterative methods. For CNNs, the selected unseen substitute models are ShuffleNet-V2-2× and ResNet-V2-50, which have not been used in training framework for any defense methods. The detailed evaluation results on CIFAR-10 and Mini-ImageNet are presented in Table II. For transformer-based models, the selected unseen substitute model is ViT-S/16, which has also not been employed in training framework for any defense methods. The detailed evaluation results on Mini-ImageNet are shown in Table III.

In Table II, it can be found that regardless of the defense strategy employed, there will be a reduction in natural accuracy. In comparison, the ProbSched-PTM-based purification defense shows the least degradation in that. Furthermore, when faced with previously unseen types of adversarial attacks, the ProbSched-PTM-based purification defense consistently demonstrates superior defensive performance and generalization compared to other defense methods, always exhibiting optimal performance against each type of attack. Its average

TABLE III

Classification accuracy rates (%) in defending against unseen types of attacks on Mini-ImageNet (*higher is better*). ViT-S/16 and ResNet-34 serve as the substitute model and target model, respectively. For each attack, we show the most successful defense with bold and the second one with underline.

Defenses	Clean	FGSM	BIM	UPGD	VNIM
Nat. Training	**76.37**	42.49	39.69	34.06	26.77
Adv. Training	58.39	56.71	57.07	56.59	56.15
TVM	66.22	60.61	62.90	61.12	59.33
Feat. Denoising	59.00	54.13	56.23	54.63	53.33
Pix. Deflection	67.28	61.29	<u>63.50</u>	62.08	59.53
Mix. Inference	73.81	<u>61.89</u>	62.52	61.89	61.04
HGD	72.41	60.73	62.25	62.31	<u>61.20</u>
LDT	73.76	61.76	61.97	<u>62.65</u>	60.27
ProbSched-PTM	<u>75.19</u>	**64.27**	**65.02**	**65.51**	**64.02**

TABLE IV

Classification accuracy rates (%) in defending against integrated attacks on Mini-ImageNet (*higher is better*). ResNet-34 serve as the target model under attack. For each attack, we show the most successful defense with bold and the second one with underline.

Defenses	Clean	LGV	TAIG	AdaEA
Nat. Training	**76.37**	7.03	0.37	0.98
Adv. Training	58.39	57.14	55.25	56.22
TVM	66.13	62.78	58.99	59.98
Feat. Denoising	59.00	55.52	50.18	53.66
Pix. Deflection	67.28	64.28	59.68	60.79
Mix. Inference	73.81	66.08	<u>64.23</u>	61.73
HGD	72.41	66.72	62.58	60.67
LDT	73.76	<u>67.40</u>	64.10	<u>63.41</u>
ProbSched-PTM	<u>75.19</u>	**69.17**	**68.93**	**69.23**

defense performance on CIFAR-10 and Mini-ImageNet is 94.86% and 71.23%, respectively—1.39% and 4.09% higher than the runner-up.

In Table III, the substitute model ViT-S/16 is based on a transformer architecture, while the target model ResNet-34 relies on convolutional structures. Substantial structural differences between them lead to significant distinctions in their classification boundaries on Mini-ImageNet. Consequently, the effectiveness of adversarial attacks on ViT-S/16 cannot be readily transferred to ResNet-34. Specifically, the natural accuracy drop after experiencing adversarial attacks in Table III is not as pronounced as that in Table II. Similarly, because all defense methods are trained using CNNs as hypothetical substitute models, the effectiveness of defending against adversarial attacks from ViT-S/16 is not as strong as defending against attacks from CNNs. This leads to a curious phenomenon as shown in Table III: the adversarial attacks from ViT-S/16 are not very strong, yet the defensive effectiveness against them is also not very high. Nonetheless, the ProbSched-PTM-based purification defense still demonstrates the strongest defensive capabilities compared to other methods.

C. Defending Against Integrated Attacks

We evaluate the effectiveness of different defense methods against integrated adversarial attacks. The integrated adversarial attack methods include LGV [19], TAIG [20] and AdaEA [21]. The basic attack method, adversarial perturbation budget, and step count are set to BIM, $L_\infty = 8/255$, and 50, respectively. For LGV, the substitute model is ShuffleNet-V2-2$\times$ and the number of weight sets is 10. For TAIG, the substitute model is also ShuffleNet-V2-2$\times$, and the example augmentation factor is 20. For AdaEA, the substitute model ensemble consists of ShuffleNet-V2-2$\times$, ResNet-V2-50, and VGG-19. For all defense methods, these substitute models have never been encountered. The detailed evaluation results on Mini-ImageNet are shown in Table IV. It is evident that when confronted with ensemble attacks involving multiple gradients or models, the ProbSched-PTM-based purification defense continues to exhibit the highest natural accuracy and defensive performance compared to other methods.

D. Transferability of Defensive Capability

We evaluated the transferability of different methods' defensive capabilities, i.e., their defensive effects on other target models with different structures. The adversarial attack methods include FGSM, BIM, UPGD, VMIM, and VNIM. The adversarial perturbation budget is $L_\infty = 8/255$, and the step count is set to 50 for iterative methods. The selected unseen substitute models are ShuffleNet-V2-2$\times$ and ResNet-V2-50. We only choose to compare HGD and LDT with the ProbSched-PTM-based purification defense because their defense and identification model can be separated. Therefore, when the original identification model is replaced with new target models VGG-19 and ViT-S/16, they can still function properly. The detailed evaluation results on CIFAR-10 and Mini-ImageNet are presented in Table V.

From Table V, it can be observed that the natural accuracy of the ProbSched-PTM-based purification defense closely aligns with that of the target model. This suggests that the proposed method rarely leads to misclassification of input examples. When compared to HGD and LDT, the ProbSched-PTM-based purification defense exhibits stronger robustness and achieves the best defensive performance under each type of attack, whether assisting the CNN model VGG-19 or the transformer model ViT-S/16.

V. Conclusion

In this paper, our approach is to deploy a purification model outside the target model to eliminate the adversarial perturbations from adversarial examples. To enhance the generalization of the proposed method, we propose the ProbSched-PTM-based purification defense, which utilizes a CAU-Net as the purification model, training it to eliminate adversarial perturbations by using adversarial examples generated from pre-trained models. Meanwhile, ProbSched-PTM integrates the model difference probability and negative momentum probability to dynamically schedule the pre-trained models, maximizing the transferability of the generated adversarial examples. The evaluation results demonstrate that the ProbSched-PTM-based purification defense can effectively defend against various types of adversarial attacks in a black-box environment.

TABLE V

Classification accuracy rates (%) of different methods for other target models in defending against unseen types of attacks on CIFAR-10 and Mini-ImageNet (*higher is better*). ShuffleNet-V2-2× and ResNet-V2-50 serve as substitute models for generating adversarial examples, VGG-19 and ViT-S/16 serve as other target models not included in the training framework. For each attack, we show the most successful defense with bold.

Target Model	Dataset	Defenses	Clean	ShuffleNet-V2-2×					ResNet-V2-50				
				FGSM	BIM	UPGD	VMIM	VNIM	FGSM	BIM	UPGD	VMIM	VNIM
VGG-19	CIFAR-10	Nat. Training	**94.69**	51.11	18.55	13.14	5.67	5.96	48.31	22.41	15.90	7.54	7.21
		HGD	90.46	89.75	90.19	89.79	90.06	90.22	92.05	91.35	91.13	91.53	91.64
		LDT	93.03	91.32	90.67	90.56	90.77	90.74	92.21	91.69	91.87	92.56	92.69
		ProbSched-PTM	93.53	**92.69**	**92.27**	**92.82**	**91.93**	**92.59**	**92.51**	**92.64**	**92.72**	**92.68**	**93.08**
	Mini-ImageNet	Nat. Training	**70.33**	21.96	11.10	8.41	4.02	3.65	22.49	29.37	24.43	12.56	10.63
		HGD	62.83	57.73	58.88	56.36	57.37	58.09	61.93	61.00	58.83	60.92	61.32
		LDT	65.82	63.97	59.64	58.25	59.58	59.88	66.71	62.91	63.32	64.93	65.11
		ProbSched-PTM	67.31	**64.35**	**62.81**	**61.89**	**62.52**	**62.62**	**67.82**	**64.23**	**64.92**	**66.89**	**66.74**
ViT-S/16	CIFAR-10	Nat. Training	**98.67**	85.30	82.77	78.26	67.31	68.07	88.01	85.03	81.21	73.14	73.58
		HGD	95.20	94.07	93.76	93.52	93.48	93.43	95.14	94.30	94.15	94.71	94.32
		LDT	97.22	96.04	95.60	95.41	95.42	95.53	96.89	95.80	95.79	96.04	96.24
		ProbSched-PTM	97.65	**96.81**	**96.03**	**95.89**	**96.12**	**96.23**	**97.28**	**96.57**	**96.38**	**96.57**	**96.30**
	Mini-ImageNet	Nat. Training	**90.53**	68.77	69.18	63.00	52.30	50.86	73.45	76.28	71.03	59.88	60.38
		HGD	84.36	80.04	79.74	78.03	79.57	79.89	82.16	81.55	80.11	81.06	81.43
		LDT	87.19	84.97	83.32	82.42	83.39	83.68	86.80	84.33	84.18	85.30	85.58
		ProbSched-PTM	88.62	**86.12**	**85.23**	**84.81**	**85.54**	**85.63**	**87.32**	**86.20**	**86.27**	**87.23**	**87.12**

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CVPR. pp. 770–778, 2016.

[2] J. Guo, K. Han, Y. Wang, H. Wu, X. Chen, C. Xu, and C. Xu, "Distilling Object Detectors via Decoupled Features," CVPR. pp. 2154-2164, 2021.

[3] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, and H. Zhang, "A comparative study of real-time semantic segmentation for autonomous driving," CVPR. pp. 700–70010, 2018.

[4] I.J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," ICLR. 2015.

[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," ICLR. 2018.

[6] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Threat of adversarial attacks on deep learning in computer vision: Survey II," CoRR abs/2108.00401. 2021.

[7] X. Wang, and K. He, "Enhancing the transferability of adversarial attacks through variance tuning," CVPR. pp. 1924–1933, 2021.

[8] A. Kurakin, I.J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," ICLR. 2017.

[9] F. Croce, and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," ICML. pp. 2206–2216, 2020.

[10] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A.L. Yuille, "Improving transferability of adversarial examples with input diversity," CVPR. pp. 2725–2734, 2019.

[11] J. Lin, C. Song, K. He, L. Wang, and J.E. Hopcroft, "Nesterov accelerated gradient and scale invariance for adversarial attacks," ICLR. 2020.

[12] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," CVPR. pp. 1778–1787, 2018.

[13] C. Xie, Y. Wu, L.v.d. Maaten, A.L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," CVPR. pp. 501–509, 2019.

[14] Y. Bahat, M. Irani, and G. Shakhnarovich, "Natural and adversarial error detection using invariance to image transformations," CoRR abs/1902.00236, 2019.

[15] Z. Liu, Q. Liu, T. Liu, N. Xu, X. Lin, Y. Wang, and W. Wen, "Feature distillation: Dnn-oriented JPEG compression against adversarial examples," CVPR. pp. 860–868, 2019.

[16] T. Pang, K. Xu, and J.Zhu, "Mixup inference: Better exploiting mixup to defend adversarial attacks," ICLR. 2020.

[17] Z. Wei, J. Chen, M. Goldblum, Z. Wu, T. Goldstein, and Y. Jiang, "Towards Transferable Adversarial Attacks on Vision Transformers," AAAI. pp. 2668-2676, 2022.

[18] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," CVPR. pp. 9185-9193, 2018.

[19] M. Gubri, M. Cordy, M. Papadakis, Y.L. Traon, and K. Sen, "LGV: boosting adversarial example transferability from large geometric vicinity," ECCV. pp. 603-618, 2022.

[20] Y. Huang, and A.W.Kong, "Transferable adversarial attack based on integrated gradients," ICLR. 2022.

[21] B. Chen, J. Yin, S. Chen, B. Chen, and X. Liu, "An adaptive model ensemble adversarial attack for boosting adversarial transferability," ICCV. pp. 4466-4475, 2023.

[22] C. Guo, M. Rana, M. Cissé, and L. van der Maaten, "Countering adversarial images using input transformations," ICLR. 2018.

[23] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J.A. Storer, "Deflecting adversarial attacks with pixel deflection," CVPR. pp. 8571-8580, 2018.

[24] J. Li, S. Zhang, J. Cao, and M. Tan, "Learning defense transformations for counterattacking adversarial examples," Neural Netw. vol.164, pp. 177-185, 2023.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," CVPR. pp. 1-9, 2015.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," CVPR. pp. 4510-4520, 2018.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," MICCAI. pp. 234-241, 2015.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," CVPR. pp. 7132–7141, 2018.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K.Q. Weinberger, "Densely connected convolutional networks," CVPR. pp. 2261–2269, 2017.

[30] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," NeurIPS. pp. 4467–4475, 2017.

[31] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," CVPR. pp. 6307–6315, 2017.

[32] I. Radosavovic, R.P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," CVPR. pp. 10425-10433, 2020.

[33] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," CVPR. pp. 5987-5995, 2017.

[34] S. Zagoruyko, and N. Komodakis, "Wide residual networks," BMVC.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," ECCV. pp. 630-645, 2016.

[36] N. Ma, X. Zhang, H. Zheng, and J. Sun, "Shufflenet V2: practical guidelines for efficient CNN architecture design," ECCV. 122-138, 2018.

[37] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," ICLR. 2015.

[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," ICLR. 2021.