A Implementation Details

In this section, we provide implementation details that are omitted in the main paper due to the page limit.

Models. We leverage the pre-trained models of PNASNet-5-Large (PNASNet), adv-Inception-v3 (Inc-v3$_{adv}$) and ens-adv-Inception-ResNet-v2 (IncRes-v2$_{ens}$) from the PyTorch Image Models library[2]. For other target models, we use the pre-trained models provided by the Torchvision library[3]. Moreover, the model architecture of FD and DA is Resnext101. For HGD and R&P, we adopt the official models provided in the corresponding papers.

Logit Loss. The logit loss directly leverages the logit output to calculate the backpropagated gradients:

$$J_{logit}(\boldsymbol{x}, y) = -l_y(\boldsymbol{x}), \tag{A.1}$$

where $l_y(\cdot)$ denotes the logit output with respect to the ground-truth class y. It addresses the vanishing gradient problem by eliminating the softmax function used in the Cross-Entropy loss.

Loss Flatness Visualization. We follow the previous work [25] to visualize the loss flatness around the given adversarial examples. Specifically, we first randomly sample a direction vector $\boldsymbol{d}$ from a Gaussian distribution and normalize it on an l_2 unit norm ball. Then, we calculate the loss flatness $L(a)$ as follows:

$$L(a) = \|J(\boldsymbol{x}^{adv} + a \cdot \boldsymbol{d}, y) - J(\boldsymbol{x}^{adv}, y)\|, \tag{A.2}$$

where a is the deviation magnitude of $\boldsymbol{x}^{adv}$, $J(\cdot, \cdot)$ is the loss function of the surrogate model, y is the ground-truth label. We repeat the above calculation 20 times with different random directions $\boldsymbol{d}$ and take the averaged value to represent the loss flatness.

B Additional Experimental Results

In this section, we provide additional experimental results to further demonstrate the effectiveness and generalizability of the proposed GSA. We first compare GSA with baseline methods on another two surrogate models. Then we evaluate GSA with smaller magnitudes of perturbations.

B.1 Evaluation on Other Surrogate Models

We compare GSA with the baseline methods using VGG16 and Inception-v3 as the surrogate models. The results of this comparison are summarized in Table B.1, which lists the attack success rates on normally trained models with a perturbation magnitude of $\epsilon = 16/255$ and 10 iterations. Among the baseline methods, TAIG-R does not demonstrate advanced adversarial transferability on these two surrogate models as it does on ResNet50, whereas Admix is the best-performing baseline method. However, GSA exceeds the performance of Admix on the surrogate models VGG16 and Inception-v3 by a clear margin of 2.3% and 13.4%, respectively.

These results demonstrate that GSA has superior adversarial transferability compared with existing state-of-the-art transfer-based attacks on various surrogate models. The significant improvement in attack success rates suggests that GSA is a powerful approach to generating transferable adversarial examples that can evade multiple models.

[2] https://github.com/rwightman/pytorch-image-models
[3] https://github.com/pytorch/vision

B.2 Evaluation with Smaller Perturbations

We evaluate the performance of GSA and the baseline methods with smaller magnitudes of perturbations to show the effectiveness of GSA in stealthiness-sensitive scenarios. As the perturbation magnitude decreases, the stealthiness of the adversarial samples increases. However, this often comes at the cost of lower attack success rates.

Table B.2 illustrates the attack success rates with the magnitude of perturbations $\epsilon = 4/255$ and $\epsilon = 8/255$. When $\epsilon = 4/255$, TAIG-R achieved a 1.7% higher attack success rate on average compared with GSA. However, when the perturbation magnitude increased to $\epsilon = 8/255$, GSA outperformed TAIG-R by an average attack success rate of 3.1%. As previously shown in the main paper, GSA also demonstrated better adversarial transferability with $\epsilon = 16/255$ while reducing the overhead of TAIG-R by 62%. These results demonstrate that GSA can generate effective adversarial examples with smaller magnitudes of perturbations and has the advantage of being stealthy while maintaining a high attack success rate. In addition, it also has better adversarial transferability with larger magnitudes of perturbations and lower overhead than state-of-the-art methods.

C Visualization of GSA Adversarial Examples

In Figure C.1, we present a visual comparison of 10 randomly selected benign images and their corresponding adversarial examples crafted by GSA. In particular, we generated adversarial examples on ResNet50 using the perturbation magnitudes of 4/255, 8/255, and 16/255. We can observe that the adversarial perturbations generated by GSA are imperceptible to human eyes, even when using relatively larger perturbations. This highlights the stealthiness of GSA-crafted adversarial examples and makes GSA a powerful tool for evaluating the robustness of models and identifying potential security vulnerabilities in sensitive scenarios.

Table B.1: Attack success rates (%) on normally trained models using VGG16 and Inception-v3 as the surrogate models. The symbol $*$ indicates the white-box results on the corresponding surrogate model. The attack success rate of the surrogate model is not included in the average attack success rate. The highest values of each column are marked in bold.

Method	ResNet50	DenseNet121	VGG16	Inception-v3	MobileNet	SENet154	PNASNet	Average
IFGSM	39.5	37.2	**100.0***	25.5	50.3	30.1	20.6	33.9
DI	44.7	44.3	**100.0***	26.8	57.1	37.3	31.2	40.2
TI	42.0	44.7	**100.0***	29.9	55.9	40.7	30.2	40.6
SI	67.5	70.3	**100.0***	52.1	75.0	53.9	49.4	61.4
Admix	75.9	79.7	**100.0***	62.4	82.6	63.1	62.0	71.0
TAIG-R	58.2	62.2	**100.0***	49.0	68.9	42.2	40.0	53.4
GSA	**81.7**	**81.0**	**100.0***	**63.7**	**85.0**	**66.1**	**62.2**	**73.3**
IFGSM	34.5	35.3	44.3	**100.0***	43.8	23.8	19.0	33.5
DI	48.4	50.2	59.7	99.8*	54.9	34.2	35.1	47.1
TI	37.2	41.2	48.6	**100.0***	48.7	28.0	20.8	37.4
SI	39.1	43.8	48.3	99.9*	50.7	26.0	18.6	37.8
Admix	48.8	51.6	57.1	**100.0***	60.2	32.8	25.6	46.0
TAIG-R	44.0	51.7	48.0	99.3*	54.7	26.4	25.6	41.7
GSA	**62.4**	**65.4**	**69.1**	**100.0***	**68.4**	**47.3**	**43.6**	**59.4**

Table B.2: Attack success rates (%) on normally trained models with $\epsilon = 4/255$ and $\epsilon = 8/255$. The symbol $*$ indicates the surrogate model used to generate adversarial examples. The attack success rate of the surrogate model is not included in the average attack success rate.

Method	ϵ	ResNet50*	DenseNet121	VGG16	Inception-v3	MobileNet	SENet154	PNASNet	Average
IFGSM	4/255	**100.0**	33.0	38.2	18.6	37.1	15.1	9.5	25.3
	8/255	**100.0**	61.6	60.8	27.4	56.4	31.8	21.6	43.3
DI	4/255	99.8	42.0	55.2	22.3	44.3	20.6	15.6	33.3
	8/255	**100.0**	73.6	81.6	36.7	71.8	48.2	40.6	58.8
TI	4/255	**100.0**	39.9	39.3	21.6	38.7	16.6	11.6	28.0
	8/255	**100.0**	68.0	64.5	32.8	61.1	36.7	27.1	48.4
SI	4/255	**100.0**	51.0	44.3	27.4	47.5	19.0	12.3	33.6
	8/255	**100.0**	79.5	70.2	47.7	73.6	42.5	30.1	57.3
Admix	4/255	**100.0**	53.6	48.8	28.7	55.2	20.5	12.9	36.6
	8/255	**100.0**	86.9	78.9	55.4	81.9	49.5	36.8	64.9
TAIG-R	4/255	**100.0**	**68.6**	60.9	**35.8**	**63.1**	**32.6**	**24.0**	**47.5**
	8/255	**100.0**	87.5	80.5	**57.5**	81.3	56.2	47.7	68.5
GSA	4/255	**100.0**	62.1	**61.7**	34.8	61.7	31.3	23.3	45.8
	8/255	**100.0**	**87.8**	**86.7**	54.9	**84.5**	**62.3**	**53.2**	**71.6**

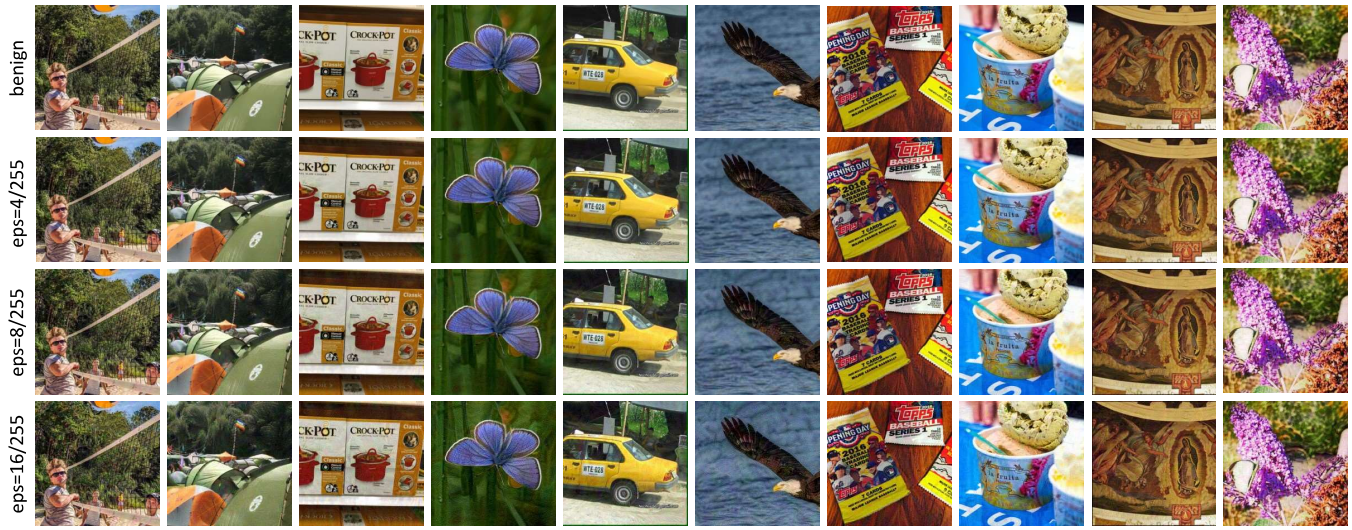

Figure C.1: Visualization of ten randomly selected benign images and their adversarial examples crafted by the proposed GSA. The surrogate model is ResNet50.