SMP-Attack: Boosting the Transferability of Feature Importance-based Adversarial Attack with Semantics-aware Multi-granularity Patchout

Supplementary Material

In this supplementary material, we provide additional explanations, experiments, and analyses to further support our findings.

A. Additional Explanations of SMP-Attack

A.1. Connection to Existing Feature-based Attacks

The proposed SMP-Attack method introduces a general attack framework that extends existing feature importance-based attack approaches by incorporating semantics-aware multi-granularity patchout and multi-stage optimization. In the following, we elaborate on the relationship between our SMP-Attack and other feature-based attack approaches.

Theorem 1. *With specific patch settings $\mathbb{P}$ and selected layers $\mathbb{L}$ across total S training stages, the proposed SMP attack can be reduced to FIA [55] and RPA [63] respectively.*

Proof of Theorem 1. The optimal attack settings for FIA and RPA are assumed to be as follows: p_m (modify probability), M (ensemble number), k (selected layer), T (total iteration), and $\mathbb{P}_{\text{FIA}}=\{0,\cdots,0\}$, $\mathbb{P}_{\text{RPA}}=\{d_1,\cdots,d_M\}$ (patch setting).

On the one hand, we modify the stage-wise iteration and layer settings (*i.e.*, $\mathbb{L}_{\text{SMP}}=\{\ell^{(1)},\cdots,\ell^{(S)}\}$, $\{T^{(1)},\cdots,T^{(S)}\}$) of multi-stage optimization in total S stages as: $T^{(1)}+\cdots+T^{(S)}=T$, $\ell^{(1)}=\cdots=\ell^{(S)}=k$. Thus, our multi-stage training strategy becomes to single-stage training with respect to k-th layer of surrogate model. On the other hand, we modify the stage-wise patch settings (*i.e.*, $\{\mathbb{P}^{(1)}_{\text{SMP}},\cdots,\mathbb{P}^{(M)}_{\text{SMP}}\}$) of semantics-aware multi-granularity patchout in total S stages as: $d_1^{(s)}=d_1,\cdots,d_M^{(s)}=d_M$ and $c_1^{(s)}=+\infty,\cdots,c_M^{(s)}=+\infty$ for $s=\{1,\cdots,S\}$. According to patch definitions in Eqs. 4-5, D_{sc} becomes to D_s as the hyperparameter c approaches to positive infinity. Thus, our multi-granularity patch becomes to patch size-based single granularity.

Because of the identical patch settings, our SMP-Attack simplifies to RPA. By further setting $d_1^{(s)}=\cdots=d_M^{(s)}=0$ for each s-th stage, the patch definition of our SMP-Attack reduces to a pixel-level representation (see Eq. 3), resulting in the equivalence between the simplified SMP-Attack and FIA, which completes the proof. □

Thus it can be seen that our SMP-Attack represents a more generalized attack framework compared to existing feature-based methods. Contrarily, FIA [55] and RPA [63] can be regarded as simplified versions of our SMP-Attack.

A.2. Properties of Multi-granularity Patchout

To overcome the limitations of single granularity, we refine the image patch generation process by incorporating color information, introducing shape-based granularity alongside the existing patch size-based granularity.

Comparison of Single-granularity vs. Multi-granularity Patches. In Fig. 5, we compare the generated patches with different settings. (a) By leveraging the color distance, SMP generates *irregular patches with varying sizes*, in contrast to regular patches produced by RPA. (b) By fixing d and varying c, SMP generates *irregular patches with consistent sizes but varying shapes*. (c) By varying d and c, SMP generates *irregular patches with varying sizes and shapes*. As depicted in the 4th row of Fig. 5, the segmentation of the "dog" object verifies that multi-granularity patch effectively preserves key semantic features relevant to the object itself. By maintaining essential features (*e.g.*, edges, textures, colors, etc.) while eliminating non-essential ones, our multi-granularity method helps reduce overfitting and interference from model-specific features.

Efficient Implementation of Multi-granularity Patch Generation. Due to the conceptual similarity with superpixels [2, 23, 28, 58], we define D_{sc} in spatial space (*Row* and *Column* components: r, c) and LAB color space (*L*, *A* and *B* components: l, a, b), *i.e.*, $d_{spatial}(\boldsymbol{x}_{ij},\boldsymbol{c}_p^n)=(\boldsymbol{x}_{ij}[r]-\boldsymbol{c}_p^n[r])^2+(\boldsymbol{x}_{ij}[c]-\boldsymbol{c}_p^n[c])^2$ and $d_{color}(\boldsymbol{x}_{ij},\boldsymbol{c}_p^n)=(\boldsymbol{x}_{ij}[l]-\boldsymbol{c}_p^n[l])^2+(\boldsymbol{x}_{ij}[a]-\boldsymbol{c}_p^n[a])^2+(\boldsymbol{x}_{ij}[b]-\boldsymbol{c}_p^n[b])^2$. Multi-granularity patch can be efficiently generated with linear time complexity using simple linear iterative clustering (SLIC) [2]. At each iteration, within a local $(2d+1)\times(2d+1)$ neighborhood, each pixel $\boldsymbol{x}_{ij}$ is assigned to its nearest patch $\boldsymbol{x}_p^n$, and the patch center is updated as $\boldsymbol{c}_p^n=1/|\boldsymbol{x}_p^n|\sum_{\boldsymbol{x}_{ij}\in\boldsymbol{x}_p^n}\boldsymbol{x}_{ij}$, where $|\boldsymbol{x}_p^n|$ is the number of pixels in the n-th patch. In practice, a few iterations (*e.g.*, $4\sim10$) are sufficient for convergence. Our core innovation lies in introducing multi-granularity into aggregated gradient computation. SLIC is employed as a tool within our SMP framework to realize this idea, as it is designed to generate superpixels with local perceptual consistency under fixed d and c. By adjusting parameters d and c, SLIC can produce superpixels of varying sizes and shapes, thereby suppressing irrelevant

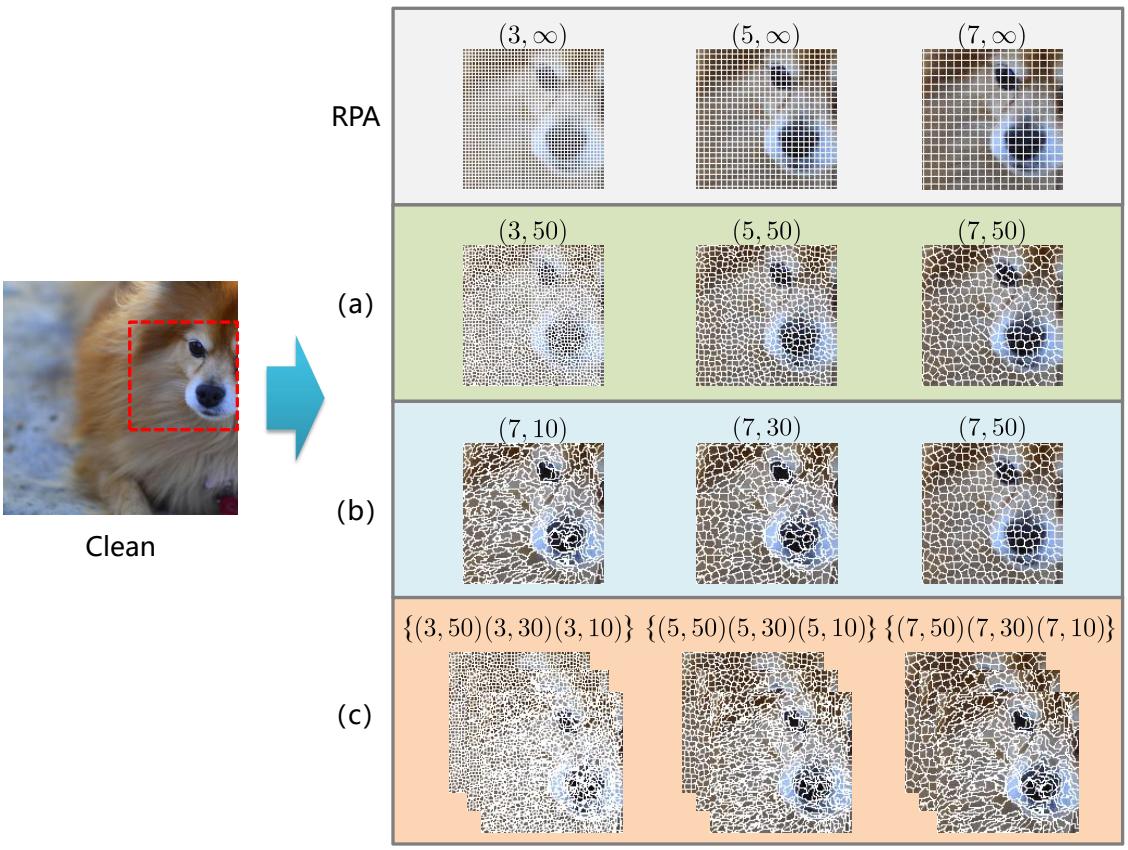


Figure 5. Comparison of the generated patches under different parameter settings. In contrast to RPA (1st row) that generates regular/non-deformable patches with varying sizes (*i.e.*, single-granularity), our proposed SMP method (4th row) generates irregular/deformable patches with varying sizes and shapes (*i.e.*, multi-granularity). For clarity, we visualize only a specific region of the clean input image, highlighted by a red dashed box.

features during aggregation, particularly along non-object-related boundaries (see Fig. 5). This exploration of uncertainty helps our SMP retain object-related features and improve gradient aggregation quality.

Exploration of Various Patch Settings. We compare the attack transferability under various patch settings in Fig. 6, where the source models are Vgg-16 (left) and IncRes-v2 (right) respectively, with the target models indicated on the x-axis. It is obvious that the multi-granularity ($\mathcal{M}_e$) attack demonstrates superior performance over single-granularity attacks ($\mathcal{M}_a$, $\mathcal{M}_b$, $\mathcal{M}_c$, $\mathcal{M}_d$).

By incorporating a semantics-aware multi-granularity mechanism, the proposed SMP-Attack enhances feature pattern diversity and improves aggregate gradient quality, thereby achieving superior adversarial transferability over existing feature-based attacks.

A.3. Properties of Multi-stage Optimization

To overcome the limitations of single-stage optimization, we refine the iterative perturbation training process by incorporating multi-layer semantic information (particularly from shallow and intermediate DNN layers), and introducing a multi-stage training strategy to replace the existing single-stage training strategy.

In the following experiments, adversarial examples are generated using the source models specified in the subfigure titles and are evaluated against the target models listed along the x-axis.

Design and Configuration of Multi-Stage Training Strategies. Specifically, we divide the total number of iterations T as (T_S, T_M, T_D), where T_S, T_M, T_D represent the iterations assigned to shallow/middle/deep layer based losses, and the corresponding patch settings are denoted as $\mathbb{P}_S$, $\mathbb{P}_M$, $\mathbb{P}_D$. We select ($Conv1\_1$, $Conv3\_3$, $Conv5\_3$) from Vgg-16 and ($Conv2d\_1a\_3x3$, $Conv2d\_4a\_3x3$, $Conv2d\_7b\_1x1$) from IncRes-v2 as shallow, middle, deep layers respectively to

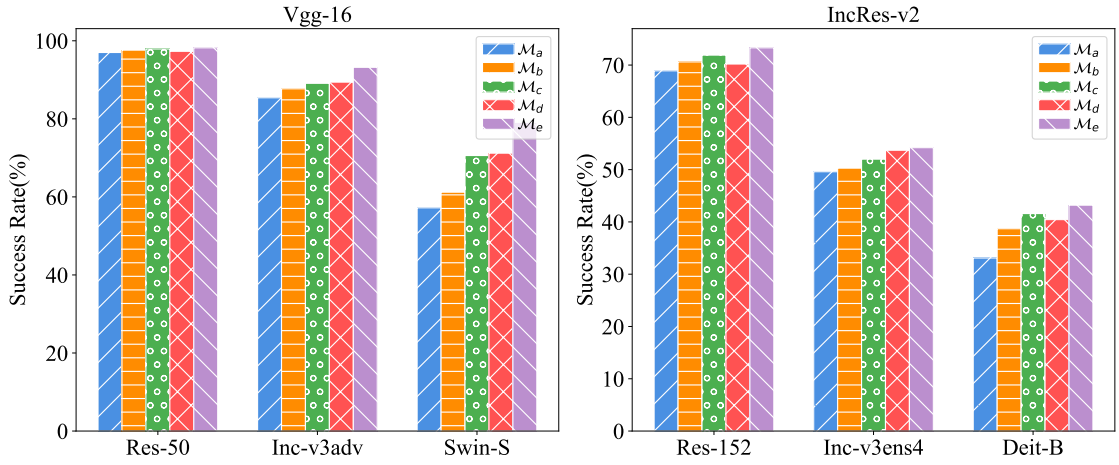


Figure 6. Comparison of the attack transferability under various patch settings. The attack model settings are defined as follows: $\mathcal{M}_a$ (pixel-based single granularity employed by FIA, *i.e.*, $d = 0$), $\mathcal{M}_b$ (regular patch size-based single granularity employed by RPA, *i.e.*, varying d and setting $c = +\infty$), $\mathcal{M}_c$ (irregular patch size-based single granularity employed by SMP, *i.e.*, varying d and fixing c), $\mathcal{M}_d$ (irregular patch shape-based single granularity employed by SMP, *i.e.*, fixing d and varying c), and $\mathcal{M}_e$ (irregular patch size and shape-based multi-granularity employed by SMP, *i.e.*, varying d and c).

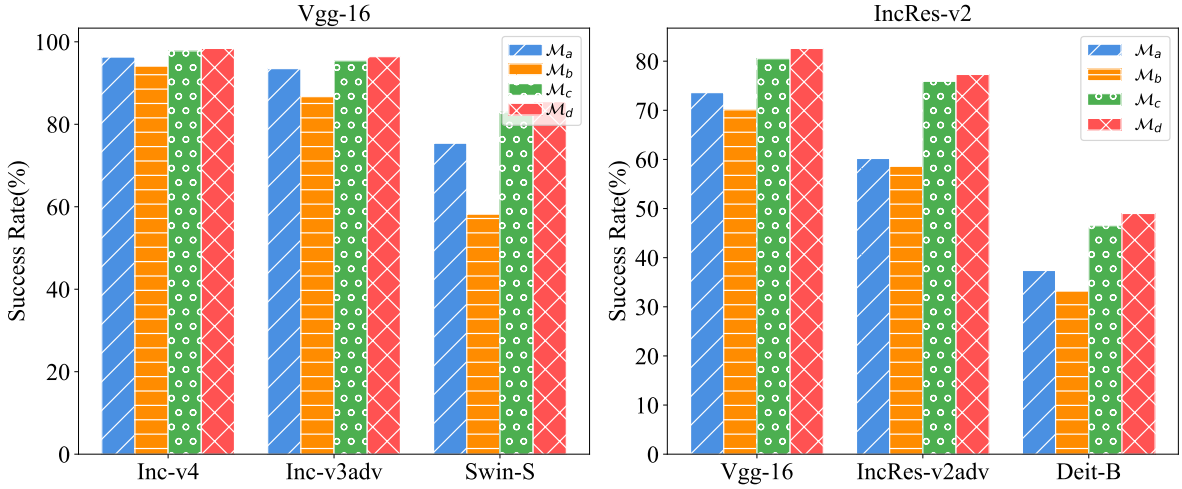


Figure 7. Comparison of the attack transferability between conventional training strategies and our multi-stage training strategy. The attack model settings are defined as follows: $\mathcal{M}_a$ (conventional joint optimization of the sum of shallow layer-based loss and intermediate layer-based loss), $\mathcal{M}_b$ (alternative optimization between shallow layer-based loss and middle layer-based loss), $\mathcal{M}_c$ (multi-stage optimization employed by SMP, $(T_S, T_M, T_D) = (1, 9, 0)$, $\mathbb{P}_S$=$\{(1, 100), (2, 100), (3, 100)\}$, $\mathbb{P}_M$=$\{(3, 50), (4, 40), (5, 30)\}$, $\mathbb{P}_D = \emptyset$), and $\mathcal{M}_d$ (multi-stage optimization with more diverse patch settings by SMP, $(T_S, T_M, T_D) = (1, 9, 0)$, $\mathbb{P}_S$=$\{(1, 95), (2, 95), (3, 95), (1, 100), (2, 100), (3, 100)\}$, $\mathbb{P}_M$=$\{(3, 40), (4, 30), (5, 20), (3, 50), (4, 40), (5, 30)\}$, $\mathbb{P}_D = \emptyset$).

construct $\mathbb{L}$. Patch settings are $\mathbb{P}_S = \{(1, 95), (2, 95), (3, 95), (1, 100), (2, 100), (3, 100)\}$, $\mathbb{P}_M = \mathbb{P}_D = \{(3, 40), (4, 30), (5, 20), (3, 50), (4, 40), (5, 30)\}$. Here, we select the patch size parameter d from the set $\{1, 2, 3, 4, 5\}$, and adjust the patch shape parameter c with slight variations around $C_{Smooth} \approx 100$ and $C_{Sharp} \approx 20$.

Comparison of Different Training Strategies. Fig. 7 illustrates the effect of different training strategies on the attack transferability. We observe that multi-stage training strategies ($\mathcal{M}_c$, $\mathcal{M}_d$) outperforms conventional training strategies ($\mathcal{M}_a$, $\mathcal{M}_b$), such as joint optimization and alternative optimization. Furthermore, we conduct the ablation study of different layer settings in multi-stage optimization framework. As shown in Fig. 8, the proposed simple yet effective two-stage training strategy ($\mathcal{M}_a$) achieves the best ASR performance compared to other multi-stage training strategies ($\mathcal{M}_b$, $\mathcal{M}_c$, $\mathcal{M}_d$, $\mathcal{M}_e$, $\mathcal{M}_f$). Compared to $\mathcal{M}_a$, $\mathcal{M}_b$ exhibits a certain degree of performance degradation, as excessive reliance on shallow layer information causes the update direction of adversarial examples to diverge greatly from the original optimization trajectory

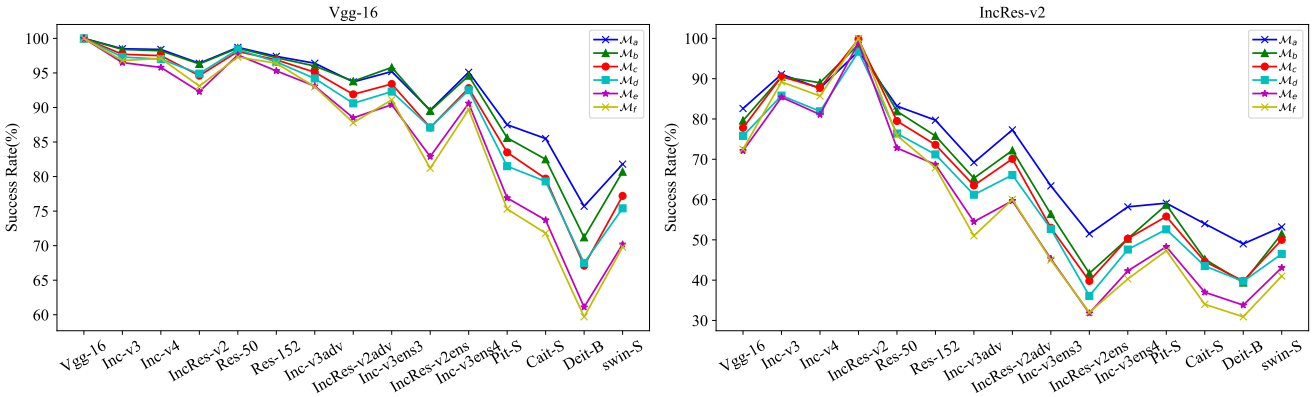


Figure 8. Comparison of the attack transferability under different attack settings within multi-stage optimization framework. The attack model settings are defined as follows: $\mathcal{M}_a$, $(T_S, T_M, T_D) = (1, 9, 0)$; $\mathcal{M}_b$, $(T_S, T_M, T_D) = (2, 8, 0)$; $\mathcal{M}_c$, $(T_S, T_M, T_D) = (1, 8, 1)$; $\mathcal{M}_d$, $(T_S, T_M, T_D) = (0, 9, 1)$; $\mathcal{M}_e$, $(T_S, T_M, T_D) = (0, 8, 2)$; $\mathcal{M}_f$, $(T_S, T_M, T_D) = (2, 6, 2)$.

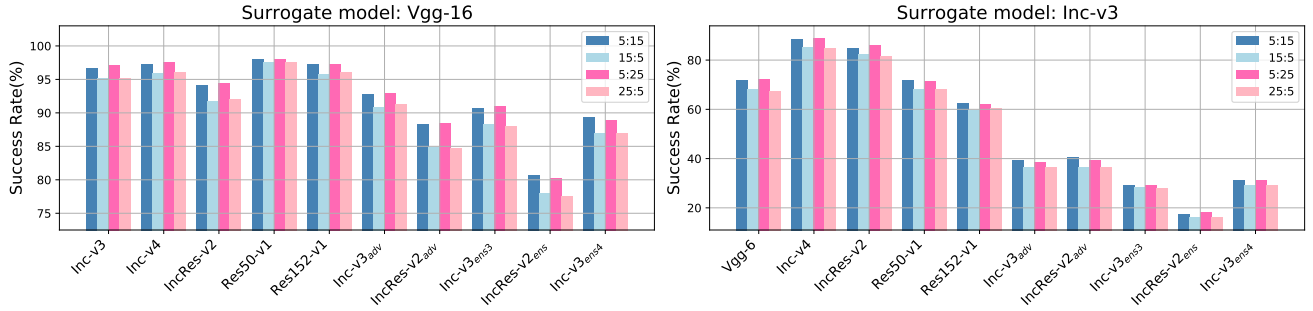


Figure 9. Comparison of attack transferability under different two-stage training configurations with extended training iterations.

guided by middle layer-based feature importance loss. On the other hand, overuse of deep layer information (*e.g.*, $\mathcal{M}_e$ and $\mathcal{M}_f$) results in overfitting to the surrogate model.

Implementation Guidelines for Extended Training Iterations. Fig. 8 validate the effectiveness of our two-stage training strategy with $(T_s, T_m) = (1, 9)$. This suggests that moderate use of shallow layers (*e.g.*, 10%–50%) is optimal, while overuse should be avoided. Building upon this, we further investigate the effectiveness of our two-stage training strategy under a more practical setting with extended training iterations. For larger iterations (*e.g.*, $T = 20$ or 30), we compare configurations $(T_s, T_m) = (5, 15)$ and $(5, 25)$ with their counterparts $(15, 5)$ and $(25, 5)$. As shown in Fig. 9, using fewer shallow-layer iterations ($T_s = 5$) consistently outperforms excessive ones ($T_s = 15$ or 25).

By optimizing multiple layer-based loss across different stages, the proposed SMP-Attack systematically explores both generic essential features from shallow layers and model-agnostic significant features from intermediate layers, thereby achieving enhanced transferability.

B. Additional Experiments of SMP-Attack

B.1. Comparison of the Transferability of Different Attack Methods using Additional Surrogate Models

In this section, we quantitatively compare our SMP methods with other feature-based attack methods using additional surrogate CNN models, including Res50-v1 [20], Vgg-19 [41], Inc-v3 [6].

For the experimental results given in Sec. 4 and the Appendix, the selected layers of each surrogate model are presented in Table 3. Firstly, the middle layer setting is completely consistent with previous methods. Secondly, the shallow layer is generally set to the first DNN layer. Exceptionally, in ResNet models, the shallow layer setting is set to the last layer of 1st block to aligns with its intermediate layer setting due to the residual connection structure. It is important to emphasize that our SMP attack follows the layer setting of previous studies and consistently select first DNN layer to incorporate generic feature patterns. Our method achieves better attack performance without any fine-tuning for individual surrogate models, thus avoiding significant parameter tuning costs. Additionally, in Sec. A.1 of this Appendix, our SMP method is theoretically

Surrogate Model	Shallow Layer	Middle Layer
Vgg-16	Conv1_1	Conv3_3
Vgg-19	Conv1_1	Conv3_4
Inc-v3	Conv2d_1a_3x3	Mixed_5b
IncRes-v2	Conv2d_1a_3x3	Conv2d_4a_3x3
Res50-v1	The last layer of 1st block	The last layer of 2nd block
Res152-v1	The last layer of 1st block	The last layer of 2nd block

Table 3. The layer configurations for our multi-stage optimization applied to each surrogate model.

	Attack	Vgg-16	Inc-v3	Inc-v4	IncRes-v2	Res50-v1	Res152-v1	Inc-v3adv	IncRes-v2adv	Inc-v3ens3	IncRes-v2ens	Inc-v3ens4	Avg_{bb}
Res50-v1	FIA	90.4	87.2	82.4	78.4	99.8	96.7	77.9	69.3	70.6	56.7	71.0	80.0
	RPA	94.1	91.5	88.0	87.9	99.9	98.0	83.3	77.5	77.9	66.1	75.0	85.4
	NAA	76.2	77.4	75.9	70.1	92.4	73.4	36.4	39.5	31.6	19.5	32.1	56.8
	DANAA	69.1	72.9	69.9	64.3	91.4	70.0	35.0	37.5	31.3	19.0	31.0	53.8
	NEAA	67.7	70.4	66.6	61.2	90.1	67.6	33.0	35.8	29.0	17.6	29.1	51.6
	BFA	81.7	75.3	68.8	67.8	99.5	88.0	62.6	55.5	58.9	45.4	58.5	66.3
	MFAA	94.2	89.5	85.4	83.5	**100.0**	97.8	78.9	69.6	72.2	56.4	68.8	81.5
	SMP(Ours)	**95.6**	**93.0**	**90.7**	**90.1**	99.9	**98.2**	**86.3**	**81.3**	**81.0**	**71.1**	**81.1**	**88.0**
	PIDI-FIA	96.5	91.8	88.1	87.2	99.7	97.8	88.1	81.9	82.7	68.9	80.1	87.5
	PIDI-RPA	97.4	94.9	93.5	92.1	**99.8**	98.8	90.4	86.8	87.8	78.7	85.4	91.4
	PIDI-NAA	89.5	89.7	85.9	84.6	96.4	88.9	51.0	57.1	45.4	28.6	45.8	69.4
	PIDI-DANAA	87.6	89.3	86.6	84.5	97.0	90.4	53.9	59.9	50.1	33.6	47.6	71.0
	PIDI-NEAA	86.5	87.4	84.1	82.3	96.5	88.5	50.2	55.4	45.3	29.7	45.0	68.3
	PIDI-BFA	83.2	78.4	69.9	68.4	99.7	89.1	64.3	57.1	59.8	46.6	59.7	67.7
	PIDI-MFAA	**98.4**	95.4	93.5	92.8	**99.8**	98.6	89.4	86.3	85.8	73.5	82.5	90.5
	PIDI-SMP(Ours)	**98.4**	**96.1**	**94.8**	**94.8**	**99.8**	**98.9**	**93.5**	**91.6**	**92.2**	**85.4**	**90.8**	**94.2**
Vgg-19	FIA	99.4	94.9	95.7	91.8	95.7	93.4	93.1	87.4	91.0	83.1	89.7	92.3
	RPA	99.8	96.3	96.2	94.6	97.3	96.2	92.4	87.4	89.7	83.4	89.6	93.0
	NAA	96.2	91.0	91.2	87.9	88.4	82.6	62.9	65.2	64.0	43.1	62.7	75.9
	DANAA	94.2	89.5	88.6	85.1	86.3	80.1	61.4	63.3	59.4	42.6	59.0	73.6
	NEAA	93.2	88.0	87.6	83.9	85.0	78.2	59.8	60.6	56.9	40.5	57.9	72.0
	BFA	99.2	91.9	91.1	86.5	92.5	90.2	87.8	81.4	85.6	76.3	83.5	87.8
	MFAA	99.8	97.5	**97.6**	**95.6**	97.6	96.8	**95.9**	91.8	94.0	89.2	93.9	95.4
	SMP(Ours)	**100.0**	**97.5**	97.2	95.5	**98.7**	**97.2**	95.4	**92.9**	**94.5**	**89.7**	**94.3**	**95.7**
	PIDI-FIA	99.8	96.4	96.5	93.7	97.1	95.6	92.9	88.8	89.8	85.2	91.0	93.3
	PIDI-RPA	99.7	97.9	97.1	95.9	98.2	96.8	94.6	90.3	91.7	86.5	92.4	94.6
	PIDI-NAA	97.4	94.0	94.1	90.1	92.6	88.1	68.6	70.2	65.9	48.0	64.2	79.4
	PIDI-DANAA	96.8	94.1	92.3	88.8	91.5	87.9	68.2	69.0	65.9	48.2	63.5	78.7
	PIDI-NEAA	96.3	92.6	91.0	88.2	91.0	86.2	66.1	68.7	65.0	46.4	63.8	77.8
	PIDI-BFA	99.7	92.4	91.4	87.1	92.8	90.9	88.2	81.8	86.9	77.6	85.4	88.6
	PIDI-MFAA	**100.0**	**99.0**	**98.5**	**96.8**	98.5	**98.3**	**96.5**	**93.8**	**94.2**	**91.2**	**95.4**	**96.6**
	PIDI-SMP(Ours)	**100.0**	97.4	97.5	95.9	**98.6**	97.1	96.1	92.3	93.7	90.2	95.0	95.8
Inc-v3	FIA	71.9	98.3	83.5	79.3	69.6	64.2	54.7	55.2	43.5	23.4	41.3	62.3
	RPA	75.3	98.6	85.7	84.0	72.7	68.4	59.0	59.5	45.5	26.3	44.1	65.4
	NAA	73.8	97.0	84.3	82.2	74.2	68.3	60.6	62.5	50.1	31.7	50.5	66.8
	DANAA	75.4	97.8	87.3	84.6	77.2	72.1	65.2	68.3	**55.4**	33.5	**55.1**	70.2
	NEAA	79.8	99.1	89.4	86.7	79.9	74.5	65.0	65.0	52.3	31.7	52.8	70.6
	BFA	79.2	**99.9**	95.8	91.7	76.8	74.4	**65.7**	62.5	50.8	**39.2**	44.7	71.0
	MFAA	81.9	97.6	86.5	84.6	78.6	72.8	65.2	64.9	51.9	32.5	44.9	69.2
	SMP(Ours)	**82.3**	99.7	**92.9**	**90.7**	**83.6**	**75.1**	62.9	**66.2**	50.4	28.4	49.4	**71.1**
	PIDI-FIA	81.4	98.5	87.9	85.0	78.9	74.9	60.0	61.9	46.3	26.8	49.8	68.3
	PIDI-RPA	83.7	98.5	89.6	88.7	83.0	79.8	64.8	65.6	49.6	29.4	53.8	71.5
	PIDI-NAA	82.1	97.2	87.2	86.3	81.2	79.0	65.7	66.9	54.3	33.5	54.3	71.6
	PIDI-DANAA	85.5	97.9	89.5	89.2	83.6	81.9	71.1	**72.5**	**62.4**	**37.9**	**61.5**	75.7
	PIDI-NEAA	88.2	**99.0**	92.7	91.5	88.4	85.1	**71.3**	71.9	55.7	35.3	58.0	**76.1**
	PIDI-BFA	81.3	**100.0**	**97.0**	**93.9**	78.4	76.1	70.8	65.3	53.4	44.3	50.8	71.1
	PIDI-MFAA	87.1	98.0	89.9	89.0	87.2	83.8	68.0	70.5	56.2	34.0	51.2	74.1
	PIDI-SMP(Ours)	**89.1**	99.0	92.4	92.7	**89.3**	**85.2**	68.1	69.6	54.5	35.0	59.3	75.8

Table 4. The attack success rate (%) of various transfer-based attacks against six CNN models and five defended CNN models. The average ASR of all black-box models are reported. The best results are highlighted in bold red.

proven to be a generalized feature-based attack approach, explaining its superior performance over other attack methods. Even though the attack performance could be further enhanced through layer-specific fine-tuning for each model, this is not the primary focus of our work. Instead, we aim to introduce a general multi-stage optimization framework to effectively learn multi-layer features.

As a supplement to the main experiments in Sec. 4, the proposed SMP attack are further compared with state-of-the-art feature-based attacks using additional surrogate DNN models. Table 4 presents the black-box ASR results on six undefended CNN models, including Vgg-16 [41], Inc-v3 [6], Inc-v4 [7], IncRes-v2 [7], Res50-v1 [20], and Res152-v1 [20], and five defended CNN models, including Inc-v3adv and IncRes-v2adv [26], Inc-v3ens3, Inc-v3ens4 and IncRes-v2ens [45]. Table 5 presents the black-box ASR results on eight undefended ViT models, including PiT-S [21], CaiT-S [44], DeiT-B [43], Swin-B[31], ViT-B [56], Visformer-S [5], Convit-B[12], and Twins-PCPVT-B [8], and three defended ViT models, including Deit-Sadv [3], Swin-Badv [36], and XCit-S12adv [9]. As shown in Tables 4-5, our SMP-Attack achieves the highest average ASR (highlighted in bold red) on all black-box settings.

	Attack	PiT-S	CaiT-S	DeiT-B	Swin-B	ViT-B/16	Visformer-S	Convit-Base	Twins-PCPVT-B	Deit-Sadv	Swin-Badv	xcit_Sadv	Avg_{bb}
Res50-v1	FIA	56.2	48.1	39.0	46.4	37.1	63.7	41.3	52.0	23.8	15.8	30.0	41.2
	RPA	65.1	60.4	48.5	56.5	45.1	71.9	51.2	64.8	**25.4**	17.5	30.5	48.8
	NAA	50.0	35.2	35.8	48.8	27.2	60.2	35.6	54.6	15.7	10.3	16.1	35.4
	DANAA	46.4	32.4	34.8	45.7	25.3	55.8	32.6	51.4	15.3	9.9	14.9	33.1
	NEAA	43.0	30.1	33.9	41.6	24.2	53.2	30.5	48.4	15.4	9.7	14.7	31.3
	BFA	50.9	41.5	38.9	44.6	30.0	50.5	40.1	47.6	22.4	14.5	26.9	37.1
	MFAA	58.0	50.3	42.3	49.0	39.7	64.1	43.5	56.2	24.2	16.2	**30.3**	43.1
	SMP(Ours)	**71.4**	**65.7**	**56.5**	**62.8**	**48.5**	**77.9**	**54.9**	**70.6**	24.4	**16.9**	28.6	**52.6**
	PIDI-FIA	68.6	61.4	53.1	57.8	52.5	74.6	52.0	67.0	33.5	25.8	37.0	53.0
	PIDI-RPA	78.2	73.0	64.5	69.3	59.6	82.5	65.5	76.3	35.3	27.8	37.4	60.9
	PIDI-NAA	70.2	57.7	58.7	66.6	46.2	77.0	57.7	72.3	25.0	18.1	24.7	52.2
	PIDI-DANAA	73.5	59.8	60.0	70.3	50.3	77.4	59.4	75.3	24.8	18.7	23.4	53.9
	PIDI-NEAA	69.9	56.3	58.7	66.3	44.5	74.7	56.0	72.0	24.6	18.4	24.6	51.5
	PIDI-BFA	71.2	58.4	57.9	65.2	50.5	77.1	57.6	72.4	32.8	27.1	30.5	54.6
	PIDI-MFAA	74.8	66.5	58.4	62.3	58.0	80.0	58.9	73.7	33.7	26.2	36.6	57.2
	PIDI-SMP(Ours)	**85.1**	**82.5**	**74.3**	**78.2**	**70.4**	**89.0**	**74.5**	**84.0**	**38.7**	**32.5**	**42.0**	**68.3**
Vgg-19	FIA	74.3	69.8	55.7	65.2	60.8	82.4	56.3	71.1	34.8	24.4	36.3	57.4
	RPA	78.9	76.0	65.7	71.8	62.3	83.5	64.0	72.7	36.4	25.6	37.1	61.3
	NAA	70.4	58.1	53.9	65.9	49.1	77.4	52.4	70.6	25.9	16.9	23.5	51.3
	DANAA	65.8	55.1	51.8	62.3	45.3	73.1	50.6	67.5	24.2	16.2	23.9	48.7
	NEAA	64.7	53.8	51.0	60.9	44.6	71.4	49.0	65.1	24.0	15.2	22.8	47.5
	BFA	71.1	67.4	57.2	63.6	54.7	77.2	55.1	68.4	31.3	21.1	31.0	54.4
	MFAA	80.4	79.0	65.6	73.1	67.9	87.4	63.2	78.7	35.7	26.6	37.5	63.2
	SMP(Ours)	**85.4**	**83.8**	**74.3**	**81.7**	**74.2**	**89.8**	**71.7**	**85.2**	**37.5**	**25.8**	**37.9**	**67.9**
	PIDI-FIA	76.6	74.1	62.4	66.5	65.0	84.2	61.5	74.0	40.6	30.2	40.4	61.4
	PIDI-RPA	79.6	75.8	67.8	70.3	62.2	83.1	65.6	73.2	43.5	31.2	42.3	63.1
	PIDI-NAA	79.1	68.3	66.2	71.6	58.3	81.3	63.4	76.0	33.9	22.3	29.0	59.0
	PIDI-DANAA	76.5	68.4	64.8	72.0	57.7	80.0	62.3	74.8	32.3	21.8	28.4	58.1
	PIDI-NEAA	76.0	67.2	63.3	70.3	54.8	77.6	62.2	74.3	32.1	21.0	28.3	57.0
	PIDI-BFA	79.4	73.7	68.5	70.1	66.7	83.4	63.7	77.1	39.2	28.5	36.5	62.3
	PIDI-MFAA	84.9	82.5	71.5	74.4	72.3	89.7	68.6	80.3	42.6	31.7	41.0	67.2
	PIDI-SMP(Ours)	**86.0**	**84.0**	**73.0**	**80.5**	**74.8**	**90.5**	**72.6**	**85.2**	**46.6**	**33.0**	**42.8**	**69.9**
Inc-v3	FIA	46.6	37.0	32.0	28.2	26.7	46.4	29.0	42.4	20.7	13.4	21.9	31.3
	RPA	50.9	41.4	36.0	32.4	30.6	54.3	36.1	50.5	20.0	13.9	20.9	35.2
	NAA	52.4	44.1	38.9	50.3	32.5	55.6	39.5	53.2	20.5	14.3	20.2	38.3
	DANAA	52.2	**46.5**	40.2	52.8	33.1	58.0	41.5	56.8	19.8	14.2	20.6	39.6
	NEAA	54.5	45.5	39.7	51.9	33.4	58.2	**42.2**	55.9	21.7	14.9	21.8	40.0
	BFA	54.6	44.7	38.2	50.4	33.7	59.1	40.6	55.5	22.3	15.0	22.1	39.7
	MFAA	54.1	44.3	38.5	49.3	**35.5**	58.7	40.3	55.0	**24.2**	**15.9**	**23.8**	40.0
	SMP(Ours)	**58.5**	43.9	**40.4**	**53.5**	32.2	**62.6**	39.9	**57.8**	19.2	12.9	21.1	**40.2**
	PIDI-FIA	57.8	48.5	44.1	36.3	35.5	55.8	38.0	50.2	29.3	18.0	30.3	40.3
	PIDI-RPA	62.4	51.1	47.5	40.2	42.7	64.2	46.7	60.8	30.0	21.2	31.1	45.3
	PIDI-NAA	52.4	44.1	38.9	50.3	42.9	63.9	50.6	62.2	29.8	19.9	27.9	43.9
	PIDI-DANAA	62.8	56.1	**55.1**	58.2	47.1	68.6	**53.7**	66.4	30.9	20.8	27.8	49.8
	PIDI-NEAA	63.0	57.2	51.2	58.4	**47.2**	69.8	53.2	**66.6**	32.3	**22.2**	30.7	50.2
	PIDI-BFA	62.1	57.3	49.7	50.4	49.5	53.5	49.2	60.6	31.3	19.1	27.6	46.4
	PIDI-MFAA	**66.8**	57.4	51.2	58.7	46.0	69.7	50.2	65.3	30.7	20.7	30.5	49.7
	PIDI-SMP(Ours)	66.3	**57.6**	54.6	**60.9**	45.1	**69.9**	51.6	66.3	**32.8**	21.8	**32.0**	**50.8**

Table 5. The attack success rate (%) of various transfer-based attacks against eight ViT models and three defended ViT models. The average ASR of all black-box models are reported. The best results are highlighted in bold red.

B.2. Additional Evaluations on the Compatibility of Multi-stage Training Strategy

In this section, we evaluate the compatibility of our multi-stage training strategy when integrated with other feature-based attacks. Fig. 10 compare the performance of existing feature-based attacks before and after integration with our multi-stage optimization (MSO). Consistent with our SMP's attack settings, existing feature-based attacks update AEs using the shallow layer for one iteration, followed by nine iterations with the intermediate layer.

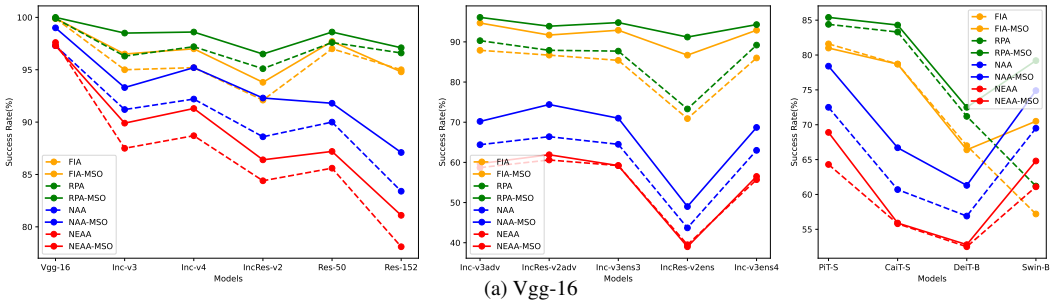
Compared to their respective counterparts (*i.e.*, FIA, RPA, NAA, NEAA), FIA-MSO, RPA-MSO, NAA-MSO, and NEAA-MSO, which generate AEs by our MSO framework, improve the attack performance by a large margin across various CNN/ViT models, thereby validating the compatibility of our multi-stage optimization framework in diverse attack settings.

B.3. Comparison of the Efficiency of Different Attack Methods

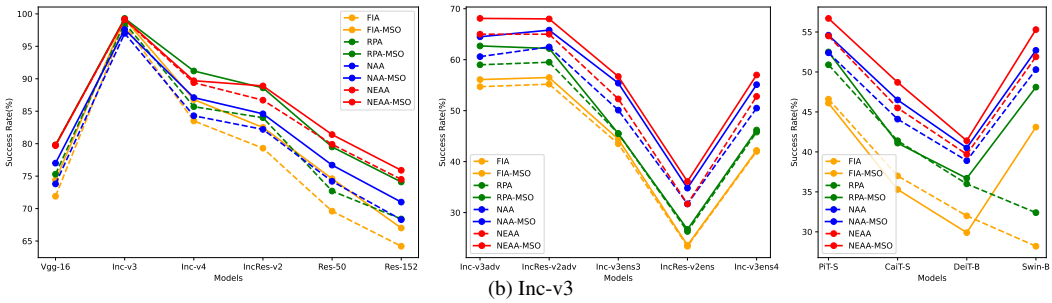
In this section, we assess the attack efficiency on the ImageNet dataset, using the average number of iterations t required for the DNN model's first misclassification and the average runtime per image as the evaluation metrics.

We calculate the metric $\hat{t}$ over all the testing images, *i.e.*, $\hat{t} = (1/N)\sum_{i=1}^{N} t_i$, where N is the total number of images and t_i represents the number of iterations required for i-th image. Table 6 shows the attack efficiency of different methods across various undefended/defended CNN models. Because of the multi-stage training mechanism, our SMP method achieves the highest attack efficiency in all the black-box settings. Furthermore, Compared to MFAA, which integrates deep and intermediate layers, the proposed SMP demonstrates better attack efficiency by leveraging shallow and intermediate layers.

In addition, we evaluate the average runtime per image (in seconds). As shown in Table 7, SMP achieves faster runtime than RPA under the single-stage setting, benefiting from its efficient C++ implementation. Under the multi-stage setting, SMP maintains competitive efficiency, as the stage-wise training design does not introduce significant computational overhead.



(a) Vgg-16



(b) Inc-v3

Figure 10. The impact of combining multi-stage optimization (MSO) with other feature-based attacks on black-box ASR against undefend CNNs (left), defended CNNs (middle), and ViT (right) target models. The source models are indicated in the sub-figure titles, while the target models are shown on the x-axis. The dashed line represents the original feature-based attack, while the solid line represents the combination method that generates AEs in our MSO framework.

Attack	Vgg-16	Inc-v3	Inc-v4	IncRes-v2	Res50-v1	Inc-v3adv	IncRes-v2adv	Inc-v3ens3	IncRes-v2ens
FIA	5.57	5.97	6.65	6.99	4.15	6.51	7.43	6.88	8.05
RPA	4.81	5.15	5.76	6.06	3.50	5.88	6.69	6.30	7.34
NAA	6.52	6.17	6.40	7.19	5.64	8.72	8.62	8.84	9.71
DANAA	8.04	7.67	8.01	8.76	7.13	9.57	9.62	9.68	10.25
NEAA	6.37	6.14	6.37	7.09	5.69	8.55	8.51	8.62	9.56
MFAA	4.79	5.11	5.72	6.06	3.32	6.03	6.78	6.35	7.52
SMP	**4.74**	**5.07**	**5.58**	**5.93**	**3.30**	**5.82**	**6.54**	**6.16**	**7.29**

Table 6. Comparison of attack efficiency of different feature-based methods. The source model is Res152-v1, and the target models are indicated in the 1st row. The best results are highlighted in bold red, where lower values represent better efficiency.

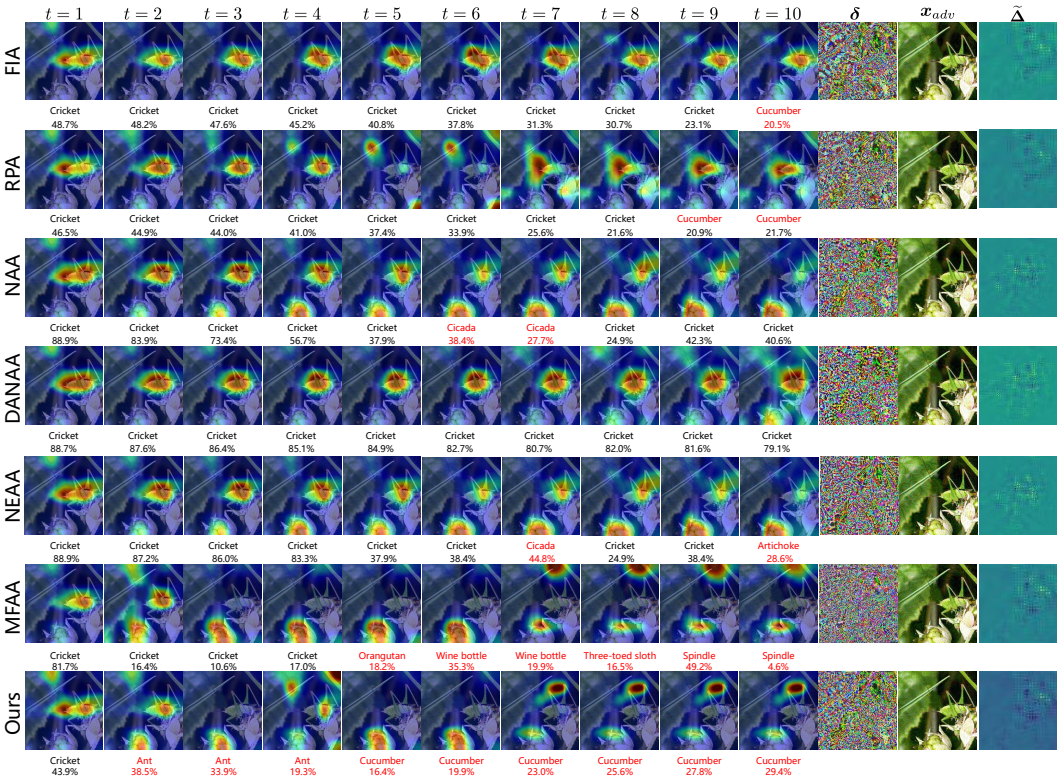
Attack	FIA	RPA	NAA	NEAA	SMP
Without Multi-Stage Optimization	2.74	6.11	2.31	17.82	5.15
With Multi-Stage Optimization	4.97	10.89	5.46	40.64	11.25

Table 7. Comparison of the average runtime across different feature-based methods under single-stage and multi-stage training strategies.

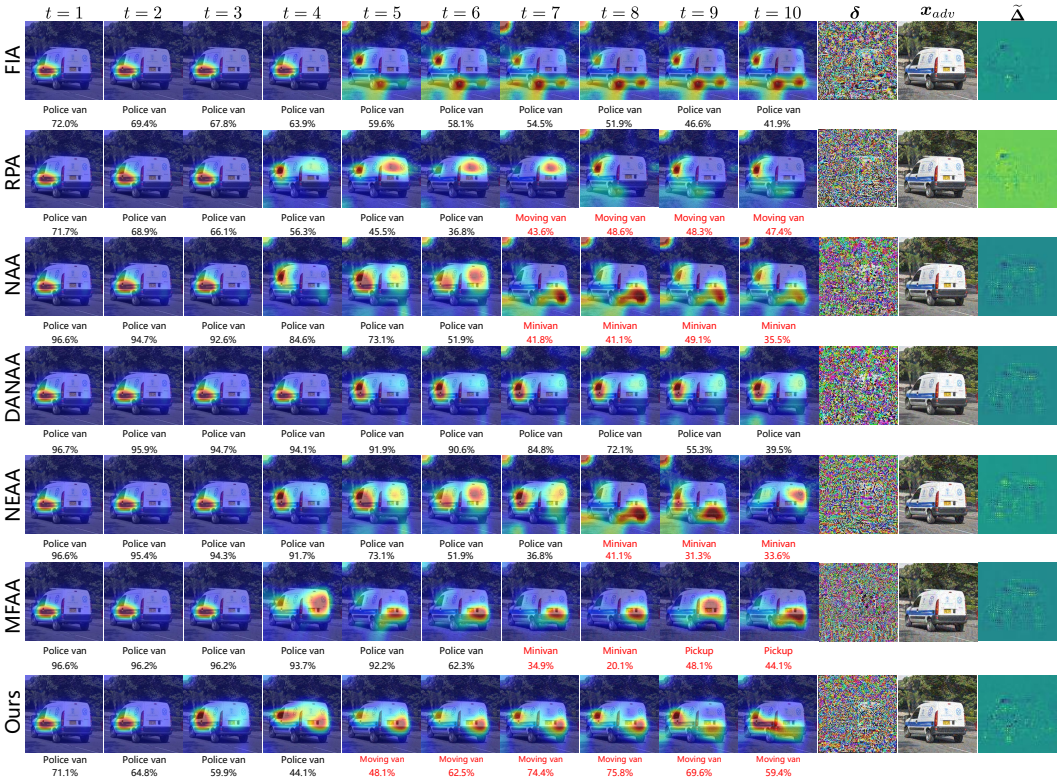
B.4. Additional Comparison of Iterative Attack Process of Different Attack Methods

Fig. 11 illustrates the iterative attack process of different feature-based methods. Adversarial examples $\boldsymbol{x}_{adv}$ generated by the source model (IncRes-v2 [7], Res50-v1 [20]) are used to attack the target model (Vgg-16 [41]). In the first ten columns, each row shows the attention transition in the class activation map [40] for each attack method during the training iterations ($t = 1, \cdots, 10$), along with predicted labels and confidence levels. Additionally, in the last three columns, we presents perturbations $\boldsymbol{\delta}$, adversarial examples $\boldsymbol{x}_{adv}$, and aggregate gradients $\widetilde{\boldsymbol{\Delta}}$ for each attack.

As shown in the last column of Fig. 11 (a), the proposed SMP produces more accurate aggregated gradients than other state-of-the-art methods, benefiting from the superiority of multi-granularity over single-granularity. Compared to other attacks, which gradually disrupts DNN prediction with multiple iterations, our SMP enables a rapid label change from 1st to 2nd iterations, *i.e.*, "Cricket"⇒"Ant". As can be seen, SMP utilizes the multi-stage training strategy to effectively refine the optimization trajectory for updating AEs from the early stage, *i.e.*, $t = 2$. Building on this, SMP continues to search the adversarial information along this refined optimization trajectory, ultimately resulting in the erroneous prediction "Cucumber" with a higher confidence 29.4%. Thus, the proposed SMP method demonstrates more effective and efficient attacks. Similar results can also be observed in Fig. 11 (b).



(a) IncRes-v2



(b) Res50-v1

Figure 11. Comparison of iterative attack process of different feature-based methods (Source: IncRes-v2/Res50-v1, Target: Vgg-16). Each row illustrates the transition of class activation maps, adversarial perturbation, adversarial example, and aggregate gradient.