

Improving Adversarial Transferability with Local Perturbation Augmentation

Supplementary Material

A. Neighborhood Sampling

In previous works [2, 3, 10, 11], the neighborhood sampling is formulated as $x'_i = x + \delta_{t-1} + \tau_i$, where $\tau \sim U[-\gamma \cdot \epsilon, \gamma \cdot \epsilon]$ denotes random noise and γ controls the noise amplitude. After incorporating the aggregation strategy, the perturbation updated solely using neighborhood gradients can be expressed as:

$$\delta_t = \delta_{t-1} + \alpha \cdot \text{sign}\left(\frac{1}{N} \sum_{i=1}^N \nabla_x \mathcal{L}(f_\theta(x'_i), y)\right) \quad (1)$$

Accordingly, the optimization objective can be rewritten as:

$$\arg \max_{\delta} \mathbb{E}_{\tau_i \sim U} [\mathcal{L}(f_\theta(x'_i), y)] \quad (2)$$

This implies that the optimized adversarial example $x_{adv} = x + \delta$ tends to achieve a consistently high loss within its neighborhood $[x_{adv} - \gamma \cdot \epsilon, x_{adv} + \gamma \cdot \epsilon]$. This also indicates that the loss landscape around x_{adv} exhibits a property of flatness. Since the target model can be viewed as a shifted version of the surrogate model’s loss landscape, such a high-loss neighborhood is more likely to preserve its adversarial effect after the shift, thereby enhancing transferability. Our experiments further demonstrate that perturbation optimization based solely on neighborhood information remains highly effective. As shown in Table 2, with $N = 20$ and $\gamma = 3.0$, the Random Sampling strategy attains competitive transferability. We attribute this to two factors: (1) reducing gradient variance through the aggregation mechanism for more stable updates; and (2) inducing a flatness property by optimizing for higher loss across the neighborhood. Both factors have been shown effective in previous works such as GRA [11] and PGN [3].

This can be interpreted as approximately optimizing the following objective: This can be interpreted as approximately optimizing the following objective:

The proposed LPAA framework adopts a conceptually similar but structurally guided neighborhood formulation. Instead of perturbing the input with random noise, LPAA samples from localized subspaces constructed by masks: $\tilde{x}_i = x + \delta \odot M_i \cdot \beta$. Here, M_i selects a subspace of the perturbation and β controls the exploration scale. This can be interpreted as approximately optimizing the following objective:

$$\arg \max_{\delta} \mathbb{E}_{M_i \sim \Delta} [\mathcal{L}(f_\theta(\tilde{x}_i), y)] \quad (3)$$

Unlike random-noise sampling, which explores isotropic neighborhoods around x_{adv} , LPAA defines a structured neighborhood aligned with the perturbation direction δ .

This formulation enables exploration of directional robustness within multiple augmented subspaces, capturing more informative variations of the loss surface. Empirical evidence supporting this behavior is provided in Appendix C.4.

B. Implementation Details

For the defense methods (Bit and NRP), we adopted the Inc-v3_{ens3} as the target model.

In the GMI_{PI} strategy, we project the direction obtained from GMI [9] onto a constrained domain $[-\eta \cdot \alpha, \eta \cdot \alpha]$ as follows:

$$\delta_M = \eta \cdot \alpha \cdot \text{sign}(\mathcal{M}(x)) \quad (4)$$

In the LPA_{PI} strategy, we further project the perturbation by constraining it within the same neighborhood, ensuring that it remains consistent with the feasible domain:

$$\delta_0 = \frac{\mathcal{I}(x + \delta_M)}{\epsilon} \cdot \eta \cdot \alpha \quad (5)$$

C. Additional Experimental Results

C.1. Evaluation on Normally Trained Models

We further evaluate LPAA using DN-121 [5], CNX-T [7], Vis-S [1], PiT-B [4], and CaiT-S[8] as surrogate models. Table 1 reports attack success rates on various target models, comparing LPAA with state-of-the-art methods. LPAA consistently achieves the highest average success rates across CNNs and ViTs, and frequently attains the best results on individual models, demonstrating its effectiveness and robustness in adversarial attacks.

C.2. Evaluation on Defense Models and Methods

We further evaluate LPAA against several defense models and methods to verify its robustness. We assess the transferability of adversarial examples generated by the surrogate models in Table 1 to these defenses. As shown in Table 3, LPAA consistently outperforms all baseline methods across all evaluated settings. Furthermore, we report the evaluation results on additional defense methods. As shown in Table 4, LPAA remains highly competitive.

C.3. Analysis of the PI Strategy

C.3.1. Compared with GMI.

Goal Differences. GMI [9] is a momentum-initialization pre-attack technique designed to stabilize the momentum during iterative attacks. By reducing fluctuations in the momentum across iterations, GMI enhances the consistency of

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.
DN-121	GRA	92.0	100*	91.1	86.4	91.3	57.2	70.7	81.6	71.3	83.9	71.8	81.6
	PGN	92.5	100*	92.1	87.8	91.4	59.1	72.0	83.2	71.7	85.4	73.3	82.6
	SIA	97.3	100*	95.7	88.8	95.4	54.5	74.1	88.5	71.9	86.6	69.1	83.8
	ANDA	89.2	100*	86.2	77.8	86.2	45.8	62.2	77.8	60.1	77.1	59.3	74.7
	MuMoDIG	94.8	99.9*	93.7	89.0	95.1	62.5	75.8	87.7	75.7	86.3	72.3	84.8
	LPAA	96.2	100*	95.8	92.6	95.5	63.4	74.5	87.8	76.2	88.1	77.1	86.1
CNX-T	GRA	85.3	88.7	85.9	98.8*	84.4	72.7	79.4	85.3	81.7	90.0	80.8	84.8
	PGN	88.3	90.9	87.9	99.3*	86.4	76.2	82.3	88.1	84.7	92.1	83.6	87.3
	SIA	93.8	93.7	91.1	99.9*	88.0	64.1	85.9	91.0	79.1	94.6	76.3	87.0
	ANDA	85.1	88.3	84.1	100*	82.8	58.1	77.7	84.4	73.2	86.4	68.9	80.8
	MuMoDIG	89.4	90.7	88.3	99.7*	88.4	68.7	82.5	87.9	80.3	90.1	75.7	85.6
	LPAA	93.5	95.6	93.6	99.8*	93.4	81.3	88.5	92.3	90.5	96.2	88.4	92.1
Vis-S	GRA	80.7	83.5	80.2	81.7	79.8	75.3	83.0	95.7*	79.6	84.0	80.8	82.2
	PGN	85.3	88.7	85.7	88.4	86.0	82.4	88.0	97.5*	87.0	90.5	86.8	87.8
	SIA	89.6	92.5	89.6	90.0	87.0	75.4	92.2	99.6*	86.5	93.7	84.6	89.2
	ANDA	80.5	86.5	81.7	82.0	79.4	67.0	84.9	99.3*	78.1	87.9	76.5	82.2
	MuMoDIG	90.3	93.3	90.1	90.9	90.2	78.8	93.3	99.2*	88.1	93.4	85.4	90.3
	LPAA	94.1	95.6	94.0	95.0	94.1	88.8	95.0	99.1*	94.6	96.1	94.8	94.7
PiT-B	GRA	75.4	78.0	75.3	77.2	76.1	72.9	94.9*	81.7	77.9	82.1	79.4	79.2
	PGN	78.2	80.1	77.1	78.1	77.0	75.3	94.6*	83.5	79.7	82.2	81.7	80.7
	SIA	87.2	88.6	86.9	86.5	82.9	76.9	99.6*	91.9	86.4	92.6	85.5	87.7
	ANDA	78	82.3	79.3	81.4	77.8	72.0	99.0*	87.6	80.2	87.4	81.3	82.4
	MuMoDIG	80.4	82.0	79.9	80.7	79.8	74.2	97.9*	85.7	81.7	85.7	83.3	82.8
	LPAA	87.8	89.1	88.1	89.4	87.8	85.1	97.3*	91.9	91.5	92.2	91.9	90.2
CaiT-S	GRA	80.4	85.4	79.1	84.8	81.4	89.6	84.0	85.6	97.7*	89.5	92.9	86.4
	PGN	83.0	87.0	83.1	87.6	83.7	91.9	87.6	88.0	98.7*	91.7	94.7	88.8
	SIA	92.9	93.9	91.9	94.5	91.8	94.9	95.6	94.7	99.4*	96.3	97.5	94.9
	ANDA	81.5	87.6	81.3	85.1	83.3	90.2	88.1	89.2	99.7*	91.3	95.7	88.5
	MuMoDIG	85.0	88.7	84.9	87.0	87.6	90.8	88.9	88.3	97.3*	91.1	92.7	89.3
	LPAA	92.4	94.9	90.9	95.9	93.5	98.3	93.9	95.3	99.7*	98.2	99.2	95.7

Table 1. The attack success rates (%) of LPAA and state-of-the-art methods on CNNs and ViTs. * indicates the white-box attack success rate; bold values denote the best results.

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
GRA	82.1	80.2	74.4	49.5	62.2	68.4	70.5	70.7	65.5	69.3
PGN	88.5	87.9	80.6	54.6	68.3	75.6	76.7	75.9	70.3	75.4
GAA	81.4	79.3	73.2	49.6	62.2	67.4	70.1	70.1	64.4	68.6
Random Sampling	79.7	78.1	71.6	49.1	61.3	67.7	68.4	68.0	62.4	67.4

Table 2. The attack success rates (%) of the Random Sampling strategy compared with other sampling-based attack methods. Adversarial examples are generated on RN-50.

the update direction, thereby improving the transferability of adversarial examples. In contrast, the PI strategy proposed in this work is introduced to address the limited initial performance of LPA. Since the first iteration of LPA is equivalent to I-FGSM (i.e., the LPA mechanism has not yet taken effect and relies on a global perturbation), the attack becomes highly sensitive to the choice of the initial perturbation. To obtain a more foresighted initial direction, PI ini-

tializes the perturbation before the main LPA optimization begins.

Implementation Differences. After the pre-attack stage, LPA does not inherit the momentum produced by PI. Instead, it uses only the resulting perturbation to initialize the optimization process. This initialization provides a more informative directional prior for LPA.

Complementarity of PI and GMI. When integrated,

RNX-50						Vis-S					
Method	Inc-v3 _{ens4}	IncRes-v2 _{ens}	NRP	HGD	Bit	Method	Inc-v3 _{ens4}	IncRes-v2 _{ens}	NRP	HGD	Bit
GRA	74.6	71.3	67.0	74.6	64.1	GRA	73.1	70.7	66.7	75.1	63.4
PGN	80.1	77.9	72.7	82.0	70.0	PGN	81.3	77.8	73.5	82.8	70.5
SIA	65.1	59.2	45.0	74.7	45.4	SIA	72.3	67.1	48.7	78.0	52.1
ANDA	56.3	51.2	37.7	62.8	43.7	ANDA	66.2	60.0	44.1	69.6	48.7
MuMoDIG	76.2	70.6	53.2	79.1	56.6	MuMoDIG	78.9	76.8	57.2	82.0	63.1
LPAAs	86.4	84.2	78.9	87.6	74.6	LPAAs	88.4	85.9	76.0	89.8	74.7
DN-121						PiT-B					
Method	Inc-v3 _{ens4}	IncRes-v2 _{ens}	NRP	HGD	Bit	Method	Inc-v3 _{ens4}	IncRes-v2 _{ens}	NRP	HGD	Bit
GRA	85.8	79.9	78.6	87.5	74.6	GRA	68.1	64.2	60.3	68.5	59.3
PGN	86.9	80.2	79.4	87.4	72.7	PGN	69.5	67.4	61.9	70.3	61.1
SIA	87.3	76.7	59.2	91.3	59.3	SIA	67.6	62.5	46.5	70.7	48.7
ANDA	76.3	64.6	47.7	82.0	53.3	ANDA	62.5	58.0	43.1	65.7	50
MuMoDIG	89.1	83.4	69.3	92.0	71.0	MuMoDIG	67.4	64.1	47.6	70.5	55.1
LPAAs	92.1	86.1	80.1	93.2	75.4	LPAAs	81.4	77.7	66.9	82.6	68.5
CNX-T						CaiT-S					
Method	Inc-v3 _{ens4}	IncRes-v2 _{ens}	NRP	HGD	Bit	Method	Inc-v3 _{ens4}	IncRes-v2 _{ens}	NRP	HGD	Bit
GRA	79.0	75.2	70.2	79.6	68.9	GRA	75.4	71.8	66.5	76.6	66.0
PGN	81.6	78.4	73.8	83.2	71.3	PGN	78.2	75.1	69.0	78.9	69.2
SIA	71.7	63.5	46.7	77.4	45.7	SIA	83.5	79.7	58.7	86.8	62.6
ANDA	66.0	60.5	42.5	71.7	47.8	ANDA	70.4	66.2	49.3	73.7	54.6
MuMoDIG	77.1	72.9	51.3	81.3	58.6	MuMoDIG	80.4	76.0	59.5	81.9	65.9
LPAAs	87.2	85.5	78.7	89.9	74.5	LPAAs	89.0	84.6	73.0	89.9	75.2

Table 3. The attack success rates (%) of LPAAs and state-of-the-art methods on defense models and methods. Best results are shown in bold.

Method	FD	JPEG	RS	Dif
GRA	70.9	69.7	40.8	43.5
PGN	77.1	76.2	41.3	46.9
SIA	59.5	63.8	25.8	23.4
ANDA	51.8	50.3	24.9	22.5
MuMoDIG	67.0	69.3	28.1	29.9
LPAAs	81.5	81.3	40.3	43.9

Table 4. The attack success rates (%) of LPAAs and state-of-the-art methods on more defense methods. Best results are shown in bold.

the two strategies complement each other and jointly improve the transferability of adversarial examples, as demonstrated in Table 5. This compatibility indicates that PI focuses on the initialization quality while GMI enhances the momentum stability, leading to complementary improvements.

C.3.2. Analysis of the Effectiveness of PI Strategy.

In the main paper, we showed that the PI strategy works effectively within the LPAAs framework and also improves the performance of existing attack methods. We hypothesize that the benefit of PI originates from gradient correction. During the first iteration, PI provides a more favorable update direction, which then influences the subsequent

optimization process. This correction effect comes from two sources: (1) the perturbation generated in the first iteration (δ_1), which directly affects subsequent updates, and (2) the momentum accumulated from the corrected gradient in the first step (m_1). To validate this hypothesis, we design two experiments that isolate the contribution of each factor. As shown in Table 5, transferability improves when we preserve only the perturbation produced by the PI step, and it also improves when we preserve only the momentum produced by δ_0 . These results indicate that the PI strategy enhances transferability through both mechanisms, namely the initial perturbation and the momentum derived from δ_0 . It is worth noting that our PI strategy does not initialize the momentum itself, meaning that $m_0 = 0$. PI only generates an improved initial perturbation δ_0 .

It is important to note that PI produces its largest performance improvement, close to 23%, when integrated into the LPAAs framework, while improving other attack methods by approximately 7%. This observation suggests that the PI strategy substantially alleviates the initial performance limitation of the LPA strategy.

Method	DN-121	RNX-50	CNX-T	ViT-B	PiT-B	Vis-S	Inc-v3 _{ens3}	Inc-v3 _{ens4}	IncRes-v2 _{ens}	Avg.
MI-FGSM	43.8	37.3	27.4	10.4	18.2	24.0	21.4	21.7	16.0	24.5
MI-FGSM + GMI	57.3	51.2	36.8	15.0	24.8	32.8	31.9	31.2	22.1	33.7
MI-FGSM + PI _{perturbation}	49.4	43.7	31.7	12.6	21.4	27.3	24.9	25.0	18.4	28.3
MI-FGSM + PI _{momentum}	47.1	42.2	30.0	12.0	21.2	26.6	24.9	24.7	17.8	27.4
MI-FGSM + PI	55.0	47.9	36.6	14.0	25.6	30.7	29.9	29.1	20.7	32.2
MI-FGSM+PI + GMI	60.4	54.0	40.2	16.7	27.1	36.8	33.5	32.2	24.1	36.1

Table 5. The attack success rates (%) of combining PI with GMI. PI_{perturbation} refers to retaining only the perturbation δ_1 obtained by the initial perturbation δ_0 , while replacing the momentum m_1 generated from δ_0 with the m_1 obtained from the original MI-FGSM. PI_{momentum}, in contrast, retains only the momentum m_1 while replacing the perturbation δ_1 with the δ_1 obtained from the original MI-FGSM. Adversarial examples are generated on RN-50. Best results are highlighted in bold.

C.4. Further Analysis of LPAA

C.4.1. Visualization of the loss landscape

To further demonstrate that LPAA escapes local optima, we visualized the loss landscapes of adversarial examples generated by GRA, GRA + LPAA, PGN, PGN + LPAA, and LPAA. As shown in Figure 1, the visualization results indicated that LPAA found a neighborhood with consistently high loss, suggesting that it avoided sharp surrogate-specific peaks. In addition, replacing the random sampling strategy in the baseline methods with LPAA further enhanced the flatness of the loss landscape. These results demonstrated that LPAA was more effective than random sampling.

C.4.2. Gradient Analysis

To demonstrate the distinction between LPAA and Random Sampling, we calculated the average cosine similarity between individual sampled gradients and the aggregated gradient across iterations. As illustrated in Figure 2, LPAA exhibited a higher cosine similarity than Random Sampling during the initial iterations, whereas the trend reversed in the later stages. Notably, a similar evolutionary trend was observed in the pairwise cosine similarity among the sampled gradients, further confirming the consistency of our findings. This phenomenon stems from the dynamic evolution of δ : in the early phase, a smaller δ restricts exploration to a localized neighborhood; as iterations progress, the gradual increase of δ significantly expands the exploration domain of LPAA subspaces, thereby reducing the similarity across different subspaces. Consequently, this mechanism allows LPAA to leverage PI strategies for transferable directions during early iterations. In the later stages, the broadened scope of exploration enables LPAA to capture more diverse gradient information across multiple subspaces, effectively facilitating the escape from local optima.

C.5. Computational Overhead

As shown in Table 6, LPAA achieved a trade-off between performance and efficiency. Generating 1000 adversarial samples took 521 s (305 gradient calls: 105 + 200) on an NVIDIA RTX 4090 GPU, which was faster than

Method	Gradient Calls	Total Runtime (s)
VMIFGSM	210	359.6
GRA	210	359.1
PGN	400	677.0
GAA	400	681.7
LPAA	305	521.2

Table 6. Comparison of gradient calls and runtime for LPAA and sampling-based baselines. The results were obtained on an RTX 4090 GPU. The surrogate model is RN-50.

Method	RN-50	DN-121	MN-v2	Inc-v3	Vgg19	Avg.
GRA	92.93*	83.89	87.29	85.07	83.37	86.51
PGN	90.24*	82.73	85.25	83.57	82.10	84.78
SIA	93.03*	82.44	87.17	84.81	81.21	85.73
ANDA	86.50*	79.76	80.31	78.97	77.86	80.68
MuMoDIG	83.55*	75.08	78.30	76.27	74.36	77.51
LPAA	96.32*	88.91	91.48	89.80	87.85	90.87

Table 7. The attack success rates (%) of LPAA and state-of-the-art methods on CIFAR-10. * indicates the white-box attack success rate; bold values denote the best results. The surrogate model is RN-50.

sampling-based methods such as PGN (677 s, 400 calls) and GAA (681 s, 400 calls). Although LPAA was slower than VMIFGSM and GRA (both around 359 s with 210 calls), its significant performance improvement (e.g., +31.3% over VMIFGSM and +9.5% over GRA) justifies the additional computational cost.

C.6. Evaluation on CIFAR-10

To evaluate hyperparameter robustness, we tested LPAA on CIFAR-10 [6] with fixed parameters ($\epsilon = 8/255$, $\alpha = 0.8$, and $T = 10$). As shown in Table 7, LPAA consistently outperforms GRA, PGN, SIA, ANDA, and MuMoDIG, demonstrating its superior performance and robustness across datasets with different resolutions.

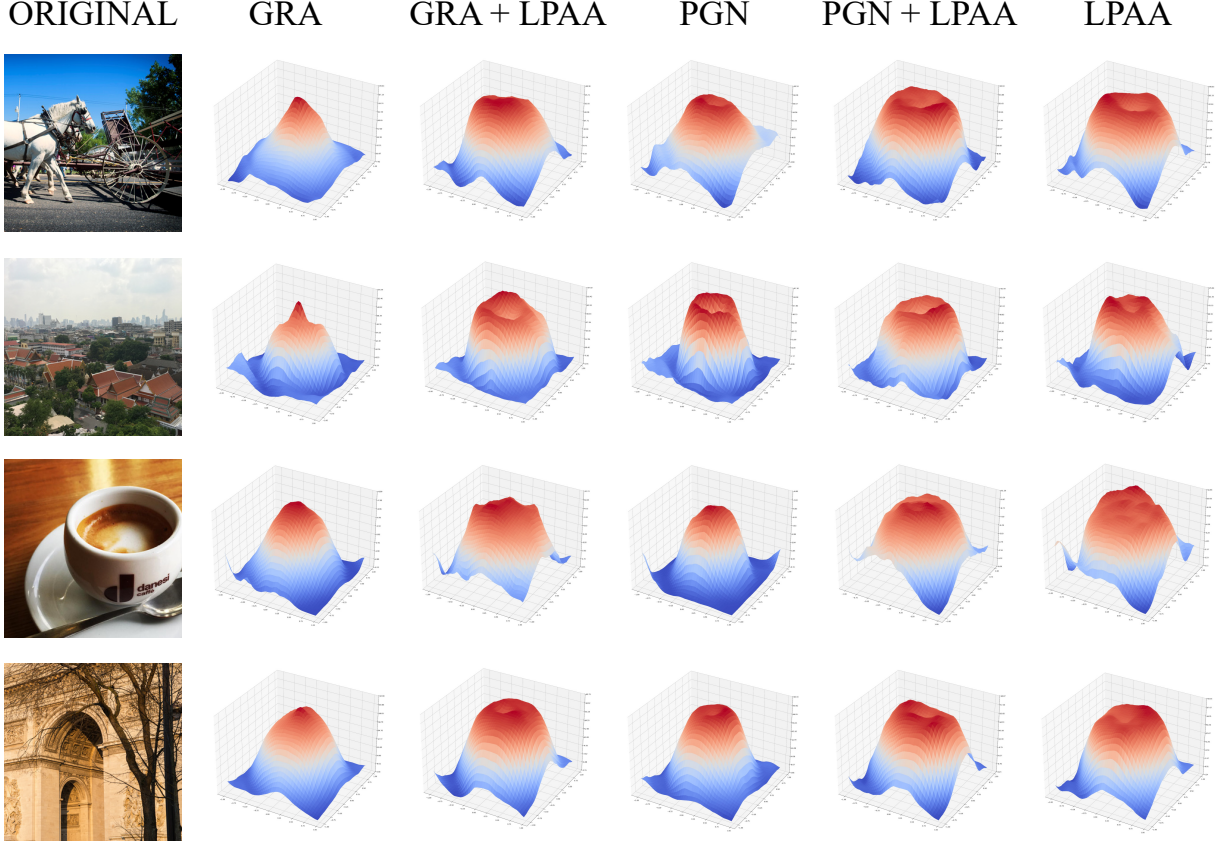


Figure 1. Visualization of the loss landscape along two random directions for four randomly sampled adversarial examples on the surrogate model Inc-v3.

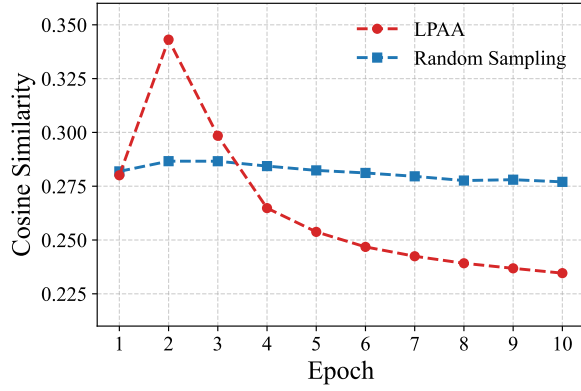


Figure 2. The cosine similarity between sampled gradients and the aggregated gradient for LPAA and Random Sampling across iterations. The surrogate model is RN-50.

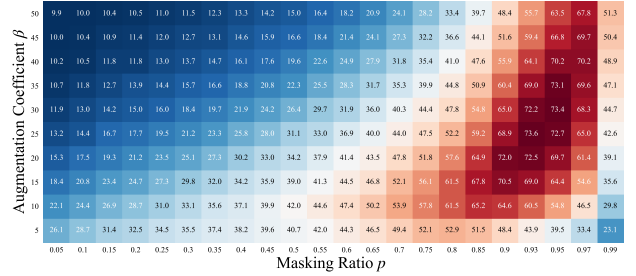


Figure 3. The average attack success rates (%) of LPAA on ViTs with different masking ratios and augmentation coefficients. The surrogate model is RN-50.

D. Ablation Study

D.1. Enhancement Coefficient β and Masking Ratio p

In addition to the results reported for CNNs in the main text, we present the attack performance of LPAA on ViTs. As shown in Figure 3, the trends observed for ViTs were

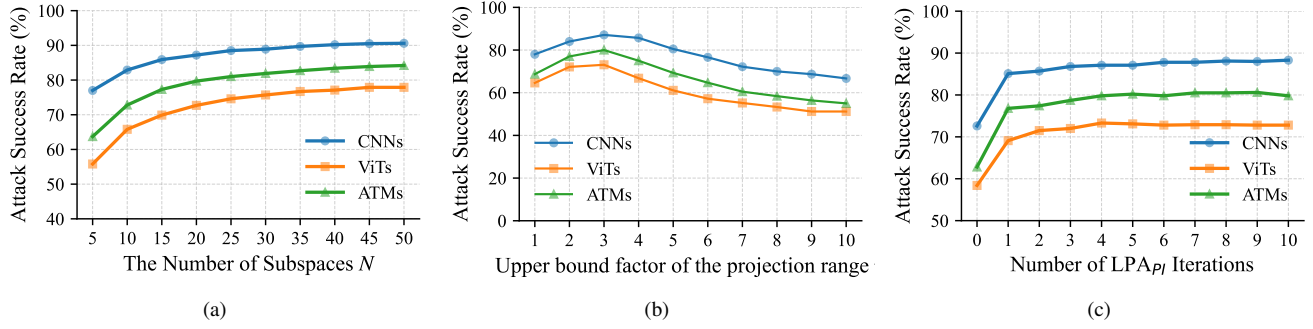


Figure 4. The average attack success rates (%) of LPAA under different numbers of subspaces, upper bound factors of the projection range, and numbers of LPA_{PI} iterations. The surrogate model is RN-50.

Parameter	CIFAR-10		ImageNet	
	$\epsilon = 8/255$		$\epsilon = 8/255$	
	CNNs	ViTs	CNNs	ViTs
$p = 0.9, \beta = 20$	92.98	62.5	40.3	
$p = 0.93, \beta = 25$	91.98	64.7	43.6	
$p = 0.93, \beta = 30$	91.70	66.2	44.6	
$p = 0.95, \beta = 30$	89.51	63.2	44.7	
$p = 0.95, \beta = 35$	89.53	65.8	45.8	

Table 8. The average attack success rates (%) of LPAA under various masking ratios p and augmentation coefficients β on CIFAR-10 and ImageNet. The surrogate model is RN-50.

Strategy	CNNs	ViTs	ATMs
Retained	85.0	61.3	72.0
Discarded	87.2	72.7	79.7

Table 9. The average attack success rates (%) of LPAA with retained vs. discarded initial perturbation. The surrogate model is RN-50.

similar to those on CNNs: when p ranged from 0.9 to 0.97, adjusting the enhancement coefficient β allowed LPAA to achieve relatively high transferability. Considering the results on CNNs, we adopted $p = 0.95$ and $\beta = 35$. Meanwhile, Table 8 provides the black-box attack results for LPAA on CIFAR-10 and ImageNet under various masking ratios and augmentation coefficients with $\epsilon = 8/255$. Based on the overall performance, we found that $p = 0.93$ and $\beta = 25$ also constituted a valid parameter configuration.

D.2. The Number of Subspaces N

LPAA obtains more comprehensive gradients by sampling N subspaces. Figure 4 (a) shows the effect of different values of N on the transferability of LPAA. We observed that the attack success rate saturated when $N = 20$, and further increasing N brought only marginal improvements. Considering both efficiency and performance, we set $N = 20$.

D.3. Upper bound factor of the projection range η

The parameter η controls the projection range of the initialization perturbation. As shown in Figure 4(b), the transferability of adversarial examples first increased and then decreased as η grew, reaching its peak at $\eta = 3$. This suggests that although a larger η expands the exploration space, it also introduces noise unrelated to transferability and drives the perturbation away from the optimal transfer direction, thereby hindering effective subsequent updates. Similarly, as shown in Table 9, discarding the initialization perturbation after the first guided update—rather than continuing to update it—further improved the success rate of transfer attacks. This observation is consistent with the negative effect of an excessively large η .

D.4. The number of iterations of LPA_{PI}

To reduce the computational overhead of the pre-attack procedure, we conducted an ablation study to examine the effect of different iteration numbers of LPA_{PI} in the PI stage on the overall transferability of LPAA. The step size in this stage was set as $\alpha_{PI} = \epsilon/T_{LPA_{PI}}$. As shown in Figure 4(c), the transferability of LPAA became relatively stable when the number of iterations for LPA_{PI} reached 5. Therefore, we set the number of LPA_{PI} pre-attack iterations to 5, with the corresponding $\alpha_{PI} = 3.2/255$.

References

- [1] 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. 1
- [2] Fuquan Gan and Yan Wo. Boosting the transferability of adversarial examples through gradient aggregation. *IEEE Transactions on Information Forensics and Security*, 2025. 1
- [3] 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

- [4] 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. [1](#)
- [5] 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](#)
- [6] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. [4](#)
- [7] 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. [1](#)
- [8] 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. [1](#)
- [9] Jiafeng Wang, Zhaoyu Chen, Kaixun Jiang, Ding kang 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. [1](#)
- [10] 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. [1](#)
- [11] 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](#)