Boosting Adversarial Transferability via Ensemble Non-Attention

Yipeng Zou[1], Qin Liu[1*], Jie Wu[2, 3], Yu Peng[4], Guo Chen[1], Hui Zhou[1], Guanghui Ye[1]

[1]College of Computer Science and Electronic Engineering, Hunan University
[2]China Telecom Cloud Computing Research Institute
[3]Department of Computer and Information Sciences, Temple University
[4]Laboratory of Intelligent Collaborative Computing, University of Electronic Science and Technology of China
{cszyp, gracelq628, guochen, huizhoucsee, yghui}@hnu.edu.cn, jiewu@temple.edu, ypeng@uestc.edu.cn

Abstract

Ensemble attacks integrate the outputs of surrogate models with diverse architectures, which can be combined with various gradient-based attacks to improve adversarial transferability. However, previous work shows unsatisfactory attack performance when transferring across heterogeneous model architectures. The main reason is that the gradient update directions of heterogeneous surrogate models differ widely, making it hard to reduce the gradient variance of ensemble models while making the best of individual model. To tackle this challenge, we design a novel ensemble attack, NAMEA, into the iterative gradient optimization process. Our design is inspired by the observation that the attention areas of heterogeneous models vary sharply, thus the non-attention areas of ViTs are likely to be the focus of CNNs and vice versa. Therefore, we merge the gradients respectively from the attention and non-attention areas of ensemble models so as to fuse the transfer information of CNNs and ViTs. Specifically, we pioneer a new way of decoupling the gradients of non-attention areas from those of attention areas, while merging gradients by meta-learning. Empirical evaluations on ImageNet dataset indicate that NAMEA outperforms AdaEA and SMER, the state-of-the-art ensemble attacks by an average of 15.0% and 9.6%, respectively. This work is the first attempt to explore the power of *ensemble non-attention* in boosting cross-architecture transferability, providing new insights into launching ensemble attacks.

Introduction

Deep neural networks (DNNs) including convolutional neural networks (CNNs) and vision transformers (ViTs) (He et al. 2016a; Dosovitskiy et al. 2020) are found to be highly vulnerable to adversarial examples. Worse still, adversarial examples crafted from surrogate models are transferable to unknown target models, making black-box attacks feasible in real-world applications. To better understand the vulnerabilities of DNNs, various transferability enhancement approaches have been proposed (Lin et al. 2019; Xie et al. 2019). Thereinto, ensemble attacks that integrate the predictions, losses, or logits of surrogate models to calculate the gradients with regard to updating adversarial examples, have shown superior adversarial transferability as they can mislead multiple surrogate models at once (Dong et al. 2018).

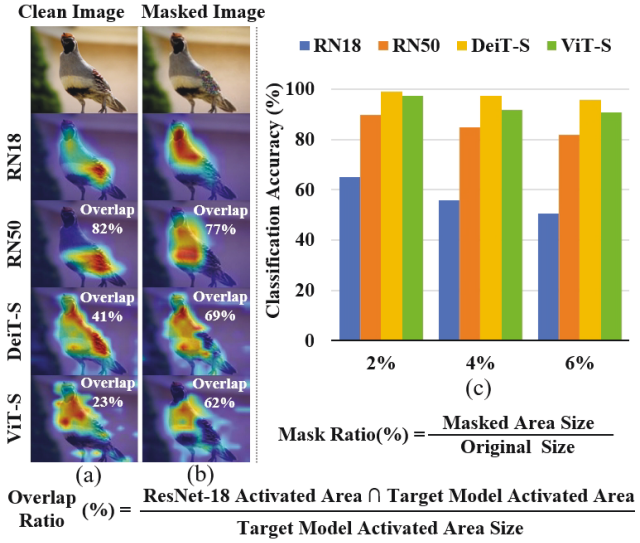

Figure 1: Attention heatmaps and classification accuracies of clean and masked images. A masked image is crafted by replacing the attention area of ResNet-18 with random noises. Target models include ResNet-50, DeiT-S, and ViT-S.

However, previous work mainly focused on transferring across models with homogeneous architectures (e.g., from surrogate CNNs to target CNNs), exhibiting poor performance when transferring across heterogeneous model architectures (e.g., from surrogate CNNs and ViTs to target CNNs and ViTs). The root cause is that the gradient update directions of heterogeneous surrogate models differ widely. For this reason, even the state-of-the-art (SOTA) ensemble attacks found it hard to balance between reducing the gradient variance of ensemble models and making the best of individual model, thus easily falling into local optimality. For instance, AdaEA (Chen et al. 2023) mitigated gradient variance across surrogate models by a discrepancy-reducing filter, which ensured stable update directions at the expense of model diversity; While SMER (Tang et al. 2024) independently optimized individual surrogate model without considering smoothing gradients, which may cause the attack

*Corresponding author.

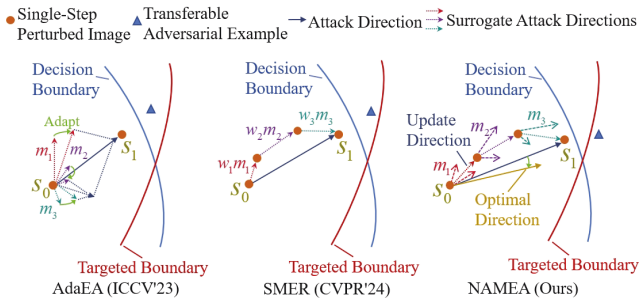

Figure 2: The attack direction search strategies of AdaEA, SMER and NAMEA. AdaEA focuses on reducing gradient discrepancies to improve attack effectiveness. SMER leverages model diversity to search the attack direction. NAMEA merges gradients of attention and non-attention areas by meta-learning to obtain a more accurate attack direction.

optimization direction to be less accurate. *Hence, the main challenge lies in how to make the best of individual model while stabilizing update direction among ensemble models.*

To tackle this challenge, we propose a non-attention enhanced meta ensemble attack, NAMEA. Our design is inspired by the observation that homogeneous models share many attention areas, but heterogeneous models focus on fairly different areas as shown in Fig. 1(a). That is, the non-attention areas of CNNs are probably to be the focus of ViTs, and vice versa. This observation is also quantitatively supported by Fig. 1(c), which shows the classification accuracies of 1,000 random ImageNet images after masking varying attention areas of ResNet-18. From this figure, we can see that as the mask ratio increases, the classification accuracies on CNNs decline substantially (up to 30%), but for ViTs, the accuracies drop slightly (within 10%). Meanwhile, we were surprised to observe that the masked image induced high ratios of attention overlaps across both homogeneous and heterogeneous models as shown in Fig. 1(b). So we have a hypothesis that cross-architecture transferability may can be improved by harnessing **the non-attention areas of ensemble models** i.e., **ensemble non-attention**.

To verify this hypothesis, we pioneer a new way of decoupling the gradients of ensemble non-attention from those of the attention areas of ensemble models, while incorporating meta-learning (Yuan et al. 2021) into Our meta-gradient optimization method consists of three steps: ① *Attention Meta-Training* that iteratively updates gradients based on the attention areas of ensemble models. ② *Non-Attention Meta-Testing* that iteratively optimizes gradients based on the non-attention areas of ensemble models. ③ *Final Update* that merges gradients calculated from both the meta-training and meta-testing steps. The first two steps encourage obtaining diverse gradients from ensemble models, while the last step aims to find a balance between stable update direction and model diversity. Especially, we construct a non-attention extraction (NAE) module based on Grad-CAM (Selvaraju et al. 2017) to extract (non-)attention areas, while designing a gradient scaling optimization (GSO) module to boost adversarial transferability in meta-testing step.

It is worth noticing that *while attention mechanism or meta-learning had been employed in adversarial attacks (Li et al. 2024; Wang et al. 2022), prior work focused on transferring across homogeneous models without considering the large gradient differences in heterogeneous ensemble models.* In contrast, **our work is the first to put forward the concept of ensemble non-attention, while merging gradients by meta learning, thus tackling the core challenge in improving cross-architecture transferability.** The major differences from the SOTA ensemble attacks are shown in Fig. 2, and our contributions are summarized as follows:

- We propose a novel ensemble attack, NAMEA, which ensures stable update direction and model diversity at once, exhibiting superior cross-architecture transferability.
- NAMEA innovatively decouples the gradients of ensemble non-attention from those of attention areas of ensemble models, while incorporating meta-learning into iterative gradient optimization process for gradient merging.
- As a plug-and-play method, NAMEA largely enhances ensemble attack performance, when combined with various gradient-based attacks. Especially for ImageNet dataset, NAMEA outperforms SOTA ensemble attacks, AdaEA and SMER, by an average of 15.0% and 9.6%, respectively. *With these encouraging results, we confirm that, ensemble non-attention contributes to boosting cross-architecture transferability, and our NAMEA provides new insights into launching ensemble attacks.*

Related Work

This section introduces the most relevant work while putting the details of adversarial attacks and defenses into APPX. A.

Ensemble Attacks. Ens (Liu et al. 2017) directly averaged the ensemble models' predictions to obtain an ensemble loss before launching gradient-based attacks. (Dong et al. 2018) further introduced the logits-based ensemble losses to enhance the adversarial transferability. SVRE (Xiong et al. 2022) reduced the gradient variance by using stochastic variance-reduced gradients. To transfer across CNNs and ViTs, AdaEA (Chen et al. 2023) adaptively fused model outputs by monitoring and adjusting gradient contributions. CWA (Chen et al. 2024) improved adversarial transferability by adjusting the flatness of the loss function. SMER (Tang et al. 2024) emphasized the diversity of surrogate models, and introduced ensemble reweighing to refine ensemble weights based on reinforcement learning. CSA (Li et al. 2025) leveraged multiple checkpoints from a single model's training trajectory to improve transferability.

Attention-based or Meta-Learning-based Attacks. For attention-based attacks, AGTA (Wu et al. 2020) guided perturbations by computing attention weights to disrupt critical features shared across CNNs. AoA (Chen et al. 2020) improved transferability by aligning perturbations with important attention areas. Attention-SA (Li et al. 2024) designed a semantic-aware attention module to guide perturbations in attention areas. But existing methods target homogeneous models and focus on perturbing the attention areas, ignoring the potential of non-attention areas in improving cross-architecture transferability. As for meta-learning-based at-

tacks, MGAA (Yuan et al. 2021) leveraged meta-learning to simulate white-box and black-box attacks. LLTA (Fang et al. 2022) used meta-learning to train perturbations over augmented tasks, simulating cross-task attack adaptation. MTA (Qin et al. 2023) trained a meta-surrogate model to simulate adaptation across attack tasks. However, existing methods normally leverage meta-learning to reduce gradient discrepancies, ignoring the gradient diversity among heterogeneous ensemble models. *In summary, our NAMEA is the first attempt to decouple the gradients of ensemble non-attention from those of attention areas, while fusing gradients via meta-gradient optimization, which boosts cross-architecture transferability from a new perspective.*

Methodology

Preliminaries

I-FGSM-based Ensemble Attacks. Given a target model $f : X \to Y$ and a clean image $x \in X$ with ground-truth label $y \in Y$, an adversarial example is crafted as $x_{adv} = x+\delta$, which fools the target model $f(x_{adv}) \neq y$, where δ is a small perturbation constrained by l_∞ norm (Dong et al. 2018). The optimization problem can be formally formulated as:

$$\arg\max_{x_{adv}} \mathcal{L}(x_{adv}, y), s.t. \|x_{adv} - x\|_\infty \leq \epsilon, \tag{1}$$

where ϵ is the perturbation budget, and $\mathcal{L}$ is often the cross-entropy loss. Let T and α be the number of iterations and the step size, respectively. To solve the optimization problem in Eq. (1), I-FGSM (Kurakin, Goodfellow, and Bengio 2017) initializes the adversarial example with clean image, i.e., $x^0_{adv} = x$ and performs iterative updates as follows:

$$x^{t+1}_{adv} = \text{Clip}^x_\epsilon(x^t_{adv} + \alpha \,\text{sign}(g^{t+1})), \tag{2}$$

where $\text{Clip}^x_\epsilon(\cdot)$ denotes clipping the perturbation within an ϵ-ball centered at the original image x, $\text{sign}(\cdot)$ is the sign function, and $g^{t+1} = \nabla_{x^t_{adv}} \mathcal{L}(x^t_{adv}, y)$ denotes the gradient of the loss function with respect to x^t_{adv}. As the gradients of target models are inaccessible, ensemble attacks craft adversarial examples from multiple surrogate models $\{f_1, \ldots, f_N\}$, where the gradients can be calculated from ensemble predications, logits, or losses (Liu et al. 2017).

Attention Extraction. Given a surrogate model f_n and an image x with label y, we apply Grad-CAM (Selvaraju et al. 2017) to derive f_n's attention map on x, denoted by $\mathbf{H}_n(x)$. Let A^c_l denote the c-th feature map in the l-th layer of model f_n, and let $A^c_l[i, j]$ be the output of the neuron with the spatial position $[i, j]$. The importance weight of feature map A^c_l can be approximated with spatially pooled gradients:

$$\alpha^c_l = \frac{1}{Z} \sum_i \sum_j \frac{\partial f_n(x)[y]}{\partial A^c_l[i, j]}, \tag{3}$$

where Z is a normalizing constant such that $\alpha^c_l \in [-1, 1]$ and $f_n(x)[y]$ is the logits of label y when feeding model f_n with input x. Then, $\mathbf{H}^l_n(x)$, the attention map at the l-th layer of model f_n can be derived by performing ReLU on the weighted combination of feature maps:

$$\mathbf{H}^l_n(x) = \text{ReLU}\left(\sum_c \alpha^c_l \cdot A^c_l\right), \tag{4}$$

where $\text{ReLU}(\cdot)$ is applied to discard negative pixels in the attention map, while retaining the features that support label y. Therefore, the attention map highlights the spatial regions most relevant to model decision. As the size of the feature maps varies across different layers and models, $\mathbf{H}^l_n(x)$ will be upsampled back to the size of the original image using bilinear interpolation. In this paper, a single layer l is chosen for attention extraction, and thus $\mathbf{H}_n(x) = \mathbf{H}^l_n(x)$.

Meta-Gradient Optimization

Following previous work (Xiong et al. 2022; Tang et al. 2024), NAMEA treats the iterative ensemble attack as a stochastic gradient descent optimization process, which consists of T outer iterations and K inner loops as shown in the left side of Fig. 3. At a high-level view, each outer iteration t invokes K inner loops to calculate the optimal meta-training gradient g^K_{tr} and the optimal meta-testing gradient g^K_{te}, while using meta-learning to obtain the merged gradient g^{t+1}. Specifically, each inner loop k picks a random model f_n from N surrogate models to extract the (non-)attention areas and performs a one-step update, ensuring each surrogate model is selected at least once in every N consecutive inner iterations. From the right side of Fig. 3, we can see that our merged gradients fuse the characteristics of ViTs and CNNs, allowing for better model diversity than AdaEA and enabling more stable update direction than SMER.

Given N surrogates $\Theta = \{f_1, \ldots, f_N\}$ and the adversarial example x^t_{adv} at the t-th outer iteration, meta-gradient optimization calculates the merged gradient g^{t+1} as follows:

① Attention Meta-Training. This step aims to find the optimal meta-training gradient g^K_{tr} based on the attention areas of selected surrogate models by running K inner loops. The meta-training adversarial example is initialized as $x^0_{tr} = x^t_{adv}$, and the meta-training gradient g^{k+1}_{tr} and adversarial example x^{k+1}_{tr} can be iteratively calculated as follows:

$$g^{k+1}_{tr} = \nabla_{x^k_{tr}} \mathcal{L}(x^k_{tr}, y), \tag{5}$$

$$x^{k+1}_{tr} = \text{Clip}^x_\epsilon\left(x^k_{tr} + \alpha \,\text{sign}(g^{k+1}_{tr})\right), \tag{6}$$

where $\mathcal{L}(x^k_{tr}, y) = -\mathbf{1}_y \cdot \log(\text{softmax}(l_n(x^k_{tr})))$ with $\mathbf{1}_y$ being the one-hot encoding of ground-truth label y, and l_n the logits of the surrogate model f_n selected at inner loop k.

② Non-Attention Meta-Testing. This step aims to find the optimal meta-testing gradient g^K_{te} based on the non-attention areas of selected surrogate models by running K inner loops. This step initializes the meta-testing adversarial example as $x^0_{te} = x^t_{adv}$. The main trick is to design a non-attention extraction (NAE) module which masks the selected models' attention areas on the meta-testing adversarial examples before gradient calculation. At the k-th inner loop, given the adversarial examples x^k_{tr} and x^k_{te} in meta-training and meta-testing, respectively, the NAE module first generates an attention mask $\mathbb{M}_k$ for the surrogate model f_n selected at the k-th inner iteration as follows:

$$\mathbb{M}_k[i, j] = \begin{cases} 1, \text{ if } \mathbf{H}_n(x^k_{tr})[i, j] >= \eta, \\ 0, \text{ otherwise}, \end{cases} \tag{7}$$

where $\mathbf{H}_n(x^k_{tr})$ is model f_n's attention map on x^k_{tr} calculated from Eq. (3)-Eq. (4), $\mathbf{H}_n(\cdot)[i, j]$ is the attention value

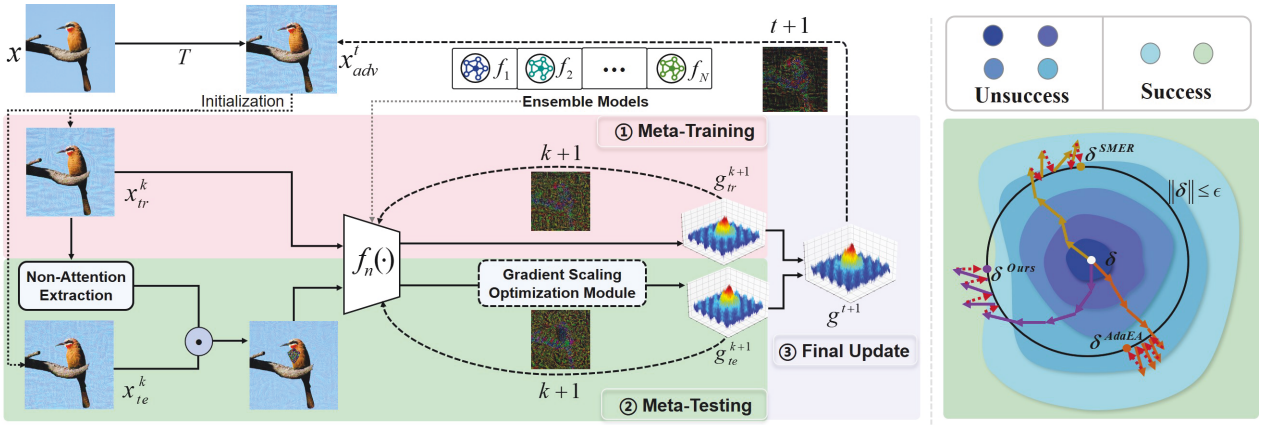

Figure 3: Overview of NAMEA. **Left:** Meta-gradient optimization process. Attention meta-training updates the gradient g_{tr}^{k+1} based on model's attention areas; Non-attention meta-testing updates the gradient g_{te}^{k+1} based on model's non-attention areas; Final update merges the gradients from meta-training and meta-testing steps to obtain the final gradient g^{t+1}. **Right:** The comparison of perturbation search process. NAMEA can quickly find the optimal direction, avoiding falling into local optimality.

at the spatial position $[i, j]$, and η is a threshold value determining if the pixel at position (i, j) of an image is important.

Let $\overline{\mathbb{M}}_k = \mathbf{1} - \mathbb{M}_k$ be the non-attention mask. The NAE module masks model f_n's attention area on the meta-testing adversarial example x_{te}^k with random Gaussian noises:

$$x_{te}^k = \overline{\mathbb{M}}_k \odot x_{te}^k + \mathbb{M}_k \odot \xi, \quad \xi \sim \mathcal{N}(0, 1), \tag{8}$$

where $\odot$ is the Hadamard product. The reason we fill random noises in the attention areas is to further distract the attention of selected models. From the experiment results shown in the right side of Fig. 5, we can see that filling random noises yields higher attack success rates compared with simply setting pixel values to 0s or 1s.

After being processed by the NAE module, the meta-testing adversarial example x_{te}^k retains only the non-attention area of model f_n. Thus, the meta-testing gradient g_{te}^{k+1} and the adversarial example x_{te}^{k+1} can be calculated as:

$$g_{te}^{k+1} = \nabla_{x_{te}^k} \mathcal{L}(x_{te}^k, y), \tag{9}$$

$$x_{te}^{k+1} = \text{Clip}_{\epsilon}^{x}\left(x_{te}^k + \alpha \,\text{sign}(g_{te}^{k+1})\right). \tag{10}$$

③ Final Update. After obtaining the optimal gradients g_{tr}^K and g_{tr}^K from the meta-training and meta-testing steps separately, the final update step obtain the fused gradient as:

$$g^{t+1} = g_{tr}^K + g_{te}^K \odot \overline{\mathbb{M}}_K. \tag{11}$$

Then, the outer loop can update the adversarial sample with Eq. (2). Note that the meta-testing gradient g_{te}^K is masked before merging. This is to ensure the transferable gradient information of attention regions will not be interfered with.

Gradient Scaling Optimization Module

Recent studies (Huang et al. 2019; Zhu et al. 2024) have proven that the intermediate-layer features of CNNs are more transferable, and the relatively small gradients in back-propagation of ViTs have negative influence on transferability. Thus, we design the gradient scaling optimization (GSO) module to further optimize the meta-testing gradients.

Layer-wise Gradient Scaling for CNN. For CNNs, the GSO module uses a scaling function to enhance the gradient contribution of intermediate layers. Let L denote the total number of layers. The scaling factor of layer l ($l \in [L/3, 2L/3]$) is defined as:

$$\lambda(l) = \lambda_1 + \lambda_2 \cdot \left(\frac{L}{l}\right), \tag{12}$$

where λ_1 controls the baseline scaling intensity, and λ_2 determines the magnitude of enhancement for each layer. Therefore, the shallower the layer, the larger the value of scaling factor. In this way, we can magnify the meta-testing gradient at layer l with the scaling factor $\lambda(l)$:

$$g_{te}[l] = g_{te}[l] \cdot \lambda(l). \tag{13}$$

Channel-wise Gradient Scaling for ViT. For ViTs, the backpropagated gradient can be decomposed into C channels, $g_{te} = \{g_{te}[1], \cdots, g_{te}[C]\}$. Thus, the GSO module uses a scaling function to reduce the contribution of channels with low gradient magnitudes. Let ϕ and σ represent the mean and standard deviation of the absolute gradient magnitudes across the C channels, respectively. If the meta-testing gradient at channel c is smaller than the average value of C channels, we can shrink the gradient magnitude as:

$$g_{te}[c] = g_{te}[c] \cdot \tanh(|\frac{g_{te}[c] - \phi}{\sigma}|). \tag{14}$$

In the Appendix, Alg. I shows the overall procedure of NAMEA, and Fig. I shows the adversarial examples crafted by NAMEA can further distract models' attention compared to all competitors, visualizing the efficacy of NAMEA.

Experiments

The attack performance is assessed on 3 benchmarks against 9 ViTs, 8 CNNs, 6 hybrid models, 6 defense models, and 9 defense methods. For ImageNet dataset, we adopt 6

Base	Attack	ViTs										CNNs								
		ViT-B	PiT-B	CaiT-S	ViS	DeiT-B	TNT-S	LeViT	ConV	Swin-B	**Avg.**	RN50	RN152	DN201	DN169	VGG16	VGG19	WRN101	BiT50	**Avg.**
I-FGSM	Ens	16.0	10.7	25.0	17.2	26.8	28.4	17.9	30.8	9.9	20.3	22.7	13.0	30.3	34.7	35.5	33.6	22.6	28.2	27.6
	SVRE	13.1	11.5	21.9	19.2	23.2	28.2	19.3	23.9	10.1	18.9	29.0	16.2	34.8	39.5	42.1	28.9	26.0	32.5	32.4
	AdaEA	25.1	17.6	39.2	27.5	40.4	40.2	28.8	42.7	15.6	30.8	38.7	21.1	47.0	50.1	53.0	48.4	34.5	39.6	41.6
	CWA	27.8	10.6	41.5	16.7	49.9	46.7	21.1	48.8	11.7	30.5	12.9	6.9	20.8	22.6	34.3	32.1	15.2	25.5	21.3
	SMER	27.4	16.4	42.6	26.0	43.9	44.7	27.7	48.9	15.4	32.6	33.2	18.4	43.1	45.7	50.0	48.4	31.4	39.6	38.7
	CSA	27.5	17.8	42.1	27.3	43.0	48.6	30.4	43.7	16.0	32.9	36.6	20.4	49.7	50.2	51.9	51.0	36.2	42.3	42.3
	Ours	**43.0**	**25.5**	**61.2**	**38.0**	**63.0**	**61.2**	**42.9**	**63.6**	**21.8**	**46.7**	**46.2**	**26.4**	**55.8**	**58.5**	**64.4**	**60.7**	**43.8**	**52.1**	**51.0**
MI-FGSM	Ens	34.0	24.9	48.5	34.7	51.7	49.8	38.7	51.2	20.6	39.3	43.4	26.5	52.8	53.6	55.2	52.9	39.6	46.4	46.3
	SVRE	31.3	24.2	43.2	35.1	44.6	50.5	38.9	46.5	19.3	37.1	49.6	30.5	58.1	60.5	59.2	58.0	45.3	50.6	51.5
	AdaEA	41.2	25.5	56.3	38.8	59.4	55.8	41.4	58.7	21.7	44.3	49.0	29.2	56.2	59.9	59.5	57.8	43.7	52.2	47.6
	CWA	35.1	18.4	53.5	28.6	55.4	56.7	38.9	58.2	18.0	40.3	37.7	22.1	48.7	51.4	58.8	53.6	37.5	44.9	44.3
	SMER	45.4	26.8	61.2	40.2	63.0	61.8	47.5	64.9	25.1	48.4	51.0	31.5	59.8	61.0	66.0	61.7	47.9	55.1	54.3
	CSA	48.5	29.8	61.3	45.4	63.2	66.2	49.0	64.2	27.1	50.5	52.0	32.0	60.4	62.3	66.1	63.6	49.6	53.8	55.0
	Ours	**56.6**	**34.9**	**72.6**	**51.1**	**74.5**	**72.5**	**59.0**	**74.5**	**32.8**	**58.7**	**59.7**	**39.7**	**69.9**	**69.9**	**73.3**	**72.2**	**57.1**	**63.4**	**63.2**
DI-MI-FGSM	Ens	42.5	38.3	56.6	50.5	56.1	62.0	53.7	59.3	31.4	50.0	59.5	41.9	70.1	71.5	71.4	70.0	60.4	63.5	63.5
	SVRE	45.2	43.1	65.4	57.0	62.5	70.5	63.0	63.3	32.2	55.8	66.8	49.1	76.7	77.8	78.2	75.4	67.7	71.7	70.4
	AdaEA	47.7	36.6	67.2	52.6	66.2	69.3	56.0	66.4	30.8	54.8	60.5	42.1	69.3	72.4	72.8	70.9	58.5	64.9	63.9
	CWA	53.6	44.4	73.6	57.9	71.1	79.4	66.1	73.7	33.1	61.4	64.3	47.8	76.7	77.9	79.6	78.5	65.0	72.9	70.3
	SMER	66.9	57.2	81.9	70.6	82.0	85.4	75.7	83.2	46.0	72.1	75.3	59.2	85.0	85.7	84.0	82.7	75.5	80.2	78.5
	CSA	54.8	46.2	68.1	60.1	69.2	73.4	63.2	68.8	38.8	60.3	63.2	48.2	76.7	75.2	76.5	73.4	66.2	72.7	69.0
	Ours	**72.7**	**63.6**	**85.9**	**77.8**	**86.5**	**89.2**	**80.8**	**86.6**	**54.1**	**77.5**	**80.9**	**68.4**	**88.6**	**89.4**	**87.7**	**87.6**	**81.1**	**86.0**	**83.7**

Table 1: Comparison of ASRs (%) between NAMEA and baselines. For all the tables, the best results are highlighted in bold.

gradient-based basic attacks. Due to limited space, this section only presents the representative results on ImageNet dataset. In the Appendix, we will provide the experiment results on CIFAR-10 and CIFAR-100 datasets, comparison of computational and memory overheads, visualization of attack performance, and supplementary results on ImageNet in terms of transferability, robustness, and ablation studies.

Experiments Setup

Datasets and Models. ImageNet (Russakovsky et al. 2015), the benchmark dataset contains 1000 categories with about 1.2 million images. To align with previous work (Zhang et al. 2023; Wei et al. 2022), we randomly select one image from each class to form the test set. Following (Chen et al. 2023), we employ ViT-T(Dosovitskiy et al. 2020), DeiT-T (Touvron et al. 2021a), ResNet-18 (RN18) (He et al. 2016a), and Inception-v3 (Inc-v3) (Szegedy et al. 2016) as the surrogate models. The target models include different architectures: ① ViT models (Heo et al. 2021; Touvron et al. 2021b; Chen et al. 2021; Han et al. 2021; Graham et al. 2021; d'Ascoli et al. 2021; Liu et al. 2021): ViT-B, PiT-B, CaiT-S, Visformer-S (ViS), DeiT-B, TNT-S, LeViT, ConViT-B (ConV), and Swin-B. ② CNN models (Huang et al. 2017; Simonyan and Zisserman 2015; Zagoruyko and Komodakis 2016; Kolesnikov et al. 2020): RN50, RN152, DenseNet-201 (DN201), DN169, VGG16, VGG19, WideResNet-101 (WRN101), and BiT-M-R50×1 (BiT50). ③ Defense models (Szegedy et al. 2016; He et al. 2016b; Tramèr et al. 2018): Inc-v4, Inc-RN-v2 (IR-v2), Inc-v3-adv (Inc-v3$_{adv}$), Inc-v3-ens3 (Inc-v3$_{ens3}$), Inc-v3-ens4 (Inc-v3$_{ens4}$), and Inc-RN-v2-ens (IR-v2$_{ens}$).

Baselines and Metrics. We compare the attack success rate (ASR) with six ensemble attacks: Ens (Liu et al. 2017), SVRE (Xiong et al. 2022), AdaEA (Chen et al. 2023), CWA (Chen et al. 2024), SMER (Tang et al. 2024), and CSA (Li et al. 2025) under the same ensemble settings and perturbation budget $\epsilon = 8/255$. Moveover, we report the average results of 5 trials, with a deviation of less than 0.6%.

Attack	Defense Models						
	Inc-v4	IR-v2	Inc-v3$_{adv}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IR-v2$_{ens}$	**Avg.**
Ens	59.6	65.5	75.1	49.3	48.9	56.3	59.1
SVRE	67.6	69.6	78.1	53.0	53.5	59.5	63.5
AdaEA	59.1	64.3	73.5	43.2	43.5	54.1	56.6
CWA	70.9	71.2	78.2	50.7	55.1	59.2	64.2
SMER	75.2	74.0	80.0	60.8	60.4	64.3	69.1
CSA	64.2	68.1	77.8	50.7	52.9	59.2	62.1
Ours	**80.9**	**79.0**	**83.7**	**66.2**	**66.3**	**68.9**	**74.2**

Table 2: Comparison of ASRs (%) against 6 defense models.

Parameters Settings. For the baselines and our NAMEA, we use I-FGSM, MI-FGSM (Dong et al. 2018), and DI-MI-FGSM (Xie et al. 2019) as the basic attacks. The hyperparameters in the baselines follow the optimal setting in the respective literature. For a fair comparison, CSA employs 7 checking points from each surrogate model, expanding the ensemble scale to 28 models. We set the number of outer iterations as $T = 10$ and the number of internal loops as $K = 16$, using step size $\alpha = 0.8/255$ and momentum decay $\mu = 1.0$. Besides, ViTs use the output of the pre-activation normalization layer in final self-attention blocks, RN18 uses the output of the last convolutional block in final residual stage, and Inc-v3 uses the output of $Mixed\_7b$ to extract attention areas, where the attention threshold is set to $\eta = 0.6$.

Main Results

Cross-Architecture Transferability. From Table 1, we can see that our NAMEA achieves superior adversarial transferability, always performing best when combining with different base attacks. While CSA achieves decent performance, it incurs huge time and memory costs to train models and save checking points, as shown in Table III of the Appendix.

Attack	Defense Methods									
	R&P	HGD	NIPS-r3	JPEG	RS	NPR	FD	Bit-RD	DiffPure	**Avg.**
Ens	57.4	43.7	58.9	52.4	19.2	21.6	50.8	51.6	26.0	42.4
SVRE	63.5	51.5	66.1	58.6	19.4	22.1	55.7	58.3	26.3	46.8
AdaEA	56.0	40.1	55.1	53.1	16.8	17.6	50.5	54.1	22.2	40.6
CWA	65.9	48.0	64.1	62.5	20.3	18.9	59.5	60.8	26.4	47.4
SMER	75.5	61.9	72.4	71.3	24.1	27.0	68.3	71.0	39.9	56.8
CSA	63.8	51.4	63.5	60.2	21.8	26.2	58.0	60.3	34.9	48.9
Ours	**80.0**	**71.4**	**78.5**	**77.6**	**29.1**	**31.6**	**74.7**	**76.9**	**50.3**	**63.3**

Table 3: Comparison of ASRs (%) against defense methods.

APIs	I-FGSM					DI-MI-FGSM				
	AdaEA	CWA	SMER	CSA	**Ours**	AdaEA	CWA	SMER	CSA	**Ours**
Google	23	22	24	24	**30**	43	47	52	46	**55**
Alibaba	21	20	23	19	**26**	39	44	48	43	**53**
Baidu	29	28	33	32	**37**	53	58	61	56	**64**

Table 4: Comparing of ASRs (%) against real-world models.

But even CSA expands surrogate model scale with various weights, our NAMEA still works better. And we can observe that among all the base attacks, NAMEA and all baselines work best under DI-MI-FGSM, followed by MI-FGSM, and finally I-FGSM. The performance gain of DI-MI-FGSM can be attributed to the input diversity that allows to better capture the universal adversarial information. In particular, SMER shows surging attack effects under DI-MI-FGSM, because ensemble reweighing makes full use of model diversity when working together with input diversity. However, even under DI-MI-FGSM, NAMEA promotes the average ASR by 5.3% compared to SMER. From the supplementary results shown in the Appendix, we can observe that NAMEA also consistently achieves the best performance under benchmark datasets CIFAR-10 and CIFAR-100 (Table I-II), additional base attacks (Table IV), hybrid target models (Table V), different perturbation budgets (Table VI), and more surrogate models (Table VII). The above results fully validate our hypothesis that ensemble non-attention contributes to improve cross-architecture transferability.

Robustness of Adversarial Examples. We compare the attack performance of NAMEA and baselines against various defense models and defense methods under base attack DI-MI-FGSM. Table 2 shows that even for adversarially trained models, our NAMEA consistently achieves the best transferability among all competitors. Besides, we evaluate the ASRs of NAMEA and baselines against 9 defense methods: R&P (Xie et al. 2018), HGD (Liao et al. 2018), NIPS-r3 (Thomas 2017), JPEG (Guo et al. 2018), RS (Cohen, Rosenfeld, and Kolter 2019), NRP (Naseer et al. 2020), FD (Liu et al. 2019), Bit-RD (Xu, Evans, and Qi 2018) and DiffPure (Nie et al. 2022). The results of Table 3 are basically consistent with those in Table 2. Even for the powerful diffusion-model-based defense DiffPure, our NAMEA outperforms baselines by 10%, indicating that NAMEA generates highly robust adversarial examples. The supplementary results in Table VIII of the Appendix demonstrate that NAMEA also has the highest robustness among all competitors under base attacks I-FGSM and MI-FGSM.

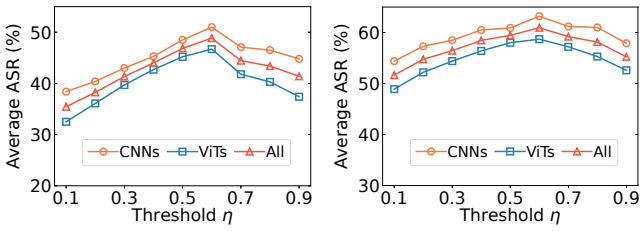

Figure 4: Average ASRs (%) of NAMEA under varying threshold. Base: I-FGSM (**Left**) and MI-FGSM (**Right**).

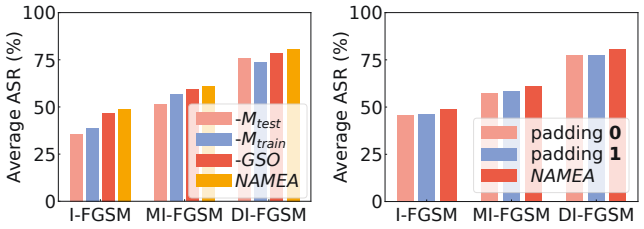

Figure 5: **Left:** Ablation study on meta-learning and GSO. **Right:** Ablation study on padding values in masked areas.

Real-World Attacks. We locally run NAMEA and five baselines, and then take the resulting adversarial examples as the inputs of authoritative image recognition APIs, i.e., Google Vision, Alibaba Cloud, and Baidu Cloud, for inference. Following (Fang et al. 2022), we consider an attack successful if the ground-truth label of a clean sample is not present in the top-5 list of the APIs' predictions. To reduce deviation, we randomly select 100 adversarial examples generated by each attack for testing. From Table 4, we can see that in real-world scenarios, NAMEA always performs best among all competitors. When we relax the success condition to top-1 list, our average ASRs under DI-MI-FGSM are 60%, 56%, and 68% for Google, Alibaba, and Baidu APIs, respectively, which are 4%, 3%, and 5% higher than the best-performing baseline SMER. Hence, NAMEA has superior transferability in various black-box scenarios.

Ablation Studies

Unless otherwise specified, the ablation experiments are assessed by the average ASRs against 9 ViTs and 8 CNNs.

Threshold η. According to Eq. (7), the smaller the value of η, the less the number of 0s in $\mathbb{M}$, thus the less non-attention areas being extracted. If the value of η is too small, the substantial semantic features of non-attention areas may be lost. Hence, we need to adjust the value to retain basic semantics for effective exploration of non-attentive areas. From Fig. 4, we can see that when $\eta = 0.6$, NAMEA achieves the optimal result, and the ASRs against CNNs and ViTs show a declining trend as η decreases or increases. This means that the attack effect on CNNs and ViTs is sensitive to η. We also investigate the impact of hyperparameters on CNN gradient scaling (Eq. (12)) in Fig. IV of the Appendix.

Meta-Learning Steps and GSO Module. Let $-M_{train}$, $-M_{test}$, and $-GSO$ denote removing the meta-training step, meta-testing step, and GSO module from NAMEA, re-

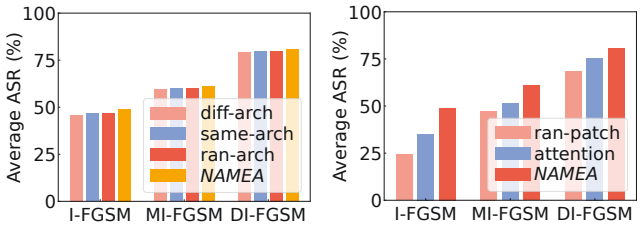

Figure 6: **Left:** Ablation study on varying meta-testing models. **Right:** Ablation study on varying meta-testing areas.

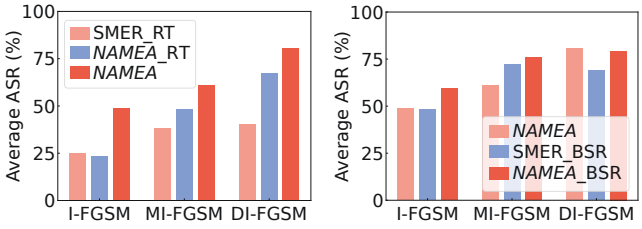

Figure 7: Average ASRs (%) of four comparison settings, which have two adversarial examples in each inner loop.

spectively. In $-M_{train}$ and $-M_{test}$, the final meta gradients are calculated as $g^{t+1} = g_{te}^K \odot \overline{\mathbb{M}}_K$ and $g^{t+1} = g_{tr}^K$, respectively. As shown in the left side of Fig. 5, if we discard the meta-testing step, there is a drop of 9.2% in the average ASR; if we discard the meta-training step, there is a drop of 7.4% in the average ASR; if we remove the GSO module, the average ASR slightly decreases (a drop of 2%). Hence we know that the GSO module has some positive effect on ASRs, and meta-learning is crucial for enhancing the performance of NAMEA. In the Appendix, Fig. II shows that the gradients of both attention and non-attention areas help craft perturbations generalizing across CNNs and ViTs, demonstrating the merged gradients fuse the transfer information of CNNs and ViTs; Fig. III shows that NAMEA achieves the best attack performance compared with different gradient aggregation strategies and different gradient weights, validating the effectiveness of meta-gradient optimization.

Padding Values. According to Eq. (8), NAMEA fills the masked areas with random noises. To evaluate the impacts of different padding values on attack effect, we evaluate the ASRs under the other two kinds of padding values: full 0s and full 1s. As shown in the right side of Fig. 5, random noises achieve the highest ASRs. For instance, when using I-FGSM as the base attack, filling random noises outperform the other two filling methods by approximately 2.7% in average ASR. This improvement may stem from the stronger disruption of models' attention caused by random noises.

Model Selection Strategies. Meta-testing uses the same surrogate model as meta-training. To test the impact of different model selection strategies on attack effect, we design the ablation settings: **diff-arch** selects a random model with different architecture; **same-arch** selects a random model with the same architecture; **ran-arch** randomly selects a model. As shown in the left side of Fig. 6, the average ASRs are almost unaffected by varying selection strategies. This is because the surrogate models in the inner loop are randomly chosen, ensuring sufficient exploration of non-attention areas of ensemble models. In the Appendix, we also provide the ablation study on different ensemble settings, and Table IX shows that NAMEA always performs best under varying ratios of CNNs to ViTs and varying ensemble scales.

Extracted Areas. Meta-testing extracts the non-attention areas from adversarial examples before gradient calculation. To validate the critical role of non-attention areas, we design the ablation settings: **attention** extracts attention areas, i.e., $x_{te}^k = \mathbb{M}_k \odot x_{te}^k$; **ran-patch** extracts random patches of size (56×56). In both settings, the final update merges the gradient as $g^{t+1} = g_{tr}^K + g_{te}^K$. From the right side of Fig. 6, we can see that non-attention areas achieve the highest ASRs, largely surpassing all the other setting. This is because non-attention areas together with attention areas can make the best of the transferable information of individual models.

Discussion

The Impact of Ensemble Non-Attention. NAMEA derives two adversarial examples, x_{tr}^k and x_{te}^k at each inner loop k, which may create the illusion that the performance gain is due to diverse inputs. To further verify the role of ensemble non-attention, we design four comparison settings with the same number of copies in each inner loop: **NAMEA_RT** replaces the NAE module with BSR (Wang et al. 2024), which randomly transforms x_{te}^k before gradient calculation, while merging the gradient as $g^{t+1} = g_{tr}^K + g_{te}^K$; **SMER_RT** applies BSR to generate diverse copies for each inner loop of SMER and updates with the average gradient. **NAMEA_BSR** and **SMER_BSR** directly combine BSR with NAMEA and SMER, respectively. From the left-side of Fig. 7, we can see that NAMEA always perform best. This is because the gradient update directions of random transformed inputs are diverse, and merging them directly will cause gradient conflicts. But the gradients of non-attention areas serve as a supplement to those of attention areas, helping to stabilize update direction and improve model diversity. From the right-side of Fig. 7, we can see that NAMEA can fully leverage both input and model diversities, thus yielding superior performance. We also observe that in DI-MI-FGSM, the ASR of **SMER_BSR** drops dramatically and that of **NAMEA_BSR** slightly declines. This may because the combination of two input transformation methods causes the inputs change too much, loosing substantial semantic features. But **NAMEA_BSR** with the help of non-attention areas enables more stable update direction. Thus, we confirm that *ensemble non-attention boosts adversarial transferability in a new angle different from input diversity*.

Conclusion

This work is the first to explore the power of ensemble non-attention in improving cross-architecture transferability. We propose a novel ensemble attack, NAMEA, which integrates ensemble non-attention and meta learning to ensure stable update direction and model diversity at once. Experiment results show that NAMEA largely surpasses the SOTA approaches, proving the validity of ensemble non-attention.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants No. 62272150, No. 62222204, the Sichuan Science and Technology Program under Grants No. 2024ZDZX0011, No. 2025ZNSFSC1472, and the Postdoctoral Fellowship Program of CPSF under Grant No. GZC20251074.

References

Chen, B.; Yin, J.; Chen, S.; Chen, B.; and Liu, X. 2023. An adaptive model ensemble adversarial attack for boosting adversarial transferability. In *Proc. of ICCV*, 4489–4498.

Chen, H.; Zhang, Y.; Dong, Y.; Yang, X.; Su, H.; and Zhu, J. 2024. Rethinking Model Ensemble in Transfer-based Adversarial Attacks. In *Proc. of ICLR*.

Chen, S.; He, Z.; Sun, C.; Yang, J.; and Huang, X. 2020. Universal adversarial attack on attention and the resulting dataset damagenet. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4): 2188–2197.

Chen, Z.; Xie, L.; Niu, J.; Liu, X.; Wei, L.; and Tian, Q. 2021. Visformer: The vision-friendly transformer. In *Proc. of ICCV*, 589–598.

Cohen, J.; Rosenfeld, E.; and Kolter, Z. 2019. Certified adversarial robustness via randomized smoothing. In *Proc. of ICML*, 1310–1320.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proc. of CVPR*, 9185–9193.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proc. of ICLR*.

d'Ascoli, S.; Touvron, H.; Leavitt, M. L.; Morcos, A. S.; Biroli, G.; and Sagun, L. 2021. Convit: Improving vision transformers with soft convolutional inductive biases. In *Proc. of ICML*, 2286–2296.

Fang, S.; Li, J.; Lin, X.; and Ji, R. 2022. Learning to learn transferable attack. In *Proc. of AAAI*, volume 36, 571–579.

Graham, B.; El-Nouby, A.; Touvron, H.; Stock, P.; Joulin, A.; Jégou, H.; and Douze, M. 2021. Levit: a vision transformer in convnet's clothing for faster inference. In *Proc. of ICCV*, 12259–12269.

Guo, C.; Rana, M.; Cisse, M.; and van der Maaten, L. 2018. Countering Adversarial Images using Input Transformations. In *Proc. of ICLR*.

Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; and Wang, Y. 2021. Transformer in transformer. *Proc. of NeurIPS*, 34: 15908–15919.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *Proc. of CVPR*, 770–778.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Deep residual learning for image recognition. In *Proc. of CVPR*, 397–406.

Heo, B.; Yun, S.; Han, D.; Chun, S.; Choe, J.; and Oh, S. J. 2021. Rethinking spatial dimensions of vision transformers. In *Proc. of ICCV*, 11936–11945.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proc. of CVPR*, 4700–4708.

Huang, Q.; Katsman, I.; He, H.; Gu, Z.; Belongie, S.; and Lim, S.-N. 2019. Enhancing adversarial example transferability with an intermediate level attack. In *Proc. of ICCV*, 4733–4742.

Kolesnikov, A.; Beyer, L.; Zhai, X.; Puigcerver, J.; Yung, J.; Gelly, S.; and Houlsby, N. 2020. Big transfer (bit): General visual representation learning. In *Proc. of ECCV*, 491–507.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017. Adversarial Machine Learning at Scale. In *Proc. of ICLR*.

Li, Q.; Hu, Q.; Fan, H.; Lin, C.; Shen, C.; and Wu, L. 2024. Attention-SA: Exploiting Model-Approximated Data Semantics for Adversarial Attack. *IEEE Transactions on Information Forensics and Security*, 19: 8673–8684.

Li, S.; He, C.; Ma, X.; Zhu, B. B.; Wang, S.; Hu, H.; Zhang, D.; and Yu, L. 2025. Enhancing Adversarial Transferability with Checkpoints of a Single Model's Training. In *Proc. of the CVPR*, 20685–20694.

Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Hu, X.; and Zhu, J. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In *Proc. of CVPR*, 1778–1787.

Lin, J.; Song, C.; He, K.; Wang, L.; and Hopcroft, J. E. 2019. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *Proc. of ICLR*.

Liu, Y.; Chen, X.; Liu, C.; and Song, D. 2017. Delving into Transferable Adversarial Examples and Black-box Attacks. In *Proc. of ICLR*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. of ICCV*, 10012–10022.

Liu, Z.; Liu, Q.; Liu, T.; Xu, N.; Lin, X.; Wang, Y.; and Wen, W. 2019. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *Proc. of CVPR*, 860–868.

Naseer, M.; Khan, S.; Hayat, M.; Khan, F. S.; and Porikli, F. 2020. A self-supervised approach for adversarial robustness. In *Proc. of CVPR*, 262–271.

Nie, W.; Guo, B.; Huang, Y.; Xiao, C.; Vahdat, A.; and Anandkumar, A. 2022. Diffusion Models for Adversarial Purification. In *Proc. of ICML*, 16805–16827.

Qin, Y.; Xiong, Y.; Yi, J.; and Hsieh, C.-J. 2023. Training meta-surrogate model for transferable adversarial attack. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 9516–9524.

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

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proc. of ICCV*, 618–626.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *Proc. of ICLR*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proc. of CVPR*, 2818–2826.

Tang, B.; Wang, Z.; Bin, Y.; Dou, Q.; Yang, Y.; and Shen, H. T. 2024. Ensemble Diversity Facilitates Adversarial Transferability. In *Proc. of CVPR*, 24377–24386.

Thomas, A. 2017. NIPS: Defense Against Adversarial Attack. https://github.com/anlthms/nips-2017/tree/master/mmd.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021a. Training data-efficient image transformers & distillation through attention. In *Proc. of ICML*, 10347–10357.

Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; and Jégou, H. 2021b. Going deeper with image transformers. In *Proc. of ICCV*, 32–42.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2018. Ensemble Adversarial Training: Attacks and Defenses. In *Proc. of ICLR*.

Wang, K.; He, X.; Wang, W.; and Wang, X. 2024. Boosting adversarial transferability by block shuffle and rotation. In *Proc. of CVPR*, 24336–24346.

Wang, Y.; Wang, J.; Yin, Z.; Gong, R.; Wang, J.; Liu, A.; and Liu, X. 2022. Generating transferable adversarial examples against vision transformers. In *Proc. of ACM MM*, 5181–5190.

Wei, Z.; Chen, J.; Goldblum, M.; Wu, Z.; Goldstein, T.; and Jiang, Y.-G. 2022. Towards transferable adversarial attacks on vision transformers. In *Proc. of AAAI*, 2668–2676.

Wu, W.; Su, Y.; Chen, X.; Zhao, S.; King, I.; Lyu, M. R.; and Tai, Y.-W. 2020. Boosting the transferability of adversarial samples via attention. In *Proc. of CVPR*, 1161–1170.

Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; and Yuille, A. 2018. Mitigating Adversarial Effects Through Randomization. In *Proc. of ICLR*.

Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; and Yuille, A. L. 2019. Improving transferability of adversarial examples with input diversity. In *Proc. of CVPR*, 2730–2739.

Xiong, Y.; Lin, J.; Zhang, M.; Hopcroft, J. E.; and He, K. 2022. Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability. In *Proc. of CVPR*, 14983–14992.

Xu, W.; Evans, D.; and Qi, Y. 2018. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *Proc. of NDSS*.

Yuan, Z.; Zhang, J.; Jia, Y.; Tan, C.; Xue, T.; and Shan, S. 2021. Meta gradient adversarial attack. In *Proc. of the ICCV*, 7748–7757.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *Proc. of BMVC*.

Zhang, J.; Huang, Y.; Wu, W.; and Lyu, M. R. 2023. Transferable Adversarial Attacks on Vision Transformers with Token Gradient Regularization. In *Proc. of CVPR*, 16415–16424.

Zhu, Z.; Wang, X.; Jin, Z.; Zhang, J.; and Chen, H. 2024. Enhancing transferable adversarial attacks on vision transformers through gradient normalization scaling and high-frequency adaptation. In *Proc. of ICLR*.