A. Implementation Details

A.1. Algorithm for Multiple Asymptotically Normal Distribution Attack

In Section 3, we propose the Multiple Asymptotically Normal Distribution Attacks (MultiANDA), a novel method that explicitly characterizes perturbations inferred from a learned distribution. Specifically, in Section 3.2, we first elaborate on the procedure of single ANDA, which approximates the posterior distribution over the adversarial perturbations by leveraging the asymptotic normality property of stochastic gradient ascent (SGA). Subsequently, in Section 3.3, we apply an ensemble strategy on ANDA to estimate a mixture of Gaussian distributions, enhancing exploration of the potential optimization space. This leads to further improved generalization performance of the attacks, facilitating the generation of more diverse and transferable adversarial examples. The detailed implementation of MultiANDA is shown in Algorithm A.1.

A.2. Datasets and Models

In this subsection, the datasets and models employed in our experiments are detailed. In addition to the evaluation experiments introduced in Section 4, we provide more results on the ViT-based [4, 6, 18] and CLIP [14] models in this supplementary material, which are used to illustrate the generalizability of our proposed methods to various model architectures and the performance on the cross-domain task.

Datasets: The dataset is ImageNet-1k, which is identical to the one described in Section 4.1.

Models: We considered two categories of target models for evaluation: seven normally trained models and seven advanced defense models. The first category includes Inception-v3 (Inc-v3) [16], Resnet-v2-50 (ResNet-50), Resnet-v2-101 (ResNet-101), Resnet-v2-152 (ResNet-152) [5], Inception-v4 (Inc-v4), Inception-ResNet-v2 (IncRes-v2) [17] and VGG-19 [15], where Inc-v3, ResNet-50, IncRes-v2 and VGG-19 are also used as the white-box source models to generate adversarial examples. The defense models contain three adversarially trained models: Inc-v3_{ens3}, Inc-v3_{ens4} and IncRes-v2_{ens} [19]; the

---

**Algorithm A.1: Multiple Asymptotically Normal Distribution Attacks**

**Input:** $x$: clean image $x \in \mathbb{R}^d$; $y$: ground-truth label; $f$: pre-trained source model;

**Parameters:** $T$: # iterations; $c$: perturbation magnitude; $n$: # batch samples for augmentation; $M$: # sampling examples; $K$: # ensemble ANDAs; $\gamma$: the radius of uniform noise;

**Output:** $x_{adv}$: Adversarial examples;

**Initialize** $x_{adv}$

for $k = 0$ to $K - 1$

Initialize the uniform noise in parallel

$\delta_{adv}^{(k)} = \frac{c}{T} x_{adv} \sim \mathcal{U}(\gamma, \gamma)$

Run $K$ ANDA loops independently and in parallel (Refer ANDA to Algorithm 1 in Section 3)

$x_{adv}(k) = \text{ANDA}(x_{adv}^{(k)}; y, f, T, M, c, n)$

end for

Average $K$ ANDA outputs of penultimate iterations:

$x_{adv}^{(T)} = \frac{1}{K} \sum_{k=0}^{K-1} x_{adv}^{(T-1)}$

Output option (a): Craft one $x_{adv}$ using the mean perturbation $\delta_{mean}$

$\bar{\delta}_{mean} = \frac{1}{K} \sum_{k=0}^{K-1} \delta_{adv}^{(k)}$

$x_{adv} = \text{Clip}_{x_{adv}}\{x_{adv}^{(T-1)} + \alpha \cdot \text{Sign} (\bar{\delta}_{mean})\}$

Output option (b): Craft $M x_{adv}$ by sampling from the learned perturbation distribution $N(\delta_k, \sigma_k)$ ($k = 0, \ldots, K - 1$)

$\{\delta_m = \frac{1}{K} \sum_{k=0}^{K-1} \delta_{k, m}, \; \delta_{k, m} \sim N(\delta_{adv}^{(k)}, \sigma_{adv}^{(k)})\}^{M}_{m=1}$

$\{x_{adv, m} = \text{Clip}_{x_{adv}}\{x_{adv}^{(T-1)} + \alpha \cdot \text{Sign} (\delta_{m})\}\}^{M}_{m=1}$

High-level representation Guided Denoiser (HGD) [9]; the Neural Representation Purifier (NRP) [13]; the Random-
ized Smoothing (RS) [1]; and the ‘Rand-3’ submission in the NIPS 2017 defense competition (NIPS-r3).

We aim to utilize ANDA/MultiAND to thoroughly test the defense effectiveness of various strategies in recently proposed defense approaches. The defense strategies and the implementation details of these selected advanced defense models are presented as follows:

- Three adversarially trained models, namely Inc-v3$_{ens}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$ [19];
- High-level representation guided denoiser (HGD, rank-1 submission in the NIPS 2017 defense competition) [9];
- Rand-3 submission in the NIPS 2017 defense competition (NIPS-r3);  
- Randomized Smoothing (RS) [1];
- Neural Representation Purifier (NRP) [13]

We adopted the implementation of NRP as described in [20], and sourced the remaining defense methods from their official implementations. For all defense methods, we utilized Inc-v3$_{ens}$ as the backbone architecture.

For the ViT series of models, four mainstream models: ViT-L/16 [4], DeiT3-B/16 [18], Swin-B/4 [11] and PiT-B [6] are selected. For the evaluation against the cross-domain task, we choose the renowned multimodal model CLIP [14].

### A.3. Hyper-Parameters

The basic attack settings were consistent with the work [2, 20]: the maximum perturbation of $\epsilon = 16$, and the number of iterations $T = 10$ and step size $\alpha = \epsilon / T = 1.6$. We carefully tuned the specific hyper-parameters for each baseline (e.g., the sampling number in VMI-FGSM, the ensemble number in FIA, etc.), and the best results were recorded.

For ANDA, we set the number of augmented samples as 25 (i.e., $n = 25$), and the image augmenting parameter Augmax (refer detailed explanation in Section C.3) as 0.3 if not specified. For MultiANDA, the default number of ANDA components is 5 (i.e., $K = 5$). To ensure the random starts of each MultiANDA component, we added small uniform noises $u$ to the original sample $x$, where $u \sim \mathcal{U}(-\gamma, \gamma)$, as shown in Algorithm A.1. To balance the diversity among the ensemble components and the original semantic information of inputs, we set the radius of uniform noise $\gamma = \frac{\alpha T}{7}$.

Regarding the adopted baselines, we employed the following corresponding hyper-parameters:

- For MI-FGSM, NI-FGSM and their variants, we set the decay factor $\mu = 1.0$. For VMI-FGSM [20], $n = 20$ and $\beta = 1.5$.
- For DIM [24], the transformation probability was set to 0.5. For SIM [20], the number of scale copies is 5. For TIM [3], we tuned the Gaussian kernel size in $\{15 \times 15, 7 \times 7\}$ with the same standard deviation $\sigma = 3$ by referring their official code.
- For FIA [21], we followed the official settings in the corresponding paper.
- For TAIG [7], we chose TAIG-S as the baseline. To ensure a fair comparison in the single form, we adhered to our basic settings ($\epsilon = 16$, $T = 10$, and the number of turning points $E = 20$). For a detailed comparison showcasing the best performance of TAIG, refer to Section B.5.

### B. Additional Experimental Results

#### B.1. Full evaluation results

Taking advantage of the extensive space in this appendix, we present the complete experimental results for the baseline methods and the proposed ANDA/MultiAND. These encompass all seven normally trained models and seven advanced defense models, as detailed in Table B.1 and Table B.2, respectively. Notably, additional experiments involving Inc-v4, ResNet-101, and Inc-v3$_{ens4}$—models with structures similar to Inc-v3, ResNet-50 and Inc-v3$_{ens3}$—further validate the effectiveness of our methods. An intriguing finding is that using ResNet-50 as the source model yields the best attacking performance. This aligns with the conclusion drawn by Wu et al. [22]. They empirically studied this observation and disclosed the skip connection provided the actual contribution to this enhancement. Wu et al. [22] and Zhu et al. [25] further boost the transferability by exploiting this finding. Moreover, the comparatively modest results on RS models have sparked our interest for future research.

We conducted a similar statistical analysis, as shown in Figure 3 in the main body, on defense models to determine the numbers of black-box target models that each generated example can deceive. The source model remains ResNet-50. Figure B.1 shows 15% and 25% of the examples generated by ANDA and MultiANDA managed to fool all seven models (as seen in the rightmost bars), respectively. In contrast, only 5% examples crafted by VMI-FGSM achieved this. In addition, the failure rate of VMI-FGSM across all seven models stands at 20%, compared to only 5% for ANDA. These results indicate that the generated adversaries by ANDA and MultiANDA are more diverse and have better transferability than those by VMI-FGSM.

#### B.2. Cross-domain attack on CLIP

Admittedly, using a surrogate with the identical dataset as target models for conducting black-box attacks is an ideal setting, typically only feasible in lab experiments or when targeting ‘general-purpose classification models’ trained on ImageNet-based datasets, as in our case. We conducted

---

1. https://github.com/anlthms/nips-2017/tree/master/mmd

2. https://github.com/dongyp13/Translation-Invariant-Attacks
| Attack       | Inc-v3 | Inc-v4 | ResNet-50 | ResNet-101 | ResNet-152 | IncRes-v2 | VGG-19 |
|-------------|--------|--------|-----------|------------|------------|-----------|--------|
| BIM         | 36.3   | 28.7   | 25.6      | 22.1       | 20.6       | 99.3*     | 37.8   |
| TIM         | 43.4   | 35.3   | 36.5      | 33.2       | 32.0       | 36.0*     | 66.2   |
| SIM         | 58.9   | 51.4   | 47.8      | 41.7       | 41.4       | 99.6*     | 49.8   |
| DIM         | 54.6   | 51.1   | 41.4      | 37.2       | 36.6       | 98.2*     | 50.0   |
| FFA         | 82.2   | 78.3   | 75.3      | 75.2       | 72.4       | 89.2*     | 80.7   |
| TAIG        | 73.9   | 69.1   | 63.4      | 60.5       | 58.4       | 95.0*     | 57.4   |
| NI-FGSN     | 61.9   | 52.5   | 49.5      | 47.5       | 44.7       | 99.2*     | 64.7   |
| MI-FGSN     | 60.3   | 53.4   | 49.3      | 45.6       | 43.0       | 98.8*     | 64.6   |
| VMI-FGSN    | 81.1   | 76.9   | 69.6      | 69.0       | 66.4       | 99.3*     | 73.5   |
| VNI-FGSN    | 80.9   | 77.2   | 70.0      | 68.6       | 65.8       | 99.4*     | 73.8   |
| ANDA        | 93.0   | 90.5   | 86.4      | 83.9       | 81.7       | 99.8*     | 82.8   |
| MultiANDA   | 93.9   | 92.1   | 87.1      | 86.1       | 85.6       | 99.8*     | 84.3   |

| Attack       | ResNet-50 | VGG-19 |
|-------------|-----------|--------|
| BIM         | 33.2      | 26.6   |
| TIM         | 47.3      | 41.3   |
| SIM         | 48.6      | 40.8   |
| DIM         | 61.9      | 55.4   |
| FFA         | 86.2      | 83.9   |
| TAIG        | 62.4      | 55.8   |
| NI-FGSN     | 59.6      | 52.7   |
| MI-FGSN     | 58.5      | 52.6   |
| VMI-FGSN    | 75.3      | 69.9   |
| VNI-FGSN    | 75.4      | 69.1   |
| ANDA        | 95.6      | 94.3   |
| MultiANDA   | 96.5      | 94.9   |

| Attack       | Inc-v3 | Inc-v4 | ResNet-50 | ResNet-101 | ResNet-152 | IncRes-v2 | VGG-19 |
|-------------|--------|--------|-----------|------------|------------|-----------|--------|
| BIM         | 100.0* | 24.1   | 20.3      | 16.4       | 15.7       | 15.6      | 34.3   |
| TIM         | 64.3*  | 35.9   | 35.9      | 31.6       | 30.6       | 25.4      | 70.4   |
| SIM         | 100.0* | 43.3   | 38.2      | 32.9       | 31.1       | 35.9      | 42.2   |
| DIM         | 100.0* | 42.5   | 31.7      | 26.9       | 25.5       | 31.4      | 45.5   |
| FFA         | 98.3*  | 85.4   | 78.4      | 76.1       | 75.3       | 81.2      | 84.5   |
| TAIG        | 99.7*  | 59.6   | 53.3      | 47.8       | 45.9       | 56.7      | 54.2   |
| NI-FGSN     | 100.0* | 44.9   | 40.0      | 37.3       | 35.2       | 39.9      | 56.9   |
| MI-FGSN     | 100.0* | 45.1   | 40.2      | 36.2       | 35.1       | 40.3      | 57.1   |
| VMI-FGSN    | 100.0* | 71.7   | 63.0      | 58.9       | 59.3       | 68.6      | 70.3   |
| VNI-FGSN    | 100.0* | 72.6   | 62.4      | 58.3       | 58.7       | 67.7      | 69.7   |
| ANDA        | 100.0* | 85.7   | 76.1      | 74.1       | 72.8       | 82.3      | 77.0   |
| MultiANDA   | 100.0* | 88.3   | 79.2      | 77.3       | 76.0       | 84.5      | 78.8   |

Table B.1. Attack success rates (%) of the proposed and baseline attacks against seven normally trained models. Sign * indicates the results against white-box models. The best/second results are shown in bold/underlined.

an empirical verification of our methods on a ‘general-purpose classification model’ trained on non-ImageNet-based dataset, CLIP [14]. Preliminary evaluation results, as shown in Table B.3, indicate that ANDA/MultiANDA also generalized well to such cross-domain model.

### B.3. Transferability on ViT-based models

Driven by the curiosity of whether the propose methods work well on ViT-based architectures, we had conducted relating experiments using MultiANDA. Preliminary results in Table B.4 demonstrate notable transferability to ViTs [4], also outperforming the selected baseline method.

#### B.4. Composite Transformation Attack

In particular, we implemented the Composite Transformation (CT) Method with ANDA and MultiANDA (ANDA-CT and MultiANDA-CT). These input transformation techniques can synergize existing transfer-based attack methods, as demonstrated in SIM, DIM, and TIM. As shown by Wang et al. [20], when integrated with VMI-FGSN and VNI-FGSN, VMI-CT-FGSN and VNI-CT-FGSN are recognized as state-of-the-art transferable-based attacks [20]. Therefore, we focused our comparison to these methods. Our results, detailed in Table B.5 and Table B.6, cover both normally trained and defense models, which demonstrate that ANDA-CT and MultiANDA-CT consistently enhance attack performance in almost all cases compared with
### Table B.2: Attack success rates (%) of the proposed and baseline attacks against seven defense models. The best/second results are shown in bold/underlined.

| Attack | Source model | Inc-v3 | Inc-v3 | Inc-v3 | HGD | RS | NRP | NIPS-r3 |
|--------|--------------|--------|--------|--------|-----|----|-----|--------|
|        | ResNet-50    |        |        |        |     |    |     |        |
| BIM    | 11.1         | 12.3   | 13.0   | 13.0   | 5.1 | 17.7| 6.0 |        |
| TIM    | 27.9         | 26.6   | 21.4   | 21.4   | 18.4| 23.9| 23.8|        |
| SIM    | 23.8         | 23.0   | 16.4   | 16.4   | 15.9| 22.6| 18.3| 18.1   |
| DIM    | 16.2         | 14.7   | 10.1   | 10.1   | 11.7| 19.5| 14.9| 12.3   |
| FIA    | 48.9         | 45.0   | 34.7   | 34.7   | 24.2| 46.1| 30.8| 42.8   |
| TAIG   | 49.0         | 46.8   | 41.2   | 41.2   | 12.2| 25.0| 16.5| 13.3   |
| MI-FGSM| 22.0         | 21.6   | 13.3   | 13.3   | 13.4| 28.8| 16.4| 17.0   |
| NI-FGSM| 21.6         | 21.9   | 13.8   | 13.8   | 13.4| 28.6| 15.5| 16.6   |
| VMI-FGSM| 49.2        | 42.9   | 38.8   | 38.8   | 36.0| 36.7| 25.2| 39.2   |
| VNI-FGSM| 49.9        | 44.1   | 37.7   | 37.7   | 35.4| 36.5| 25.6| 39.3   |
| ANDA   | 63.3         | 58.2   | 47.3   | 47.3   | 37.8| 43.9| 28.5| 57.9   |
| MultiANDA | 70.3 | 66.3 | 61.9 | 61.9 | 41.5 | 31.2 | 67.8 |

| Attack | Source model | ResNet-50   | ResNet-50   | VGG-19 |
|--------|--------------|-------------|-------------|--------|
|        |              | BIM         | TIM         |        |
| BIM    | 12.3         | 11.1        | 9.0         | 4.4    |
| TIM    | 34.0         | 33.9        | 31.7        | 30.5   |
| SIM    | 21.4         | 20.9        | 19.5        | 18.1   |
| DIM    | 20.8         | 21.4        | 19.6        | 18.4   |
| FIA    | 44.4         | 39.5        | 33.8        | 32.6   |
| TAIG   | 39.2         | 40.8        | 36.7        | 35.6   |
| MI-FGSM| 26.3         | 23.9        | 22.1        | 20.8   |
| NI-FGSM| 26.2         | 25.2        | 21.7        | 19.8   |
| VMI-FGSM| 47.4        | 45.3        | 39.4        | 37.8   |
| VNI-FGSM| 46.5        | 45.1        | 37.4        | 36.8   |
| ANDA   | 71.4         | 65.9        | 58.9        | 56.3   |
| MultiANDA | 79.7   | 76.8   | 66.8 | 66.8 |

| Attack | Source model | VGG-19   |
|--------|--------------|----------|
|        | VGG-19       |          |
| BIM    | 9.6          | 10.1     |
| TIM    | 23.0         | 25.3     |
| SIM    | 11.4         | 12.0     |
| DIM    | 10.7         | 10.7     |
| FIA    | 13.3         | 12.2     |
| TAIG   | 17.7         | 18.1     |
| MI-FGSM| 13.0         | 13.3     |
| NI-FGSM| 13.9         | 14.2     |
| VMI-FGSM| 21.9        | 20.5     |
| VNI-FGSM| 21.6        | 20.5     |
| ANDA   | 20.9         | 18.0     |
| MultiANDA | 24.6   | 22.2   |

| Attack | Source model | Inc-v3 |
|--------|--------------|--------|
|        | VGG-19       |        |
| BIM    | 11.1         | 11.6   |
| TIM    | 27.5         | 27.6   |
| SIM    | 18.1         | 18.6   |
| DIM    | 13.1         | 13.3   |
| FIA    | 37.4         | 36.7   |
| TAIG   | 38.0         | 36.8   |
| MI-FGSM| 18.3         | 17.2   |
| NI-FGSM| 18.6         | 17.3   |
| VMI-FGSM| 36.9        | 36.9   |
| VNI-FGSM| 36.4        | 37.5   |
| ANDA   | 44.4         | 43.0   |
| MultiANDA | 54.4   | 54.4   |

Table B.3. Success rates (%) on CLIP. Best results are shown in bold/underlined. The zero-shot performance of CLIP on clean ImageNet dataset is 73.7%, i.e., the clean baseline (misclassification) is 26.3%.

### B.5. Comprehensive Comparison with TAIG

In the experiments illustrated in Table B.1 and Table B.2, we used the same parameter settings for all baselines including TAIG and our proposed methods for a fair comparison. To showcase TAIG’s best performance, we replicated the experiments following TAIG’s official settings. In particular, we set $\epsilon = 0.03, 0.05, 0.1$ and $T = 20, 50, 100$. The results presented in Table B.7 shows the exceptional transferability of the baselines. Specifically, our proposed methods have consistently achieved an average success rate increase of 3.5% against black-box normally trained models and 3.6% against advanced defense models. This significant improvement highlights that the approximated posterior distribution over perturbations is more effective in crafting diverse adversarial examples than simply adopting data augmentation techniques.
Table B.4. Success rates (%) of VMI-FGSM/ANDA/MultiANDA on ViT-based target model. The best results are shown in bold.

| Source model | Target model | ViT-L/16 [4] | DeiT3-B/16 [18] | Swin-B/4 [11] | PiT-B [6] |
|--------------|--------------|--------------|-----------------|--------------|----------|
| ResNet-50    | 23.2/28.4/30.5 | 29.7/45.0/46.6 | 24.9/38.7/39.3 | 34.3/52.2/56.3 |
| IncRes-v2    | 29.0/28.4/30.8 | 30.9/41.5/43.5 | 29.8/36.1/38.3 | 38.6/49.9/53.1 |
| VGG-19       | 14.7/14.9/14.8 | 16.5/16.7/16.9 | 20.5/21.5/22.1 | 24.9/26.7/28.6 |

Table B.5. Attack success rates (%) against seven normally trained models using the proposed method and the various selected attacks enhanced by CT with corresponding source models. The best results are shown in bold. Sign * indicates the results against white-box models.

| Attack     | Target model | VGG-19 |
|------------|--------------|--------|
| ANDA-CT    | 84.7/87.9/91.7 | VGG-19 |
| MultiANDA-CT | 84.9/88.9/91.4 | 100*   |
| Inc-v3     | 99.9/91.2/85.8 | 82.3/83.8/82.8 |
| ANDA-CT    | 100*         |
| MultiANDA-CT | 100*         |

B.6. Perturbation Visualization

In previous sections, we compared the attack success rates of our methods with baseline approaches. This section offers a visual analysis of the adversarial perturbations created to assess their effects. We select and display images where adversarial examples generated by ANDA and MultiANDA successfully attack all target models, in contrast to those by VMI-FGSM, which failed to deceive any. These experiments were conducted across both normally trained black-box models and various defense models. Figures B.2 and B.3 demonstrate examples that successfully fooled six black-box models and seven defense models, respectively.

The illustrations reveal that the perturbations generated by ANDA and MultiANDA (Rows 5 and 7) target the semantic features of objects more precisely than VMI-FGSM (Row 3). For instance, in the first column of Figure B.2 and the third column of Figure B.3, ANDA and MultiANDA...
### Table B.6

| Attack                  | Target model | VNI-CT-FGSM | VMI-CT-FGSM | ANDA-CT | MultiANDA-CT |
|-------------------------|--------------|-------------|-------------|----------|--------------|
| IncRes-v2               | Inc-v3       | 86.9        | 85.8        | 92.5     | 93.6         |
| VNI-CT-FGSM             | Inc-v4       | 83.4        | 84.3        | 91.1     | 91.7         |
| VMI-CT-FGSM             | ResNet-50    | 83.8        | 83.8        | 79.9     | 92.1         |
| ANDA-CT                 | IncRes-v2    | 81.1        | 89.3        | 76.9     | 81.5         |
| MultiANDA-CT            | Inc-v3       | 91.3        | 88.9        | 81.2     | 92.9         |
| VMI-CT-FGSM             | IncRes-v3    | 71.1        | 73.3        | 74.4     | 84.3         |
| ANDA-CT                 | HGD          | 84.8        | 84.3        | 84.3     | 84.3         |
| MultiANDA-CT            | RS           | 73.3        | 74.4        | 74.4     | 74.4         |
| VMI-CT-FGSM             | NRP          | 84.8        | 84.3        | 84.3     | 84.3         |
| ANDA-CT                 | NIPS-r3      | 81.3        | 81.3        | 81.3     | 81.3         |

Table B.6. Attack success rates (%) against seven advanced defense models using the proposed method and the various selected attacks enhanced by CT techniques with corresponding source models. The best results are shown in bold.

### Table B.7

| Settings | Target | Inc-v3 | Inc-v4 | ResNet-50 | ResNet-101 | ResNet-152 | IncRes-v2 | VGG-19 |
|----------|--------|--------|--------|-----------|------------|------------|-----------|--------|
| ε = 0.08 | T = 20 | IncRes-v2 | 82.0/49.5 | 79.2/41.2 | 70.4/39.9 | 66.4/37.9 | 65.5/35.5 | 99.2/93.0* | 66.5/38.6 |
|          |        | ResNet-50 | 82.8/37.0 | 76.8/29.0 | 99.7/100.0* | 95.0/67.3 | 93.9/62.3 | 76.8/23.8 | 73.6/40.3 |
|          |        | VGG-19 | 46.8/29.3 | 53.6/27.4 | 39.6/26.4 | 35.1/22.5 | 31.7/18.2 | 31.6/15.5 | 100.0/100.0* |
|          |        | Inc-v3 | 99.9/99.3* | 67.4/34.9 | 54.6/30.5 | 50.7/28.6 | 46.7/24.0 | 58.0/27.6 | 58.6/36.7 |
| ε = 0.05 | T = 50 | IncRes-v2 | 90.4/66.8 | 88.7/60.0 | 82.7/55.6 | 80.8/52.5 | 79.2/49.8 | 99.8/95.5* | 78.2/48.4 |
|          |        | ResNet-50 | 78.5/54.5 | 75.6/46.1 | 92.8/100.0* | 87.1/85.1 | 85.8/81.6 | 74.3/42.4 | 73.9/50.9 |
|          |        | VGG-19 | 68.0/42.7 | 73.7/44.7 | 57.9/37.6 | 52.4/33.2 | 48.7/28.7 | 54.2/27.9 | 100.0/100.0* |
|          |        | Inc-v3 | 100.0/99.7* | 77.6/51.1 | 68.3/45.7 | 65.0/41.7 | 61.4/36.2 | 72.4/45.8 | 69.0/45.3 |
| ε = 0.1  | T = 100 | IncRes-v2 | 95.9/80.7 | 95.0/75.6 | 90.7/70.7 | 90.2/69.3 | 91.1/66.3 | 99.9/97.6* | 89.3/61.8 |
|          |        | ResNet-50 | 98.2/71.6 | 97.2/62.0 | 100.0/100.0* | 99.9/94.4 | 99.8/91.8 | 97.3/63.2 | 94.1/64.6 |
|          |        | VGG-19 | 87.7/59.3 | 92.2/63.7 | 81.5/55.9 | 78.6/45.7 | 77.0/44.6 | 80.5/44.0 | 100.0/100.0* |
|          |        | Inc-v3 | 100.0/100.0* | 88.7/67.6 | 83.0/59.1 | 80.0/53.7 | 78.8/53.4 | 86.7/64.1 | 82.9/55.9 |

Table B.7. Attack success rates (%) against seven normally trained models using the proposed method and TAIG, under official TAIG settings (denoted as ANDA/TAIG). The higher success rate of each comparison is highlighted in bold. Sign * indicates the results on white-box models.

impose perturbations to decisive regions, such as the bird’s head and the flowers, which are crucial for model prediction. This indicates that our methods precisely approximate the optimal solution for the maximization problem introduced in Section 2.1, as alterations in these key areas significantly increase the loss values. Furthermore, the pronounced denoising effect shown in the visualizations of perturbations by MultiANDA highlights the enhanced performance enabled by the mixture of Gaussian models.

### B.7. Time and Memory Analysis

After thorough analysis of the proposed algorithm, we found that the computational overhead largely depends on the number of batch samples during augmentations. The
Table B.8. Attack success rates (%) against seven defense models using the proposed method and TAIG, under official TAIG settings (denoted as ANDA/TAIG). The higher success rate of each comparison is highlighted in bold. The best result for every target model within each setting is underlined.

| Settings | Source | Target | Inc-v3_{ena3} | Inc-v3_{ena4} | IncRes-v2_{ena3} | HGD | RS | NRP | NIPS-rt3 |
|----------|--------|--------|---------------|---------------|--------------------|-----|----|-----|---------|
| t \(= 0.05 \) | IncRes-v2 | 45.5 / 32.3 | 42.8 / 31.8 | 34.1 / 25.3 | 42.7 / 21.6 | 25.1 / 24.8 | 18.3 / 20.2 | 39.7 / 23.4 |
| t \(= 0.05 \) | ResNet-50 | 47.2 / 23.3 | 47.3 / 25.2 | 31.8 / 15.8 | 50.0 / 14.7 | 28.4 / 25.0 | 20.9 / 18.3 | 41.8 / 16.7 |
| t \(= 0.05 \) | VGG-19 | 11.4 / 11.3 | 13.1 / 14.3 | 6.4 / 6.7 | 11.1 / 7.5 | 19.2 / 20.8 | 12.8 / 13.8 | 8.5 / 7.4 |
| t \(= 0.05 \) | Inc-v3 | 27.3 / 24.7 | 29.6 / 26.8 | 14.0 / 12.8 | 21.3 / 23.2 | 21.5 / 22.6 | 15.8 / 19.2 | 18.4 / 13.5 |
| t \(= 0.1 \) | IncRes-v2 | 58.1 / 48.6 | 55.1 / 47.6 | 43.1 / 42.1 | 55.0 / 36.2 | 35.1 / 33.0 | 24.2 / 29.4 | 53.8 / 37.4 |
| t \(= 0.1 \) | ResNet-50 | 48.7 / 38.8 | 46.7 / 37.8 | 32.2 / 28.3 | 48.1 / 28.4 | 35.1 / 32.4 | 22.6 / 25.3 | 44.7 / 28.1 |
| t \(= 0.1 \) | VGG-19 | 14.6 / 18.3 | 15.0 / 18.4 | 9.0 / 11.6 | 18.4 / 13.1 | 22.2 / 24.0 | 13.5 / 14.8 | 12.4 / 12.9 |
| t \(= 0.1 \) | Inc-v3 | 36.2 / 36.5 | 35.0 / 36.0 | 17.9 / 22.9 | 26.8 / 22.3 | 27.0 / 28.0 | 19.3 / 25.9 | 27.2 / 25.5 |

Table B.9. CPU Time, GPU Time, GPU Memory (GB) and attacking performance of different methods. The source model and target model are ResNet-50 and IncRes-v2, respectively.

| SIM | FIA | ANDA | VMI | VNI |
|-----|-----|------|-----|-----|
| CPU Time (s) | 1.366 | 1.043 | 1.968 | 2.629 | 2.648 |
| GPU Time (s) | 1.314 | 0.982 | 2.007 | 2.608 | 2.624 |
| GPU Memory (GB) | 1.390 | 7.360 | 7.242 | 5.080 | 5.070 |
| Suc. rates (%|IncRes-v2) | 35.3 | 80.4 | 94.0 | 68.3 | 67.9 |

B.8. Optimization Trajectory

To evaluate the optimization efficacy of ANDA, we analyzed the optimization trajectories of ANDA, and baseline methods VMI-FGSM and VNI-FGSM. For enhanced visualization, we employed Principal Component Analysis (PCA) [8] and projected the iteratively optimized samples (including original images and their corresponding adversarial examples of 10 iterations against Inc-v3) along the top four principal components (v1-v2 and v3-v4). We randomly chose the initial inputs and selected the ones that succeeded in deceiving the target black-box models. We visualized the iteration paths of these algorithms on the 2D loss landscape against eight black-box models. The results, as depicted in Figure B.4, demonstrate that ANDA finds more efficient optimization paths than VMI/VNI-FGSM, leading to enhanced optimization outcomes.

C. Ablation Study

In ANDA, we integrate two components, namely data augmentations and the approximation of perturbation distribution, to enhance the transferability of adversarial examples. This section examines the individual contributions of these components to ANDA’s attack performance and the impact of key hyper-parameters. Note that white-box attack performance may reach 100%. Therefore, we focus on black-box settings to precisely evaluate the efficacy of each component. We select seven representative models, including four normally and three adversarially trained models, as targets.

C.1. Number of Augmentation Batches

To determine the contribution of augmentation to ANDA, we compared results with and without augmentation, i.e., \( S \) includes one sample \( \{ x^{(i)}_{adv} \} \) or \( n \) samples...
Figure B.2. Perturbation visualization of the examples that fool the normally trained models

Figure B.3. Perturbation visualization of the examples that fool the defense models

\{\text{AUG}_i(x_{adv}^{(t)})\}_{i=1}^n\), respectively. Figure C.1 (a) illustrates that batch augmentation significantly boosts performance in various black-box models. The main reason is that augmentation introduces stochasticity into the iterative optimization procedure, forming a SGA procedure which help to characterize the asymptotic Gaussian distributions that effectively enhances the robustness of the results.

Furthermore, we analyzed the impact of varying the augmented batch number \(n\). Considering both the attack performance and computational overhead, we adjusted \(n\) to represent the perfect squares of the number of translated pixels in one dimension, ranging from 4 to 49 (refer detailed augmentation implementation in Section C.3). We observe from Figure C.2 that the overall attack performance generally improves with an increase in \(n\).

C.2. Effect of Perturbation Distribution

We further examined the impact of the approximated perturbation distribution, specifically the accumulation of historical gradients along the optimization trajectory. The results, depicted in both Figure C.1 (b) and Figure C.2, demonstrate that the attack effectiveness is significantly enhanced by the statistical analysis of historical gradients. Notably, the dashed lines in Figure C.2 are consistently higher than their corresponding solid lines of the same color, underscoring the considerable value of incorporating historical gradient data into the optimization process.

C.3. Magnitude of augmentations

Another important aspect to explore is the extent to which image translations provide the most benefits. For these translations, we utilize \(tx\) for the horizontal direction and \(ty\) for the vertical direction in the affine transformation, representing the degree of translation. Consider a scenario where we have a source image \(x_{src}\) and a translated image \(x_{tgt}\). We can then formulate the translation process using the following affine transformation in homogeneous representation. This process involves mapping the \(i_{th}\) pixel points from \(x_{src}\) to \(x_{tgt}\):

\[
\begin{pmatrix}
1 & 0 & tx \\
0 & 1 & ty \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
x_i \\
y_i \\
1
\end{pmatrix}
= 
\begin{pmatrix}
x'_i \\
y'_i \\
1
\end{pmatrix}
\]

The magnitudes of \(tx\) and \(ty\) represent the extent of translations. Specifically, we define two parameters for
Figure B.4. Visualization of optimization trajectories for generated adversarial examples by ANDA, VMI-FGSM, and VNI-FGSM.

This process: \( n \), representing the number of augmentations, and \( \text{Augmax} \), defined as \( \max(\text{abs}(tx), \text{abs}(ty)) \). It is important to note that the valid range for \( \text{Augmax} \) is \([0, 2]\), and \( n \) takes on values that are perfect squares in our experiments. For example, with \( n = 25 \) and \( \text{Augmax} = 0.3 \), we evenly distribute \( \sqrt{n} \) samples across the interval \((-\text{Augmax}, \text{Augmax})\) for both \( tx \) and \( ty \). This results in a set of \( tx \) values, denoted as \( T_x \), and a set of \( ty \) values, denoted as \( T_y \). To illustrate, consider \( T_x \) as an example:

\[
tx_i = -\text{Augmax} + i \times \frac{2\text{Augmax}}{\sqrt{n} - 1}
\]

\[T_x = \{tx_i|i = 0, 1, \ldots, \sqrt{n} - 1\}\]

The process to form \( T_y \) follows the same approach as \( T_x \). We then calculate the Cartesian product of these two sets to
obtain a series of \((tx, ty)\) pairs, which are used for further translation operations. The function \(\text{AUG}(x)\) represents a set of translated images generated from these operations:

\[
T_x \times T_y = \{(tx, ty) | tx \in T_x \land ty \in T_y\}
\]

\[
\text{AUG}(x) = \{\text{AUG}_i(x)\}, \text{where } i \text{ is the index of } T_x \times T_y
\]

As illustrated in Figure C.3, we present the Attack Success Rates (ASRs) on various models while varying the value of \(\text{Augmax}\). The results indicate that image translations within a suitable range can enhance black-box transferability. However, overly intense transformations lead to a loss of intrinsic image information, resulting in decreased attack performance. To illustrate this effect, we extended the range of \(\text{Augmax}\) and plotted the attack success rates, along with specifically translated images at different values of \(\text{Augmax}\), as shown in Figure C.4. It is observed that moderate translation extents, such as \(\text{Augmax} = 0.3\), are beneficial. Yet, when the value of \(\text{Augmax}\) exceeds 0.5, there is a significant drop in attack performance, even on a white-box model like VGG-19. This suggests that while appropriate image transformations can promote adversarial
translatability, excessive alterations can result in the loss of crucial information.

C.4. Augmentation Types

We also empirically explored the impact of different types of augmentations. Translation operations yielded effects similar to random resize and padding, as discussed in [24, 26]. Although our methods could potentially benefit from stochastic augmentations, this paper focuses solely on deterministic operations, such as translations, to enable more stable quantitative analysis. While scale operations, as in [10], are feasible, their effectiveness is inherently limited by the nature of the transformation, showing no further improvement beyond \( n > 5 \) augmentations. Additionally, augmentations like uniform additive noise [12, 20], Gaussian additive noise [23], and Bernoulli multiplicative noise [21] resulted in only modest performance enhancements in our experiments.

C.5. Number of Ensemble Models for MultiANDA

We investigated the influence of the number of components (\( K \)) in MultiANDA on performance across our selection of models, including six black-box normally trained models and seven defense models. The results, as shown in Figure C.5, indicate that the performance of MultiANDA steadily improves with an increasing number of ANDAs. This is particularly evident with defense models, where MultiANDA achieves approximately a 10% improvement in success rates for five of the defense models. These findings further suggest that sampling from a Gaussian mixture distribution enhances sample diversity, thereby boosting the translatability of the perturbed samples.

**References**

[1] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[2] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018.

[3] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.
