

## APPENDIX

### A THEORETICAL JUSTIFICATION OF FEATURE-SENSITIVE GRADIENT SCALING (FSGS)

We provide a formal argument to support the hypothesis that modulating gradients based on token-level activation norms enhances adversarial transferability. Our analysis is grounded in the relationship between semantic informativeness and gradient alignment across models.

#### A.1 PRELIMINARIES

Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$  be a surrogate classifier and  $f' : \mathbb{R}^d \rightarrow \mathbb{R}^K$  a target (black-box) classifier. An adversarial perturbation  $\delta \in \mathbb{R}^d$  is added to input  $\mathbf{x}$  such that  $\|\delta\|_\infty \leq \epsilon$  and  $f(\mathbf{x} + \delta) \neq y$ .

Assume  $\mathbf{x}$  is decomposed into  $T$  tokens with embeddings  $\mathbf{z}_1, \dots, \mathbf{z}_T \in \mathbb{R}^D$ . Denote the loss gradients w.r.t. each token as  $\mathbf{g}_i = \nabla_{\mathbf{z}_i} \mathcal{L}(f(\mathbf{x}), y)$ , and similarly for  $f'$ .

#### A.2 SEMANTIC TOKENS AND GRADIENT ALIGNMENT

Let  $S_{\text{sem}} \subseteq \{1, \dots, T\}$  be the set of semantically informative tokens (e.g., foreground object, discriminative parts). Let  $S_{\text{bg}}$  be its complement (background or irrelevant tokens).

We define the inter-model gradient alignment at token  $i$  as:

$$\text{Align}_i = \cos \theta_i = \frac{\langle \nabla_{\mathbf{z}_i} \mathcal{L}_f, \nabla_{\mathbf{z}_i} \mathcal{L}_{f'} \rangle}{\|\nabla_{\mathbf{z}_i} \mathcal{L}_f\| \cdot \|\nabla_{\mathbf{z}_i} \mathcal{L}_{f'}\|}$$

**Assumption 1.** *Gradients at semantically important tokens exhibit higher cross-model alignment:*

$$\mathbb{E}_{i \in S_{\text{sem}}} [\text{Align}_i] > \mathbb{E}_{i \in S_{\text{bg}}} [\text{Align}_i]$$

This is supported by empirical findings in model interpretability (Abnar & Zuidema, 2020; Lin & Parikh, 2016; Raghu et al., 2021) and our own Grad-CAM visualizations (see Section 4.4).

#### A.3 FEATURE-SENSITIVE GRADIENT SCALING (FSGS)

FSGS assigns a scaling factor  $s_i$  to each token based on its activation norm  $\alpha_i = \|\mathbf{z}_i\|_2$ :

$$s_i = \gamma_{\text{base}} + \lambda(1 - \hat{\alpha}_i), \quad \hat{\alpha}_i = \frac{\alpha_i - \min_j \alpha_j}{\max_j \alpha_j - \min_j \alpha_j + \varepsilon}$$

Tokens with high  $\alpha_i$  (assumed to lie in  $S_{\text{sem}}$ ) receive larger gradients, while low-importance tokens are suppressed.

**Theorem 1** (FSGS Improves Expected Gradient Alignment). *Let  $G = \sum_{i=1}^T \mathbf{g}_i$  be the unscaled gradient and  $G_{\text{FSGS}} = \sum_{i=1}^T s_i \cdot \mathbf{g}_i$  the FSGS-scaled gradient. Under Assumption 1, the cosine alignment between  $G_{\text{FSGS}}$  and the target model’s gradient  $G'$  satisfies:*

$$\cos \theta(G_{\text{FSGS}}, G') > \cos \theta(G, G')$$

*Sketch.* We decompose the total gradient into two subsets:

$$G = \sum_{i \in S_{\text{sem}}} \mathbf{g}_i + \sum_{i \in S_{\text{bg}}} \mathbf{g}_i$$

FSGS scales  $i \in S_{\text{sem}}$  by higher  $s_i$  than those in  $S_{\text{bg}}$ , thus:

$$G_{\text{FSGS}} = \sum_{i \in S_{\text{sem}}} s_i \mathbf{g}_i + \sum_{i \in S_{\text{bg}}} s_i \mathbf{g}_i$$

Since  $\mathbb{E}_{i \in S_{\text{sem}}} [\text{Align}_i] > \mathbb{E}_{i \in S_{\text{bg}}} [\text{Align}_i]$ , amplifying contributions from  $S_{\text{sem}}$  increases the expected alignment between  $G_{\text{FSGS}}$  and  $G'$ . Therefore:

$\cos \theta(G_{\text{FSGS}}, G') > \cos \theta(G, G')$  (by Jensen’s inequality over positively weighted aligned vectors)

□

#### A.4 IMPLICATION

Theorem 1 provides theoretical support for the design of FSGS: boosting gradients from semantically salient tokens leads to improved alignment with gradients from unseen models, thereby enhancing adversarial transferability. This also explains the empirical advantage of FSGS in black-box settings, particularly when transferring from ViTs to CNNs or hybrid models.

## B THE OVERALL FRAMEWORK OF OPTIMIZATION ALGORITHM

### B.1 ALGORITHM

We now present the full optimization framework used to generate adversarial examples in our method. Our algorithm builds on the momentum-based PGD attack (Dong et al., 2018a), and integrates three coordinated components: (1) *Module and Layer-Wise Gradient Modulation* to adjust the contribution of each transformer layer and suppress noisy deep-layer gradients, (2) *Feature-Sensitive Gradient Scaling (FSGS)* to selectively enhance semantically important token gradients, and (3) *Spectral Smoothness Regularization (SSR)* to constrain the perturbation’s frequency content.

Let  $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$  be a clean input,  $y \in \{1, \dots, K\}$  its ground-truth label, and  $f$  the surrogate model. We seek a perturbation  $\delta$  satisfying  $\|\delta\|_\infty \leq \epsilon$ , such that the adversarial input  $\mathbf{x}^{\text{adv}} = \mathbf{x} + \delta$  misleads  $f$  and transfers effectively to other black-box models.

#### OPTIMIZATION PROCEDURE

The perturbation is optimized over  $T$  steps using projected gradient descent with momentum. At each step  $t \in \{1, \dots, T\}$ , the perturbed input is smoothed using a differentiable Gaussian blur operator:

$$\mathbf{x}^{(t)} = \mathcal{G}_\sigma(\mathbf{x} + \delta^{(t-1)})$$

where  $\mathcal{G}_\sigma(\cdot)$  denotes Gaussian blurring with standard deviation  $\sigma$ , enforcing low-frequency spectral structure (SSR).

Next, the gradient of the loss is computed:

$$\mathbf{g}^{(t)} = \nabla_{\mathbf{x}} \mathcal{L}(f(\mathbf{x}^{(t)}), y)$$

This gradient is intercepted via backward hooks at key ViT modules (Attention, QKV, MLP). For each transformer block  $l$ , the following sequence is applied to each module:

1. **Module-wise Weakening:** The gradient  $\mathbf{g}^{(l)}$  for each module is first scaled using a module-specific weakening factor  $\omega^{(l)} \in (0, 1]$  (e.g.,  $\omega_{\text{attn}}^{(l)}, \omega_{\text{qkv}}^{(l)}, \omega_{\text{mlp}}^{(l)}$ ). This captures prior knowledge about the sensitivity of each module.
2. **Layer-wise Modulation:** The weakened attention gradient is then further modulated by a layer-specific coefficient  $\tau_l \in [0, 1]$ , which reduces the influence of deeper transformer layers:

$$\mathbf{g}^{(l)} \leftarrow \tau_l \cdot (\omega^{(l)} \cdot \mathbf{g}^{(l)})$$

3. **Feature-Sensitive Gradient Scaling (FSGS):**

A layer-aware gradient modulation mechanism that scales gradients based on token-wise importance. FSGS promotes perturbation alignment with semantically salient features while suppressing low-level, architecture-specific signals that degrade cross-model transferability. Let  $\mathbf{Z} \in \mathbb{R}^{T \times D}$  be the token embedding matrix at a given transformer block. We define the raw importance score of token  $i$  as:  $\alpha_i = \|\mathbf{z}_i\|_2$ . The importance scores are min-max normalized across tokens:  $\hat{\alpha}_i = \frac{\alpha_i - \min_j \alpha_j}{\max_j \alpha_j - \min_j \alpha_j + \epsilon}$ , where  $\epsilon$  is a small constant to avoid division by zero.

Let  $l \in \{1, \dots, L\}$  denote the index of the current transformer block, and let  $\mathcal{E} \subset \{1, \dots, L\}$  be the set of early layers (e.g.,  $\mathcal{E} = \{1, \dots, k\}$ ). Define an indicator function:

$$\beta^{(l)} = \begin{cases} 1 & \text{if } l \in \mathcal{E} \quad (\text{early layer}) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The final scaling factor for token  $i$  at layer  $l$  is then computed as:  $s_i^{(l)} = \gamma_{\text{base}} + \lambda \cdot [(1 - \beta^{(l)}) \cdot \hat{\alpha}_i + \beta^{(l)} \cdot (1 - \hat{\alpha}_i)]$ . And the FSGS-modulated gradient is:  $\mathbf{g}_i^{(l), \text{FSGS}} = s_i^{(l)} \cdot \mathbf{g}_i^{(l)}$ .

All module gradients are aggregated to form the total input gradient  $\mathbf{g}^{(t)}$ , and momentum is applied:

$$\mathbf{m}^{(t)} = \mu \cdot \mathbf{m}^{(t-1)} + \frac{\mathbf{g}^{(t)}}{\|\mathbf{g}^{(t)}\|_1}$$

The perturbation is updated and projected onto the  $\ell_\infty$ -norm ball:

$$\delta^{(t)} = \text{Clip}_\epsilon \left( \delta^{(t-1)} + \eta \cdot \text{sign}(\mathbf{m}^{(t)}) \right)$$

**Description.** Algorithm 1 summarizes our full optimization loop. The key innovation lies in the sequential application of module-wise weakening, layer-wise modulation, and semantic-aware scaling via FSGS. All components are implemented via backward hooks, ensuring compatibility with any transformer-based model.

**Hyper-parameters.** Table 6 summarizes the model-specific hyperparameters used in TESSER for each architecture. These include module-wise weakening factors ( $\omega^{(\cdot)}$ ), FSGS scaling parameters ( $\lambda$ ), spectral smoothness regularization strength ( $\sigma$ ), attention truncation depth ( $l_{\text{cut}}$ ), base gradient scaling ( $\gamma_{\text{base}}$ ), as well as optimization parameters: momentum decay ( $\mu$ ) and step size ( $\eta$ ). Values are carefully selected to balance gradient modulation and attack stability per architecture.

## B.2 COMPUTATIONAL COST

To evaluate the efficiency of our proposed method, we report the average time (in seconds) required to generate a single adversarial example using FSGS, FSGS+SSR, and the ATT across different models. As shown in Table 7, our methods incur minimal overhead compared to ATT, with only a slight increase when applying SSR. In particular, even in deeper architectures like CaiT-S/24, FSGS+SSR remains significantly more efficient than ATT.

We provide the detailed environment configuration used for all evaluations. All experiments were conducted on NVIDIA Tesla T4 GPUs hosted on Google Colab. We present the key software dependencies and their corresponding versions:

- Python: 3.11.12
- PyTorch: 2.6.0
- Torchvision: 0.21.0
- NumPy: 2.0.2
- Pillow: 11.2.1
- Timm: 1.0.15
- SciPy: 1.15.3

## C ADDITIONAL EXPERIMENTS

### C.1 QUANTITATIVE ANALYSIS FOR SSR

To evaluate the effect of spectral regularization strength, we conduct an ablation study by varying the Gaussian blur standard deviation  $\sigma$  used in Spectral Smoothness Regularization (SSR). Table 8 reports the average attack success rates (ASR) on ViTs, CNNs, and defended CNNs for  $\sigma \in \{0.5, 0.6, 0.7, 0.8\}$  across all surrogate models.

We observe a consistent trend: increasing  $\sigma$  improves transferability to CNNs and defended CNNs, while slightly reducing ASR on ViTs. This trade-off reflects the role of SSR in suppressing high-frequency architecture-specific noise: improving cross-architecture generalization but marginally

**Algorithm 1: TESSER: Transfer-Enhancing Adversarial Optimization from Vision Transformers via Spectral and Semantic Regularization**


---

**Input:** Input image  $\mathbf{x}$ , label  $y$ , model  $f$ ,  
Steps  $T$ , step size  $\eta$ , perturbation bound  $\epsilon$ ,  
Gaussian blur  $\mathcal{G}_\sigma$ , momentum  $\mu$ ,  
Base scale  $\gamma_{\text{base}}$ ,  
Module-specific FSGS strengths  $\lambda_{\text{qkv}}$ ,  $\lambda_{\text{attn}}$ ,  $\lambda_{\text{mlp}}$ ,  
Early-layer set  $\mathcal{E}$ , attention cutoff layer  $l_{\text{cut}}$ ,  
Module weakening factors  $\omega^{(l)}$ ,  
SSR loss function  $\mathcal{L}_{\text{SSR}}$   
**Output:** Adversarial example  $\mathbf{x}^{\text{adv}}$   
**Initialize:**  $\delta^{(0)} = 0$ ,  $\mathbf{m}^{(0)} = 0$   
**for**  $t = 1$  **to**  $T$  **do**  
  **1. Apply SSR:**  
   $\mathbf{x}^{(t)} = \mathcal{G}_\sigma(\mathbf{x} + \delta^{(t-1)})$   
  **2. Forward pass and compute classification loss:**  
   $\mathcal{L}_{\text{cls}}^{(t)} = \mathcal{L}(f(\mathbf{x}^{(t)}), y)$   
  **3. Backward pass with hooks at QKV, Attention, and MLP modules:**  
  **foreach**  $\text{block } l \in \{1, \dots, L\}$  **do**  
    **foreach**  $\text{module } m \in \{\text{qkv}, \text{attn}, \text{mlp}\}$  **do**  
      **3.1 Extract token features and gradients:**  
       $\mathbf{Z}^{(l,m)} = [\mathbf{z}_1^{(l,m)}, \dots, \mathbf{z}_T^{(l,m)}]$   
       $\mathbf{G}^{(l,m)} = [\mathbf{g}_1^{(l,m)}, \dots, \mathbf{g}_T^{(l,m)}]$   
      **3.2 Compute token importance:**  
       $\alpha_i = \|\mathbf{z}_i^{(l,m)}\|_2$ ,  $\hat{\alpha}_i = \frac{\alpha_i - \min_j \alpha_j}{\max_j \alpha_j - \min_j \alpha_j + \epsilon}$   
      **3.3 Apply module-wise weakening:**  
       $\mathbf{G}^{(l,m)} \leftarrow \omega^{(l)} \cdot \mathbf{G}^{(l,m)}$   
      **3.4 Selective Attention Truncation (only if  $m = \text{attn}$ ):**  
      **if**  $l \geq l_{\text{cut}}$  **then**  
      |  $\mathbf{G}^{(l,\text{attn})} \leftarrow 0$   
      **end**  
      **3.5 Compute FSGS scaling:**  
      **foreach**  $\text{token } i \in \{1, \dots, T\}$  **do**  
      | **if**  $l \in \mathcal{E}$  **then**  
      | |  $s_i = \gamma_{\text{base}} + \lambda_m \cdot (1 - \hat{\alpha}_i)$   
      | **end**  
      | **else**  
      | |  $s_i = \gamma_{\text{base}} + \lambda_m \cdot \hat{\alpha}_i$   
      | **end**  
      |  $\mathbf{g}_i^{(l,m)} \leftarrow s_i \cdot \mathbf{g}_i^{(l,m)}$   
      **end**  
    **end**  
  **end**  
  **4. Aggregate gradients across all modules:**  
   $\mathbf{g}^{(t)} = \sum_{l,m} \text{Aggregate}(\mathbf{G}^{(l,m)})$   
  **5. Momentum update:**  
   $\mathbf{m}^{(t)} = \mu \cdot \mathbf{m}^{(t-1)} + \frac{\mathbf{g}^{(t)}}{\|\mathbf{g}^{(t)}\|_1}$   
  **6. Perturbation update with projection:**  
   $\delta^{(t)} = \text{Clip}_\epsilon(\delta^{(t-1)} + \eta \cdot \text{sign}(\mathbf{m}^{(t)}))$   
**end**  
**return**  $\mathbf{x}^{\text{adv}} = \mathbf{x} + \delta^{(T)}$ 


---

Table 6: Model-specific hyperparameter settings used for TESSER.  $\omega^{(\cdot)}$  denotes the weakening factor for each module,  $\lambda$  is the FSGS scaling parameter,  $\sigma$  controls the strength of spectral regularization,  $l_{\text{cut}}$  is the attention truncation depth,  $\gamma_{\text{base}}$  is the minimum gradient scaling factor,  $\mu$  is the momentum decay used in PGD, and  $\eta$  is the step size for perturbation update.

| Hyperparameter           | ViT-B/16 | PiT-B   | CaiT-S/24 | Visformer-S |
|--------------------------|----------|---------|-----------|-------------|
| $\omega^{(\text{attn})}$ | 0.45     | 0.25    | 0.3       | 0.4         |
| $\omega^{(\text{qkv})}$  | 0.5      | 0.5     | 1.0       | 0.8         |
| $\omega^{(\text{mlp})}$  | 0.7      | 0.7     | 0.6       | 0.5         |
| $\lambda_{\text{attn}}$  | 0.4      | 0.45    | 0.5       | 0.45        |
| $\lambda_{\text{qkv}}$   | 0.5      | 0.5     | 0.5       | 0.5         |
| $\lambda_{\text{mlp}}$   | 0.55     | 0.55    | 0.65      | 0.6         |
| $\sigma$ (SSR)           | 0.5      | 0.7     | 0.7       | 0.7         |
| $l_{\text{cut}}$         | 10       | 9       | 4         | 8           |
| $\gamma_{\text{base}}$   | 0.5      | 0.5     | 0.5       | 0.5         |
| $\mu$                    | 1.0      | 1.0     | 1.0       | 1.0         |
| $\eta$                   | 1.6/255  | 1.6/255 | 1.6/255   | 1.6/255     |

Table 7: Computational cost (in seconds) for generating a single adversarial example across different models and methods. FSGS refers to our feature-sensitive gradient scaling, SSR refers to spectral smoothness regularization, and ATT denotes state of the art.

| Model       | FSGS | FSGS + SSR | ATT (Ming et al., 2024) |
|-------------|------|------------|-------------------------|
| ViT-B/16    | 0.5  | 0.52       | 0.93                    |
| PiT-B       | 0.54 | 0.6        | 1.05                    |
| CaiT-S/24   | 1.24 | 1.27       | 1.88                    |
| Visformer-S | 0.35 | 0.38       | 1.14                    |

weakening model-specific alignment. For instance, in ViT-B/16, increasing  $\sigma$  from 0.5 to 0.8 decreases ViT ASR from 83.21% to 81.21%, but improves CNN ASR from 58.77% to 61.82% and defended CNN ASR from 46.63% to 52.33%. A similar pattern is observed in PiT-B and CaiT-S/24.

Notably, the improvement on defended CNNs is particularly pronounced. For Visformer-S, the ASR on defended models improves from 46.96% at  $\sigma = 0.5$  to 61.5% at  $\sigma = 0.8$ , a gain of over 14%. These results confirm that SSR strengthens black-box transferability and robustness by encouraging low-frequency perturbations that are less dependent on the surrogate model’s internal architecture.

In practice, setting  $\sigma$  between 0.6 and 0.8 offers a favorable trade-off, preserving sufficient ViT ASR while achieving substantial improvements on CNNs and defended models. This ablation supports the effectiveness of spectral regularization and its role in enhancing generalization under diverse adversarial settings.

## C.2 COMPARISON OF ATTACK EFFICIENCY WHEN USING INPUT DIVERSITY TECHNIQUE

To further assess the effectiveness and generality of our proposed TESSER framework, we evaluate its performance when combined with an input diversity enhancement strategy, specifically PatchOut (PO) (Wei et al., 2022). This technique introduces random masking during inference to improve the robustness and transferability of adversarial perturbations.

Table 9 presents the average Attack Success Rate (ASR) of different attack methods augmented with PO, tested across ViTs, CNNs, and defended CNNs. The experiments span four representative surrogate models: ViT-B/16, CaiT-S/24, PiT-B, and Visformer-S.

Across all surrogate models and evaluation categories, TESSER+PO consistently achieves the highest ASR. For example, using PiT-B as the surrogate, TESSER+PO achieves an ASR of **94.83%** on ViTs, **87.7%** on CNNs, and **61.43%** on defended CNNs, representing improvements of more than **+10%** over the strongest baseline ATT+PO. Similar trends are observed with the other surrogate models,

Table 8: Average attack success rate (ASR) (%) against ViTs, CNNs, and defended CNNs across varying Gaussian blur strength  $\sigma$ . Increasing  $\sigma$  generally improves transferability to CNNs and defended models by enforcing low-frequency perturbations, while slightly reducing white-box ASR on ViTs.

| Model    | $\sigma$ | ViTs  | CNNs  | Def-CNNs | Model       | $\sigma$ | ViTs  | CNNs  | Def-CNNs |
|----------|----------|-------|-------|----------|-------------|----------|-------|-------|----------|
| ViT-B/16 | 0.5      | 83.21 | 58.77 | 46.63    | CaiT-S/24   | 0.5      | 94.82 | 68.12 | 45.6     |
|          | 0.6      | 83.12 | 61.85 | 49.86    |             | 0.6      | 94.57 | 71.9  | 51.76    |
|          | 0.7      | 81.95 | 61.62 | 51.13    |             | 0.7      | 94.2  | 73.87 | 54.86    |
|          | 0.8      | 81.21 | 61.82 | 52.33    |             | 0.8      | 93.82 | 73.55 | 57.06    |
| PiT-B    | 0.5      | 90.3  | 80.85 | 49       | Visformer-S | 0.5      | 75.93 | 76.77 | 46.96    |
|          | 0.6      | 91.63 | 83.4  | 54.9     |             | 0.6      | 78.47 | 80.45 | 53.9     |
|          | 0.7      | 91.36 | 83.27 | 57.06    |             | 0.7      | 78.67 | 81.67 | 58.46    |
|          | 0.8      | 90.02 | 83.45 | 58.6     |             | 0.8      | 83.33 | 81.22 | 61.5     |

Table 9: The average attack success rate (%) against ViTs, CNNs, and defended CNNs by various transfer-based attacks with input diversity enhancement strategy. The best results are highlighted in bold. “PO” denotes PatchOut (Wei et al., 2022).

| Model    | Attack         | ViTs                              | CNNs                              | Def-CNNs                          | Model       | Attack         | ViTs                              | CNNs                             | Def-CNNs                          |
|----------|----------------|-----------------------------------|-----------------------------------|-----------------------------------|-------------|----------------|-----------------------------------|----------------------------------|-----------------------------------|
| ViT-B/16 | MIM+PO         | 61.3                              | 31.3                              | 21.7                              | CaiT-S/24   | MIM+PO         | 70.3                              | 44.0                             | 29.3                              |
|          | VMI+PO         | 69.1                              | 42.8                              | 30.9                              |             | VMI+PO         | 76.8                              | 57.8                             | 38.4                              |
|          | SGM+PO         | 64.8                              | 29.2                              | 18.9                              |             | SGM+PO         | 85.1                              | 49.2                             | 29.3                              |
|          | PNA+PO         | 70.8                              | 42.6                              | 29.9                              |             | PNA+PO         | 81.6                              | 56.6                             | 39.3                              |
|          | TGR+PO         | 76.0                              | 46.7                              | 33.3                              |             | TGR+PO         | 88.8                              | 60.5                             | 40.5                              |
|          | ATT+PO         | 77.1                              | 51.7                              | 37.1                              |             | ATT+PO         | 91.1                              | 71.9                             | 54.3                              |
|          | <b>Ours+PO</b> | <b>85.18<math>\uparrow</math></b> | <b>64.17<math>\uparrow</math></b> | <b>52.16<math>\uparrow</math></b> |             | <b>Ours+PO</b> | <b>91.15<math>\uparrow</math></b> | <b>72.9<math>\uparrow</math></b> | <b>56.46<math>\uparrow</math></b> |
| PiT-B    | MIM+PO         | 47.3                              | 32.5                              | 17.5                              | Visformer-S | MIM+PO         | 54.9                              | 45.7                             | 23.4                              |
|          | VMI+PO         | 59.5                              | 46.2                              | 35.8                              |             | VMI+PO         | 64.8                              | 56.6                             | 32.6                              |
|          | SGM+PO         | 70.0                              | 45.6                              | 21.3                              |             | SGM+PO         | 51.6                              | 44.3                             | 15.0                              |
|          | PNA+PO         | 73.1                              | 57.8                              | 32.7                              |             | PNA+PO         | 68.8                              | 61.8                             | 32.3                              |
|          | TGR+PO         | 82.3                              | 68.9                              | 41.3                              |             | TGR+PO         | 70.4                              | 64.3                             | 33.5                              |
|          | ATT+PO         | 84.2                              | 75.2                              | 48.4                              |             | ATT+PO         | 70.5                              | 79.3                             | 44.5                              |
|          | <b>Ours+PO</b> | <b>94.83<math>\uparrow</math></b> | <b>87.7<math>\uparrow</math></b>  | <b>61.43<math>\uparrow</math></b> |             | <b>Ours+PO</b> | <b>84.42<math>\uparrow</math></b> | <b>79.4<math>\uparrow</math></b> | <b>58.06<math>\uparrow</math></b> |

including CaiT-S/24 and ViT-B/16, where TESSER+PO continues to outperform baselines by wide margins.

These results demonstrate two key insights: (1) TESSER is orthogonal to input diversity methods, as its performance improves further when used with PO, and (2) our gradient modulation and spectral regularization strategies remain effective under randomized input transformations, indicating strong generalization.

In particular, on defended CNNs, traditionally difficult targets due to adversarial training, TESSER + PO outperforms all baselines by significant margins (e.g., + 7% over ATT + PO with ViT-B/16). This highlights that FSGS and SSR lead to perturbations that survive stochastic augmentations while preserving transferability and robustness.

### C.3 ADVERSARIAL ATTACK EFFICIENCY AND CONFIDENCE DYNAMICS

To better assess the quality of adversarial examples beyond final attack success rate (ASR), we evaluate the efficiency and effectiveness of the generated perturbations in terms of iteration-wise model response. Specifically, we compare TESSER and ATT based on:

- **Attack efficiency:** How quickly the model’s prediction flips and stabilizes to an adversarial label across iterations.
- **Attack effectiveness:** The final confidence of the model in the adversarial label after optimization completes.

Table 10 summarizes the average iteration at which the target model stabilizes on the adversarial label (i.e., no further label flipping) and the average confidence on the adversarial class after 10 attack steps.

Table 10: Comparison of attack efficiency and effectiveness between TESSER and ATT. We report the average iteration where the model prediction stabilizes on the adversarial label (lower is better) and the final model confidence (%) in the adversarial class (higher is better).

| Method        | Stabilization Iteration ( $\downarrow$ ) | Final Confidence (%) ( $\uparrow$ ) |
|---------------|--|-------------------------------------|
| ATT           | 6.8                                      | 87.93                               |
| TESSER (Ours) | <b>5.1</b>                               | <b>91.37</b>                        |

Table 11: TESSER attack success rate (%) with and without module-wise gradient weakening  $\omega$  against eight ViT models and the average attack success rate (%) of all black-box models. The best results are highlighted in **bold**.

| Model    | Attack       | ViT-B/16     | PiT-B         | CaIT-S/24   | Visformer-S | DeiT-B      | TNT-S       | LeViT-256   | ConViT-B    | Avg <sub>bb</sub>                |
|----------|--------------|--------------|---------------|-------------|-------------|-------------|-------------|-------------|-------------|----------------------------------|
| ViT-B/16 | w/o $\omega$ | <b>99.6*</b> | 40.9          | 71          | 45.7        | 69.3        | 64.8        | 40.2        | 72.5        | 63                               |
|          | w $\omega$   | <b>100*</b>  | <b>61.7</b>   | <b>94</b>   | <b>68.3</b> | <b>92.5</b> | <b>85.6</b> | <b>72.2</b> | <b>91.4</b> | <b>83.2<math>\uparrow</math></b> |
| PiT-B    | w/o $\omega$ | 58.1         | <b>99.9*</b>  | 69.2        | 74.7        | 69          | 74          | 66.9        | 70.3        | 72.76                            |
|          | w $\omega$   | <b>74.9</b>  | <b>100.0*</b> | <b>91.6</b> | <b>93.2</b> | <b>92.1</b> | <b>95</b>   | <b>92.4</b> | <b>91.7</b> | <b>91.4<math>\uparrow</math></b> |

TESSER reaches a stable adversarial label approximately 1.7 iterations earlier than ATT, confirming its improved gradient alignment and optimization direction. Additionally, the final adversarial confidence achieved by TESSER is consistently higher, indicating stronger and more decisive misclassification. This validates that our semantic gradient modulation not only accelerates convergence but also increases attack effectiveness by pushing perturbations toward model-relevant, transferable features.

## D ADDITIONAL ABLATION STUDIES

### D.1 QUALITATIVE COMPARISON

To further analyze the effectiveness and interpretability of our proposed method, we present qualitative comparisons between TESSER (FSGS+SSR) and the state-of-the-art ATT (Ming et al., 2024) across a diverse set of samples from ImageNet. Figure 3 shows clean and adversarial images, Grad-CAM heatmaps, and FFT visualizations for perturbations.

**Semantic Alignment.** In nearly all examples, the adversarial images generated by TESSER show stronger alignment with semantically meaningful regions (e.g., bird bodies, faces, objects of interest) compared to ATT. This is reflected in the Grad-CAM visualizations guided by the adversarial label. Despite being misclassified, the Grad-CAM of TESSER adversarial examples remains spatially focused on relevant visual features, validating the effectiveness of FSGS in preserving semantically informative gradients during attack optimization.

**Spectral Coherence.** The FFT visualizations reveal that TESSER perturbations exhibit smoother and more coherent frequency profiles, with lower high-frequency energy content than those generated by ATT. This is consistently supported by the computed High-Frequency Energy Ratio (HFER), which is reduced by 6–16% across examples when using FSGS+SSR. Lower HFER confirms that SSR suppresses architecture-specific, high-frequency noise that often undermines transferability.

These additional ablations reinforce our core claim: FSGS guides perturbations toward transferable, semantically meaningful features, while SSR regularizes their spectral profile to avoid overfitting to model-specific noise. Together, these properties lead to adversarial examples that are more interpretable and more effective in black-box transfer scenarios.

### D.2 IMPACT OF MODULE-WISE GRADIENT WEAKENING

We compare ASR with and without  $\omega$  (i.e., setting all  $\omega = 1$  disables gradient weakening). On ViT-B/16, using  $\omega$  improves ASR from 63.0% to 83.2% (Table 11), confirming the effectiveness of selective gradient suppression.

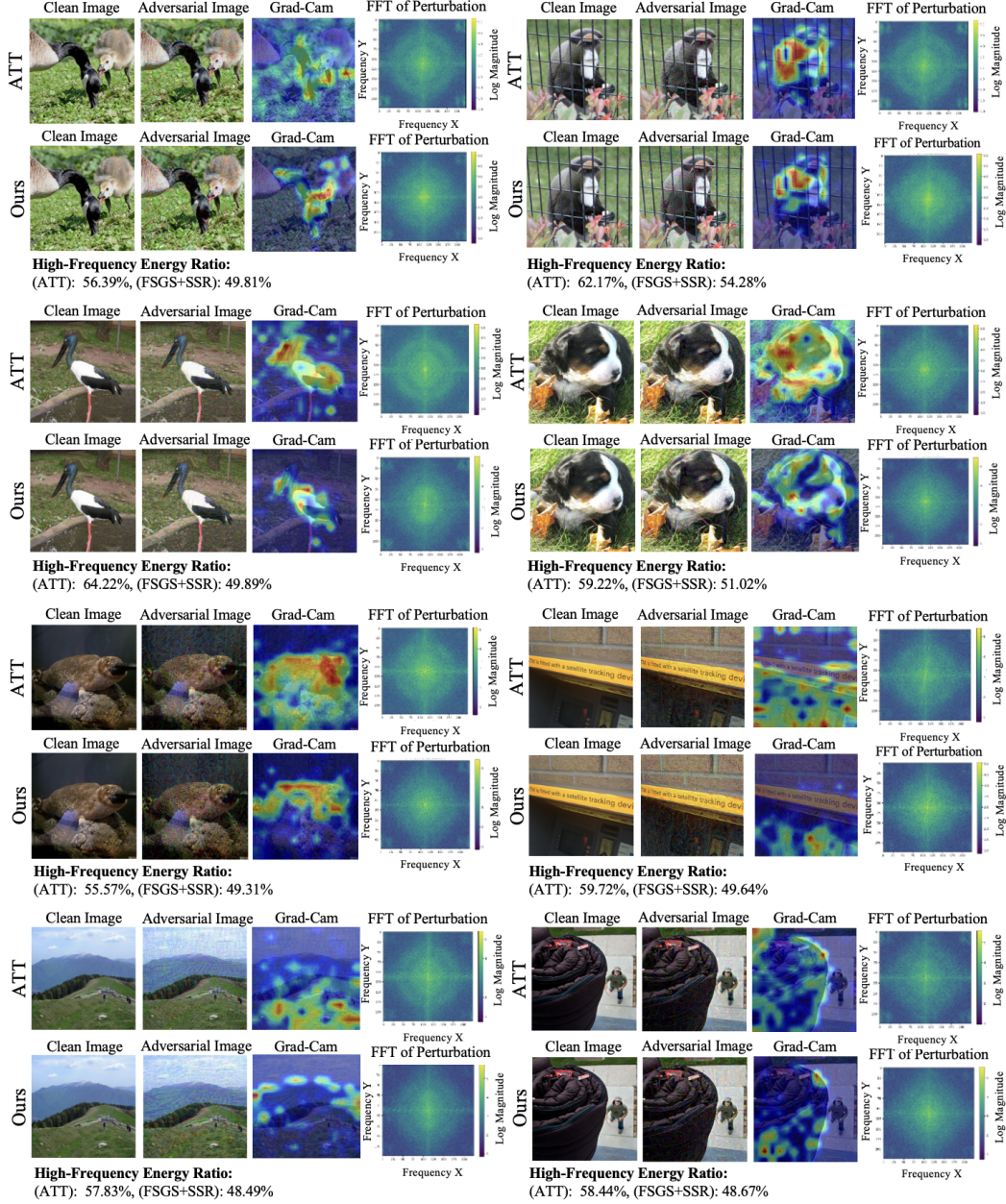


Figure 3: Qualitative comparison between ATT and our TESSER method (FSGS+SSR). Each block shows clean image, adversarial image, Grad-Cam heatmap, and FFT of the perturbation. TESSER yields semantically aligned and spectrally smooth perturbations, with consistently lower high-frequency energy ratios.



Table 12: TESSER attack success rate (%) with and without selective attention truncation  $l_{\text{cut}}$  against eight ViT models and the average attack success rate (%) of all black-box models. The best results are highlighted in **bold**.

| Model    | Attack               | ViT-B/16    | PiT-B         | CaiT-S/24   | Visformer-S | DeiT-B      | TNT-S       | LeViT-256   | ConViT-B    | Avg <sub>bb</sub> |
|----------|----------------------|-------------|---------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------------|
| ViT-B/16 | w/o $l_{\text{cut}}$ | <b>100*</b> | 60.4          | 87.3        | 67.7        | 88.3        | 85.4        | 65.4        | 89.9        | 80.55             |
|          | w $l_{\text{cut}}$   | <b>100*</b> | <b>61.7</b>   | <b>94</b>   | <b>68.3</b> | <b>92.5</b> | <b>85.6</b> | <b>72.2</b> | <b>91.4</b> | <b>83.2</b> ↑     |
| PiT-B    | w/o $l_{\text{cut}}$ | 73.1        | <b>99.7*</b>  | 82.1        | 86          | 84          | 86.4        | 83.2        | 84.1        | 84.82             |
|          | w $l_{\text{cut}}$   | <b>74.9</b> | <b>100.0*</b> | <b>91.6</b> | <b>93.2</b> | <b>92.1</b> | <b>95</b>   | <b>92.4</b> | <b>91.7</b> | <b>91.4</b> ↑     |

Table 13: TESSER attack success rate (%) with and without rescaling factor  $\lambda$  against eight ViT models and the average attack success rate (%) of all black-box models. The best results are highlighted in **bold**.

| Model    | Attack        | ViT-B/16     | PiT-B         | CaiT-S/24   | Visformer-S | DeiT-B      | TNT-S       | LeViT-256   | ConViT-B    | Avg <sub>bb</sub> |
|----------|---------------|--------------|---------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------------|
| ViT-B/16 | w/o $\lambda$ | <b>84.1*</b> | 32.1          | 54.5        | 42.3        | 55.1        | 56          | 43.1        | 56.9        | 53.01             |
|          | w $\lambda$   | <b>100*</b>  | <b>61.7</b>   | <b>94</b>   | <b>68.3</b> | <b>92.5</b> | <b>85.6</b> | <b>72.2</b> | <b>91.4</b> | <b>83.2</b> ↑     |
| PiT-B    | w/o $\lambda$ | 51.2         | <b>92.9*</b>  | 61.2        | 67.9        | 63.6        | 67.9        | 66.5        | 62.6        | 66.72             |
|          | w $\lambda$   | <b>74.9</b>  | <b>100.0*</b> | <b>91.6</b> | <b>93.2</b> | <b>92.1</b> | <b>95</b>   | <b>92.4</b> | <b>91.7</b> | <b>91.4</b> ↑     |

### D.3 IMPACT OF SELECTIVE ATTENTION TRUNCATION

On PiT-B, disabling attention truncation (i.e., no  $l_{\text{cut}}$ ) leads to an average 7% drop in ASR (Table 12), validating the importance of focusing on early-layer token gradients.

### D.4 IMPACT OF RESCALING FACTOR

Setting  $\lambda = 0$  disables adaptive FSGS scaling (only  $\gamma_{\text{base}}$  is used as a fixed multiplier). On ViT-B/16, enabling  $\lambda$  improves ASR by an average of 30% (Table 13), highlighting the value of adaptive gradient modulation in improving attack effectiveness.

In addition, we empirically validate the effectiveness of our scaling strategy by comparing it to a random scaling baseline. As shown in Table 14, our method significantly outperforms random scaling across all target models, achieving consistently higher ASR and demonstrating stronger transferability.

## E EVALUATING THE TRANSFERABILITY OF DIFFERENT ATTACK METHODS FOR TARGETED ATTACKS

While our main experiments focus on untargeted attacks, both FSGS and SSR are model-agnostic and loss-independent components applied during backpropagation. Therefore, they are fully compatible with targeted attack formulations—only the loss needs to be adapted. To validate this, we conducted targeted attack experiments using the target label set as (true label + 1). As shown in Table 15, our method (TESSER) achieves a significantly higher targeted ASR of 43.08%, outperforming PGD (16.06%), MIM (22.01%), and ATT (33.33%), demonstrating the effectiveness and transferability of our approach in targeted settings as well. The table will be included in the revised version.

Table 14: TESSER Attack Success Rate (%) with our Scaling strategy vs. Random Scaling. **Bold** = better of the two scalings for the same surrogate–target pair. \* denotes white-box (surrogate equals target).

| Surrogate | Scaling | ViT-B/16     | PiT-B         | CaiT-S/24   | Visformer-S | DeiT-B      | TNT-S       | LeViT-256   | ConViT-B    | Avg           |
|-----------|---------|--------------|---------------|-------------|-------------|-------------|-------------|-------------|-------------|---------------|
| ViT-B/16  | Random  | <b>86.4*</b> | 30.6          | 52.3        | 37.8        | 53.0        | 55.4        | 37.8        | 55.7        | 51.12         |
|           | ours    | <b>100*</b>  | <b>61.7</b>   | <b>94.0</b> | <b>68.3</b> | <b>92.5</b> | <b>85.6</b> | <b>72.2</b> | <b>91.4</b> | <b>83.2</b> ↑ |
| PiT-B     | Random  | 28.4         | <b>100*</b>   | 34.6        | 44.8        | 33.9        | 44.1        | 38.3        | 38.2        | 45.28         |
|           | ours    | <b>74.9</b>  | <b>100.0*</b> | <b>91.6</b> | <b>93.2</b> | <b>92.1</b> | <b>95.0</b> | <b>92.4</b> | <b>91.7</b> | <b>91.4</b> ↑ |

Table 15: The attack success rate (%) of various transfer-based targeted attacks against eight ViT models and the average attack success rate (%) of all black-box models. The best results are highlighted in **bold**.

| Model    | Attack | ViT-B/16     | PiT-B         | CaiT-S/24   | Visformer-S | DeiT-B      | TNT-S       | LeViT-256   | ConViT-B    | Avg <sub>bb</sub> |
|----------|--------|--------------|---------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------------|
| ViT-B/16 | PGD    | <b>96.1*</b> | 2.1           | 6.8         | 2.7         | 5.6         | 6           | 1.7         | 7.5         | 16.06             |
|          | MIM    | <b>99.4*</b> | 6.6           | 16.1        | 6.3         | 13.9        | 11.8        | 4.4         | 17.6        | 22.01             |
|          | ATT    | <b>99.5*</b> | 8.7           | 35.6        | 9.6         | 33.6        | 30.9        | 6.7         | 42.1        | 33.33             |
|          | Ours   | <b>99.6*</b> | <b>17.9</b>   | <b>47.3</b> | <b>21</b>   | <b>48.6</b> | <b>42.1</b> | <b>16.1</b> | <b>52.1</b> | <b>43.08↑</b>     |
| PiT-B    | PGD    | 0.8          | <b>96.6*</b>  | 1.1         | 2.3         | 0.9         | 2.1         | 1.3         | 0.7         | 13.22             |
|          | MIM    | 5.3          | <b>99.9*</b>  | 5.1         | 8           | 4.9         | 6.5         | 4           | 5.5         | 17.4              |
|          | ATT    | 12.1         | <b>100*</b>   | 14.6        | 20.2        | 13.2        | 18.8        | 12.2        | 16.8        | 25.98             |
|          | Ours   | <b>20.4</b>  | <b>100.0*</b> | <b>26</b>   | <b>33</b>   | <b>28.4</b> | <b>30.2</b> | <b>26.3</b> | <b>27.2</b> | <b>47.86↑</b>     |

Table 16: The attack success rate (%) of Autoattack (AA) vs. TESSER against eight ViT models and the average attack success rate (%) of all black-box models. The best results are highlighted in **bold**.

| Model    | Attack | ViT-B/16     | PiT-B         | CaiT-S/24   | Visformer-S | DeiT-B      | TNT-S       | LeViT-256   | ConViT-B    | Avg <sub>bb</sub> |
|----------|--------|--------------|---------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------------|
| ViT-B/16 | AA     | <b>99.9*</b> | 10.5          | 42.9        | 13.2        | 33.6        | 38.4        | 17          | 40.4        | 36.98             |
|          | Ours   | <b>100*</b>  | <b>61.7</b>   | <b>94</b>   | <b>68.3</b> | <b>92.5</b> | <b>85.6</b> | <b>72.2</b> | <b>91.4</b> | <b>83.2↑</b>      |
| PiT-B    | AA     | 10           | <b>98.5*</b>  | 13.5        | 24          | 11.3        | 21.2        | 22.4        | 13.9        | 26.85             |
|          | Ours   | <b>74.9</b>  | <b>100.0*</b> | <b>91.6</b> | <b>93.2</b> | <b>92.1</b> | <b>95</b>   | <b>92.4</b> | <b>91.7</b> | <b>91.4↑</b>      |

## F EVALUATING TESSER PERFORMANCE VS. AUTOATTACK

While AutoAttack (AA) is a strong white-box evaluation benchmark, it is not optimized for transfer-based black-box settings. To enable a fair comparison, we evaluate both TESSER and AutoAttack under the same transfer setup with a fixed perturbation budget of  $\epsilon = 16/255$ . As reported in Table 16, TESSER achieves over 50% higher average ASR compared to AutoAttack across multiple target models. For example, when attacking PiT-B from a ViT-B/16 surrogate, TESSER achieves an ASR of 61.7%, compared to only 10.5% for AutoAttack. This gap is expected, as TESSER is explicitly designed to optimize black-box transferability, whereas AutoAttack is tailored for white-box robustness evaluation.

## G EVALUATION TESSER TRANSFERABILITY TO VISUAL STATE SPACE MODELS

To further increase the architectural dissimilarity, we evaluate TESSER transferability to Vision Mamba (Zhu et al., 2024), a state-space-based architecture with bidirectional SSMs and position-aware embeddings, representing a class of models distinct from transformers. As shown in Table 17, TESSER consistently achieves the highest ASR 87.1% and 76.7% on both Vim-Tiny and Vim-Small, respectively compared to 80.9% and 69% for sota ATT attack, demonstrating robust transfer even under significant architectural and representational divergence.

Table 17: Comparative experiments of different attack methods on VIM. “clean” indicates that clean images are classified and all results indicate the percentage of classification errors (*i.e.*, ASR).

| Model    | Attack | VIM-tiny     | VIM-small    |
|----------|--------|--------------|--------------|
| ViT-B/16 | clean  | 3.1          | 0.9          |
|          | MIM    | 45.6         | 42           |
|          | ATT    | 80.9         | 69           |
|          | TESSER | <b>87.1↑</b> | <b>76.7↑</b> |
| PiT-B    | MIM    | 32.3         | 34.9         |
|          | ATT    | 53.4         | 55.1         |
|          | TESSER | <b>77.4↑</b> | <b>80.7↑</b> |