MAP: Masked Adversarial Perturbation for Boosting Black-box Attack Transferability

Kaige Li, Maoxian Wan, Qichuan Geng, Weimin Shi, Xiaochun Cao, *Senior Member, IEEE,* and Zhong Zhou

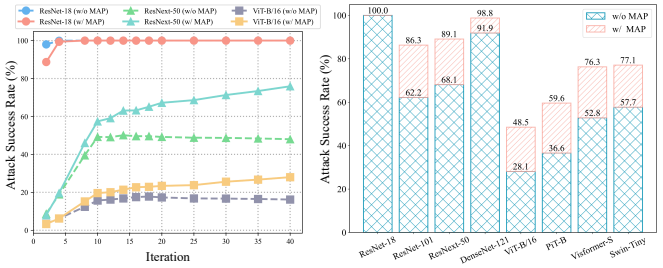
***Abstract*—The transferability of adversarial examples is vital for black-box attacks, as it enables the adversary to deceive the target model without knowing its internals. Despite numerous methods focusing on transferability, they still struggle with transferring across models with distinct architectural components (e.g., CNNs and ViTs). In this work, we argue that the limited adversarial perturbation diversity leads to overfitting of the surrogate model, which acts as a key factor in reducing transferability. To this end, we propose a Masked Adversarial Perturbation (MAP) method to boost adversarial transferability across various architectures from a novel perspective of diversifying perturbation. Specifically, MAP randomly masks perturbation patches during iterations and compels the remaining ones to retain the attack effect, which diversifies perturbations to mitigate their overfitting to the surrogate model. Naturally, MAP spreads perturbation over local patches to alleviate their co-adaptation and prevent perturbations from overly relying on specific patterns. Consequently, it can deceive convolution operation and self-attention mechanism indiscriminately by attacking their basic input units, i.e., a single patch, showing superior transferability over previous methods. Extensive experiments illustrate that MAP consistently and significantly boosts diverse black-box attacks to achieve state-of-the-art performance.**

***Index Terms*—Adversarial examples, Black-box attack, Adversarial transferability, Masked Perturbation.**

I. Introduction

DEEP neural networks (DNNs) [1–3] are highly vulnerable to adversarial examples (AEs), where small, well-designed perturbations can cause their incorrect predictions [4, 5]. This raises concerns about their reliability in real-world applications and highlights the need for effective attacks to expose these vulnerabilities. More importantly, AEs show some transferability, i.e., AEs crafted for a surrogate model remain adversarial for others [6]. Such transferability enables black-box attacks for real-world applications without accessing the target model, thus posing a greater security issue. However, existing attacks [7, 8] demonstrate superior white-box attack performance but relatively poor transferability.

Recently, numerous black-box methods have emerged to boost adversarial transferability, including gradient-based methods [5, 8, 9, 12], input transformations [10, 13–15], advanced objective functions [16, 17], model-related [18, 19] and ensemble-based attacks [20, 21]. Despite their progress, there is still a large performance gap compared to white-box attacks [4, 7, 22] with access to the knowledge of the target model. We attribute this to insufficient perturbation diversity, which can lead to their overfitting to the surrogate model and limit their transferability. For example, while attack performance improves on the surrogate model, it declines on the target models as in Fig. 1(a). Especially, they also fail in transferring across models with huge architecture differences (e.g., CNNs and ViTs) as in Fig. 1(b). To address the above issues, we propose enhancing black-box attack transferability with more diverse adversarial perturbations. Although previous attacks [6, 10, 14, 15] have employed input transformations to diversify the input to mitigate overfitting, they still fail to reach the full potential of input diversity as they perform transformations at the input level rather than perturbation.



(a) MI-FGSM Attack using ResNet-18 (b) DIM Attack using ResNet-18

Fig. 1. Comparison of attack performance on different model architectures. (a) Previous attacks (e.g., MI-FGSM [9] and DIM [10]) suffer from overfitting to the surrogate model. With iteration, the performance on the surrogate model, ResNet-18 [2], improves significantly, while decreases on the target model (e.g., ViT-B/16 [11]). (b) They also struggle with huge architecture differences, where adversarial examples crafted for CNNs often exhibit poor transferability to ViTs. By contrast, the proposed method, MAP, can effectively alleviate overfitting and markedly boost the attack success rate.

Therefore, we introduce a novel strategy to explicitly encourage perturbation diversity. Specifically, we propose Masked Adversarial Perturbation (MAP), a plugin for black-box attacks (Fig. 2(a)), to boost adversarial transferability across architectures. While masking techniques have been adopted in some works [6, 23], where they improve transferability by masking AEs, the introduced masks, especially hard masks [24], directly modify the image content, bringing *serious statistical shifts* (between training and testing) and *unexpected gradient feedback* (during training), leading to their lackluster performance. By contrast, we take a novel perspective of masking adversarial perturbation instead of masking AEs, as illustrated in Fig. 2(b). Despite being a seemingly minor change, we ensure a more consistent and stable training and testing process, mitigating the above issues. Specifically, MAP masks out a random selection of perturbation patches and constrains the remaining patches to maintain the attack effectiveness. On the one hand, this acts like Dropout [25] on perturbation patches, which reduces co-adaptation between them to prevent perturbations from overly

The first two authors contributed equally to this work.

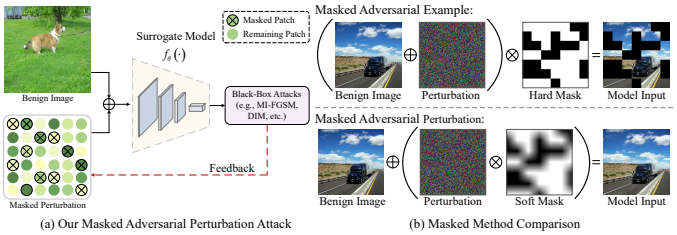


Fig. 2. Mask-based adversarial attack methods. (a) Our MAP explicitly diversifies adversarial perturbations by randomly masking perturbations during iterations to mitigate overfitting and boost transferability. (b) Comparison of previous masked adversarial attacks and our proposed method, highlighting the differences in masking adversarial examples versus perturbations.

relying on specific patterns. On the other hand, over training iterations, different perturbation patches are masked out to generate diverse perturbation patterns, alleviating overfitting between the surrogate model and perturbations. Naturally, MAP spreads out perturbations over local patches, and promotes the attack intensity of each patch, which fools convolution and self-attention mechanism by attacking their basic input units, i.e., a single patch, thus showing better black-box attack transferability over prior methods.

Further, vanilla mask generation methods [6, 24] use hard masks with a fixed mask ratio, leading to two issues. One is hard masks lead to statistical shifts [26] and zero gradient [27] of the perturbation, hindering its effective learning. Another is that using fixed mask ratios poses a trade-off dilemma: a lower one improves optimization stability but fail to introduce sufficient perturbation diversity, while a higher one reduces overfitting but destabilizes training. To this end, we design a novel mask generation method that combines Soft Mask Generation (SMG) and Curriculum Mask Learning (CML). SMG generates soft, continuous-valued masks that preserve partial responses in masked regions, mitigating gradient vanishing. Meanwhile, CML progressively increases the mask ratio from a low initial value during training, which enhances diversity and ensures stable perturbation learning by gradually increasing task complexity. In the MAP, SMG and CML work jointly to further enhance black-box transferability.

To our knowledge, MAP is the first method to exploit masked adversarial perturbation to boost adversarial transferability. Due to its simple and universal concept, MAP can be straightforwardly integrated into various black-box attacks and consistently improve their performance, as shown in Fig. 1. In summary, the main contributions are summarized as follows:

1) We propose to mask out a random selection of perturbation patches to increase perturbation diversity for better adversarial transferability, which sheds new light on how to craft more transferable adversarial examples.
2) We design a novel mask generation method that introduces soft masks with progressively increasing mask ratios, alleviating statistical shift and zero gradient issues caused by hard masks with a fixed mask ratio.
3) Extensive experiments on the ImageNet dataset illustrate that our method is generally compatible with the state-of-the-art black-box attacks and consistently boosts their transferability to both CNNs and ViTs.

II. Related Works

In this section, we provide a concise review of adversarial attacks and adversarial defenses.

A. Adversarial Attacks

Since Szegedy et al. [28] identify the existence of adversarial examples, numerous black-box attacks [29–32] have emerged to discover the vulnerability of DNNs. Existing methods can be classified into query-based [31–33] and transfer-based attacks [14–16, 34]. Among them, the latter does not require any information about the target model, making it more practical for real-world scenes. Transfer-based black-box attack methods mainly include:

Gradient-based methods. MI-FGSM [9] incorporate momentum into iterations to stabilize update directions and escape from poor local optima. Variance tuning [12] uses gradient variance of previous iteration to tune the current gradient in MI-FGSM [9] to boost transferability. Differently, GRA [5] proposes two gradient relevance frameworks to exploit the information in the neighborhood to adaptively correct the update direction.

Input transformation-based attacks. DIM [10] applies random resizing and padding to inputs to alleviate overfitting to white-box models. TIM [22] adopts an ensemble of translated inputs to avoid overfitting. Spectrum Simulation Attack (SSA) [13] apply a spectrum transformation to diversify input image, thus generating more transferable adversarial examples. Block Shuffle and Rotation (BSR) [14] splits the image into multiple blocks, then randomly shuffles and rotates these blocks to craft a set of diverse images for gradient calculation. Structure Invariant Attack (SIA) [15] applies random image transformations to each image block to generate a variety of images for gradient calculation. Wang et al. [35] propose to select and constrain adversarial optimization in a subset of frequency components that are more critical to model prediction. MaskBlock [23] repeatedly masks a patch of adversarial images to generate multiple masked images for collaboratively crafting more transferable adversarial examples. Learnable Patch-wise Mask (LPM) [6] proposes to drop out selected patches of adversarial images to prune the model-specific regions during perturbation generation, thus avoiding overfitting the surrogate model. Unlike the above methods [6, 35], MAP introduces a random perturbation masking strategy, eliminating the need to optimize specific frequency components or image masks based on surrogate models. This randomness effectively avoids overfitting of perturbations to specific models or image regions. In particular, MAP significantly alleviates statistical shifts and unexpected gradient feedback by masking perturbations rather than images, further improving cross-model transferability and versatility.

Advanced objective functions. While many attacks use cross-entropy (CE) loss as the cost function, some works find that regularization terms are conducive to transferability. For example, Transferable Adversarial Perturbation (TAP) [16] introduces two regularization terms to alleviate gradient vanishing and remove the high-frequency perturbations to promote

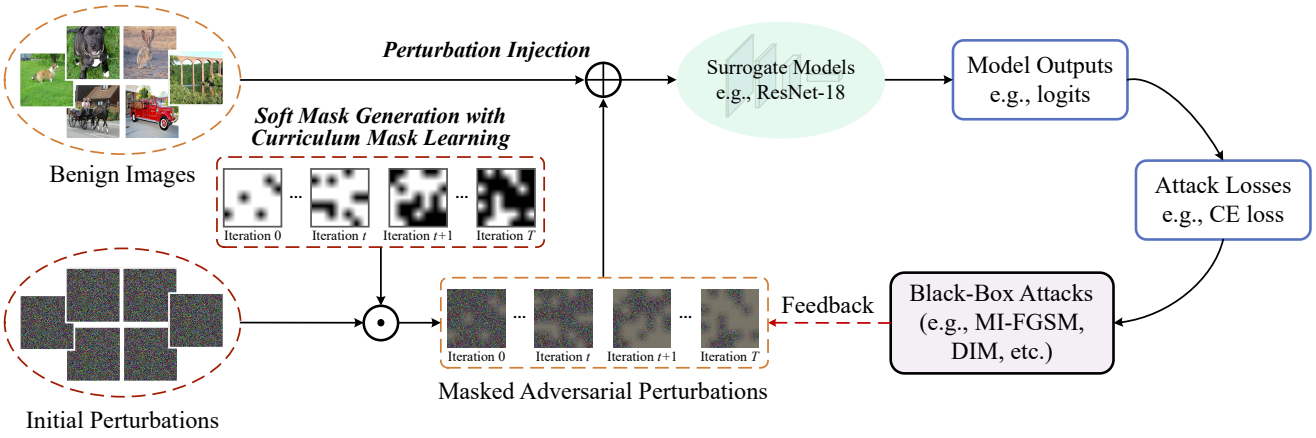


Fig. 3. Black-box attack with the proposed Masked Adversarial Perturbation (MAP). During iterations, we generate diverse soft masks with different mask ratios according to iterations t, which masks several patches in the adversarial perturbation to reduce its complexity and improve its diversity.

transferability. Random Patch Attack (RPA) [17] obtains critical features of objects by random patch transformations, thus adjusting the weight of feature loss to improve transferability. In contrast, we improve transferability by masking adversarial perturbations to explicitly increase their diversity.

Model-related attacks. Some attacks adjust the surrogate model's architecture to promote transferability. BPA [18] recovers the truncated gradient of non-linear layers to boost transferability. Self-Attention Patches Restructure (SAPR) [29] randomly permutes input tokens at each attention layer in ViTs to improve transferability. Zhang et al. [36] recomposes the ViTs to add Virtual Dense Connections (VDC) to back-propagate deeper gradients, enhancing AEs' transferability.

Ensemble-based attacks. Liu et al. [20] found that averaging the predictions of multiple models to get an ensemble loss resulted in more transferable samples. Recently, Chen et al. [21] propose adaptively adjusting the weight of the outputs from each model to further improve transferability. Further, SMER [30] adopts reinforcement learning to reweigh ensemble models to optimize adversarial transferability. Compared with other methods, ensemble-based attacks show promising performance. However, finding multiple proper models for same task is tricky, and training on multiple models is time and resource-intensive, making them prohibitive in some cases.

As failing to encourage perturbation diversity, existing methods still struggle with transferring across models with huge differences, even for input transformation-based attacks (e.g., SIA [15] and BSR [14]). In this work, we propose a novel plug-in method that boosts black-box attacks by explicitly diversifying perturbation, achieving highly transferable attacks.

B. Adversarial Defenses

Adversarial defenses have been extensively studied to mitigate the risks posed by adversarial attacks. Adversarial Training (AT) [37] is one of the most effective methods by injecting AEs into training to enhance model robustness. Another prominent way involves preprocessing inputs before feeding them into the target model. For instance, High-level representation Guided Denoiser (HGD) [38], integrates multiple networks (e.g., Inception-ResNetV2 [3], InceptionV3 [1]), each with a denoising component based on U-Net to eliminate the adversarial perturbation, to make predictions. Neural Representation Purifier (NRP) [39] trains a neural representation purifier by a self-supervised adversarial training mechanism to purify the input sample. DiffPure [40] leverages diffusion models for adversarial purification by diffusing adversarial examples with noise and recovering clean images through a reverse generative process. R&P [41] uses random resizing and random padding to defend adversarial threat. Aside from the above methods, certified defenses provide provable robustness guarantees within a given radius. For example, RS [42] trains a robust ImageNet classifier (i.e., ResNet-50 [2]) with a tight robustness guarantee to defense adversarial examples.

III. Methodology

In this section, we first introduce the preliminaries. Then we detail our MAP with two proposed mask strategies. Finally, we provide theoretical analysis about different methods. The black-box attack process with MAP is illustrated in Fig. 3.

A. Preliminaries

Given a benign image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$ with its label $\boldsymbol{y}$, let f_ϕ be a classification model. Adversarial attack aims to find a small perturbation $\boldsymbol{\delta}$ to generate the adversarial example $\boldsymbol{x}^{adv} = \boldsymbol{x} + \boldsymbol{\delta}$, which is indistinguishable from $\boldsymbol{x}$ (i.e., $\|\boldsymbol{\delta}\|_p \leq \epsilon$) but can fool the network, i.e., $f_\phi\left(\boldsymbol{x}^{adv}\right) \neq \boldsymbol{y}$. Here ϵ is the maximum perturbation, and $\|\cdot\|_p$ is the ℓ_p norm distance. We use ℓ_∞ to align with previous methods [9, 10, 15]. The generation of the adversarial examples can be formalized as:

$$\boldsymbol{x}^{adv} = \underset{\|\boldsymbol{\delta}\|_\infty \leq \epsilon}{\arg\max}\, J\left(f_\phi\left(\boldsymbol{x}+\boldsymbol{\delta}\right), \boldsymbol{y}\right), \tag{1}$$

where $J(\cdot,\cdot)$ is the loss function of f_ϕ (e.g., cross-entropy loss). However, under the black-box setting, it is impractical to directly optimize Eq. 1 via the target model f_ϕ as it is inaccessible. To address this issue, a common practice is to craft adversarial examples via an accessible surrogate model f_θ and rely on the transferability to deceive the target model. Taking I-FGSM [7] as an example, it generates small

perturbations $\boldsymbol{\delta}_{t+1}$ in the direction of the gradient sign to craft an adversarial example at iteration $t+1$:

$$\boldsymbol{\delta}_{t+1} = \text{Clip}\left(\boldsymbol{\delta}_t + \alpha \cdot \text{sign}\left(\nabla_{\boldsymbol{\delta}_t} J\left(f_\theta\left(\boldsymbol{x}+\boldsymbol{\delta}_t\right), \boldsymbol{y}\right)\right), -\epsilon, +\epsilon\right), \tag{2}$$

where $\text{Clip}(\cdot)$ restricts perturbation into the range $[-\epsilon, \epsilon]$, α denotes step size.

B. Masked Adversarial Perturbation

Due to limited perturbation diversity, existing transfer-based attacks [10, 43] tend to overfit the surrogate model and exhibit poor transferability. To this end, we propose a Masked Adversarial Perturbation (MAP) method to explicitly diversify perturbations. MAP randomly masks out perturbation patches at each iteration to construct more diverse perturbation patterns, thus preventing it from overfitting to a specific model. From another perspective, MAP spreads perturbations across various local patches, alleviating excessive co-adapting between patches to prevent the adversarial nature of perturbations from overly relying on specific patterns.

Specifically, MAP withholds local perturbations by randomly masking out patches of the entire adversarial perturbation at each iteration. For that purpose, a patch mask $\mathcal{M}$ is randomly sampled from a uniform distribution:

$$\mathcal{M}_{\substack{mb+1:(m+1)b \\ nb+1:(n+1)b}} = [v > r] \;\; with \; v \in \mathcal{U}(0,1), \tag{3}$$

where b denotes the patch size, $m \in [0, H/b-1]$ and $n \in [0, W/b-1]$ index the patch, $[\cdot]$ denotes the Iverson bracket, r is the mask ratio, $\mathcal{U}$ is the uniform distribution. The masked adversarial perturbation $\boldsymbol{\delta}^M$ is obtained by element-wise multiplication between the mask and perturbation:

$$\boldsymbol{\delta}^M = \mathcal{M} \odot \boldsymbol{\delta}. \tag{4}$$

Thus, MAP replaces Eq. 1 with:

$$\boldsymbol{x}^{adv} = \underset{\|\boldsymbol{\delta}\|_\infty \leq \epsilon}{\arg\max}\, J\left(f_\theta\left(\boldsymbol{x}+\boldsymbol{\delta}^M\right), \boldsymbol{y}\right). \tag{5}$$

The masked adversarial attack only utilizes limited information of the unmasked regions, as in Fig. 3. In this way, MAP spreads perturbation over local patches to increase their attack intensity. Meanwhile, as MAP masks various patches during training, it promotes perturbation diversity explicitly to prevent it from overfitting the surrogate model.

C. Soft Mask Generation

Hard masks generated by Eq. 3 will lead to statistical shifts [26], where the mean and variance of the perturbation distributions differs between the training and testing phases, potentially leading to suboptimal attack results. In addition, hard masks will also cause issues with zero gradients and patch death [27], which increases the risk of overreliance on specific patches. To this end, we further propose a Soft Mask Generation (SMG) strategy to mitigate the above issues by using smoother transitions on the mask boundaries.

To generate soft mask $\mathcal{M}^{soft}$, we first generate a low-resolution binary matrix $\mathcal{B} \in \mathbb{R}^{\frac{H}{b} \times \frac{W}{b}}$:

$$\mathcal{B}_{m,n} = [v > r] \;\; with \; v \in \mathcal{U}(0,1), \tag{6}$$

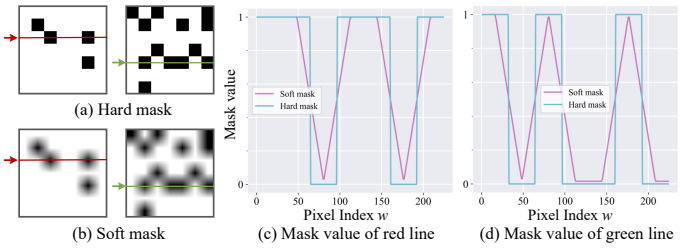


Fig. 4. Comparisons between hard and soft masks. We visualize the mask values of the red and green lines in both masks respectively.

and then use bilinear interpolation to upsample $\mathcal{B}$ to the same size as adversarial perturbation $\boldsymbol{\delta}$ to obtain soft mask:

$$\mathcal{M}^{soft}_{h,w} = \sum_{m}^{\frac{H}{b}} \sum_{n}^{\frac{W}{b}} \mathcal{B}_{m,n} \cdot max(0, 1-|h/b-m|) \cdot max(0, 1-|w/b-n|), \tag{7}$$

where $h \in [0, \ldots, H-1]$, $w \in [0, \ldots, W-1]$. As in Fig. 4, hard masks have obvious step phenomena at the junction of the masked and unmasked regions, whereas soft masks allow for a smoother transition. That is, soft masks preserve a certain degree of response in the masked regions, which relieves zero gradient and statistical shift issues, helping to boost the generalization of the perturbation and training stability.

D. Curriculum Mask Learning

By randomly masking adversarial perturbations, MAP dramatically improves attack transferability. However, finding a suitable mask ratio is non-trivial. A lower mask ratio facilitates early convergence of perturbation learning, but limits its generalization to unseen models. By contrast, a higher one reduces overfitting risk, but makes it harder to capture basic perturbation patterns, leading to lower learning efficiency and unstable updates. To this end, we propose a Curriculum Mask Learning (CML) strategy that gradually increases the mask ratio during training. This balances diversity and stability to guide perturbations from simple patterns toward more generalizable ones, thereby enhancing transferability.

Specifically, instead of manually specifying a fixed value to mask ratio r, we propose to dynamically set r based on the number of attack iterations. The intuition behind the dynamic schedule is to start with an easier learning task (low mask ratio) in early iterations to stabilize optimization and accelerate convergence. As training progresses, the increasing mask ratio introduces greater task complexity to create more general perturbation patterns that improve transferability and adversarial robustness by reducing reliance on specific patterns. We list some instances of dynamic schedules:

$$\begin{aligned} r(t) &= r_s + \frac{r_e - r_s}{T} \times t, \\ r(t) &= r_s + (r_e - r_s) \times \left(2^{t/T} - 1\right), \\ r(t) &= r_s + (r_e - r_s) \times (t/T)^{1.2}, \end{aligned} \tag{8}$$

where r_s and r_e denote the starting and ending mask ratio, respectively, t is the current iteration and T is the total number

of iterations. Experiments show that all instances are equally effective. In this paper, we mainly adopt the first simple linear schedule. Fig. 5 illustrates the mask evolution during training, where CML generates varying mask patterns across iterations, promoting more diverse perturbation inputs.

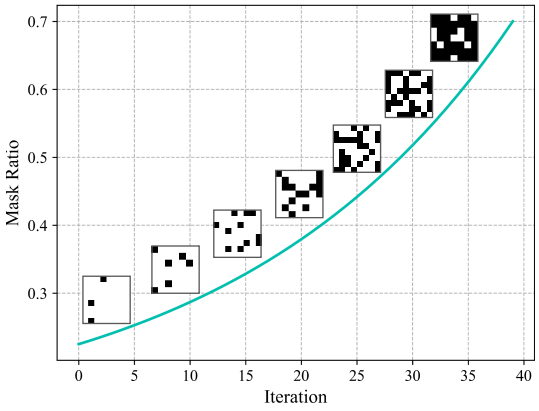


Fig. 5. Illustration of the Curriculum Mask Learning strategy.

Our MAP is widely applicable to various existing transfer-based black-box attacks. Here we integrate MAP into MI-FGSM [9] and summarize the algorithm in Algorithm 1. Furthermore, MAP introduces negligible overhead during training, as its masking step involves only simple tensor operations, and it incurs no extra cost during testing, as the generated perturbations can be directly applied without further processing.

Algorithm 1 MAP algorithm.

Input: A model f_θ with loss function J; A benign example $\boldsymbol{x}$ with label $\boldsymbol{y}$; initial perturbation $\boldsymbol{\delta}_0$; step size α; maximum perturbation ϵ; iteration number T; decay factor μ; patch size b; starting mask ratio r_s and ending mask ratio r_e.

Output: Adversarial example $\boldsymbol{x}^{adv}$.

$\boldsymbol{x}_0^{adv} = \boldsymbol{x} + \boldsymbol{\delta}_0,\ \boldsymbol{g}_0 = 0$

for $t = 0$ to $T-1$ **do**

Calculate the mask ratio $r(t) = r_s + \frac{r_e - r_s}{T} \times t$ based on Eq. 8 defined by CML

Generate soft mask $\mathcal{M}^{soft}$ based on Eq. 7 and $r(t)$

$\boldsymbol{x}_t^{adv} = \boldsymbol{x} + \mathcal{M}^{soft} \odot \boldsymbol{\delta}_t$

Input $\boldsymbol{x}_t^{adv}$ to f_θ and get the gradient $\nabla_{\boldsymbol{\delta}} J\left(\boldsymbol{x}_t^{adv}, \boldsymbol{y}\right)$

Update the momentum:

$$\boldsymbol{g}_{t+1} = \mu \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{\delta}} J\left(\boldsymbol{x}_t^{adv}, \boldsymbol{y}\right)}{\left\|\nabla_{\boldsymbol{\delta}} J\left(\boldsymbol{x}_t^{adv}, \boldsymbol{y}\right)\right\|_1} \tag{9}$$

Update perturbation $\boldsymbol{\delta}$ by applying the gradient sign:

$$\boldsymbol{\delta}_{t+1} = \text{Clip}\left(\boldsymbol{\delta}_t + \alpha \cdot \text{sign}\left(\boldsymbol{g}_{t+1}\right), -\epsilon, +\epsilon\right) \tag{10}$$

end for

$\boldsymbol{x}^{adv} = \boldsymbol{x} + \boldsymbol{\delta}_T$

return $\boldsymbol{x}^{adv}$

E. Theoretical Analysis

Previous methods (e.g., LPM [6]) propose the Masked Adversarial Examples (MAE) strategy, i.e., $\boldsymbol{x}^{adv} = (\boldsymbol{x} + \boldsymbol{\delta}) \times \mathcal{M}$, leading to significant statistical shifts between training and testing and unexpected gradient feedback during optimization. In contrast, MAP masks adversarial perturbations instead, i.e., $\boldsymbol{x}^{adv} = \boldsymbol{x} + (\boldsymbol{\delta} \times \mathcal{M})$, aiming to reduce these shifts and stabilize gradient feedback, causing more transferable AEs. To support this claim, we theoretically compare the statistical shift and gradient feedback of both masking strategies.

Statistical Shift Analysis. We derive the mean shift $\Delta\mathcal{E}$ and variance shift $\Delta\mathcal{D}$ for both methods as follows:

$$\begin{aligned}
\Delta\mathcal{E}_{MAE} &= \left|\mathbb{E}[(\boldsymbol{x} + \boldsymbol{\delta}_1) \times \mathcal{M}] - \mathbb{E}[\boldsymbol{x} + \boldsymbol{\delta}_1]\right| \\
&= \left|\mathbb{E}\left[\mathcal{M}\boldsymbol{x}\right] + \mathbb{E}\left[\mathcal{M}\boldsymbol{\delta}_1\right] - \mathbb{E}\left[\boldsymbol{x}\right] - \mathbb{E}\left[\boldsymbol{\delta}_1\right]\right|, \\
\Delta\mathcal{E}_{MAP} &= \left|\mathbb{E}[\boldsymbol{x} + (\mathcal{M} \times \boldsymbol{\delta}_2)] - \mathbb{E}[\boldsymbol{x} + \boldsymbol{\delta}_2]\right| \\
&= \left|\mathbb{E}\left[\boldsymbol{x}\right] + \mathbb{E}\left[\mathcal{M}\boldsymbol{\delta}_2\right] - \mathbb{E}\left[\boldsymbol{x}\right] - \mathbb{E}\left[\boldsymbol{\delta}_2\right]\right|, \\
\Delta\mathcal{D}_{MAE} &= \left|\mathbb{D}[(\boldsymbol{x} + \boldsymbol{\delta}_1) \times \mathcal{M}] - \mathbb{D}[\boldsymbol{x} + \boldsymbol{\delta}_1]\right| \\
&= \Big|\mathbb{E}\left[\left(\mathcal{M}\boldsymbol{x} + \mathcal{M}\boldsymbol{\delta}_1\right)^2\right] - \left(\mathbb{E}\left[\mathcal{M}\boldsymbol{x} + \mathcal{M}\boldsymbol{\delta}_1\right]\right)^2 \\
&\quad -\mathbb{E}\left[\left(\boldsymbol{x} + \boldsymbol{\delta}_1\right)^2\right] + \left(\mathbb{E}\left[\boldsymbol{x} + \boldsymbol{\delta}_1\right]\right)^2\Big|, \\
\Delta\mathcal{D}_{MAP} &= \left|\mathbb{D}[\boldsymbol{x} + (\mathcal{M} \times \boldsymbol{\delta}_2)] - \mathbb{D}[\boldsymbol{x} + \boldsymbol{\delta}_2]\right| \\
&= \Big|\mathbb{E}\left[\left(\boldsymbol{x} + \mathcal{M}\boldsymbol{\delta}_2\right)^2\right] - \left(\mathbb{E}\left[\boldsymbol{x} + \mathcal{M}\boldsymbol{\delta}_2\right]\right)^2 \\
&\quad -\mathbb{E}\left[\left(\boldsymbol{x} + \boldsymbol{\delta}_2\right)^2\right] + \left(\mathbb{E}\left[\boldsymbol{x} + \boldsymbol{\delta}_2\right]\right)^2\Big|.
\end{aligned} \tag{11}$$

where $\mathbb{E}[\cdot]$ and $\mathbb{D}[\cdot]$ denotes the expectation and variance function, respectively. In this case, we assume $\mathcal{M}$ is independent of $\boldsymbol{x}$ and $\boldsymbol{\delta}$ after training, then we have:

$$\begin{aligned}
\Delta\mathcal{E}_{MAE} &= \left|\mathbb{E}\left[\mathcal{M}\right]\mathbb{E}\left[\boldsymbol{x}\right] + \mathbb{E}\left[\mathcal{M}\right]\mathbb{E}\left[\boldsymbol{\delta}_1\right] - \mathbb{E}\left[\boldsymbol{x}\right] - \mathbb{E}\left[\boldsymbol{\delta}_1\right]\right|, \\
\Delta\mathcal{E}_{MAP} &= \left|\mathbb{E}\left[\boldsymbol{x}\right] + \mathbb{E}\left[\mathcal{M}\right]\mathbb{E}\left[\boldsymbol{\delta}_2\right] - \mathbb{E}\left[\boldsymbol{x}\right] - \mathbb{E}\left[\boldsymbol{\delta}_2\right]\right|, \\
\Delta\mathcal{D}_{MAE} &= \Big|\mathbb{E}\left[\mathcal{M}^2\right]\mathbb{E}\left[\boldsymbol{x}^2\right] + 2\mathbb{E}\left[\mathcal{M}^2\right]\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_1\right] + \mathbb{E}\left[\mathcal{M}^2\right] \times \\
&\quad \mathbb{E}\left[{\boldsymbol{\delta}_1}^2\right] - \left(\mathbb{E}\left[\mathcal{M}\right]\right)^2\left(\mathbb{E}\left[\boldsymbol{x}\right]\right)^2 - 2\left(\mathbb{E}\left[\mathcal{M}\right]\right)^2\mathbb{E}\left[\boldsymbol{x}\right]\mathbb{E}\left[\boldsymbol{\delta}_1\right] \\
&\quad - \left(\mathbb{E}\left[\mathcal{M}\right]\right)^2\left(\mathbb{E}\left[\boldsymbol{\delta}_1\right]\right)^2 - \Big(\mathbb{E}\left[\boldsymbol{x}^2\right] + 2\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_1\right] + \mathbb{E}\left[\boldsymbol{\delta}_1^2\right] \\
&\quad - \left(\mathbb{E}\left[\boldsymbol{x}\right]\right)^2 - \left(\mathbb{E}\left[\boldsymbol{\delta}_1\right]\right)^2 - 2\mathbb{E}\left[\boldsymbol{x}\right]\mathbb{E}\left[\boldsymbol{\delta}_1\right]\Big)\Big|, \\
\Delta\mathcal{D}_{MAP} &= \Big|\mathbb{E}\left[\boldsymbol{x}^2\right] + \mathbb{E}\left[\mathcal{M}^2\right]\mathbb{E}\left[\boldsymbol{\delta}_2^2\right] + 2\mathbb{E}\left[\mathcal{M}\right]\mathbb{E}\left[\boldsymbol{\delta}_2\boldsymbol{x}\right] - \\
&\quad \Big(\left(\mathbb{E}\left[\boldsymbol{x}\right]\right)^2 + \left(\mathbb{E}\left[\mathcal{M}\right]\right)^2\left(\mathbb{E}\left[\boldsymbol{\delta}_2\right]\right)^2 + 2\mathbb{E}\left[\boldsymbol{x}\right]\mathbb{E}\left[\mathcal{M}\right]\mathbb{E}\left[\boldsymbol{\delta}_2\right]\Big) \\
&\quad - \Big(\mathbb{E}\left[\boldsymbol{x}^2\right] + 2\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_2\right] + \mathbb{E}\left[\boldsymbol{\delta}_2^2\right] - \left(\mathbb{E}\left[\boldsymbol{x}\right]\right)^2 - \\
&\quad 2\mathbb{E}\left[\boldsymbol{x}\right]\mathbb{E}\left[\boldsymbol{\delta}_2\right] - \left(\mathbb{E}\left[\boldsymbol{\delta}_2\right]\right)^2\Big)\Big|.
\end{aligned} \tag{12}$$

Here, we have $\mathbb{E}[Z] = \mu_z$, $\mathbb{D}[Z] = \sigma_z^2$, $\mathbb{D}(Z) = \mathbb{E}\left(Z^2\right) - \left(\mathbb{E}[Z]\right)^2$. For hard mask, we have $\mathcal{M}^2 = \mathcal{M}$, and $\mathbb{E}[\mathcal{M}] = \frac{1}{N}\left(\sum_{i=1}^{K}(m_i = 1) + \sum_{i=K+1}^{N}(m_i = 0)\right) = \frac{K}{N} = 1 - r$, where we denote $1 - r$ as α. Then, Eq. 12 is re-written as:

$$\begin{aligned}
\Delta\mathcal{E}_{MAE} &= \left|\alpha\mu_{\boldsymbol{x}} + \alpha\mu_{\boldsymbol{\delta}_1} - \mu_{\boldsymbol{x}} - \mu_{\boldsymbol{\delta}_1}\right| = (1-\alpha)\left|\mu_{\boldsymbol{x}} + \mu_{\boldsymbol{\delta}_1}\right|, \\
\Delta\mathcal{E}_{MAP} &= \left|\mu_{\boldsymbol{x}} + \alpha\mu_{\boldsymbol{\delta}_2} - \mu_{\boldsymbol{x}} - \mu_{\boldsymbol{\delta}_2}\right| = (1-\alpha)\left|\mu_{\boldsymbol{\delta}_2}\right|, \\
\Delta\mathcal{D}_{MAE} &= \Big|-\alpha^2\left(\mu_{\boldsymbol{x}}^2 + 2\mu_{\boldsymbol{x}}\mu_{\boldsymbol{\delta}_1} + \mu_{\boldsymbol{\delta}_1}^2\right) + \alpha\left(\sigma_{\boldsymbol{x}}^2 + \mu_{\boldsymbol{x}}^2 + \right. \\
&\quad \left. 2\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_1\right] + \sigma_{\boldsymbol{\delta}_1}^2 + \mu_{\boldsymbol{\delta}_1}^2\right) - \left(\sigma_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{\delta}_1}^2 + 2\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_1\right] - \right. \\
&\quad \left. 2\mu_{\boldsymbol{x}}\mu_{\boldsymbol{\delta}_1}\right)\Big|, \\
\Delta\mathcal{D}_{MAP} &= \Big|-\alpha^2\mu_{\boldsymbol{\delta}_2}^2 + \alpha\left(\sigma_{\boldsymbol{\delta}_2}^2 + \mu_{\boldsymbol{\delta}_2}^2\right) + (2\alpha - 2)\left(\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_2\right] - \right. \\
&\quad \left. \mu_{\boldsymbol{x}}\mu_{\boldsymbol{\delta}_2}\right) - \sigma_{\boldsymbol{\delta}_2}^2\Big|.
\end{aligned} \tag{13}$$

From the above derivations, we obtain:

1. **Mean Shift.** It is evident that $\Delta\mathcal{E}_{MAE} > \Delta\mathcal{E}_{MAP}$ since $\mu_{\boldsymbol{x}} \gg \mu_{\boldsymbol{\delta}}$ in general, indicating that MAP exhibits a smaller mean shift compared to MAE.

2. **Variance Shift.** To compare $\Delta\mathcal{D}_{MAE}$ and $\Delta\mathcal{D}_{MAP}$, we proceed in two steps:

(a) Determine the sign intervals of $\Delta\mathcal{D}$. Since $\Delta\mathcal{D}$ is a quadratic function of α, we can get that $\Delta\mathcal{D}_{MAE} \geq 0$ at $[\alpha_1, 1]$, and $\Delta\mathcal{D}_{MAP} \geq 0$ at $[1, \alpha_2]$. According to the properties of the function, the values of α_1, α_2 can be calculated as:

$$\begin{aligned} \alpha_1 &= \frac{\sigma_{\boldsymbol{x}}^2 + \mu_{\boldsymbol{x}}^2 + 2\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_1\right] + \sigma_{\boldsymbol{\delta}_1}^2 + \mu_{\boldsymbol{\delta}_1}^2}{\mu_{\boldsymbol{x}}^2 + 2\mu_{\boldsymbol{x}}\mu_{\boldsymbol{\delta}_1} + \mu_{\boldsymbol{\delta}_1}^2} - 1, \\ \alpha_2 &= \frac{2\left(\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_2\right] - \mu_{\boldsymbol{x}}\mu_{\boldsymbol{\delta}_2}\right) + \sigma_{\boldsymbol{\delta}_2}^2}{\mu_{\boldsymbol{\delta}_2}^2}, \end{aligned} \tag{14}$$

From ImageNet statistics [44], we can obtain the value of $\mu_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{x}}$, and $\mu_{\boldsymbol{x}} \gg \mu_{\boldsymbol{\delta}}$. Meanwhile, $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}\right] - \mu_{\boldsymbol{x}}\mu_{\boldsymbol{\delta}} \approx 0$ as $|\boldsymbol{\delta} - 0| \leq \epsilon$. Hence, we have that $\alpha_1 \approx 0.22$ and $\alpha_2 > 1.0$. That is, when $\alpha \in [\alpha_1, 1] \approx [0.22, 1]$, $\Delta\mathcal{D}_{MAE} \geq 0$, and when $\alpha \in [0, 1]$, $\Delta\mathcal{D}_{MAP} \leq 0$.

(b) Compare $\Delta\mathcal{D}_{MAE}$ and $\Delta\mathcal{D}_{MAP}$. For simplicity, we compare them only at $\alpha \in [0.22, 1]$ since other intervals yield similar conclusions:

$$\begin{aligned} \Delta &= \Delta\mathcal{D}_{MAE} - \Delta\mathcal{D}_{MAP} = -\alpha^2\left(\mu_{\boldsymbol{x}}^2 + 2\mu_{\boldsymbol{x}}\mu_{\boldsymbol{\delta}_1} + \mu_{\boldsymbol{\delta}_1}^2 + \mu_{\boldsymbol{\delta}_2}^2\right) \\ &+ \alpha\left(\sigma_{\boldsymbol{x}}^2 + \mu_{\boldsymbol{x}}^2 + 2\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_1\right] + \mu_{\boldsymbol{\delta}_1}^2 + \sigma_{\boldsymbol{\delta}_1}^2 + \mu_{\boldsymbol{\delta}_2}^2 + \sigma_{\boldsymbol{\delta}_2}^2\right. \\ &\left.+2\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_2\right] - 2\mu_{\boldsymbol{x}}\mu_{\boldsymbol{\delta}_2}\right) - \left(\sigma_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{\delta}_1}^2 + 2\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_1\right] - 2\mu_{\boldsymbol{x}}\mu_{\boldsymbol{\delta}_1}\right) \\ &- 2\left(\mathbb{E}\left[\boldsymbol{x}\boldsymbol{\delta}_2\right] - \mu_{\boldsymbol{x}}\mu_{\boldsymbol{\delta}_2}\right) - \sigma_{\boldsymbol{\delta}_2}^2, \\ &\approx -\alpha^2\mu_{\boldsymbol{x}}^2 + \alpha\left(\sigma_{\boldsymbol{x}}^2 + \mu_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{\delta}_1}^2 + \sigma_{\boldsymbol{\delta}_2}^2\right) - \left(\sigma_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{\delta}_1}^2 + \sigma_{\boldsymbol{\delta}_2}^2\right), \end{aligned} \tag{15}$$

where $\Delta \geq 0$ at $[\alpha_3, 1]$, and $\alpha_3 = \frac{\sigma_{\boldsymbol{x}}^2 + \mu_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{\delta}_1}^2 + \sigma_{\boldsymbol{\delta}_2}^2}{\mu_{\boldsymbol{x}}^2} - 1 \approx 0.26$. That is, when mask ratio $r = 1 - \alpha \in [0, 0.74]$, our method enjoys smaller variance shift, which is better performance as previous methods [6] generally set $r \leq 0.3$ to avoid unexpected gradient feedback, as discussed below. Particularly, our SMG strategy can further reduces the statistical shifts as it preserves some responses in the masked regions, and we provide the derivation in the Supplementary Material.

Gradient Stability Analysis. We now analyze how MAP and MAE influence gradient feedback during the initial training phase. Given that perturbation $\boldsymbol{\delta}$ is generally initialized to 0, we derive the gradients under both methods:

$$\frac{\partial f_\theta\left(\boldsymbol{x} + \mathcal{M}\boldsymbol{\delta}\right)}{\partial\boldsymbol{\delta}} = \frac{\partial f_\theta\left(\boldsymbol{x} + \mathcal{M}\boldsymbol{\delta}\right)}{\partial\left(\boldsymbol{x} + \mathcal{M}\boldsymbol{\delta}\right)}\mathcal{M} \overset{\boldsymbol{\delta}=0}{=} \frac{\partial f_\theta\left(\boldsymbol{x}\right)}{\partial\boldsymbol{x}}\mathcal{M}, \tag{16}$$

$$\frac{\partial f_\theta\left(\left(\boldsymbol{x} + \boldsymbol{\delta}\right)\mathcal{M}\right)}{\partial\boldsymbol{\delta}} = \frac{\partial f_\theta\left(\left(\boldsymbol{x} + \boldsymbol{\delta}\right)\mathcal{M}\right)}{\partial\left(\left(\boldsymbol{x} + \boldsymbol{\delta}\right)\mathcal{M}\right)}\mathcal{M} \overset{\boldsymbol{\delta}=0}{=} \frac{\partial f_\theta\left(\mathcal{M}\boldsymbol{x}\right)}{\partial\left(\mathcal{M}\boldsymbol{x}\right)}\mathcal{M}. \tag{17}$$

As observed, our MAP only masks the gradient of the perturbation, while MAE directly modify the overall gradient, which brings unexpected gradient feedback as the surrogate model typically has not encountered masked images.

Experimental Analysis. Finally, we experimentally verify the above derivation. As in Fig. 6(a) and (b), MAP exhibits smaller mean and variance changes under different mask ratios, with SMG further reducing these changes. In addition, MAE causes large deviations in model output as it directly modify the image content, while MAP does not, as in Fig. 6(c). Output deviation can lead to unexpected behavior of the surrogate model, since the model often has never encountered masked images, causing the model to produce optimization targets and gradient feedbacks that are skewed from what is expected. Hence, when the mask ratio becomes larger, its performance performance drops sharply, as in Fig. 6(d). By contrast, MAP maintains outstanding performance even with higher mask ratios, and SMG can further boost performance, demonstrating the robustness and effectiveness of our method.

In summary, our MAP ensures smaller statistical changes and accurate gradient feedback by masking perturbations instead of adversarial examples, improving the stability and transferability of the attack.

IV. Experiments

A. Experimental Settings

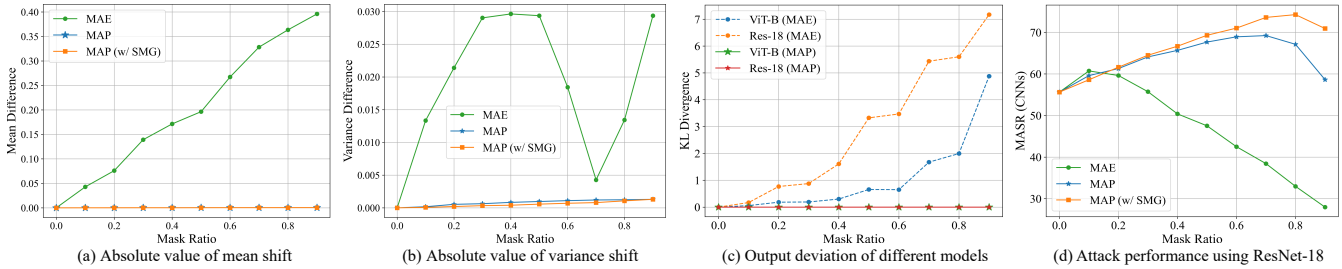
Datasets. Similar to previous methods [9, 12, 14], we randomly sample 1,000 images from the ImageNet validation set [44], where each image is from one category and can be correctly classified by the employed model.

Baseline Attacks. Our MAP can be combined with existing transfer-based black-box attacks. To verify its effectiveness, we choose existing state-of-the-art (SOTA) methods, including gradient-based methods, e.g., MI-FGSM [9], GRA [5], and input transformation-based attacks, e.g., SIA [15], L2T [48], and advanced objective-based, e.g., TAP [16], RPA [17], and model-related attacks, e.g., BPA [18], VDC [36] as our baselines. For fairness, all methods, except gradient-based ones, are integrated into MI-FGSM [9].

Models. We choose Convolutional Neural Networks (CNNs), including ResNet-18 (Res-18) [2], ResNet-101 (Res-101) [2], ResNext-50 (ReX-50) [50], DenseNet-121 (DN-121) [51], and Vision Transformers (ViTs), including ViT-B/16 (ViT-B) [11], PiT-B [52], Visformer-S (Vis-S) [53], Swin-Tiny (Swin-T) [54], and CNN-ViT hybrid models, including MaxViT-T [55], MobileViTv2-2.00 [56], and MLP-Mixers, including MLP-Mixer-B/16 [57], and MambaOut-Tiny [58] to evaluate the attack performance.

Metrics: We use Attack Success Rate (ASR), which denotes the misclassification rates of the target model on the adversarial examples, to evaluate attack methods. We also adopt the Mean Attack Success Rate (MASR) to measure the transferability of an attack method across multiple target models (except for the surrogate model).

Implementation Details. During training, we use randomly masked perturbations to boost adversarial transferability, while during evaluation we use complete perturbations. By default, we set patch size $b = 16$, initial mask ratio $r_s = 0.4$ and ending mask ratio $r_e = 0.9$ for CNNs as the surrogate model, while $r_s = 0.3$ and $r_e = 0.7$ for ViTs as the surrogate model, iteration number $T = 40$. Regarding other hyper-parameters of the baseline methods, for fair comparison, we strictly follow their original settings. For example, for MI-FGSM [9], the maximum perturbation $\epsilon = 16$, decay factor $\mu = 1$.



(a) Absolute value of mean shift (b) Absolute value of variance shift (c) Output deviation of different models (d) Attack performance using ResNet-18

Fig. 6. Comparison of different mask methods. In (a) and (b), we list the absolute shift values of the statistics of the three methods at different mask ratios. In (c), we visualize the prediction deviation of the model for the images before and after masking at the initial phase. In (d), we present the Mean Attack Success Rate (MASR) of the three methods on the CNN models.

TABLE I
ASR (%) OF VARIOUS TRANSFER-BASED ATTACKS AGAINST NORMALLY TRAINED MODELS, AND THEIR ENHANCED VERSION BY OUR METHOD, USING RES-18 [2] AS THE SURROGATE MODEL.

Category	Attacks	CNNs			ViTs				MASR	MASR
		Res-101	ReX-50	DN-121	ViT-B	PiT-B	Vis-S	Swin-T	(CNNs)	(ViTs)
Gradient-based	MI-FGSM [9]	41.6	49.2	78.7	17.3	24.5	36.2	42.3	56.5	30.1
	+Ours	**69.3**	**75.9**	**94.7**	**28.0**	**39.9**	**57.7**	**58.5**	**80.0**$_{23.5\uparrow}$	**46.0**$_{15.9\uparrow}$
	VMI-FGSM [12]	70.1	75.7	94.2	31.8	43.0	60.2	62.0	80.0	49.3
	+Ours	**86.6**	**89.2**	**98.4**	**41.5**	**56.2**	**75.4**	**75.6**	**91.4**$_{11.4\uparrow}$	**62.2**$_{12.9\uparrow}$
	EMI-FGSM [45]	56.6	62.6	90.6	21.7	33.4	47.5	53.0	69.9	38.9
	+Ours	**85.7**	**88.5**	**98.6**	**39.1**	**53.3**	**74.0**	**74.6**	**90.9**$_{21.0\uparrow}$	**60.3**$_{21.4\uparrow}$
	AI-FGTM [46]	34.6	40.5	70.1	12.7	20.1	28.9	34.9	48.4	24.1
	+Ours	**55.2**	**61.1**	**88.4**	**20.8**	**30.6**	**46.7**	**51.5**	**68.2**$_{19.8\uparrow}$	**37.4**$_{13.3\uparrow}$
	GRA [5]	64.6	70.3	94.1	30.5	40.5	55.0	61.1	76.3	46.8
	+Ours	**83.0**	**86.3**	**97.8**	**43.4**	**54.6**	**69.1**	**75.5**	**89.0**$_{12.7\uparrow}$	**60.7**$_{13.9\uparrow}$
	PGN [8]	68.5	72.2	94.3	31.9	42.5	58.6	65.1	78.3	49.5
	+Ours	**72.9**	**77.2**	**94.4**	**38.2**	**49.7**	**63.1**	**70.6**	**81.5**$_{3.2\uparrow}$	**55.4**$_{5.9\uparrow}$
Input transformation-based	TIM [22]	39.6	46.8	78.1	17.0	19.4	30.4	40.0	54.8	26.7
	+Ours	**58.0**	**65.9**	**90.2**	**23.2**	**27.9**	**45.1**	**52.3**	**71.4**$_{16.6\uparrow}$	**37.1**$_{10.4\uparrow}$
	DIM [10]	69.9	74.7	93.4	32.7	40.8	59.3	60.5	79.3	48.3
	+Ours	**86.3**	**89.1**	**98.8**	**48.5**	**59.6**	**76.3**	**77.1**	**91.4**$_{12.1\uparrow}$	**65.4**$_{17.1\uparrow}$
	SSA [13]	69.8	73.6	94.6	31.1	42.0	56.8	63.5	79.3	48.4
	+Ours	**87.6**	**90.0**	**97.7**	**45.1**	**56.7**	**75.5**	**77.6**	**91.8**$_{12.5\uparrow}$	**63.7**$_{15.3\uparrow}$
	DeCowA [47]	91.6	92.3	99.6	59.4	69.5	85.9	82.9	94.5	74.4
	+Ours	**97.4**	**98.3**	**99.9**	**74.9**	**83.0**	**93.0**	**91.7**	**98.5**$_{4.0\uparrow}$	**85.7**$_{11.3\uparrow}$
	BSR [14]	90.2	93.2	99.6	44.2	59.7	80.8	76.5	94.3	65.3
	+Ours	**97.7**	**98.2**	**99.9**	**65.2**	**77.1**	**92.9**	**90.8**	**98.6**$_{4.3\uparrow}$	**81.5**$_{16.2\uparrow}$
	SIA [15]	93.9	95.9	99.6	45.2	60.6	83.7	77.3	96.5	66.7
	+Ours	**98.4**	**99.0**	**100.0**	**65.2**	**78.6**	**94.0**	**91.0**	**99.1**$_{2.6\uparrow}$	**82.2**$_{15.5\uparrow}$
	L2T [48]	93.3	95.7	100.0	58.0	70.8	87.7	85.5	96.3	75.5
	+Ours	**98.2**	**98.7**	**100.0**	**73.7**	**84.2**	**95.0**	**92.7**	**99.0**$_{2.7\uparrow}$	**86.4**$_{10.9\uparrow}$
Advanced objective-based	TAP [16]	36.1	43.4	69.9	13.6	17.3	26.1	33.0	49.8	22.5
	+Ours	**61.0**	**67.0**	**93.5**	**18.0**	**23.8**	**43.4**	**46.3**	**73.8**$_{24.0\uparrow}$	**32.9**$_{10.4\uparrow}$
	RPA [17]	64.9	68.6	92.5	26.2	35.5	53.0	58.6	75.3	43.3
	+Ours	**74.2**	**76.0**	**95.1**	**31.2**	**41.0**	**59.4**	**63.7**	**81.8**$_{6.5\uparrow}$	**48.8**$_{5.5\uparrow}$
	TAIG [49]	40.6	47.0	78.7	13.7	22.5	32.7	41.1	55.4	27.5
	+Ours	**77.5**	**80.8**	**96.9**	**32.7**	**44.6**	**64.7**	**66.3**	**85.1**$_{29.7\uparrow}$	**52.1**$_{24.6\uparrow}$
Model-related	SGM [19]	47.2	52.7	81.6	21.1	29.8	42.1	48.7	60.5	35.4
	+Ours	**71.5**	**76.8**	**95.2**	**33.9**	**46.3**	**62.7**	**64.3**	**81.2**$_{20.7\uparrow}$	**51.8**$_{16.4\uparrow}$
	BPA [18]	61.4	68.0	92.7	24.1	36.6	52.2	58.9	74.0	43.0
	+Ours	**85.7**	**88.3**	**98.6**	**37.6**	**51.9**	**72.1**	**72.7**	**90.9**$_{16.9\uparrow}$	**58.6**$_{15.6\uparrow}$

B. Evaluating Adversarial Transferability

To verify the effectiveness of our method, we integrate MAP with various black-box attack methods. Concretely, we generate the adversarial examples on a single model, i.e., Res-18 [2] and evaluate them on the other models. The evaluation results of the target model on the crafted adversarial examples are summarized in Tab. I. As expected, our MAP consistently demonstrates much better transferability than the baselines on all models with different architectures, ranging from $+2.6$ up to $+29.7$ MASR for CNNs, and $+5.5$ up to $+24.6$ MASR for ViTs. For example, compared to SIA [15], MAP enhances transferability to ViTs by 15.5%. Even compared to previous SOTA method L2T [48], MAP still achieves a notable gain, boosting transferability from $75.5\% \rightarrow 86.4\%$. Similar results are also observed when using ViT-B [11] as the surrogate model, as in Tab. II. For example, our MAP also boosts SIA [15] by a large margin of 5.9% on the MASR for CNNs. Even for model-related attacks (e.g., VDC [36]), MAP also

TABLE II
ASR (%) OF TRANSFER-BASED ATTACKS AGAINST THE NORMALLY TRAINED MODELS, AND THEIR ENHANCED VERSION BY OUR METHOD, USING VIT-B [11] AS THE SURROGATE MODEL.

Category	Attacks	CNNs				ViTs			MASR (CNNs)	MASR (ViTs)
		Res-18	Res-101	ReX-50	DN-121	PiT-B	Vis-S	Swin-T		
Gradient-based	MI-FGSM [9]	57.1	36.9	40.2	54.9	43.3	47.6	58.9	47.3	49.9
	+Ours	**70.5**	**51.2**	**54.5**	**67.6**	**61.7**	**64.9**	**75.9**	**61.0**$_{13.7\uparrow}$	**67.1**$_{17.2\uparrow}$
	VMI-FGSM [12]	70.3	52.7	59.0	68.2	64.6	67.1	76.0	62.6	69.2
	+Ours	**77.3**	**67.3**	**69.5**	**78.5**	**77.5**	**78.2**	**86.5**	**73.2**$_{10.6\uparrow}$	**80.7**$_{11.5\uparrow}$
	EMI-FGSM [45]	77.6	61.1	63.2	77.4	73.0	75.8	85.7	69.8	78.2
	+Ours	**86.4**	**78.7**	**79.1**	**87.4**	**87.0**	**88.9**	**93.5**	**82.9**$_{13.1\uparrow}$	**89.8**$_{11.6\uparrow}$
	AI-FGTM [46]	51.1	32.9	38.5	49.9	42.4	44.7	58.4	43.1	48.4
	+Ours	**55.0**	**41.9**	**45.1**	**54.7**	**51.9**	**52.7**	**65.1**	**49.2**$_{6.1\uparrow}$	**56.6**$_{8.2\uparrow}$
	GRA [5]	79.4	68.3	70.7	79.2	80.0	80.9	85.8	74.4	82.2
	+Ours	**85.7**	**79.5**	**79.8**	**85.5**	**87.8**	**87.8**	**91.7**	**82.6**$_{8.2\uparrow}$	**89.1**$_{6.9\uparrow}$
Input transformation-based	TIM [22]	55.7	33.2	41.2	54.6	38.6	42.2	50.3	46.2	43.7
	+Ours	**63.8**	**45.4**	**51.3**	**62.6**	**51.3**	**55.6**	**62.3**	**55.8**$_{9.6\uparrow}$	**56.4**$_{12.7\uparrow}$
	DIM [10]	70.9	60.8	63.1	71.6	71.5	73.0	76.7	66.6	73.7
	+Ours	**76.4**	**69.6**	**71.5**	**77.9**	**79.6**	**77.5**	**81.2**	**73.9**$_{7.3\uparrow}$	**79.4**$_{5.7\uparrow}$
	SSA [13]	75.8	61.2	63.9	74.6	72.1	72.7	83.1	68.9	76.0
	+Ours	**84.1**	**73.3**	**76.1**	**85.0**	**83.7**	**84.7**	**91.0**	**79.6**$_{10.7\uparrow}$	**86.5**$_{10.5\uparrow}$
	BSR [14]	88.6	81.5	85.9	91.3	90.9	90.1	90.7	86.8	90.6
	+Ours	**91.9**	**90.8**	**92.0**	**93.4**	**94.2**	**94.1**	**94.3**	**92.0**$_{5.2\uparrow}$	**94.2**$_{3.6\uparrow}$
	SIA [15]	89.2	81.7	83.9	91.3	91.0	90.2	93.2	86.5	91.5
	+Ours	**92.9**	**90.5**	**91.5**	**94.5**	**94.4**	**94.4**	**95.5**	**92.4**$_{5.9\uparrow}$	**94.8**$_{3.3\uparrow}$
	DeCowA [47]	93.1	85.1	88.2	95.0	94.1	93.3	93.5	90.4	93.6
	+Ours	**95.7**	**92.5**	**93.4**	**97.3**	**96.8**	**97.0**	**96.8**	**94.7**$_{4.3\uparrow}$	**96.9**$_{3.3\uparrow}$
	L2T [48]	94.2	97.1	97.0	96.3	98.0	97.3	98.1	96.2	97.8
	+Ours	**98.8**	**97.8**	**98.4**	**99.2**	**99.2**	**99.3**	**99.4**	**98.6**$_{2.4\uparrow}$	**99.3**$_{1.5\uparrow}$
Advanced objective-based	TAP [16]	29.8	15.6	19.2	26.5	18.2	21.0	31.8	22.8	23.7
	+Ours	**57.3**	**34.8**	**37.8**	**54.0**	**42.8**	**47.0**	**65.2**	**46.4**$_{23.6\uparrow}$	**51.7**$_{28.0\uparrow}$
	TAIG [49]	50.5	30.3	35.2	47.3	37.1	41.5	57.8	40.8	45.5
	+Ours	**77.1**	**62.7**	**66.6**	**77.8**	**73.7**	**76.6**	**88.0**	**71.1**$_{30.3\uparrow}$	**79.4**$_{33.9\uparrow}$
Model-related	VDC [36]	76.3	58.0	60.1	75.4	67.4	72.1	82.5	67.5	79.8
	+Ours	**86.0**	**68.1**	**69.5**	**83.0**	**76.7**	**80.1**	**89.8**	**76.7**$_{9.2\uparrow}$	**82.2**$_{2.4\uparrow}$
	SAPR [29]	79.3	52.2	58.1	73.2	64.3	67.6	82.5	65.7	71.5
	+Ours	**85.9**	**74.7**	**77.9**	**87.7**	**84.6**	**87.5**	**94.0**	**81.6**$_{15.9\uparrow}$	**88.7**$_{17.2\uparrow}$

TABLE III
ASR (%) OF VARIOUS ENSEMBLE-BASED ATTACKS. THE ADVERSARIAL EXAMPLES ARE CRAFTED ON THE RES-18 [2], RES-101 [2], REX-50 [50], AND DN-121 [51].

Attacks	ViT-B	PiT-B	Vis-S	Swin-T	MASR (ViTs)
ENS [20]	37.9	53.2	67.8	67.6	56.6
+Ours	**62.5**	**79.1**	**91.0**	**86.9**	**79.9**$_{23.3\uparrow}$
AdaEA [21]	38.6	53.1	68.7	66.8	56.8
+Ours	**63.6**	**78.4**	**90.2**	**86.8**	**79.8**$_{23.0\uparrow}$
SMER [30]	42.3	56.5	71.3	70.8	60.2
+Ours	**62.1**	**73.5**	**81.3**	**79.9**	**74.2**$_{14.0\uparrow}$

TABLE IV
ASR (%) OF VARIOUS ATTACKS ON MODELS OF DIFFERENT ARCHITECTURES. THE ADVERSARIAL EXAMPLES ARE CRAFTED ON THE RES-18 [2].

Attacks	MaxViT	MobViTV2	Mixer	MambaOut	MASR
MI-FGSM [9]	26.8	53.1	40.3	35.3	40.9
+Ours	**43.9**	**75.8**	**52.5**	**57.6**	**57.5**$_{16.6\uparrow}$
DIM [10]	47.3	81.2	57.5	61.0	63.4
+Ours	**65.3**	**91.2**	**70.0**	**74.6**	**75.3**$_{11.9\uparrow}$
BSR [14]	66.5	96.4	69.6	77.2	77.9
+Ours	**88.7**	**98.8**	**81.8**	**90.9**	**90.1**$_{12.2\uparrow}$
SIA [15]	70.8	97.5	67.5	81.6	79.6
+Ours	**88.2**	**99.0**	**84.1**	**91.8**	**90.8**$_{11.2\uparrow}$

provides considerable performance gains. Moreover, we compare various ensemble-based attacks in Tab. III. By integrating ours, substantial improvements in ASR are observed across all methods. For example, MAP improves MASR (ViTs) by up to 23.3% compared to SMER [30]. Finally, we evaluate the performance of MAP on more diverse architectural paradigms. As in Tab. IV, MAP still shows strong generalizability across architectures, such as MLP-Mixers [57] and hybrid models like MaxViT [55]. These improvements highlight the robustness and effectiveness of MAP in boosting transferability.

To further emphasize the generalization of the MAP, we evaluate its performance using more surrogate models (both CNN-based and ViT-based) and under varying hyper-parameters (e.g., mask ratios). The results, summarized in the Supplementary Material, demonstrate that our method has explored a perspective of diversifying perturbations that existing methods have not yet considered, and therefore is general to almost all methods to boost their transferability without bells and whistles.

TABLE V
ASR (%) OF VARIOUS DEFENSE METHODS ON THE ADVERSARIAL EXAMPLES GENERATED BY DIFFERENT ATTACKS. THE ADVERSARIAL EXAMPLES ARE CRAFTED ON THE RES-18 [2]. W/O AND W/ RESPECTIVELY INDICATE WHETHER MAP IS ADOPTED.

Method	AT [37]		HGD [38]		RS [42]		NRP [39]		R&P [41]		DiffPure [40]	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/
MI-FGSM [9]	33.3	$35.4_{2.1\uparrow}$	33.2	$57.6_{24.4\uparrow}$	23.4	$24.8_{1.4\uparrow}$	27.8	$38.2_{10.4\uparrow}$	39.5	$68.4_{28.9\uparrow}$	28.5	$34.7_{6.2\uparrow}$
TIM [22]	40.1	$43.6_{3.5\uparrow}$	41.3	$55.4_{14.1\uparrow}$	34.5	$42.3_{7.8\uparrow}$	40.0	$51.4_{11.4\uparrow}$	40.3	$56.5_{16.2\uparrow}$	59.5	$70.7_{11.2\uparrow}$
DIM [10]	37.4	$40.5_{3.1\uparrow}$	64.9	$82.1_{17.2\uparrow}$	26.6	$29.5_{2.9\uparrow}$	42.5	$54.5_{12.0\uparrow}$	70.4	$88.2_{17.8\uparrow}$	42.5	$53.5_{11.0\uparrow}$
BSR [14]	40.4	$42.7_{2.3\uparrow}$	83.1	$96.1_{13.0\uparrow}$	28.0	$31.5_{3.5\uparrow}$	51.4	$68.5_{17.1\uparrow}$	88.5	$98.1_{9.6\uparrow}$	45.1	$57.0_{11.9\uparrow}$
SIA [15]	40.6	$43.4_{2.8\uparrow}$	87.1	$95.6_{8.5\uparrow}$	28.8	$32.1_{3.3\uparrow}$	53.8	$70.2_{16.4\uparrow}$	91.1	$98.0_{6.9\uparrow}$	47.5	$58.5_{11.0\uparrow}$
TAIG [49]	33.2	$38.7_{5.5\uparrow}$	32.2	$65.0_{32.8\uparrow}$	22.1	$27.3_{5.2\uparrow}$	25.9	$48.1_{22.2\uparrow}$	41.8	$75.0_{33.2\uparrow}$	19.3	$43.0_{23.7\uparrow}$

TABLE VI
ASR (%) OF VARIOUS DEFENSE METHODS ON THE ADVERSARIAL EXAMPLES GENERATED BY DIFFERENT ATTACKS. THE ADVERSARIAL EXAMPLES ARE CRAFTED ON THE VIT-B [11]. W/O AND W/ RESPECTIVELY INDICATE WHETHER MAP IS ADOPTED.

Method	AT [37]		HGD [38]		RS [42]		NRP [39]		R&P [41]		DiffPure [40]	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/
MI-FGSM [9]	34.2	$35.2_{1.0\uparrow}$	30.9	$40.6_{9.7\uparrow}$	22.9	$24.7_{1.8\uparrow}$	28.5	$32.5_{4.0\uparrow}$	48.1	$63.5_{15.4\uparrow}$	23.1	$25.0_{1.9\uparrow}$
TIM [22]	39.0	$40.1_{1.1\uparrow}$	29.4	$39.3_{9.9\uparrow}$	29.1	$34.7_{5.6\uparrow}$	33.8	$40.8_{7.0\uparrow}$	41.0	$51.8_{10.8\uparrow}$	38.3	$43.8_{5.5\uparrow}$
DIM [10]	36.2	$37.5_{1.3\uparrow}$	57.4	$67.9_{10.5\uparrow}$	26.4	$28.5_{2.1\uparrow}$	41.2	$48.9_{7.7\uparrow}$	72.4	$77.0_{4.6\uparrow}$	32.8	$38.6_{5.8\uparrow}$
BSR [14]	37.6	$40.7_{3.1\uparrow}$	76.3	$85.4_{9.1\uparrow}$	29.3	$32.1_{2.8\uparrow}$	52.0	$63.0_{11.0\uparrow}$	88.2	$93.1_{4.9\uparrow}$	37.6	$47.0_{9.4\uparrow}$
SIA [15]	38.3	$40.5_{2.2\uparrow}$	76.5	$88.1_{11.6\uparrow}$	29.3	$32.8_{3.5\uparrow}$	54.6	$65.5_{10.9\uparrow}$	91.7	$95.1_{3.4\uparrow}$	37.8	$47.4_{9.6\uparrow}$
TAIG [49]	32.2	$35.3_{3.1\uparrow}$	24.4	$54.9_{30.5\uparrow}$	22.0	$27.1_{5.1\uparrow}$	24.0	$39.6_{15.6\uparrow}$	47.7	$78.3_{30.6\uparrow}$	14.6	$29.1_{14.5\uparrow}$

TABLE VII
MASR (%) OF OUR PROPOSED METHOD FOR DIFFERENT BLACK-BOX ATTACK METHODS. THE EXPERIMENT IS CONDUCTED BY USING RES-18 [2] AS THE SURROGATE MODEL. MASR IS OBTAINED BY AVERAGING ASRS OF DIFFERENT TYPES OF BLACK-BOX TARGET MODELS.

MAP	CML	SMG	MI-FGSM [9]		DIM [10]		RPA [17]		BPA [18]	
			MASR (CNNs)	MASR (ViTs)	MASR (CNNs)	MASR (ViTs)	MASR (CNNs)	MASR (ViTs)	MASR (CNNs)	MASR (ViTs)
			56.5	30.1	79.3	48.3	75.3	43.3	74.0	43.0
✓			70.5	40.7	83.7	55.8	76.6	44.7	84.6	52.9
✓	✓		73.0	40.6	85.2	58.0	77.5	45.6	86.6	53.2
✓		✓	75.5	42.7	90.1	62.4	81.3	46.9	88.8	57.2
✓	✓	✓	**80.0**	**46.0**	**91.4**	**65.4**	**81.8**	**48.8**	**90.9**	**58.6**

C. Evaluating on Defense Methods

MAP performs excellently on diverse normally trained models when attacking both single and ensemble models. Recently, several defense methods have been proposed to deal with the threat of AEs. To fully validate the effectiveness of MAP, we use the AEs generated on a single model using various attacks to attack the defense models, including AT [37], HGD [38], RS [42], NRP [39], R&P [41], and DiffPure [40].

As presented in Tab. V, MAP achieves consistent improvements across various attacks and defenses, ranging from $+1.4$ up to $+33.2$ ASR. In particular, BSR [14] (w/ MAP) achieves 96.1% ASR on the powerful denoising method HGD [38] with a weaker surrogate model, i.e., Res-18 [2]. Besides, similar results are also observed when using ViT-B [11] as the surrogate model, as shown in Tab. VI. For example, SIA [15] (w/ MAP) achieves 88.1% ASR on the HGD, which outperforms vanilla SIA by 11.6%. Such improvement can be attributed to the fact that MAP makes AEs more generalizable by enhancing perturbation diversity while distributing perturbations across individual patches and directly targeting the basic processing units (patch) of existing models, thus effectively resisting existing defense methods. This characteristic further validates MAP's effectiveness against models equipped with carefully designed defense strategies and helps identify the shortcomings of existing defense methods.

D. Ablation Studies

To gain further insights into the superior performance of MAP, we perform detailed ablation studies to validate its effectiveness and two proposed strategies, i.e., CML and SMG. As in Tab. VII, using any of the components can boost transferability. Concretely, MAP individually leads to noticeable improvement over the baseline, verifying its effectiveness. For example, by adding MAP to MI-FGSM [9], MASR is improved by 14.0% for CNNs (70.5% vs. 56.5%) and 10.6% for ViTs (40.7% vs. 30.1%). Further, both CML and SMG can bring impressive improvements on the basis of MAP. For instance, CML increase MASR (CNNs) and MASR (ViTs) by 1.5% (85.2% vs. 83.7%) and 2.2% (58.0% vs. 55.8%) respectively on DIM [10] with MAP, while SMG increase them by 6.4% (90.1% vs. 83.7%) and 6.6% (62.4% vs. 55.8%), respectively. The improvement proves that updating the mask ratio via curriculum learning and generating soft masks can indeed boost performance. Finally, combining MAP with CML and SMG achieves the best performance, indicating the latter two complement MAP to further improve transferability.

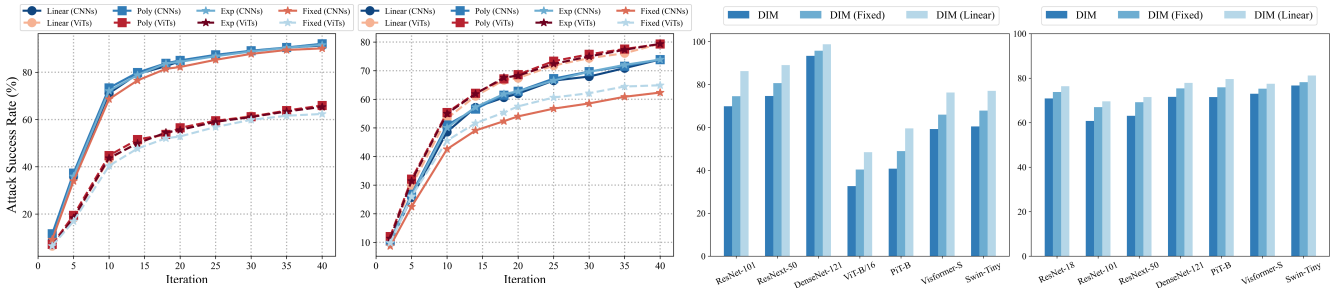


Fig. 7. ASR (%) of the CML strategy under different settings. For the first and third figures, we adopt Res-18 [2] as the surrogate model, while for others, we adopt ViT-B [11]. Linear (CNNs) denotes the MASR for CNNs based on DIM [10] with the proposed MAP and Linear schedule, otherwise similar.

TABLE VIII
PERFORMANCE COMPARISON OF DIFFERENT MASKED ADVERSARIAL ATTACKS AND OUR METHOD. THE SURROGATE MODEL IS RES-18 [2].

Attacks	CNNs			ViTs				MASR	MASR
	Res-101	ReX-50	DN-121	ViT-B	PiT-B	Vis-S	Swin-T	(CNNs)	(ViTs)
MI-FGSM [9]	41.6	49.2	78.7	17.3	24.5	36.2	42.3	56.5	30.1
+ Oval Mask	41.8	47.9	73.3	15.4	22.0	32.4	38.7	$54.3_{2.2\downarrow}$	$27.1_{3.0\downarrow}$
+ Circular Mask	42.0	47.1	74.1	15.6	21.4	33.5	39.3	$54.4_{2.1\downarrow}$	$27.5_{2.6\downarrow}$
+ Triangular Mask	42.0	46.2	73.4	15.6	21.9	31.8	40.1	$53.9_{2.6\downarrow}$	$27.4_{2.7\downarrow}$
+ MaskBlock [23]	49.2	51.4	78.6	18.0	25.1	38.1	45.6	$59.7_{3.2\uparrow}$	$31.7_{1.6\uparrow}$
+ LPM [6]	53.0	59.6	85.1	17.1	24.9	41.3	45.3	$65.9_{9.4\uparrow}$	$32.2_{2.1\uparrow}$
+ Ours	**69.3**	**75.9**	**94.7**	**28.0**	**39.9**	**57.7**	**58.5**	$\mathbf{80.0}_{23.5\uparrow}$	$\mathbf{46.0}_{15.9\uparrow}$

Ablation on Mask Ratio Schedules. In this paper, we propose a Curriculum Mask Learning strategy to dynamically adjust mask ratio during iterations. Initially, the mask ratio is lower to ensure effective learning of adversarial perturbations. With iteration, the mask ratio is gradually increased to cope with overfitting on specific patterns. In Eq. 8, we list three schedule instances, i.e., Linear, Exp, and Poly schedules, respectively. We represent the case as a baseline where the mask ratio is fixed to a constant (e.g., 0.5 or 0.7) during iterations, which is denoted as Fixed schedule. We report ASR of DIM [10] attacks (known as classic input transformation-based attack) under various ratio schedules in Fig. 7. We observe that the ASR improves with more iterations, and our dynamic schedule consistently outperforms the Fixed schedule. In particular, the three schedules exhibit similar performance and are all better than the Fixed schedule. Given its simplicity, we adopt Linear schedule in our method and leave the exploration of more dedicated mask ratio schedules in future work. Finally, we find that while the fixed schedule outperforms the baseline (DIM), its additional gain is somewhat lower compared to the dynamic schedule, likely due to limited perturbation variation and insufficient exploration of diverse perturbation patterns.

Ablation on Mask Strategies. We further analyze the impact of different masking strategies on adversarial transferability. As in Tab. VIII, compared to MAE strategy (e.g., LPM and MaskBlock), our MAP significantly improves transferability by avoiding large statistical shifts and unexpected gradient feedback. In addition, we evaluate different mask shapes (e.g., triangle, circle) and find that rectangular masks we used achieve the best performance, likely due to their better alignment with the feature extraction mechanisms of CNNs and ViTs. These results highlight the effectiveness of MAP's perturbation masking strategy and its superior adaptability across models. Finally, we study the performance of different mask localization strategies in Tab. IX. The results show that random masks (RM) achieve the best transferability across CNNs and ViTs, while foreground (FM) and background masks (BM) perform worse due to limited perturbation diversity caused by over-emphasizing specific image regions.

TABLE IX
COMPARATIVE EXPERIMENTS ON DIFFERENT MASK LOCATION STRATEGIES. THE ADVERSARIAL EXAMPLES ARE CRAFTED ON THE RES-18 [2]. FM, BM, AND RM DENOTE THE USE OF FOREGROUND MASKS, BACKGROUND MASKS, AND RANDOM MASKS, RESPECTIVELY.

Attacks	MASR(CNNs)			MASR(ViTs)		
	FM	BM	RM	FM	BM	RM
MI-FGSM [9]	73.3	75.8	**80.0**	40.0	41.8	**46.0**
DIM [10]	89.0	90.0	**91.4**	61.3	62.3	**65.4**
BSR [14]	97.1	97.3	**98.6**	74.3	75.1	**81.5**
SIA [15]	98.2	98.5	**99.1**	77.5	80.1	**82.2**

E. Parameter Studies

We perform parameter studies to discuss the effect of the hyper-parameters of MAP, namely mask patch size b and mask ratio r. They determine the learning efficiency and effect of the adversarial examples during iterations. To find the proper combination for b and r, we evaluate MAP on two classic attack methods (i.e., DIM [10] and TIM [22]) with various b and r, and the results are shown in the Tab. X and Tab. XI.

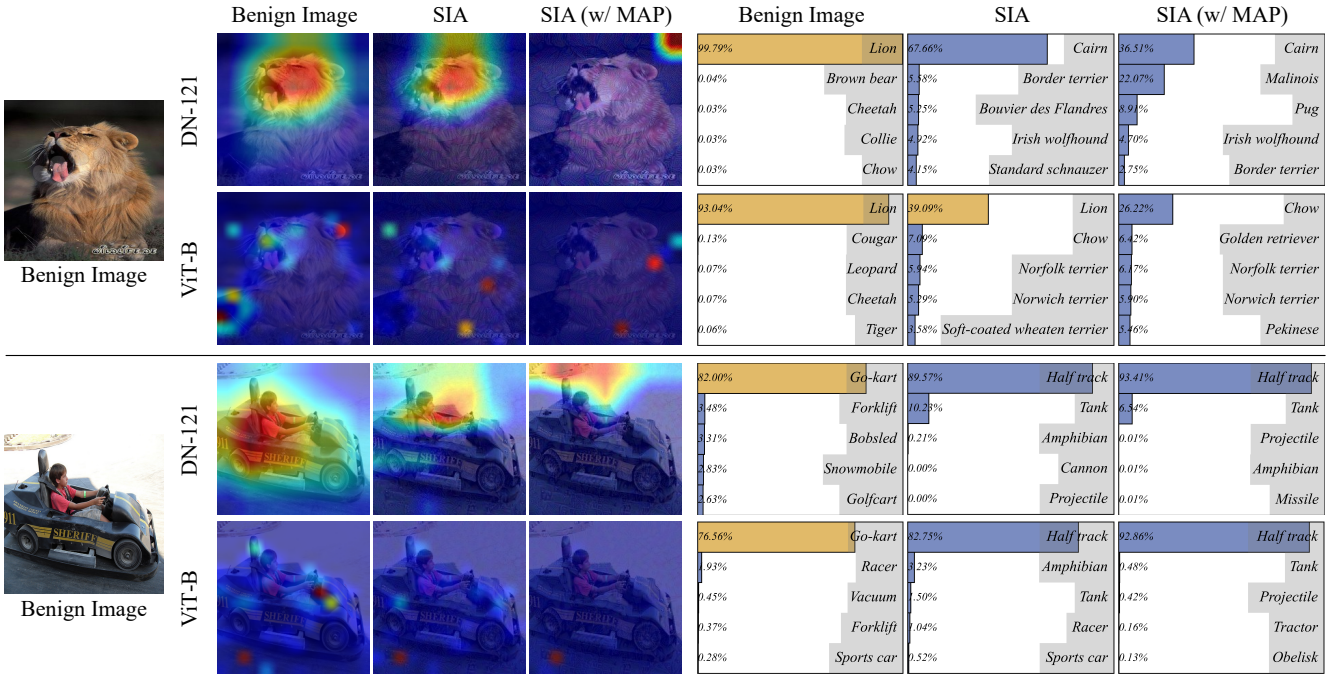
For Tab. X, we adopt DIM [10] (w/ MAP) to generate the adversarial examples on Res-18 [2]. Compared to the baseline DIM attack, which enjoys 79.3% MASR for CNNs and 48.3% MASR for ViTs, MAP achieves remarkable improvements in a range of b between 4 and 32 and r between 0.3 and 0.7. In

TABLE X
PARAMETER STUDY OF THE PATCH SIZE AND THE MASK RATIO OF MAP WITH DIM [10]. WE USE RES-18 [2] AS THE SURROGATE MODEL.

MASR (CNNs)/ MASR (ViTs)		Mask Ratio						
		0.3	0.5	0.7	0.9	[0.3-0.5]	[0.3-0.7]	[0.4-0.9]
Patch Size	4	86.7/55.7	88.4/58.9	87.6/60.3	78.5/52.6	88.8/59.1	91.2/62.4	89.9/64.2
	8	86.9/56.4	89.2/60.0	89.0/61.7	80.4/53.4	89.0/59.9	91.3/64.1	90.7/64.6
	16	87.6/56.5	88.8/60.0	89.2/61.8	81.4/54.6	90.1/60.0	91.2/63.5	**91.4/65.4**
	32	86.6/55.0	89.0/59.1	89.1/61.3	76.7/49.6	89.2/59.6	90.0/61.5	90.3/63.1

TABLE XI
PARAMETER STUDY OF THE PATCH SIZE AND THE MASK RATIO OF MAP WITH TIM [22]. WE USE VIT-B [11] AS THE SURROGATE MODEL.

MASR (CNNs)/ MASR (ViTs)		Mask Ratio						
		0.3	0.5	0.7	[0.2-0.5]	[0.2-0.6]	[0.3-0.6]	[0.3-0.7]
Patch Size	4	50.4/47.3	53.2/51.9	50.3/49.2	53.3/51.0	53.9/52.4	54.8/52.5	54.9/53.7
	8	51.5/49.5	52.6/53.2	51.0/50.0	54.5/51.9	54.8/55.7	54.5/55.2	**56.0**/55.9
	16	51.8/50.6	52.9/53.8	49.5/48.3	54.6/55.0	54.8/56.3	55.6/56.2	55.8/**56.4**
	32	51.7/50.5	52.1/53.5	49.2/49.3	53.7/53.8	53.9/54.8	53.8/55.0	53.9/55.7



(a) Attention Maps (b) Top-5 Confidence Distribution Plots

Fig. 8. Comparisons between SIA [15] and SIA (w/ MAP) attacks using Res-18 [2] as the surrogate model. We visualize the benign input, as well as Grad-CAM map (for DN-121 [51]) and self-attention map (for ViT-B [11]) of adversarial examples generated by two attacks in (a), respectively. The ground-truth labels of the two benign images are *Lion* and *Go-kart*, and are marked as orange in (b). Best viewed in color and zoom in.

particular, as discussed earlier, a lower mask ratio ($r = 0.3$) is inferior to a higher one ($r = 0.7$), but an excessively high mask ratio ($r = 0.9$) rather leads to performance degradation. For example, when $b = 16$, $r = 0.7$, MAP achieves 61.8% MASR (ViTs), outperforming $r = 0.3$ by 5.3% and $r = 0.9$ by 7.2%. In contrast, CML achieves better performance with various mask ratio ranges, indicating its robustness to mask ratios. The best performance is achieved for $b = 16$, $r_s = 0.4$, and $r_e = 0.9$. Hence, we adopt this setting by default when using CNNs (e.g., Res-18 [2]) as the surrogate model.

For Tab. XI, we adopt TIM [22] (w/ MAP) to generate the adversarial examples on ViT-B [11]. Similarly, compared to the baseline TIM attack, which exhibits 46.2% MASR for CNNs and 43.7% MASR for ViTs, MAP also achieves noticeable improvements in a range of b between 4 and 32 and r between 0.3 and 0.7. In particular, we also found that an excessively high mask ratio ($r = 0.7$) decays the attack performance, which can be attributed to underfitting to the surrogate model. To strike a balance between overfitting (an excessively low mask ratio) and underfitting (an excessively high mask ratio), we introduce the CML strategy for better performance, as in Tab. XI. When $r_s = 0.3$, $r_e = 0.6$ or 0.7, $b = 8$ or 16, the black-box attack performance is relatively optimal. Hence, we set $r_s = 0.3$, $r_e = 0.7$, and $b = 16$ in our experiments when

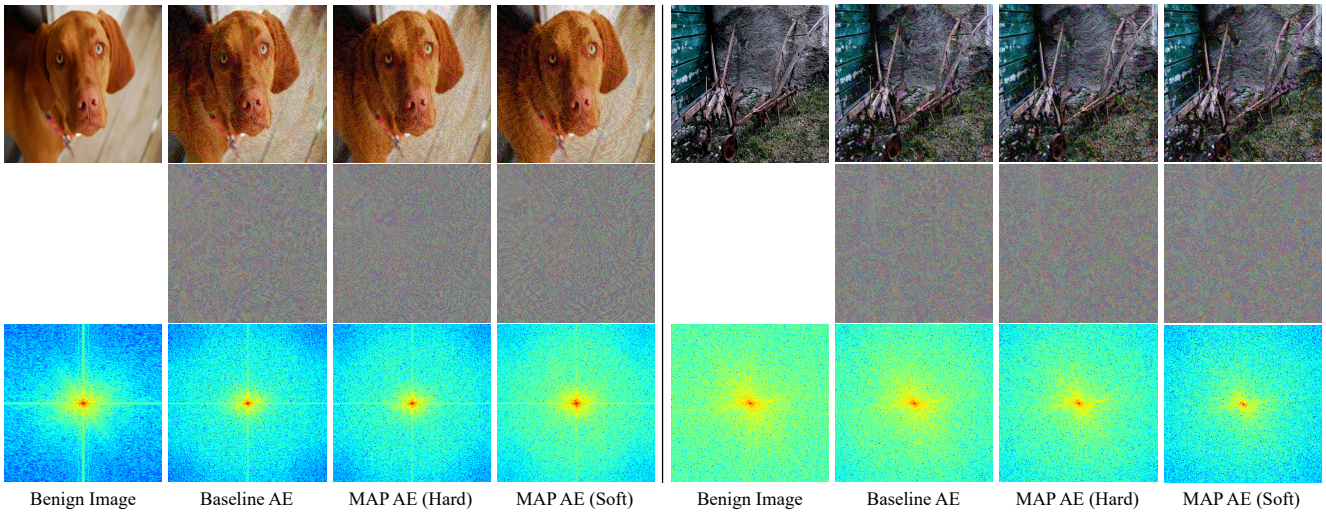


Fig. 9. Visualization of adversarial examples, perturbations, and corresponding spectrums from various methods. "Baseline AE" refers to adversarial examples from the baseline attack, while "MAP AE (Hard)" and "MAP AE (Soft)" denote those generated by MAP with hard and soft masks, respectively.

using ViTs (e.g., ViT-B [11]) as the surrogate model.

F. Visualizations

To analyze the effectiveness of our MAP, we visualize the attack effects of the SIA [15] (w/ and w/o MAP), with attention maps, which can highlight the discriminative regions for the classification network. Specifically, we use Grad-CAM [59] to visualize the heatmaps of the CNN-based model, DN-121 [51], and use self-attention maps to visualize the attention of the ViT-based model, ViT-B [11]. As shown in Fig. 8(a), both attacks perturb the network's attention maps. However, by comparison, MAP weakens the network's attention to the objects more effectively. For example, the attention maps of the adversarial examples generated by SIA [15] still highlight object regions to some extent, while SIA (w/ MAP) focuses on highlighting non-object regions to perturb the network.

In addition, we present the Top-5 confidence scores of the two black-box target network outputs under both attacks. As in Fig. 8(b), when the difference in model architecture is small (i.e., the target model is also a CNN-based model), both methods can perturb the classification confidence. For example, both methods, i.e., SIA without and with MAP, can deceive DN-121 [51]. However, when the model difference is large (i.e. the target model is a ViT-based model), we find that vanilla SIA [15] sometimes fails to mislead ViT-B [11]. For instance, for the first image, although the adversarial examples generated by SIA [15] weaken the confidence of the ground-truth label, it still cannot make the target model ViT-B [11] output wrong predictions. By contrast, SIA (w/ MAP) can effectively misguide the networks. This merit can be attributed to the intrinsic mechanism of the proposed MAP, which increases the attack effect of each patch, and disturbs the target model by attracting its attention patterns to the regions corresponding to the non-ground-truth classes.

Finally, to further illustrate the impact of MAP, we visualize some adversarial examples, perturbations, and their corresponding spectrums using Fast Fourier Transform (FFT) [60]. FFT is used to analyze the frequency components of AEs, offering insights into how MAP affects model features across different frequency ranges. As in Fig. 9, MAP (Soft) induces broader spectrum changes across both low- and high-frequency components compared to the baseline (MI-FGSM [9]) and MAP (Hard), relative to the original benign images. As various models (e.g., CNNs) process frequency information differently [61], this frequency diversification contributes to MAP's superior attack effect and cross-model generalization.

V. Conclusion

In this paper, we present Masked Adversarial Perturbation (MAP), a universal method to boost black-box adversarial transferability. MAP progressively masks a random selection of adversarial perturbation patches and requires the remaining patches to still retain the attack effect. As MAP masks various patches in each iteration, it diversifies perturbations explicitly to prevent overfitting between perturbation and surrogate model, and co-adapting between patches, thus showing superior transferability for various architectures (either CNNs or ViTs). With an extensive evaluation, we have proven that MAP achieves noticeable performance improvements on various black-box attack methods. We hope that, due to its simplicity, MAP can be adopted as part of future black-box methods to narrow the gap between black-box and white-box attacks.

Future. In the future, we will investigate more reasonable mask generation methods (e.g., model-shared discriminative region mining) and apply MAP to more black-box attacks.

References

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the*

IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[5] H. Zhu, Y. Ren, X. Sui, L. Yang, and W. Jiang, "Boosting adversarial transferability via gradient relevance attack," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4741–4750, 2023.

[6] X. Wei and S. Zhao, "Boosting adversarial transferability with learnable patch-wise masks," *IEEE Transactions on Multimedia*, 2023.

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

[8] Z. Ge, H. Liu, W. Xiaosen, F. Shang, and Y. Liu, "Boosting adversarial transferability by achieving flat local maxima," *Advances in Neural Information Processing Systems*, vol. 36, pp. 70141–70161, 2023.

[9] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

[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," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2730–2739, 2019.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[12] X. Wang and K. He, "Enhancing the transferability of adversarial attacks through variance tuning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1924–1933, 2021.

[13] Y. Long, Q. Zhang, B. Zeng, L. Gao, X. Liu, J. Zhang, and J. Song, "Frequency domain model augmentation for adversarial attack," in *European conference on computer vision*, pp. 549–566, Springer, 2022.

[14] K. Wang, X. He, W. Wang, and X. Wang, "Boosting adversarial transferability by block shuffle and rotation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2024.

[15] X. Wang, Z. Zhang, and J. Zhang, "Structure invariant transformation for better adversarial transferability," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4607–4619, 2023.

[16] W. Zhou, X. Hou, Y. Chen, M. Tang, X. Huang, X. Gan, and Y. Yang, "Transferable adversarial perturbations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 452–467, 2018.

[17] 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*, pp. 1672–1678, 2022.

[18] W. Xiaosen, K. Tong, and K. He, "Rethinking the backward propagation for adversarial transferability," *Advances in Neural Information Processing Systems*, vol. 36, pp. 1905–1922, 2023.

[19] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, "Skip connections matter: On the transferability of adversarial examples generated with resnets," in *International Conference on Learning Representations*, 2020.

[20] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *International Conference on Learning Representations*, 2017.

[21] B. Chen, J. Yin, S. Chen, B. Chen, and X. Liu, "An adaptive model ensemble adversarial attack for boosting adversarial transferability," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4489–4498, 2023.

[22] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4312–4321, 2019.

[23] M. Fan, C. Chen, X. Liu, and W. Guo, "Maskblock: Transferable adversarial examples with bayes approach," *arXiv preprint arXiv:2208.06538*, 2022.

[24] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] X. Li, S. Chen, X. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2682–2690, 2019.

[27] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

[28] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[29] H. Zhou, Y.-a. Tan, Y. Wang, H. Lyu, S. Wu, and Y. Li, "Improving the transferability of adversarial examples with restructure embedded patches," *arXiv preprint arXiv:2204.12680*, 2022.

[30] B. Tang, Z. Wang, Y. Bin, Q. Dou, Y. Yang, and H. T. Shen, "Ensemble diversity facilitates adversarial transferability," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24377–24386, 2024.

[31] Y. Shi, Y. Han, Y.-a. Tan, and X. Kuang, "Decision-based black-box attack against vision transformers via patch-wise adversarial removal," *NeurIPS*, vol. 35, pp. 12921–12933, 2022.

[32] Y. Shi, Y. Han, Q. Hu, Y. Yang, and Q. Tian, "Query-efficient black-box adversarial attack with customized iteration and sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2226–2245, 2022.

[33] Y. Bai, Y. Wang, Y. Zeng, Y. Jiang, and S.-T. Xia, "Query efficient black-box adversarial attack on deep neural networks," *Pattern Recognition*, vol. 133, p. 109037, 2023.

[34] Y. Gan, C. Wu, D. Ouyang, S. Tang, M. Ye, and T. Xiang, "Lesep: Boosting adversarial transferability via latent encoding and semantic embedding perturbations," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[35] Y. Wang, Y. Wu, S. Wu, X. Liu, W. Zhou, L. Zhu, and C. Zhang, "Boosting the transferability of adversarial attacks with frequency-aware perturbation," *IEEE Transactions on Information Forensics and Security*, 2024.

[36] J. Zhang, Y. Huang, Z. Xu, W. Wu, and M. R. Lyu, "Improving the adversarial transferability of vision transformers with virtual dense connection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 7133–7141, 2024.

[37] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *International Conference on Learning Representations*, 2020.

[38] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1778–1787, 2018.

[39] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli, "A self-supervised approach for adversarial robustness," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 262–271, 2020.

[40] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion models for adversarial purification," in *ICML*, 2022.

[41] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *ICLR*, 2018.

[42] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *international conference on machine learning*, pp. 1310–1320, PMLR, 2019.

[43] J. Byun, S. Cho, M.-J. Kwon, H.-S. Kim, and C. Kim, "Improving the transferability of targeted adversarial examples through object-based diverse input," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15244–15253, 2022.

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[45] X. Wang, J. Lin, H. Hu, J. Wang, and K. He, "Boosting adversarial transferability through enhanced momentum," in *British Machine Vision Conference*, 2021.

[46] J. Zou, Y. Duan, B. Li, W. Zhang, Y. Pan, and Z. Pan, "Making adversarial examples more transferable and indistinguishable," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 3662–3670, 2022.

[47] Q. Lin, C. Luo, Z. Niu, X. He, W. Xie, Y. Hou, L. Shen, and S. Song, "Boosting adversarial transferability across model genus by deformation-constrained warping," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 3459–3467, 2024.

[48] R. Zhu, Z. Zhang, S. Liang, Z. Liu, and C. Xu, "Learning to transform dynamically for better adversarial transferability," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24273–24283, 2024.

[49] Y. Huang and A. W.-K. Kong, "Transferable adversarial attack based on integrated gradients," in *International Conference on Learning Representations*, 2022.

[50] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[51] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[52] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11936–11945, 2021.

[53] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian, "Visformer: The vision-friendly transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 589–598, 2021.

[54] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[55] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "Maxvit: Multi-axis vision transformer," in *European conference on computer vision*, pp. 459–479, Springer, 2022.

[56] S. Mehta and M. Rastegari, "Separable self-attention for mobile vision transformers," *arXiv preprint arXiv:2206.02680*, 2022.

[57] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in neural information processing systems*, vol. 34, pp. 24261–24272, 2021.

[58] W. Yu and X. Wang, "Mambaout: Do we really need mamba for vision?," *arXiv preprint arXiv:2405.07992*, 2024.

[59] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam,

D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[60] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *International conference on machine learning*, pp. 5301–5310, PMLR, 2019.

[61] S. Wang, R. Veldhuis, C. Brune, and N. Strisciuglio, "What do neural networks learn in image classification? a frequency shortcut perspective," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1433–1442, 2023.

Kaige Li received the M.S. degree from the Ocean University of China, Qingdao, China, in 2020. He is currently pursuing the Ph.D. degree at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China. His current research interests include computer vision, artificial intelligence and smart city.

Maoxian Wan received the B.S. degree from Beijing University of Posts and Telecommunications in 2022. He is currently pursuing the M.S. degree at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China. His research interests include computer vision and artificial intelligence.

Qichuan Geng received the B.S. degree in Automation Science in 2012 and the Ph.D. degree in Technology of Computer Application in 2021 from Beihang University, Beijing, China. He is currently a Lecturer and the Master's Instructor with the Information Engineering College, Capital Normal University. His main research interests include computer vision, artificial intelligence, and scene geometry recovery.

Weimin Shi received the MS. degree from Beijing University of Chemical Technology in 2021. He is currently pursuing the Ph.D. degree in the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China. His current research interests include deep learning, multi-modal learning, computer vision, and smart city.

Xiaochun Cao (Senior Member, IEEE) is a Professor of School of Cyber Science and Technology, Sun Yat-sen University. He received the B.E. and M.E. degrees both in computer science from Beihang University (BUAA), China, and the Ph.D. degree in computer science from the University of Central Florida, USA, with his dissertation nominated for the university-level Outstanding Dissertation Award. After graduation, he spent about three years at ObjectVideo Inc. as a Research Scientist. From 2008 to 2012, he was a professor at Tianjin University. Before joining SYSU, he was a professor at the Institute of Information Engineering, Chinese Academy of Sciences. He has authored and co-authored over 200 journal and conference papers. In 2004 and 2010, he was the recipient of the Piero Zamperoni best student paper award at the International Conference on Pattern Recognition. He is a fellow of IET and a Senior Member of IEEE. He is on the editorial boards of IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE TRANSACTIONS ON MULTIMEDIA, and was on the editorial board of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Zhong Zhou received the B.S. degree from Nanjing University and the Ph.D. degree from Beihang University in 1999 and 2005 respectively. He is currently Professor of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, and also of Zhongguancun Laboratory, Beijing, China. His main research interests include Virtual Reality/Augmented Reality/Mixed Reality, Computer Vision, Artificial Intelligence and Cognitive Security.