Improving Adversarial Transferability with Local Perturbation Augmentation

Jian-Xun Mi[1,2,3*] Xuanhui Zhong[1,2,3] Weisheng Li[1,2,3]
[1]Key Laboratory of Big Data Intelligent Computing, [2]Chongqing Key Laboratory of Image Cognition
[3]School of Computer Science and Technology,
Chongqing University of Posts and Telecommunications, China

Abstract

Adversarial examples expose fundamental vulnerabilities within deep neural networks, and their transferability highlights shared weaknesses across diverse models. Existing mainstream attack methods often rely on iterative processes with various strategies to improve transferability, but the limited knowledge of the target model restricts the success of these approaches. In this paper, we reveal that the iterative optimization process tends to over-specialize adversarial perturbations to the local gradient characteristics of the surrogate model, thereby hindering their transferability to other models. To address this limitation, we propose a novel attack method called Local Perturbation Augmentation Attack (LPAA). The key innovation of our approach lies in constructing multiple augmented local subspaces during each iteration, which steers perturbation updates towards a more generalizable direction, reducing over-reliance on the surrogate model. Additionally, to improve the initial performance and overcome sensitivity to initial perturbation, we introduce a dedicated perturbation initialization strategy that ensures the optimization process starts from a direction with greater transferability. Compared with existing random neighborhood sampling strategies, LPAA serves as an effective approach that leverages the characteristics of perturbations to overcome their limitations. Extensive experiments on CNNs and ViTs demonstrate that LPAA consistently generates highly transferable adversarial examples, significantly surpassing the performance of state-of-the-art methods.

1. Introduction

Deep neural networks (DNNs) have achieved remarkable performance across many tasks [9, 16, 29] but remain vulnerable to adversarial examples, where small, human-imperceptible perturbations can lead to incorrect predictions [14, 19]. This vulnerability poses serious risks in safety-critical applications such as autonomous driving [5] and facial recognition [2, 53], making it essential to study adversarial examples to enhance the reliability and robustness of DNNs.

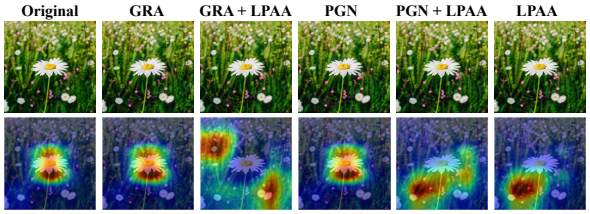


Figure 1. Attention maps [32] on DenseNet-121 [18] for clean images and adversarial examples produced by GRA [52], PGN [13], their combinations with LPAA (GRA + LPAA, PGN + LPAA), and LPAA alone. The adversarial examples were generated on ResNet-50 [16]. Despite their high visual similarity, only GRA + LPAA, PGN + LPAA, and LPAA cause misclassification.

Adversarial example generation methods can be categorized into white-box and black-box attacks based on the adversary's knowledge of the target model. In the white-box setting, the adversary has full access to the model's architecture, parameters, and training data, making it easier to craft adversarial examples [4, 19]. In contrast, in the black-box setting, the adversary has only limited knowledge, making the attack more challenging. Black-box attacks are typically divided into query-based and transfer-based approaches. Query-based attacks [1, 3] require numerous queries to the target model, whereas transfer-based attacks [7, 10, 13] generate adversarial examples on a surrogate model and transfer them to the target. Since extensive querying is often infeasible in practice, transfer-based attacks have received greater attention.

Much research has aimed to improve the transferability of adversarial examples. The iterative version of FGSM [14], I-FGSM [19], employs multi-step loss maximization. Although this enhances the performance in the white-box setting, it reduces transferability, indicating that naive iterative optimization can overfit the surrogate model [7].

*Corresponding author. Email: mijianxun@gmail.com

To mitigate this, later works incorporate auxiliary strategies, such as stabilizing update directions with neighborhood information [39, 52], flattening the loss landscape [11, 13], applying input transformations for diverse gradients [22, 46], or ensembling multiple models [48]. However, these approaches still optimize in the global perturbation space and are constrained by the surrogate model's gradients, making it difficult to escape local optima and further improve transferability.

In this paper, we propose the Local Perturbation Augmentation Attack (LPAA) to mitigate the overfitting of adversarial examples to the surrogate model. Instead of relying on global perturbations, our approach constructs multiple subspaces via local perturbations. This design prevents the optimization from being trapped in local optima caused by following a single gradient path in the global space. To facilitate effective exploration within these subspaces and to escape local optima, we introduce an augmentation coefficient to dynamically expand the perturbation search range. Furthermore, to enhance initial performance and alleviate sensitivity to the initial perturbation, we incorporate a dedicated perturbation initialization strategy, which supplies a highly transferable starting direction for the subsequent attack process. Compared with existing random neighborhood sampling strategies [13, 52], LPAA can be viewed as a perturbation-based neighborhood sampling strategy. Its distinctive advantage lies in its ability to strategically leverage both the direction and magnitude of perturbations, as opposed to existing methods that often rely on random neighborhood sampling. An example demonstrating the misclassification capability of LPAA is shown in Figure 1. Extensive experiments on both CNNs and ViTs demonstrate the high transferability of adversarial examples generated by LPAA, where it consistently outperforms current state-of-the-art methods.

In summary, our contributions are as follows:

- We demonstrate that naive iterative optimization of perturbations may result in overfitting to the surrogate model. To address this issue, we propose a local perturbation augmentation strategy that leverages multiple augmented subspaces of the perturbation to mitigate overfitting and enhance transferability.
- We propose a novel perturbation initialization strategy to enhance initial performance and mitigate the sensitivity of iterative updates to initial perturbations. This strategy provides a highly transferable starting direction, thereby improving the overall transferability of adversarial attacks.
- We propose the Local Perturbation Augmentation Attack (LPAA) framework. Extensive experiments demonstrate that LPAA significantly improves transferability compared to existing baselines. Moreover, it is compatible with input transformation-based methods, further improving transferability.
- LPAA can be regarded as a perturbation-based neighborhood sampling strategy that leverages both the directional and magnitude characteristics of perturbations, leading to notable improvements over existing attack methods.

2. Related Work

2.1. Gradient-based Attack

The transfer phenomenon of adversarial examples was first confirmed by FGSM [14], which utilizes a single-step perturbation along the direction that maximizes the loss function. Its iterative version, I-FGSM [19], significantly enhances white-box attack capability. However, by greedily iterating toward the direction that maximizes the loss function, it falls into the local optimum of the surrogate model, resulting in lower transferability. Subsequent work revealed the similarity between model training and optimizing adversarial examples. MI-FGSM [7] introduced momentum from model training to stabilize the perturbation update direction, and momentum has since become a fundamental component of a series of existing methods. NI-FGSM [22] further improves the transferability of adversarial examples by incorporating Nesterov's accelerated gradient. VMI-FGSM [39] reduces gradient variance, while GRA [52] utilizes the gradients of neighboring sample points to correct the current update direction and employs a decay indicator to reduce step size, further stabilizing the update direction. PGN [13] seeks flatter local optima, effectively enhancing transferability. ANDA [10] leverages the asymptotic normality property of stochastic gradient ascent to characterize perturbations sampled from a learned distribution. MIG [26] uses integrated gradients to steer the generation of adversarial perturbations. MuMoDIG [30] improves on MIG by refining the integration path through multiplicity, monotonicity, and diversity, significantly boosting adversarial transferability.

2.2. Input Transformation-based Attack

Similar to data augmentation in model training, transforming inputs during adversarial optimization has proven to be an effective strategy. In the process of generating adversarial examples, DIM [46] utilizes random resizing and padding on the input to obtain gradients for the update. TIM [8] leverages CNNs' translation invariance by convolving image gradients with predefined kernels to simulate translated gradients. SIM [22] computes the average of gradients obtained by scaling the input image to multiple different scales. Differing from spatial domain transformations, SSA [25] transforms images in the frequency domain. Admix [40] blends the original input with randomly sampled images to generate adversarial examples. SIA [41] enhances adversarial transferability by dividing an image into local

blocks and applying random transformations to each block independently.

2.3. Others

Model-related methods enhance adversarial transferability by altering network architecture or computation. LinBP [15] improves transferability by omitting ReLU layers during backpropagation to increase CNNs' linearity, while BPA [45] modifies backward computations to address gradient truncation. SGM [43] emphasizes gradients from skip connections. For ensemble attacks, Ghost [20] uses ghost networks and longitudinal ensembles for gradient diversity with low overhead, and SVRE [48] reduces gradient variance for better ensemble direction. Methods like ATA [44], FIA [42], and MFAA [51] redesign loss functions based on intermediate model layers.

3. Methodology

3.1. Preliminary

Given a clean image x with dimensions $C \times H \times W$ and its ground-truth label y, as well as a model f_θ parameterized by θ. In the white-box setting, adversarial attacks aim to generate an adversarial example $x_{\text{adv}} = x + \delta$ such that the model's prediction deviates from the true label, i.e., $f_\theta(x + \delta) \neq y$. Meanwhile, to ensure the imperceptibility of the perturbation, the constraint $\|\delta\|_\infty \leq \epsilon$ must be satisfied, where ϵ represents the perturbation budget. Taking I-FGSM [19] as an example, this objective is typically achieved by iterative optimization to maximize the loss function, which can be formulated as:

$$\delta_t = \delta_{t-1} + \alpha \cdot \text{sign}\left(\nabla_x \mathcal{L}(f_\theta(x + \delta_{t-1}), y)\right) \quad (1)$$

where $\mathcal{L}(\cdot, \cdot)$ denotes the cross-entropy loss, $\alpha = \epsilon/T$ is the step size of each iteration, T is the total number of iterations, and $\delta_0 = 0$. In contrast, the goal of a transfer-based attack is to utilize adversarial examples generated on a surrogate model to mislead one or more unknown target models. This objective can be expressed as:

$$\arg\max_{\delta} \mathbb{E}_{f_{\theta_i} \in \mathcal{F}} \left[\mathcal{L}(f_{\theta_i}(x + \delta), y)\right] \quad (2)$$

where f_{θ_i} denotes a target model, and $\mathcal{F}$ represents the set of target models. However, due to the structural and parametric differences among models, adversarial examples that perform well on the surrogate model often fail to achieve comparable effects on target models. This is because the adversarial perturbation is updated continuously along the direction of the maximum loss, which tends to fall into sharp (high-curvature) regions of the surrogate model's loss landscape [7, 13]. As a result, the perturbation overfits the surrogate model, thereby degrading the transferability of adversarial examples.

3.2. Local Perturbation Augmentation

Let $g(x+\delta_{t-1}) = \nabla_x \mathcal{L}(f_\theta(x+\delta_{t-1}), y)$. Each perturbation update can then be written as $\delta_t = \delta_{t-1} + \alpha \cdot u_{t-1}$, where $u_{t-1} = \text{sign}(g(x + \delta_{t-1}))$. Thus, each update step depends on the gradient of the surrogate model evaluated at $x+\delta_{t-1}$. As the iteration count t increases, the final perturbation can be expressed as a linear combination of all encountered gradient directions, i.e., $\delta_T = \sum_{t=1}^{T} \alpha \cdot u_{t-1}$. Consequently, iterative updates based on the full perturbation vector tend to accumulate curvature-related information that is highly specific to the surrogate model. This model-specific adaptation reduces the transferability of the resulting perturbation across different models. To mitigate this overfitting effect, we introduce stochastic masking during perturbation optimization. The random mask constrains each update to rely on a randomly sampled subspace, regularizing the optimization trajectory and encouraging exploration of more diverse directions in the input space. Therefore, the accumulation process of the perturbation can be expressed as:

$$\delta_t = \delta_{t-1} + \alpha \cdot u_{t-1}, u_{t-1} = \text{sign}(g(x + \delta_{t-1} \odot M_{t-1})) \quad (3)$$

The binary mask M_{t-1} has the same dimensions as δ_{t-1}. Each element is 0 with probability p and 1 otherwise. The masking ratio p controls the proportion of masked perturbation, thereby adjusting the effective dimensionality of the optimization subspace. However, gradients computed within a single subspace capture only limited, local information about the loss landscape. To obtain more comprehensive and stable gradient estimates, we aggregate information from multiple randomly sampled subspaces. As illustrated in Figure 2, the aggregation strategy samples multiple masks to cover different dimensions, resulting in a more comprehensive gradient update direction. Accordingly, we employ the averaged gradient over N independently sampled subspaces:

$$\bar{g} = \frac{1}{N} \sum_{i=1}^{N} g_{i-1}(x + \delta_{t-1} \odot M_{i-1}) \quad (4)$$

While stochastic masking encourages exploration in different subspaces, the overall perturbation magnitude is typically limited. Consequently, the exploration remains within a small range, making adversarial examples prone to local optima. Prior studies [12, 50] have shown that moderately relaxing the step-size constraint can improve the transferability of adversarial examples. Unlike these methods that directly enlarge the step size, we enhance the perturbation by expanding its exploration range to obtain more diverse gradients, thereby approaching the decision boundary more effectively. Specifically, we introduce an augmentation coefficient β to scale the perturbation and enlarge the search space. This process is illustrated in Figure 2. Accordingly,

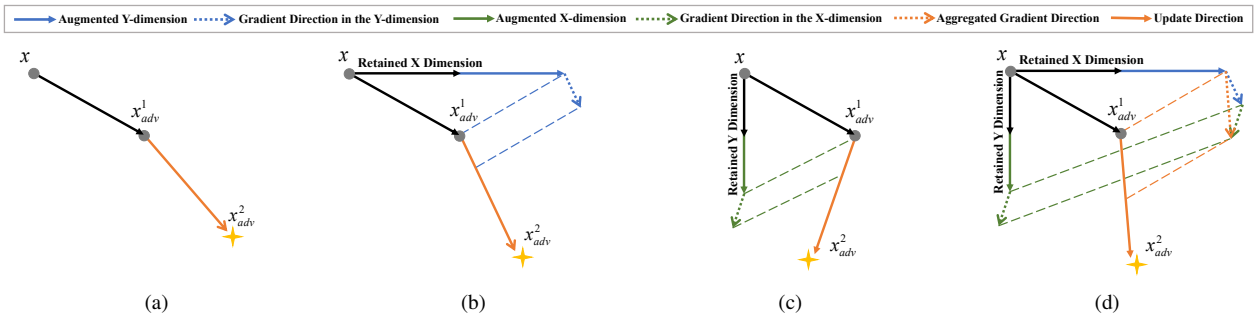


Figure 2. Schematic of Local Perturbation Augmentation. (a) shows the plain update of x^1_{adv} using its own gradient to obtain x^2_{adv}. (b) and (c) show updates using the augmented subspace gradients formed only along the X-dimension and only along the Y-dimension, respectively. (d) shows the perturbation update using aggregated gradient information from the augmented subspaces along both the X and Y dimensions.

Eq. (4) can be rewritten as:

$$\bar{g} = \frac{1}{N}\sum_{i=1}^{N} g_{i-1}(x + \delta_{t-1} \odot M_{i-1} \cdot \beta) \tag{5}$$

and the perturbation update becomes:

$$\delta_t = \delta_{t-1} + \alpha \cdot \text{sign}\left(\frac{1}{N}\sum_{i=1}^{N} g_{i-1}(x + \delta_{t-1} \odot M_{i-1} \cdot \beta)\right) \tag{6}$$

Moreover, as the perturbation accumulates across iterations, the effective exploration range dynamically changes, further mitigating overfitting in adversarial example generation. We refer to this approach as the Local Perturbation Augmentation (LPA) strategy. Experimental results validate the effectiveness of the proposed strategy.

3.3. Proposed Perturbation Initialization

Building upon the formulation in Eq. (6), when the iteration step $t = 0$, the LPA strategy has not yet taken effect since $\delta_0 = 0$. Moreover, because each iteration relies on locally guided perturbations, the overall performance is highly sensitive to the choice of the initial perturbation. Experimental results also confirm this observation. Therefore, we propose a perturbation initialization strategy that provides a more transferable starting direction.

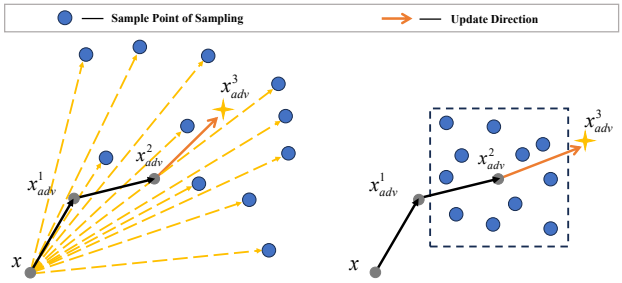
Recent studies [27, 33] have explored the impact of initialization perturbation on adversarial attacks. Hybrid Batch Attack [33] utilizes perturbations from transfer-based attacks as a starting point to support query-based attacks, while Texture-Adv [27] employs handcrafted efficient adversarial textures as the initialization to enhance transferability. However, the former is specifically designed for query-based attacks, and the latter purely leverages prior knowledge without considering methodological differences.

Algorithm 1 Local Perturbation Augmentation Attack

Input: A clean image x with ground-truth label y; a classifier f_θ; the loss function $\mathcal{L}$; adversarial perturbation δ.
Parameter: The number of iterations T; the step size α; decay factor μ; the perturbation budget ϵ; the number of subspaces N; upper bound factor of the projection range η; augmentation coefficient β; masking ratio p.
Output: An adversarial example x_{adv}.

$g_0 = 0$, $\delta_0 = 0$, $m_0 = 0$.
Calculate the initial perturbation δ_M :
$\delta_M = \Pi_{[-\eta\cdot\alpha,\ \eta\cdot\alpha]}\big(\mathcal{M}(x)\big)$
Calculate the initial perturbation δ_0 :
$\delta_0 = \Pi_{[-\eta\cdot\alpha,\ \eta\cdot\alpha]}\big(\mathcal{I}(x + \delta_M)\big)$
for t = 1,2,. . . ,T **do**
  $\overline{g} = 0$
  for i = 1,2,. . . ,N **do**
    Calculating the gradient in a single subspace:
    $g_{i-1} = \nabla_x \mathcal{L}(f_\theta(x + \delta_{t-1} \odot M_{i-1} \cdot \beta), y)$
    Calculate average gradient in multiple subspaces:
    $\overline{g} = \overline{g} + g_{i-1} \cdot \frac{1}{N}$
  end for
  if $t = 1$ **then**
    $\delta_{t-1} = 0$
  end if
  Update m_t by $m_t = \mu \cdot m_{t-1} + \frac{\overline{g}}{\|\overline{g}\|_1}$
  Update δ_t by $\delta_t = \delta_{t-1} + \alpha \cdot sign(m_t)$
end for
$x_{adv} = x + \delta_T$
return x_{adv}

We note that Global Momentum Initialization (GMI) [38] employs a pre-attack procedure to initialize momentum, effectively stabilizing gradient directions and guiding momentum-based attacks toward more foresighted directions. Owing to its foresightedness, the momentum pro-



(a) LPAA (b) Random sampling strategy

Figure 3. Comparison of different sampling strategies for gradient aggregation. We ignore the differences from the first two perturbation updates. (a) illustrates the characteristics of LPAA as a sampling strategy, while (b) shows the neighborhood random sampling strategy, where the dashed box represents the boundary of the sampling region.

vided by GMI reveals the vulnerability direction of deep neural networks. Inspired by this idea but differing from GMI, which initializes the momentum itself, we convert the momentum direction obtained by GMI into an initial perturbation to support our strategy. We term this as GMI_{PI}. Specifically, we define it as:

$$\delta_M = \Pi_{[-\eta\cdot\alpha,\ \eta\cdot\alpha]}(\mathcal{M}(x)) \tag{7}$$

where $\mathcal{M}(x)$ represents the momentum direction computed by GMI, and $\Pi_{[-\eta\cdot\alpha,\ \eta\cdot\alpha]}(\cdot)$ projects it onto the perturbation domain $[-\eta\cdot\alpha,\ \eta\cdot\alpha]$, and the coefficient η controls the upper bound of this projection.

However, δ_M only provides a general direction and cannot be effectively integrated into our framework. To rectify this initialization, we further apply the LPA strategy introduced in the previous subsection:

$$\delta_0 = \Pi_{[-\eta\cdot\alpha,\ \eta\cdot\alpha]}\big(\mathcal{I}(x+\delta_M)\big) \tag{8}$$

where $\mathcal{I}(\cdot)$ denotes the LPA operation. We refer to $\Pi_{[-\eta\cdot\alpha,\ \eta\cdot\alpha]}\big(\mathcal{I}(x)\big)$ as LPA_{PI}. The combination of GMI_{PI} and LPA_{PI}, as defined in Eq. (8), constitutes our overall Perturbation Initialization (PI) strategy. Importantly, the initialized perturbation δ_0 serves only as a directional prior for the first iteration, and subsequent updates proceed independently.

Finally, by integrating LPA and PI and further incorporating a momentum mechanism, we form the complete Local Perturbation Augmentation Attack (LPAA) framework. The full algorithm is summarized in **Algorithm 1**.

3.4. Compared with Random Neighborhood Sampling

In essence, our strategy can be regarded as a perturbation-based neighborhood sampling strategy. In previous works [11, 13, 39, 52], neighborhood sampling is formulated as $x+\delta_{t-1}+\tau$, where $\tau \sim U[-\gamma\cdot\epsilon, \gamma\cdot\epsilon]$ denotes random noise, and γ controls its amplitude.

Methods such as VMI-FGSM [39] and GRA [52] adopt this strategy along with an aggregation mechanism, and this combined approach is further utilized by PGN [13] and GAA [11] to locate flat local optima. In fact, this combined strategy inherently aims to achieve high loss values within the local neighborhood, thereby reflecting the property of a flat local optimum (see the discussion in **Appendix A.**).

In contrast, our method introduces additional stochasticity through M_{t-1}, which controls the size of the subspace, and utilizes β to regulate the sampling range. As illustrated in Figure 3, our approach performs neighborhood sampling directly on the perturbation itself through the augmented subspace, resulting in a broader sampling range and more consistent update directions. In comparison, previous neighborhood sampling strategies conduct sampling around the iterative point, yielding a much smaller sampling region.

Our empirical results demonstrate that replacing the neighborhood sampling strategy of existing state-of-the-art methods with ours leads to a more substantial improvement in the transferability of adversarial examples, highlighting the effectiveness of our proposed approach.

4. Experiments

4.1. Experimental Settings

4.1.1. Dataset.

We randomly selected 1,000 images from different categories in the ILSVRC 2012 validation dataset [31]. These images are correctly classified by nearly all models discussed in this paper and have been widely used in prior work [30, 41, 52].

4.1.2. Baseline.

We adopted state-of-the-art gradient-based attack methods including GRA [52], PGN [13], ANDA [10], and MuMoDiG [30], as well as input transformation-based methods such as DIM [46], TIM [8], SIM [22], Admix [40], and SIA [41], as baselines for comparison.

4.1.3. Models and Defenses.

To extensively evaluate our proposed method, we conducted experiments on a diverse set of architectures. Specifically, we adopted five CNNs: RN-50 [16], DN-121 [18], RNX-50 [47], CNX-T [24], and Inc-v3 [34], as well as six ViTs: ViT-B [9], PiT-B [17], Visformer-S (Vis-S) [6], CaiT-S [36], Swin-T [23], and DeiT-S [35]. We also considered three adversarially trained models (ATMs), including Inc-v3_{ens3}, Inc-v3_{ens4}, and IncRes-v2_{ens} [37]. Three defense methods, namely NRP [28], HGD [21], and Bit-Red (Bit) [49], were also incorporated for comprehensive performance evaluation.

Model	Method	RN-50	DN-121	RNX-50	CNX-T	Inc-v3	ViT-B	PiT-B	Vis-S	CaiT-S	Swin-T	DeiT-S	Avg.
RN-50	GRA	91.5*	82.1	80.2	74.0	75.2	49.4	62.6	69.1	60.4	70.5	60.2	70.5
	PGN	**96.9***	89.4	87.9	80.9	82.4	54.2	68.8	76.6	67.1	76.6	65.6	76.9
	SIA	95.9*	89.1	87.3	77.5	76.7	41.8	64.5	76.2	54.3	73.1	56.2	72.1
	ANDA	94.6*	72.6	72.2	59.5	62.1	33.1	49.2	57.6	42.2	56.2	41.6	58.3
	MuMoDIG	94.0*	85.0	82.5	75.1	78.9	49.7	65.4	73.2	61.1	71.5	58.3	72.2
	LPAA	94.9*	**90.4**	**88.7**	**84.6**	**85.0**	**58.3**	**74.7**	**80.4**	**72.0**	**80.1**	**70.5**	**80.0**
RNX-50	GRA	85.3	85.7	93.6*	79.0	79.4	58.1	70.0	74.9	68.0	75.8	67.7	76.1
	PGN	91.6	90.8	97.1*	85.7	86.0	64.1	77.2	81.7	73.9	81.8	74.2	82.2
	SIA	91.7	89.5	96.8*	81.6	80.7	48.2	72.8	80.3	62.9	76.5	61.6	76.6
	ANDA	83.0	81.9	**97.7***	68.7	67.7	37.2	57.0	66.7	49.7	63.4	48.6	65.6
	MuMoDIG	89.1	90.0	96.4*	80.8	84.1	55.6	72.2	80.5	69.1	77.7	65.0	78.2
	LPAA	**93.8**	**93.6**	96.9*	**90.6**	**89.6**	**70.0**	**82.3**	**86.5**	**80.9**	**87.2**	**80.2**	**86.5**
ViT-B	GRA	72.1	78.7	71.7	77.4	73.7	97.8*	78.2	79.3	88.2	83.9	88.0	80.8
	PGN	74.2	81.2	75.5	80.4	75.7	98.1*	82.1	82.7	91.3	87.2	90.8	83.6
	SIA	80.5	85.3	80.2	82.1	78.0	97.9*	85.5	85.6	89.2	89.1	90.0	85.8
	ANDA	66.7	76.5	65.7	69.5	70.0	98.0*	73.2	75.1	83.6	78.5	82.2	76.3
	MuMoDIG	76.9	81.9	77.5	79.1	77.7	95.4*	83.1	84.1	88.1	86.0	88.0	83.4
	LPAA	**85.2**	**89.5**	**85.0**	**89.6**	**86.2**	**99.4***	**90.3**	**90.4**	**97.5**	**95.6**	**97.6**	**91.5**

Table 1. The attack success rates (%) of LPAA and state-of-the-art methods on CNNs and ViTs. * indicates white-box attack success rates, and bold values denote the best results. See more results in **Appendix C.1**.

RN-50						ViT-B					
Method	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	NRP	HGD	Bit	Method	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	NRP	HGD	Bit
GRA	71.1	64.8	64.3	69.4	62.7	GRA	67.8	63.8	58.2	67.2	57.2
PGN	76.3	70.1	67.6	75.1	65.6	PGN	71.0	67.1	62.1	70.9	60.8
SIA	60.5	54.8	41.5	65.8	42.8	SIA	69.1	64.1	47.4	72.3	53.8
ANDA	48	43.2	35.5	52.4	37.9	ANDA	56.9	51.7	40.7	57.1	48.1
MuMoDIG	67.5	64.5	49.0	71.4	50.8	MuMoDIG	71.8	69.5	53.6	72.1	58.3
LPAA	**80.4**	**77.4**	**73.5**	**81.3**	**70.5**	LPAA	**79.9**	**75.5**	**64.6**	**80.2**	**65.5**

Table 2. The attack success rates (%) of LPAA and state-of-the-art methods using RN-50 and ViT-B as surrogate models on defense models and methods. Best results are shown in bold. See more results **Appendix C.2**.

4.1.4. Parameter Setting.

Following prior works [7, 13, 41, 52], we set the number of iterations T to 10, the perturbation budget ϵ to $16/255$, the step size α to $1.6/255$ and the momentum decay factor μ to 1.0. To ensure a fair comparison, we set $N_T = 20$ in MuMoDIG [30], which achieves higher attack success rates than the original configuration. Regarding LPAA, in the PI strategy, the GMI$_{\text{PI}}$ follow the default settings in GMI [38]. For LPA$_{\text{PI}}$, we set $T_{\text{LPA}_{\text{PI}}} = 5$ and $\alpha_{\text{LPA}_{\text{PI}}} = 3.2/255$, $\eta = 3$. In contrast, in the LPA strategy, the parameters are set as follows: $N = 20$, $p = 0.95$, and $\beta = 35$.

4.2. Evaluation on Normally Trained Models

To evaluate the transfer-based attack capability of the generated adversarial examples, we employed CNN and ViT as surrogate models to generate adversarial examples and tested their attack success rates on various target models. As shown in Table 1, the adversarial examples generated by LPAA under both surrogate model architectures achieve higher attack success rates than the state-of-the-art methods in most cases. Specifically, when using RNX-50 as the surrogate model, LPAA achieves an average attack success rate of 86.5%, outperforming the second-best method PGN by 4.3%. When using ViT-B as the surrogate model, LPAA attains an average attack success rate of 91.5%, surpassing the second-best method SIA by 5.7%. These results demonstrate the superior performance of LPAA in cross-model transfer attack tasks.

4.3. Evaluation on Defense Models and Methods

We further compared LPAA with other state-of-the-art methods on two ensemble adversarially trained models (Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$) and three defense methods (NRP, HGD, and Bit). As shown in Table 2, LPAA achieves higher attack success rates. Specifically, with RN-50 as surrogate, LPAA achieves a 76.6% average attack success rate,

Method	DN-121	RNX-50	CNX-T	ViT-B	PiT-B	Vis-S	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Avg.
DIM	59.3	54.2	43.4	22.1	32.7	41.2	38.7	37.3	31.6	40.1
LPAA+DIM	**92.0**	**90.5**	**88.1**	**72.8**	**82.1**	**86.0**	**87.5**	**85.8**	**84.4**	**85.5**
TIM	45.9	37.2	29.6	13.6	19.7	24.1	28.4	27.8	24.2	27.8
LPAA+TIM	**92.0**	**89.2**	**84.0**	**61.8**	**71.1**	**79.9**	**83.7**	**83.4**	**80.9**	**80.7**
SIM	57.4	54.2	39.3	18.8	30.2	39.2	33.0	31.8	26	36.7
LPAA+SIM	**94.9**	**93.9**	**92.1**	**71.9**	**83.2**	**89.5**	**91.1**	**90.4**	**87.7**	**88.3**
Admix	65.4	61.2	45.9	20.7	34.6	43.3	38.1	37.2	30.1	41.8
LPAA+Admix	**96.8**	**95.6**	**94.0**	**72.7**	**85.4**	**91.3**	**93.0**	**91.7**	**89.7**	**90.0**

Table 3. The attack success rates (%) of LPAA when combined with DIM, TIM, SIM, and Admix on CNNs, ViTs, and ATMs. The surrogate model is RN-50. The best results are highlighted in bold.

Method	DN-121	RNX-50	CNX-T	ViT-B	PiT-B	Vis-S	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Avg.
VMI-FGSM	59.9	58.5	48.6	27.2	40.0	45.3	43.8	42.2	37.2	44.7
VMI-FGSM + PI	73.6	72.3	63.4	**37.6**	**52.1**	59.7	58.4	56.0	49.7	58.1
VMI-FGSM + PI + LPA	**78.7**	**76.2**	**66.7**	35.1	50.9	**60.6**	**61.3**	**59.0**	**52.4**	**60.1**
GRA	82.1	80.2	74.4	49.5	62.2	68.4	70.5	70.7	65.5	69.3
GRA + PI	87.2	86.9	81.6	55.9	68.1	75.1	78.0	78.1	74.0	76.1
GRA + PI + LPA	**90.7**	**88.9**	**84.9**	**59.9**	**74.0**	**79.9**	**82.4**	**80.3**	**76.0**	**79.7**
PGN	88.5	87.9	80.6	54.6	68.3	75.6	76.7	75.9	70.3	75.4
PGN + PI	90.4	90.1	84.0	59.2	72.8	80.9	80.5	80.4	76.7	79.4
PGN + PI + LPA	**92.6**	**90.7**	**87.2**	**62.7**	**77.1**	**82.8**	**83.6**	**83.3**	**79.6**	**82.2**
GAA	81.4	79.3	73.2	49.6	62.2	67.4	70.1	70.1	64.4	68.6
GAA + PI	87.5	86.0	80.6	54.4	67.6	74.0	78.2	77.6	73.1	75.4
GAA + PI + LPA	**90.3**	**89.0**	**84.7**	**59.7**	**73.6**	**80.7**	**81.7**	**81.3**	**75.9**	**79.7**

Table 4. The attack success rates (%) of existing state-of-the-art methods based on neighborhood sampling combined with PI and LPA. RN-50 is used as the surrogate model, and the best results are highlighted in bold.

outperforming the second-best method PGN's 70.9%. With ViT-B as surrogate, LPAA attains 73.1%, surpassing PGN's 66.4%. These results further validate the strong transferability and effectiveness of LPAA, even against robust defenses.

4.4. Combined with Other Attacks

4.4.1. Combined with Input Transformation-based Attack.

To verify the compatibility of the proposed method while further enhancing its attack capability, we combined LPAA with various input transformation-based attack methods (including DIM, TIM, SIM, and Admix) to generate adversarial samples on the RN-50 model. As shown in Table 3, when applied to existing input transformation-based methods, LPAA significantly enhances their transferability, achieving an average improvement of nearly 50%. Simultaneously, the performance of LPAA itself is further improved compared to its performance reported in Table 1 and Table 2. These results fully demonstrate the strong compatibility of the LPAA attack method.

4.4.2. Combined with Neighborhood-based Sampling Attack.

To demonstrate the superior performance of LPAA as a neighborhood sampling strategy, we replaced the neighborhood sampling mechanism in existing state-of-the-art neighborhood-based methods with the LPAA strategy. Furthermore, to more clearly highlight the advantages of LPAA, we combined each method with both the PI and LPAA (LPA + PI) strategies separately. As shown in Table 4, the PI strategy significantly enhanced the transferability of existing state-of-the-art methods, with an overall improvement of approximately 7.75%. This indicates that the proposed PI strategy is not only compatible with LPA but also applicable to a broader range of methods (further results and discussion on the PI strategy are provided in **Appendix C.3**). When combined with LPA, the transferability was further improved in most cases, with an average increase of about 3.2%. It is worth noting that we did not fine-tune the hyperparameters of LPAA for each method. Even so, except for VMI-FGSM with ViT-B and PiT-B as target models, all other methods showed significant performance gains when integrated with LPAA. This further

Noise	GMI_{PI}	LPA_{PI}	CNNs	ViTs	ATMs
			64.6	50.1	53.3
✓			64.5	49.2	52.9
	✓		82.2	66.4	73.4
		✓	82.6	68.6	74.8
✓		✓	78.2	63.9	69.0
	✓	✓	**87.2**	**72.7**	**79.7**

Table 5. The average attack success rates (%) of LPAA using different initialization strategies. Noise denotes initialization using random noise. GMI_{PI} is defined as initializing perturbations with Eq. (7), and LPA_{PI} as initializing with the LPA strategy. The notation GMI_{PI} + LPA_{PI} indicates initialization based on Eq. (8).

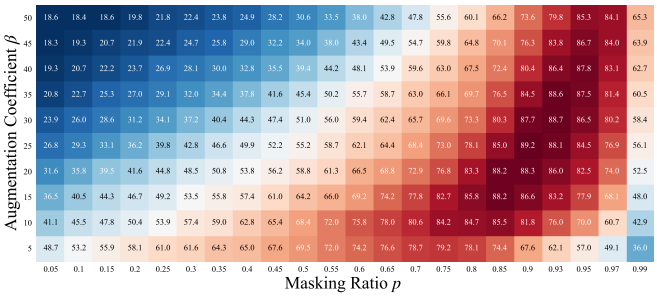


Figure 4. The average attack success rates (%) of LPAA on CNNs with different masking ratios and augmentation coefficients. More results on ViTs are shown in our **Appendix D.1**.

demonstrates the effectiveness of LPAA's sampling mechanism, which accounts for both the direction and magnitude of perturbations.

4.5. Ablation Study

In this subsection, we examined how different parameters and initialization strategies affected LPAA. Adversarial examples were generated on RN-50, and their average attack success rates were evaluated against other CNNs, ViTs, and ATMs. Additionally, ablation studies on the number of subspaces N, the upper bound factor of the projection range η, and the number of iterations of LPA_{PI} are presented in **Appendix D**.

4.5.1. Initialization Strategies.

To explore the impact of initialization perturbations on attack performance, we introduced multiple initialization strategies into the LPAA method, with the corresponding results shown in Table 5. The experiments demonstrated that different initialization methods significantly influenced the transferability of LPAA, indicating that LPAA exhibits sensitivity to perturbation directions. Specifically, initializing with random noise alone failed to introduce new information, resulting in transfer attack performance that was even slightly lower than that without initialization. Moreover, when random noise was combined with LPA_{PI} for initialization, it undermined the improvement in transferability brought by LPA_{PI} initialization. In contrast, GMI_{PI} demonstrated a degree of foresight, and initializing with GMI_{PI} could moderately improve the transferability of LPAA. However, due to the differences in foresight strategies between GMI_{PI} and our method, the overall performance gain remained limited. Notably, when we integrated GMI_{PI} with LPA_{PI}, the transferability of adversarial examples was enhanced, achieving the best attack performance.

4.5.2. Enhancement Coefficient β and Masking Ratio p.

Masking ratio p controls the degree to which the mask is applied, and thereby indirectly controls the size of the augmented subspace. The enhancement coefficient β governs the range that each subspace can explore. Figure 4 reports the combined effect of p and β on the average attack success rate across CNNs. From the figure, we observed that, when the augmented subspace was relatively small (roughly $p = 0.75$ to $p = 0.97$), LPAA achieved a high attack success rate by appropriately limiting the subspace exploration range. However, when p became too large (e.g., $p = 0.99$), the constructed subspace was excessively small, in other words, the iterative perturbation information became too sparse, preventing effective exploitation of the surrogate model's information and consequently reducing transferability. Considering the transferability on ViT models, we ultimately chose $p = 0.95$ and $\beta = 35$.

5. Conclusion

In this work, we introduced the Local Perturbation Augmentation Attack (LPAA) framework to alleviate the overfitting of adversarial examples to surrogate models. Instead of applying global perturbations, LPAA performs exploration within multiple augmented subspaces, guiding updates toward more generalizable directions. Furthermore, the proposed perturbation initialization strategy provides a highly transferable starting direction, addressing the initialization challenge in utilizing augmented subspaces. In addition, LPAA serves as a neighborhood sampling strategy that can be seamlessly integrated into existing sampling-based attack methods to further enhance their performance. Extensive experiments demonstrate that LPAA achieves superior transferability compared with state-of-the-art methods. Moreover, the effectiveness of LPAA underscores the feasibility of performing iterative optimization using only partial perturbations, providing a new perspective for future research.

Acknowledgments

This work was supported by State Grid science and technology project (5700-202327286A-1–1-ZN).

References

[1] Yang Bai, Yisen Wang, Yuyuan Zeng, Yong Jiang, and Shu-Tao Xia. Query efficient black-box adversarial attack on deep neural networks. *Pattern Recognition*, 133:109037, 2023. 1

[2] Fadi Boutros, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Elasticface: Elastic margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1578–1587, 2022. 1

[3] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017. 1

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 1

[5] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[6] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 589–598, 2021. 5

[7] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 1, 2, 3, 6

[8] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4312–4321, 2019. 2, 5

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 5

[10] Zhengwei Fang, Rui Wang, Tao Huang, and Liping Jing. Strong transferable adversarial attacks via ensembled asymptotically normal distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24841–24850, 2024. 1, 2, 5

[11] Fuquan Gan and Yan Wo. Boosting the transferability of adversarial examples through gradient aggregation. *IEEE Transactions on Information Forensics and Security*, 2025. 2, 5

[12] Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. Patch-wise attack for fooling deep neural network. In *European Conference on Computer Vision*, pages 307–322. Springer, 2020. 3

[13] Zhijin Ge, Hongying Liu, Wang Xiaosen, Fanhua Shang, and Yuanyuan Liu. Boosting adversarial transferability by achieving flat local maxima. *Advances in Neural Information Processing Systems*, 36:70141–70161, 2023. 1, 2, 3, 5, 6

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

[15] Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. *Advances in neural information processing systems*, 33:85–95, 2020. 3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5

[17] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11936–11945, 2021. 5

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1, 5

[19] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018. 1, 2, 3

[20] Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. Learning transferable adversarial examples via ghost networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11458–11465, 2020. 3

[21] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1778–1787, 2018. 5

[22] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. *arXiv preprint arXiv:1908.06281*, 2019. 2, 5

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 5

[24] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 5

[25] Yuyang Long, Qilong Zhang, Boheng Zeng, Lianli Gao, Xianglong Liu, Jian Zhang, and Jingkuan Song. Frequency domain model augmentation for adversarial attack. In *European conference on computer vision*, pages 549–566. Springer, 2022. 2

[26] Wenshuo Ma, Yidong Li, Xiaofeng Jia, and Wei Xu. Transferable adversarial attack for both vision transformers and

convolutional networks via momentum integrated gradients. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4630–4639, 2023. 2

[27] Ningping Mou, Binqing Guo, Lingchen Zhao, Cong Wang, Yue Zhao, and Qian Wang. No-box universal adversarial perturbations against image classifiers via artificial textures. *IEEE Transactions on Information Forensics and Security*, 2024. 4

[28] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 262–271, 2020. 5

[29] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1

[30] Yuchen Ren, Zhengyu Zhao, Chenhao Lin, Bo Yang, Lu Zhou, Zhe Liu, and Chao Shen. Improving integrated gradient-based transferable adversarial examples by refining the integration path. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6731–6739, 2025. 2, 5, 6

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5

[32] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1

[33] Fnu Suya, Jianfeng Chi, David Evans, and Yuan Tian. Hybrid batch attacks: Finding black-box adversarial examples with limited queries. In *29th USENIX security symposium (USENIX Security 20)*, pages 1327–1344, 2020. 4

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 5

[35] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 5

[36] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 32–42, 2021. 5

[37] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 5

[38] Jiafeng Wang, Zhaoyu Chen, Kaixun Jiang, Dingkang Yang, Lingyi Hong, Pinxue Guo, Haijing Guo, and Wenqiang Zhang. Boosting the transferability of adversarial attacks with global momentum initialization. *Expert Systems with Applications*, 255:124757, 2024. 4, 6

[39] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1924–1933, 2021. 2, 5

[40] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16158–16167, 2021. 2, 5

[41] Xiaosen Wang, Zeliang Zhang, and Jianping Zhang. Structure invariant transformation for better adversarial transferability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4607–4619, 2023. 2, 5, 6

[42] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7639–7648, 2021. 3

[43] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. *arXiv preprint arXiv:2002.05990*, 2020. 3

[44] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R Lyu, and Yu-Wing Tai. Boosting the transferability of adversarial samples via attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1161–1170, 2020. 3

[45] Wang Xiaosen, Kangheng Tong, and Kun He. Rethinking the backward propagation for adversarial transferability. *Advances in Neural Information Processing Systems*, 36:1905–1922, 2023. 3

[46] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2730–2739, 2019. 2, 5

[47] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 5

[48] Yifeng Xiong, Jiadong Lin, Min Zhang, John E Hopcroft, and Kun He. Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14983–14992, 2022. 2, 3

[49] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017. 5

[50] Ming Zhang, Xiaohui Kuang, Hu Li, Zhendong Wu, Yuanping Nie, and Gang Zhao. Improving transferability of adversarial examples with virtual step and auxiliary gradients. In *IJCAI*, pages 1629–1635, 2022. 3

[51] Desheng Zheng, Wuping Ke, Xiaoyu Li, Yaoxin Duan, Guangqiang Yin, and Fan Min. Enhancing the transferability of adversarial attacks via multi-feature attention. *IEEE Transactions on Information Forensics and Security*, 2025. 3

[52] Hegui Zhu, Yuchen Ren, Xiaoyan Sui, Lianping Yang, and Wuming Jiang. Boosting adversarial transferability via gradient relevance attack. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4741–4750, 2023. 1, 2, 5, 6

[53] Zheng Zhu, Guan Huang, Jiankang Deng, Yun Ye, Junjie Huang, Xinze Chen, Jiagang Zhu, Tian Yang, Jiwen Lu, Dalong Du, et al. Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10492–10502, 2021. 1