applied sciences

MDPI

Article

Segment Shards: Cross-Prompt Adversarial Attacks against the Segment Anything Model

Shize Huang [1,2], Qianhui Fan [1,2,*], Zhaoxin Zhang [1], Xiaowen Liu [1], Guanqun Song [1] and Jinzhe Qin [1]

[1] The Key Laboratory of Road and Traffic Engineering, Ministry of Education, Tongji University, 4800 Caoan Rd., Shanghai 201804, China; zhangzhaoxin@ust.hk (Z.Z.); 2131306@tongji.edu.cn (J.Q.)
[2] Shanghai Key Laboratory of Rail Infrastructure Durability and System Safety, 4800 Caoan Rd., Shanghai 201804, China
* Correspondence: 2310848@tongji.edu.cn

Abstract: Foundation models play an increasingly pivotal role in the field of deep neural networks. Given that deep neural networks are widely used in real-world systems and are generally susceptible to adversarial attacks, securing foundation models becomes a key research issue. However, research on adversarial attacks against the Segment Anything Model (SAM), a visual foundation model, is still in its infancy. In this paper, we propose the prompt batch attack (PBA), which can effectively attack SAM, making it unable to capture valid objects or even generate fake shards. Extensive experiments were conducted to compare the adversarial attack performance among optimizing without prompts, optimizing all prompts, and optimizing batches of prompts as in PBA. Numerical results on multiple datasets show that the cross-prompt attack success rate (ASR*) of the PBA method is 17.83% higher on average, and the attack success rate (ASR) is 20.84% higher. It is proven that PBA possesses the best attack capability as well as the highest cross-prompt transferability. Additionally, we introduce a metric to evaluate the cross-prompt transferability of adversarial attacks, effectively fostering research on cross-prompt attacks. Our work unveils the pivotal role of the batched prompts technique in cross-prompt adversarial attacks, marking an early and intriguing exploration into this area against SAM.

Keywords: adversarial attack; deep learning; foundation model

Citation: Huang, S.; Fan, Q.; Zhang, Z.; Liu, X.; Song, G.; Qin, J. Segment Shards: Cross-Prompt Adversarial Attacks against the Segment Anything Model. *Appl. Sci.* **2024**, *14*, 3312. https://doi.org/10.3390/app14083312

Academic Editor: Luis Javier Garcia Villalba

Received: 19 March 2024
Revised: 8 April 2024
Accepted: 9 April 2024
Published: 15 April 2024

1. Introduction

In recent years, many researchers have suggested that using "foundation models" [1] as support for various downstream tasks is a promising trend in the development of AI. Well-known foundation models, such as BERT [2], GPT-3 [3], CLIP [4], and ViT [5], demonstrate remarkable feature learning and expression capabilities in crucial tasks within the domains of natural language processing (NLP), cross-modality matching (CMM), and computer vision (CV). Recently, the segment anything model (SAM) [6], proposed by Meta, demonstrated remarkable and versatile capabilities in visual segmentation tasks, similar to the aforementioned foundation models. SAM is expected to become a crucial foundation model for basic image segmentation tasks and has the potential to serve as a supporting module for numerous downstream tasks.

However, the robustness and security of these foundation models have become crucial research topics that cannot be overlooked, as foundation models are widely used in downstream tasks. Many studies [7–11] have indicated that almost all DNNs are vulnerable to attacks from adversarial examples. Recently, researchers have also started to pay attention to the robustness of foundation models. Paul and Chen [12] conducted a study on the robustness of the ViT model and found that it has superior robustness compared to other models, as it is better able to defend against adversarial attacks. The researchers believe that the main reasons include the following: (1) the attention mechanism of ViT is capable of extracting rich global contextual information from images, which enhances the model's

robustness; (2) training the model on large-scale datasets increases its robustness. Similarly, SAM uses ViT as the backbone of its image encoder, and it is trained on an extremely large-scale dataset. Therefore, the robustness of SAM is likely to be higher than other DNNs. Huang et al. [13] conducted adversarial attacks against SAM with single-point prompts using FGSM [14], PGD [15], BIM [16], and SegPGD [17] methods, and as a result, the background area is little affected.

What is particularly noteworthy is that SAM introduced a prompt mechanism, which requires adversarial examples to have some degree of transferability across different prompts to successfully perform the attack task. Unlike [13], Zhang et al. [18] explored the transferability of the cross-prompts of adversarial attacks against SAM by gradually increasing the number of point prompts. The examination of cross-prompt transferability in attacks, exploring variations in the number of point prompts, was conducted in [19]. Based on the outcomes, the authors infer that it is challenging to enhance the cross-prompt transferability of attacks by simply further increasing the number of point prompts.

Similar to recent scholarly discussions [18,19], this paper focuses on the following question: How can one generate adversarial examples that possess higher cross-prompt transferability?

However, current methods [18,19] insufficiently delve into leveraging prompt information techniques. They primarily focus on attacking prominent foreground objects, neglecting attacks on the entirety of image scenes. Based on this understanding, our motivation comes from the need to make more use of prompt information to generate more cross-prompt transferable adversarial examples in the generation stage of an adversarial attack. Therefore, we propose a new adversarial attack method called PBA (prompt batch attack) to further improve the cross-prompt transferability of adversarial attacks against SAM. In this method, we leverage prompt information in a good way instead of attacking without prompts or with all prompts, which improves the attack success rate as well as the cross-prompt transferability of adversarial examples. Furthermore, our PBA method can cause the SAM segmentation results to fail across the entire image, shifting from 'segment anything' to 'segment shards'.

In summary, our work makes the following contributions:

1. We design three adversarial attack methods with different ways of utilizing prompt information to perform adversarial attacks on SAM. The most effective method (PBA) exhibits both a high attack success rate and excellent cross-prompt transferability.
2. We propose an effective and clear metric (cross-prompt attack success rate, ASR^*) to evaluate the cross-prompt transferability of adversarial attacks. This metric takes into account both the degree of prompt variance and the attack success rate.

The rest of this paper is organized as follows. Section 2 describes some background information about adversarial attacks, visual foundation models, and the adversarial robustness of the foundation model. Section 3 deals with the definitions of adversarial attack, cross-prompt adversarial attack, and cross-prompt attack success rate. Section 4 proposes the prompt batch attack (PBA) method along with two basic comparison algorithms: The no prompt attack (NPA) method and the prompt attack (PA) method. Section 5 presents the experimental and numerical results, and some discussions of the experimental results are presented in Section 6. Finally, Section 7 presents the summary and future prospects for the entire paper.

2. Related Work

In this section, we first provide a concise overview of the pertinent background of this paper. Then, we dive into the details from three perspectives: adversarial attacks, visual foundation models, and adversarial attacks against foundation models.

2.1. Background

Adversarial attacks and foundation models constitute the two main background themes of this paper. An adversarial attack involves adding a small perturbation to the model's input, causing a severe degradation in the neural network model's performance. Inputs

with this small perturbation are referred to as adversarial examples. On the other hand, the foundation model represents a recent paradigm of deep neural networks with strong generalization capabilities. In this paper, a novel adversarial attack method against a typical visual foundation model with cross-prompt transferability is proposed and investigated.

2.2. Adversarial Attack

In 2014, Szegedy et al. [20] found the existence of adversarial examples and proposed the L-BFGS attack method. FGSM [14], proposed by Goodfellow, is an adversarial perturbation generation method based on gradient backpropagation computation. In order to solve the problem of the FGSM attack's instability and the obviousness of the perturbation, DeepFool [21] and C&W [22] methods were proposed, which are dedicated to finding a minimal adversarial perturbation. Furthermore, JSMA [23] performed the adversarial attack by only changing the values of a few pixels. However, these methods [14,20–23] focused on classifiers and lacked exploration in the more complex and practical deep neural network models. After that, DAG [24] completed the adversarial attack on the object detection and instance segmentation model by the dense attack method. Then, many studies extended adversarial attacks to models in different domains, including object detection [25], instance segmentation [26,27], human pose estimation [28], person re-identification [29], person detector [30], visual language model [31,32], remote sensing [33], and 3D point cloud processing [34,35]. These works show that adversarial attacks can threaten the security of various neural network-based application models.

Many research works show that adversarial examples have cross-model transferability. Zhou et al. [36] proposed two methods to improve cross-model transferability: filtering high-frequency perturbations and maximizing the distance between the clean image and the adversarial example. References [37,38] generated more transferable adversarial examples by adding a variance-adjusted regularization module. In addition, some studies [39–42] have suggested that the transferability of adversarial examples can be improved by increasing the diversity of inputs.

With the emergence of SAM featuring prompt-guided inputs, few recent studies [18,19] have started to focus on the cross-prompt transferability of adversarial attack methods. (We compare these attack methods against SAM in detail in Section 2.4). However, these research studies are confined to attacking only the image encoder or enhancing cross-prompt transferability by increasing the number of point prompts. Consequently, there is a lack of research on ways to leverage prompt information to improve cross-prompt transferability.

2.3. Visual Foundation Models

Foundation models, which are considered the next wave in AI, can be used for numerous downstream tasks with minimal fine-tuning. In the field of computer vision, foundation models are still in an early stage [1]. However, quite a few studies [4,43–46] have effectively contributed to the development process of Visual foundation models. Compared to traditional supervised models, foundation models can directly use super large-scale unlabeled raw data. The development of unsupervised learning in computer vision efficiently reduced the dependence on manually labeled datasets, which facilitated the development of visual foundation models; Chen et al. [43] proposed an unsupervised learning technique using a contrast learning framework to train models. He et al. [44] introduced self-supervised techniques in vision tasks and achieved results equivalent to the supervised model. Then, He et al. [45] introduced Transformer to vision tasks and accomplished training on massive data with exciting effects. The pre-trained model (ViT) is widely reused in vision downstream tasks to extract the features of images. In addition, foundation models have contributed to the development of foundation models in visual generation. For example, the CLIP [4] foundation model has inspired a range of CLIP-based visual generation studies. The DALL·E 2 model [46], among them, shows an impressive ability to create images.

Recently, the segment anything model (SAM) [6], a foundation model proposed by Meta for the visual segmentation task, has received a lot of attention from researchers. By building a special data engine with semi-supervised training, SAM has accomplished training on extremely large-scale data and has demonstrated extraordinary ability in the zero-shot segmentation task.

2.4. Adversarial Attack against the Foundation Model

The ability of a deep neural network model to resist attacks from adversarial examples is referred to as the model's adversarial robustness. Given the extensive reuse of foundation models, their adversarial robustness is a significant and crucial topic. Bommasani et al. [1] points out that improving the adversarial robustness of foundation models presents an important opportunity. Shafahi et al. [47] suggested that robust feature extractors can be useful for transferring robustness to other domains.

Existing studies [12,15,48,49] have shown that large datasets and large model capacities are beneficial for improving adversarial robustness. Schmidt et al. [48] suggested that training an adversarial robust model requires more data. References [15,49] found that improving adversarial robustness requires larger model parameters and capacity. Paul and Chen [12] showed that Visual Transformers have higher adversarial robustness than other models. Unfortunately, it has been demonstrated that the CLIP foundation model does not exhibit sufficient robustness against adversarial attacks [50].

Researchers still need to study more about the adversarial robustness of foundation models in different domains [1]. The SAM, meanwhile, has not been sufficiently studied for its adversarial robustness as an important foundation model in the field of computer vision. As shown in Table 1, we collected the latest adversarial attack methods against SAM and focused on their comparison regarding the topic of cross-prompt transferability. Huang et al. [13] did not consider the influence of prompt changes on their attack success rate, and the attack results have limited effect in the background part of the image. Zhang et al. [18] conducted targeted attacks against SAM, including mask removal, mask enlargement, and mask manipulation. Additionally, initial observations revealed an increase in the cross-prompt transferability of adversarial attacks against SAM, as the number of point prompts increased. Zheng and Zhang [19] contend that an increase in the number of point prompts has a limited effect on the improvement of cross-prompt transferability. Therefore, they propose a prompt-agnostic attack method, which only attacks the image encoder of SAM. In this paper, we introduce a technique termed PBA (prompt batch attack) against SAM. The PBA method enhances the cross-prompt transferability of adversarial examples by attacking SAM with dense point prompts in batches. Furthermore, the attack effect generated by PBA, resembling fragmented glass, impacts the entire image.

Table 1. Comparison of adversarial attack methods against SAM.

Study	Year	Cross-Prompt Transferability
Huang et al. [13]	2023	No discussion.
Zhang et al. [18]	2023	The increase in the number of point prompts can improve the cross-prompt transferability.
Zheng and Zhang [19]	2023	An increase in the number of point prompts has a limited effect on the improvement of cross-prompt transferability.

3. Problem Definition

3.1. Adversarial Attack Problem Definition

Let X denote the input image, which follows a large-scale image data distribution, G_{data}, i.e., $X \sim G_{data}$. For an input image sample, x, where $x \in X$, its ground truth for the segmentation task is a set of pixel coordinate sets, denoted as $y \sim Y$, where $y = \{M_1, M_2, ..., M_N\}$ and M_i share the same shape with X. The learning system for image segmentation tasks, such as SAM, can be represented as an abstract function $f : X, P \rightarrow Y$,

where P denotes the distribution of prompt. The task of segmenting anything can be described as follows:

$$\mathbf{Y} = f(\mathbf{X}, \mathbf{P}; \theta) \tag{1}$$

The SAM conceptualizes the abstract function, f, into three components: the image encoding system, $g_{im} : X \rightarrow E$, where E signifies the distribution of image embedding; the prompt encoding system, $g_{pt} : P \rightarrow F$, where F signifies the distribution of prompt embedding; and the decoding system, $h_{dec} : E, F \rightarrow Y$. The task of segmenting anything using SAM can be mathematically described by the following equation:

$$\mathbf{Y} = h_{dec}(g_{im}(\mathbf{X}), g_{pt}(\mathbf{P})) \tag{2}$$

The problem of generating adversarial examples to attack SAM can be described by the following:

$$x^{adv} = arg \min_{x^{adv}} \mathcal{L}(h_{dec}(g_{im}(x^{adv}), g_{pt}(\mathbf{P})), y') \tag{3}$$

where x^{adv} denotes the image with adversarial perturbation, $\mathbf{P}$ denotes any possible input prompt, y' denotes the error output of h_{dec}, and $\mathcal{L}$ denotes the loss function of the adversarial attack.

3.2. Cross-Prompt Attack Problem Definition

For consistent understanding, in this section, we provide an intuitive description and formal definition of the cross-prompt adversarial attack (CAA) and the cross-prompt adversarial success rate (ASR^*).

3.2.1. Cross-Prompt Adversarial Attack

As shown in Figure 1, the cross-prompt adversarial attack (CAA) refers to the scenario in which adversarial examples generated using certain prompt information are required to attack the model under various prompts. The process of CAA can be divided into two steps: adversarial example generation and adversarial attack implementation. In the first step, attackers typically utilize specific and limited prompts. However, in the second step, attackers aim for the generated adversarial examples to succeed across diverse prompts, rather than being effective only under specific prompts.

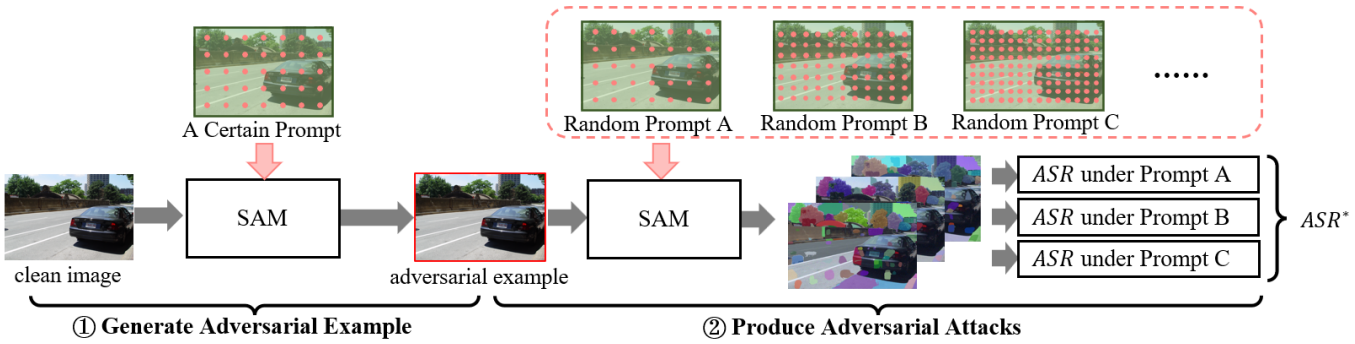

Figure 1. Cross-prompt adversarial attack. The ASR denotes the attack success rate and the ASR* denotes the cross-prompt attack success rate.

3.2.2. Cross-Prompt Attack Success Rate

In this section, we introduce a metric, ASR*, for evaluating the cross-prompt transferability of adversarial examples. For the sake of discussion, we assume that all prompts take the form of point coordinates.

Before entering the discussion of the definition of the cross-prompt attack success rate (ASR*), we first provide a complete definition of the attack success rate (ASR). Following some previous work [20,21,24], ASR is defined by the drop in mIoU as Equation (4).

$$ASR = \frac{mIoU_{nos} - mIoU_{atk}}{mIoU_{nos}} \tag{4}$$

where $mIoU_{nos}$ denotes the mIoU value of the segmentation results on the image with randomly added noise, and $mIoU_{atk}$ denotes the mIoU value of the segmentation results on adversarial examples.

$$mIoU = \frac{1}{n}\sum_{i=0}^{n}\max_{j=0}^{m}\frac{M_i \cap M_j}{M_i \cup M_j} \tag{5}$$

where M_i denotes the i-th object mask of the segmentation results of SAM in the adversarial image or noise image, and M_j denotes the j-th object mask of the segmentation results of SAM in the clean image.

Next, we start with the illustration of the definition of ASR*, where all ASR_i without the * sign are defined by Equation (4) above.

First, we use d_{ab} to measure the degree of difference between prompt a and prompt b. This d_{ab} can be calculated as Equation (6). Here, J_δ denotes the Jaccard Distance, p_a denotes the set of coordinates of all points in prompt a, and p_b denotes the set of coordinates of all points in prompt b. Finally, n denotes the number of points in prompt a and m denotes the number of points in prompt b.

$$d_{ab} = J_\delta(p_a, p_b) + \sum_{i=0}^{n}\min_{j=0}^{m}\frac{\sqrt{(p_a^{ix} - p_b^{jx})^2 + (p_a^{iy} - p_b^{jy})^2}}{\sqrt{H^2 + W^2}} \tag{6}$$

Then, suppose that the prompt t, is the prompt used in generating the adversarial example, and a total of k types of prompts are used to produce adversarial attacks. The cross-prompt transferability of the adversarial example across k different types of prompts can be calculated using Equations (7) and (8), as follows:

$$ASR_t^* = \sum_{i=0}^{k}\left(\frac{d_{it}}{D_t} \times ASR_i\right) \tag{7}$$

$$D_t = \sum_{i=0}^{k} d_{it} \tag{8}$$

Here, d_{it} denotes the degree of difference between prompt i and prompt t, ASR_i denotes the attack success rate of the adversarial example under prompt i, and ASR_t^* denotes the cross-prompt attack success rate of the adversarial example.

4. Method

In this section, we introduce three different algorithms, namely NPA, PA, and PBA, to attack SAM in a white box setting. The different degrees of utilization of SAM's prompt information are the main differences between these three algorithms.

As shown in Figure 2, in the no prompt attack (NPA) method, gradient-based adversarial perturbation optimization only involves the image encoder structure of SAM. In the prompt attack (PA) method, gradient-based adversarial perturbation optimization involves all structures of SAM with prompt input as a whole. In the prompt batch attack (PBA) method, each optimization iteration uses only one of the prompt batches as input and keeps using different prompt batches during optimization iterations. Additionally, the PBA adds momentum information between different prompt batch iterations to stabilize the update direction. It is worth noting that momentum information between iterations is also used in the PA method but this momentum information is not obtained under the influence

of different prompt batches. We will elaborate on these three attack algorithms (NPA, PA, and PBA) in Sections 4.1–4.3. When reading these sections, one may refer to the symbol table (Table 2) to help with comprehension.

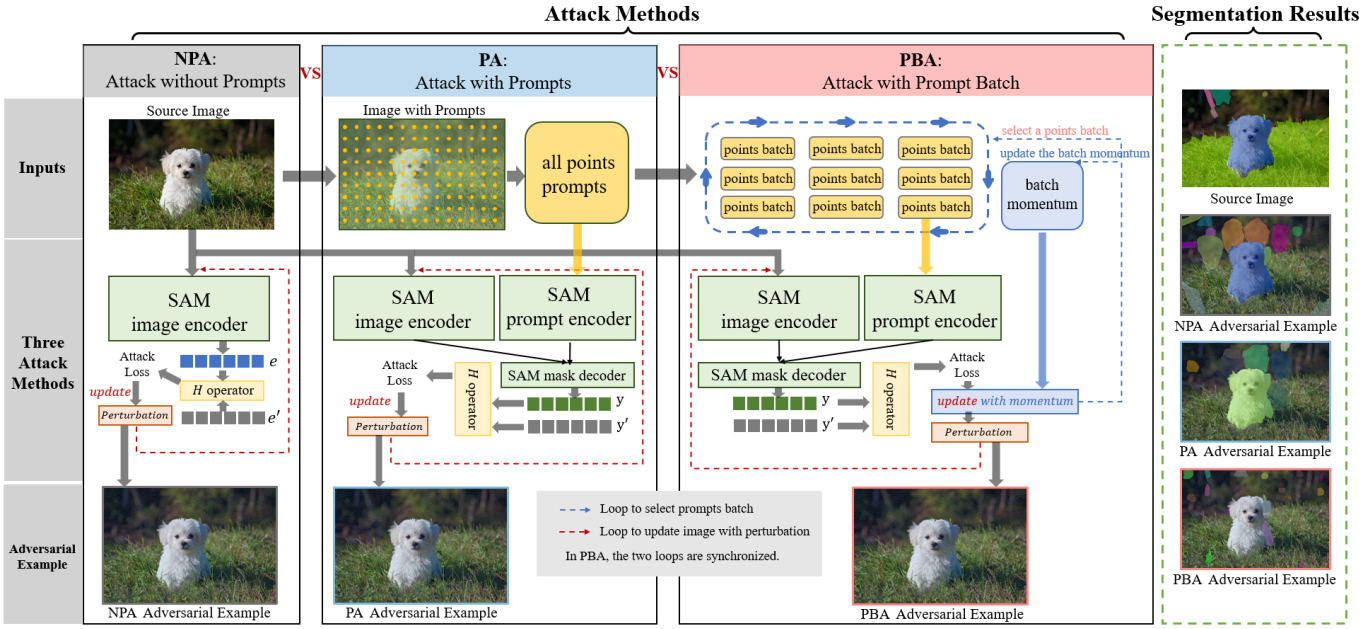

Figure 2. The three different algorithms (NPA, PA, PBA) aim to generate adversarial examples for attacking the SAM in a white-box setting.

Table 2. Algorithmic symbols table.

Signal	Explanation	Used by
X	input image	NPA, PA, PBA
X'	arbitrary real image	NPA, PA, PBA
K	maximum iterations	NPA, PA, PBA
g_{im}	the image encoder of SAM	NPA, PA, PBA
g_{pt}	the prompt encoder of SAM	PA, PBA
h_{dec}	the mask decoder of SAM	PA, PBA
N	the number of prompt batch	PBA
p	prompt	PA
$P = \{p_1^*, p_2^*, ..., p_N^*\}$	batches of prompt	PBA
α	attack rate	NPA, PA, PBA
β	momentum decay factor	NPA, PA, PBA
μ	maximum perturbation value	NPA, PA, PBA
X_{adv}	the adversarial example	NPA, PA, PBA

4.1. NPA: Attack without Prompts

In the no prompt attack (NPA) method, we only use the output of g_{im} to calculate $\mathcal{L}$ as in Equation (9), without considering g_{pt}, h_{dec} and the prompt input, p. The overall pipeline of the NPA algorithm is illustrated in Algorithm 1.

$$\mathcal{L}_{NPA}(x^{adv}, e') = \|g_{im}(x^{adv}) - e'\|_2^2 \tag{9}$$

where x^{adv} denotes the image with adversarial perturbation, and e' denotes a constant in the image embedding space, which can be generated from an arbitrary real image.

Algorithm 1: NPA

Data: input image $\mathbf{X}$, arbitrary real image $\mathbf{X}'$, maximum iterations K, the image encoder of SAM g_{im}, the attack rate α, the momentum decay factor β, the maximum perturbation value μ
Result: the adversarial example $\mathbf{X}^{adv}$
$\mathbf{X}_0 \leftarrow \mathbf{X}$;
$l \leftarrow 0$;
$m_0 \leftarrow 0$;
$e' \leftarrow g_{im}(X')$;
for $i = 0$ *to* K **do**
  $\mathbf{g} \leftarrow \nabla_{\mathbf{X}_l} \mathcal{L}_{NPA}(\mathbf{X}_l, e')$;
  $\mathbf{m}_{l+1} \leftarrow \beta \cdot \mathbf{m}_l + \frac{\mathbf{g}}{\|\mathbf{g}\|}$;
  $\mathbf{X}_{l+1} \leftarrow \mathbf{X}_l + \alpha \cdot sgn(m_{l+1})$;
  $\mathbf{X}_{l+1} \leftarrow \max(\min(\mathbf{X}_{l+1}, \mathbf{X} + \mu), \mathbf{X} - \mu)$;
  $l \leftarrow l + 1$;
end
$\mathbf{X}_{adv} \leftarrow \mathbf{X}_l$;
return $\mathbf{X}_{adv}$

4.2. PA: Attack with Prompts

As a further step, the PA method not only considers adversarial attacks against the image encoder but also takes into account the influence of the mask decoder on the final results. Therefore, the PA method defines $\mathcal{L}$ using the output of h_{dec} as Equation (10). The overall pipeline is illustrated in Algorithm 2.

$$\mathcal{L}_{PA}(x^{adv}, P, y') = \|h_{dec}(g_{im}(x^{adv}), g_{pt}(p)) - y'\|_2^2 \tag{10}$$

where x^{adv} denotes the image with adversarial perturbation, p denotes a special prompt with dense point coordinates of the input image, and y' denotes a constant in the space of output of h_{dec}, which can be generated from an arbitrary real image.

Algorithm 2: PA

Data: input image $\mathbf{X}$, arbitrary real image $\mathbf{X}'$, prompt p, maximum iteration K, the image encoder of SAM g_{im}, the prompt encoder of SAM g_{pt}, the mask decoder of SAM h_{dec}, attack rate α, momentum decay factor β, maximum perturbation value μ
Result: the adversarial example $\mathbf{X}^{adv}$
$\mathbf{X}_0 \leftarrow \mathbf{X}$;
$l \leftarrow 0$;
$m_0 \leftarrow 0$;
$e' \leftarrow g_{im}(X')$;
$y' \leftarrow h_{dec}(e', p)$;
for $i = 0$ *to* K **do**
  $\mathbf{g} \leftarrow \nabla_{\mathbf{X}_l} \mathcal{L}_{PA}(\mathbf{X}_l, p, y')$;
  $\mathbf{m}_{l+1} \leftarrow \beta \cdot \mathbf{m}_l + \frac{\mathbf{g}}{\|\mathbf{g}\|}$;
  $\mathbf{X}_{l+1} \leftarrow \mathbf{X}_l + \alpha \cdot sgn(m_{l+1})$;
  $\mathbf{X}_{l+1} \leftarrow \max(\min(\mathbf{X}_{l+1}, \mathbf{X} + \mu), \mathbf{X} - \mu)$;
  $l \leftarrow l + 1$;
end
$\mathbf{X}_{adv} \leftarrow \mathbf{X}_l$;
return $\mathbf{X}_{adv}$

4.3. PBA: Attack with Prompt Batch

In order to improve the transferability of adversarial examples across different prompts, we promote the PA method by (1) dividing the special input prompts with dense point coordinates into small batches during the iterations of the attack, (2) using only one batch of the prompt to calculate $\mathcal{L}$ as Equation (11). The overall pipeline is illustrated in Algorithm 3.

$$\mathcal{L}_{PBA}(x^{adv}, p^*, y') = \|h_{dec}(g_{im}(x^{adv}), g_{pt}(p^*)) - y'\|_2^2 \tag{11}$$

where x^{adv} denotes the image with adversarial perturbation, p^* denotes one batch of p, and y' denotes a constant in the space of output of h_{dec}, which can be generated from an arbitrary real image.

Algorithm 3: PBA

Data: input image $\mathbf{X}$, arbitrary real image $\mathbf{X}'$, batches of prompt $P = \{p_1^*, p_2^*, ..., p_N^*\}$, maximum iteration K, the image encoder of SAM g_{im}, the prompt encoder of SAM g_{pt}, the mask decoder of SAM h_{dec}, attack rate α, momentum decay factor β, maximum perturbation value μ

Result: the adversarial example $\mathbf{X}^{adv}$

$\mathbf{X}_0 \leftarrow \mathbf{X}$;
$l \leftarrow 0$;
$m_0 \leftarrow 0$;
$e' \leftarrow g_{im}(X')$;
$y'_1, y'_2, ..., y'_N \leftarrow h_{dec}(e', g_{pt}(p_1^*)), h_{dec}(e', g_{pt}(p_2^*)), ..., h_{dec}(e', g_{pt}(p_N^*))$;
for $i = 0$ *to* K **do**
  for $j = 1$ *to* N **do**
    $\mathbf{g} \leftarrow \nabla_{\mathbf{X}_l} \mathcal{L}_{BPAM}(\mathbf{X}_l, p_j^*, y'_j)$;
    $\mathbf{m}_{l+1} \leftarrow \beta \cdot \mathbf{m}_l + \frac{\mathbf{g}}{\|\mathbf{g}\|}$;
    $\mathbf{X}_{l+1} \leftarrow \mathbf{X}_l + \alpha \cdot sgn(m_{l+1})$;
    $\mathbf{X}_{l+1} \leftarrow \max(\min(\mathbf{X}_{l+1}, \mathbf{X} + \mu), \mathbf{X} - \mu)$;
    $l \leftarrow l + 1$;
  end
end
$\mathbf{X}_{adv} \leftarrow \mathbf{X}_l$;
return $\mathbf{X}_{adv}$

4.4. Algorithms Analysis

In this section, we present a comparative analysis of NPA, PA, SPA, and PBA algorithms, and, we attempt to explain the superiority of the PBA algorithm.

As shown in Table 3, the differences between the algorithms are mainly reflected in five aspects, which are the use of the SAM structure, the use of prompt information, the loss function, the number of iterations, and the process affecting the momentum.

Table 3. Key differences between the algorithms.

Algorithm	Used SAM Structure	Used Prompt	Loss Function [2]	Number of Iteration [3]	Momentum Affected by
NPA	g_{im}	No	L_{NPA}	K	No prompt
PA	g_{im}, g_{pt}, h_{dec}	p	L_{PA}	K	$p, p, p, ..., p, p, p$
SPA [1]	g_{im}, g_{pt}, h_{dec}	p_1^*, p_2^*, p_3^*	L_{PA}	3*K	$p_1^*, p_1^*, p_1^*, p_2^*, p_2^*, p_2^*, p_3^*, p_3^*, p_3^*$
PBA	g_{im}, g_{pt}, h_{dec}	$P = \{p_1^*, p_2^*, p_3^*\}$	L_{PBA}	K*3	$p_1^*, p_2^*, p_3^*, p_1^*, p_2^*, p_3^*, p_1^*, p_2^*, p_3^*$

[1] In order to exclude the effect of the number of iterations and prompt selection on the results, SPA sequentially completes the attack using different prompt batches, which is equivalent to multiple PA attacks on the same image. [2] The definitions of loss functions can be found in Equations (9)–(11). [3] In this table, the number of prompt batches, N, is 3.

Among these aspects, the most essential is that different algorithms affect the momentum in different processes. In Figure 3, we show the intuitive impact of the different processes affecting the momentum. In gradient-based adversarial perturbation generation processes, momentum serves as a critical factor influencing the direction of perturbation generation. It can be succinctly understood that the introduction of varying information during the iterative process affects the direction of momentum.

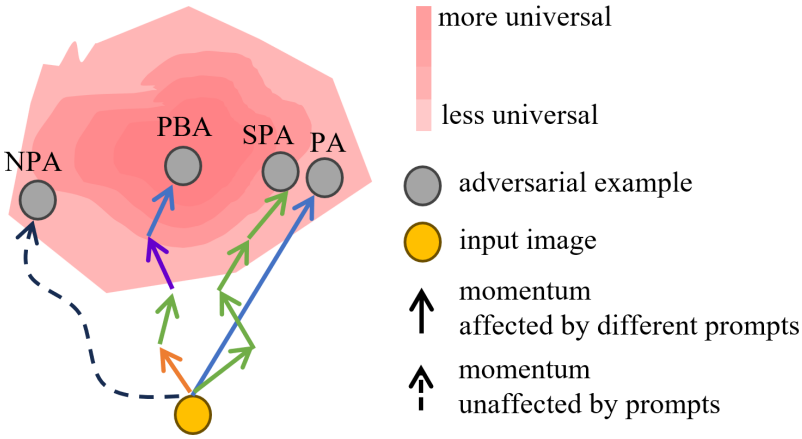

Figure 3. Illustration of the comparison of the algorithms.

In the NPA algorithm, the direction of momentum is not influenced by any prompt information, thereby making it difficult for the generated adversarial examples to reach more universal prompt-dense regions (adversarial examples capable of successful attacks under a wider range of prompt conditions). In the PA algorithm, the direction of momentum is solely influenced by a single prompt, thereby significantly impacting the universality of the adversarial examples due to potential bias inherent in the specific prompt itself. In the PBA algorithm, the direction of momentum is continuously influenced by the combined effects of different prompt batch information, allowing the synthesis of multiple prompt information to correct the direction of momentum in a shorter period. Therefore, the momentum direction of PBA consistently approaches the correct direction. Conversely, in the SPA algorithm, despite having the same number of iterations and identical prompt information as PBA, it fails to promptly synthesize multiple prompt batch information to adjust the momentum direction. Consequently, the momentum direction of SPA remains unstable.

In conclusion, adversarial examples generated by the PBA method consistently converge toward the correct direction, demonstrating superior generalization across different prompt conditions. This highlights PBA cross-prompt transferability.

5. Experiments

In this section, we conduct a quantitative evaluation of the proposed PBA adversarial example generation algorithm and compare its attack capability and cross-prompt transferability with those of the NPA and PA algorithms. Furthermore, in order to demonstrate the image quality of the adversarial examples, we conduct a quantitative evaluation of the quality of the generated adversarial examples by comparing their image similarities with a clean image. Additionally, we used two NVIDIA V100 PCIe GPUs to complete all experiments.

5.1. Experimental Settings

5.1.1. Task

We use points as prompts to drive SAM for the segmentation task on the entire image. Specifically, we set different numbers of points as prompts, including 8×8, 16×16, and 20×20, which are, respectively, named **Pts8**, **Pts16**, and **Pts20**. By selecting different prompt inputs and attack methods, we conduct three sets of experiments, namely **PBA**

(Pts8, prompt batch size 16), **PA** (Pts8, with only one batch of prompt), and **NPA** (Pts8, without prompt). Additionally, to demonstrate the effectiveness of the PBA method more clearly, we add an experimental group called **SPA**(sequential prompt attack). **SPA** sequentially completes the attack using different prompt batches, which is equivalent to multiple PA attacks on the same image. In the evaluation phase, we attack the SAM under different prompt settings (**Pts8**, **Pts16**, and **Pts20**) by adversarial examples generated from the four methods (**PBA**, **PA**, **NPA**, and **SPA**).

5.1.2. Hyperparameters

The hyperparameters used in SAM are listed in Table 4. These settings follow the default settings of SAM. For the adversarial example generation task, otherwise specified, we set the hyperparameters to 10 attack iterations, a 0.01 attack rate, a 0.2 maximum perturbation amplitude, and 0.9 momentum decay factor.

Table 4. Hyperparameters in SAM

Hyperparameter Name	Pred_Iou_Thresh	Stability_Score_Thresh	Stability_Score_Offset	Box_Nms_Thresh	Crop_N_Layers
Hyperparameter value	0.88	0.85	1.0	0.7	0

5.1.3. Datasets

We conducted experiments on a diverse set of datasets, including the CBCL Street Scenes Dataset [51], CrowdHuman Dataset [52], and the UNIMIB2016 Food Dataset [53]. To facilitate the experiments without sacrificing accuracy, we selected the first 300, 200, and 200 images from the validation sets of the three datasets, respectively, as our experimental data. We conducted experiments on these data, including attack capability, cross-prompt transferability, and an evaluation of the image quality of adversarial examples. For ease of reference, we will refer to these three datasets as **CBCL300**, **CH200**, and **UNIMIB200**.

5.1.4. Metrics

The quantitative metrics used in the experiments include the mean intersection-over-union (mIoU), attack success rate (ASR), cross-prompt attack success rate (ASR*), and structural similarity index measure (SSIM). As SAM performs well in the segmentation task, we use the SAM segmentation results as the ground truth and calculate the mIoU value of the segmentation results of adversarial examples, defined by Equation (5). Following previous work [20,21,24], the ASR metric is defined by the drop in mIoU as Equation (4). According to Section 3.2, the ASR* metric is defined as Equation (7). The SSIM [54] is a similarity metric that measures the quality of an image based on the similarity to a reference image and is in line with human perception. In these experiments, we use SSIM to measure the degree of distortion in the adversarial examples generated by different algorithms.

5.2. Attack Results

We conducted experiments on three datasets, as mentioned earlier. For the four experimental groups under the same model hyperparameters settings, the mIoU and ASR results in different prompt settings can be observed in Table 5. Additionally, we randomly added noise to images of the three datasets and calculated their mIoU by SAM under the Pts8 setting. Table 5 indicates that the PBA method exhibits the lowest mIoU, the highest ASR value, and the highest ASR* value across the three prompt settings on the three datasets. These results suggest that the PBA method demonstrates the best adversarial attack ability and cross-prompt transferability compared to the other three methods.

In Figure 4, we present several instances of adversarial examples along with their SAM segmentation mask outputs, as well as the original images with their SAM mask outputs. Figure 4 shows that the mask outputs of adversarial examples are fragmented into shards. Furthermore, it indicates that the adversarial examples generated by the PBA method exhibit better cross-prompt transferability performance, suggesting that

their attack capability remains stable under different prompt settings. Specifically, the differences in the attack results in columns 2 to 4 are relatively small, all displaying high cross-prompt transferability. However, the effectiveness of attacks in columns 5 to 7 gradually deteriorates, indicating lower cross-prompt transferability.

Table 5. The mIoU and ASR results of different adversarial attack methods on different datasets.

Metrics	CBCL300				CH200				UNIMIB200			
	PBA	**SPA**	**PA**	**NPA**	**PBA**	**SPA**	**PA**	**NPA**	**PBA**	**SPA**	**PA**	**NPA**
mIoU Pts8	0.3419	0.5774	0.5145	0.6935	0.4768	0.6438	0.5988	0.7359	0.3153	0.5218	0.4645	0.6721
mIoU Pts16	0.4739	0.6029	0.6122	0.6881	0.5761	0.6573	0.6718	0.7193	0.4197	0.5486	0.5358	0.6500
mIoU Pts20	0.4636	0.5999	0.6101	0.6860	0.5630	0.6526	0.6624	0.7216	0.4007	0.5408	0.5231	0.6460
mIoU Noisy	0.8226				0.8263				0.8347			
ASR Pts8	0.5844	0.2981	0.3745	0.1569	0.4230	0.2209	0.2753	0.1094	0.6223	0.3749	0.4435	0.1948
ASR Pts16	0.4239	0.2671	0.2558	0.1635	0.3028	0.2045	0.1870	0.1295	0.4972	0.3428	0.3581	0.2213
ASR Pts20	0.4364	0.2707	0.2583	0.1661	0.3186	0.2102	0.1984	0.1267	0.5199	0.3521	0.3733	0.2261
ASR*	0.4317	0.2693	0.2574	0.1651	0.3126	0.2081	0.1940	0.1278	0.5113	0.3486	0.3675	0.2243

Figure 4. Adversarial examples with segmentation mask results of SAM. The text above illustrates the source of the images below it. The text below the images indicates the prompt setting (either Pts8, Pts16, or Pts20) used to generate the segmentation results on those images.

6. Discussion

6.1. Attack Iteration

In this section, we conducted experiments to demonstrate the impact of different attack iterations on the mean intersection-over-union (mIoU), attack success rate (ASR), and structural similarity index measure (SSIM). Specifically, to ensure consistency in the frequency of adding the perturbation by different methods, the number of attack iterations for all methods, except PBA, is set to four times the corresponding number shown in Figure 5.

Through Figure 5, we can observe the following: (1) The advantage of PBA over other attack methods becomes more pronounced as the number of attack iterations increases. (2) The PA method outperforms the PBA method when the number of attack iterations is small, which may be due to the fact that the PBA method utilizes only partial information from the prompt at the initial stage. (3) The perturbation sizes of the adversarial examples generated by the three methods are essentially the same, as evidenced by the consistency of SSIM values.

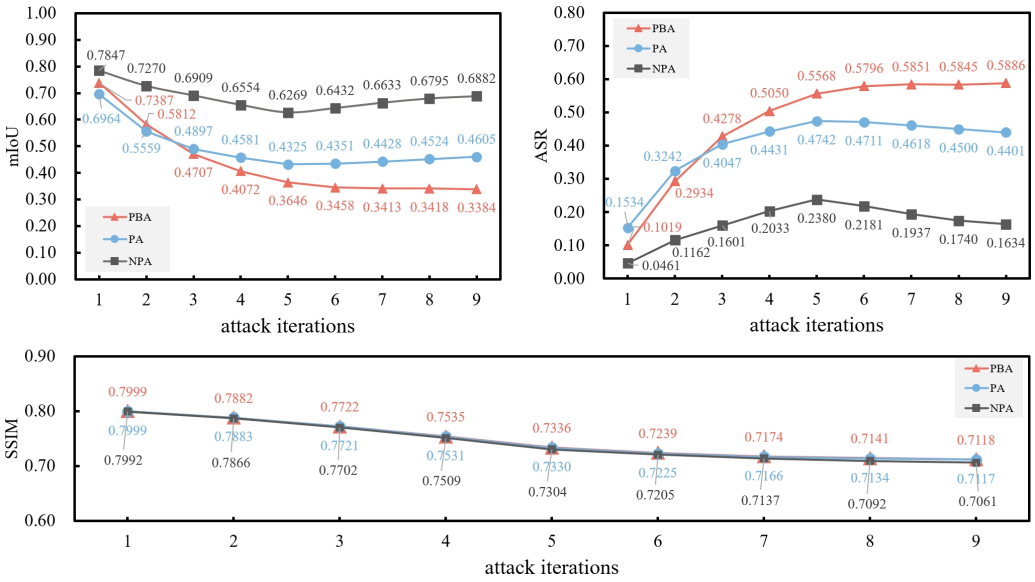

Figure 5. Comparative analysis of mIoU, ASR, and SSIM across attack iterations and methods. To ensure consistency in the actual number of image updates, for both PA and NPA methods, the number of attack iterations is four times the number shown in the horizontal coordinate in this figure.

6.2. Image Quality

In this section, we will discuss the relationship between image quality and the attack success rate of adversarial examples generated by the PBA method and PA method.

We varied the hyperparameter (the maximum perturbation value μ) during the generation of adversarial examples to obtain adversarial examples with different image qualities and attack effectiveness. We evaluated the image quality by calculating their similarity to the original images. In these experiments, we used the structural similarity index measure (SSIM) metric to quantify the image quality of adversarial examples while assessing their attack effectiveness through the attack success rate (ASR) metric. This experiment was conducted on a dataset of 300 images, sourced from the validation set of the CBCL Street Scenes Dataset [51].

The experimental results are presented in Table 6 and Figure 6. From Table 6, it can be observed that in most cases, the PBA method exhibits higher SSIM values and lower mIoU values compared to the PA method, especially when the perturbation is smaller ($\mu < 0.2$). In a more intuitive manner, from the relationship graph depicted in Figure 6, it can be observed that under similar image quality conditions (equal SSIM), the PBA method demonstrates higher attack effectiveness (lower mIoU). Conversely, under incomparable

attack effectiveness conditions (equal mIoU), the PBA method exhibits superior image quality (higher SSIM).

To summarize, the experimental results from this section demonstrate that the PBA method achieves both effective adversarial attacks and high-quality image generation simultaneously.

Table 6. Comparison results on SSIM and mIoU metrics between the adversarial examples generated by the PBA and PA methods. The μ hyperparameter represents the maximum perturbation value that can be added during the adversarial example generation process.

μ		μ0.05	μ0.1	μ0.15	μ0.2	μ0.25	μ0.3	μ0.35
PBA	SSIM	0.8606	0.8265	0.7818	0.7358	0.6981	0.6723	0.6583
	mIoU	0.5854	0.3119	0.2098	0.1761	0.155	0.1503	0.1444
PA	SSIM	0.8604	0.8258	0.7804	0.7351	0.6990	0.6746	0.6614
	mIoU	0.6143	0.3936	0.2917	0.2405	0.2187	0.2022	0.1972

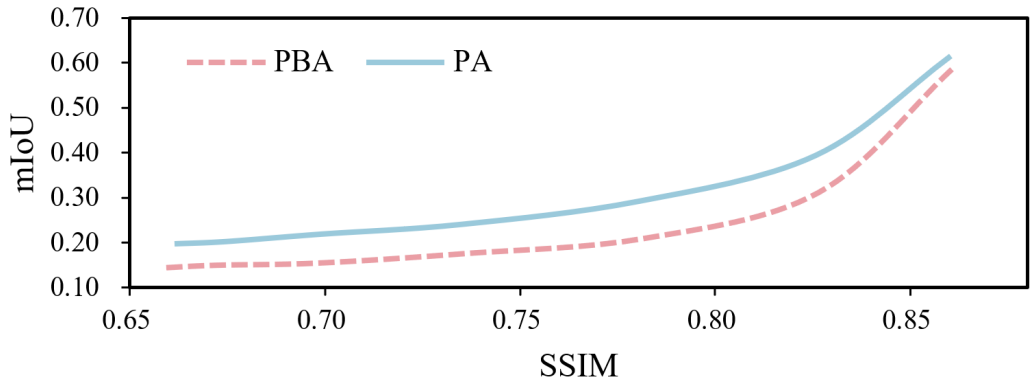

Figure 6. Illustration of the relationship between mIoU and SSIM under the PBA and PA methods. The dataset utilized in this experiment is CBCL300, with the range of variation for the maximum perturbation value, μ, ranging from 0.05 to 0.35.

7. Conclusions

In this paper, we propose a method (the PBA method) to attack the significant visual foundation model (SAM), indicating that SAM has room for improvement in adversarial robustness. The experimental results demonstrate that the PBA method can successfully generate adversarial examples that perform well in both cross-prompt transferability and attack success rates. Numerical results on multiple datasets show that the cross-prompt attack success rate (ASR*) of the PBA method is 17.83% higher on average, and the attack success rate (ASR) is 20.84% higher. Generating adversarial examples with prompt batching can effectively promote the cross-prompt transferability of adversarial examples. Additionally, we find that enhancing the cross-prompt transferability of adversarial examples is crucial for attacking visual foundation models equipped with the prompt mechanism.

Additionally, we believe that using adversarial examples generated by the PBA method could be risky for real-world systems based on SAM. On the one hand, the PBA method exhibits strong cross-prompt transferability. On the other hand, SAM is a widely used visual foundation model. Therefore, we recommend adopting adversarial training when using SAM. This involves incorporating adversarial examples into SAM's training dataset to enhance its security.

In future research, we will extensively focus on adversarial attacks and defenses targeting various types of foundation models, and explore defense methods against adversarial attacks in different application scenarios.

Author Contributions: Methodology, Q.F.; Software, Q.F.; Validation, Z.Z. and X.L.; Data curation, Z.Z. and J.Q.; Writing—original draft, Q.F.; Writing—review & editing, S.H., X.L. and G.S.; Visualization, G.S. and J.Q.; Funding acquisition, S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of Chongqing, China: CSTB2022 NSCQ-MSX1454.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: CBCL300, CH200, UNIMIB200. The code of this paper can be found in SegmentShards.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the Opportunities and Risks of Foundation Models. *arXiv* **2022**, arXiv:2108.07258.
2. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
3. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 6–12 December 2020.
4. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021.
5. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
6. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. *arXiv* **2023**, arXiv:2304.02643.
7. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2805–2824. [CrossRef] [PubMed]
8. Ren, K.; Zheng, T.; Qin, Z.; Liu, X. Adversarial Attacks and Defenses in Deep Learning. *Engineering* **2020**, *6*, 346–360. [CrossRef]
9. Biggio, B.; Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 2154–2156.
10. Zhang, W.E.; Sheng, Q.Z.; Alhazmi, A.; Li, C. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.* **2020**, *11*, 1–41. [CrossRef]
11. Luo, Z.; Li, Q.; Zheng, J. A Study of Adversarial Attacks and Detection on Deep Learning-Based Plant Disease Identification. *Appl. Sci.* **2021**, *11*, 1878. [CrossRef]
12. Paul, S.; Chen, P.Y. Vision Transformers Are Robust Learners. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 2071–2081. [CrossRef]
13. Huang, Y.; Cao, Y.; Li, T.; Juefei-Xu, F.; Lin, D.; Tsang, I.W.; Liu, Y.; Guo, Q. On the Robustness of Segment Anything. *arXiv* **2023**, arXiv:2305.16220.
14. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arXiv:1412.6572.
15. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
16. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2017**, arXiv:1607.02533.
17. Gu, J.; Zhao, H.; Tresp, V.; Torr, P.H.S. SegPGD: An Effective and Efficient Adversarial Attack for Evaluating and Boosting Segmentation Robustness. In Proceedings of the Computer Vision—ECCV 2022, Tel Aviv, Israel, 23–27 October 2022; pp. 308–325.
18. Zhang, C.; Zhang, C.; Kang, T.; Kim, D.; Bae, S.H.; Kweon, I.S. Attack-SAM: Towards Attacking Segment Anything Model With Adversarial Examples. *arXiv* **2023**, arXiv:2305.00866.
19. Zheng, S.; Zhang, C. Black-box Targeted Adversarial Attack on Segment Anything (SAM). *arXiv* **2023**, arXiv:2310.10010.
20. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
21. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582. [CrossRef]
22. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 39–57. [CrossRef]

23. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. In Proceedings of the 2016 IEEE European symposium on security and privacy (EuroS&P), Saarbrucken, Germany, 21–24 March 2016; pp. 372–387.
24. Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; Yuille, A. Adversarial Examples for Semantic Segmentation and Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1378–1387. [CrossRef]
25. Xiao, Y.; Pun, C.M.; Liu, B. Fooling deep neural detection networks with adaptive object-oriented adversarial perturbation. *Pattern Recognit.* **2021**, *115*, 107903. [CrossRef]
26. Zhang, Z.; Huang, S.; Liu, X.; Zhang, B.; Dong, D. Adversarial Attacks on YOLACT Instance Segmentation. *Comput. Secur.* **2022**, *116*, 102682. [CrossRef]
27. Lee, W.; Kim, Y. Enhancing CT Segmentation Security against Adversarial Attack: Most Activated Filter Approach. *Appl. Sci.* **2024**, *14*, 2130. [CrossRef]
28. Zhang, Z.; Huang, S.; Liu, X.; Fan, Q.; Dong, D. Adversarial attack on human pose estimation network. *J. Electron. Imaging* **2024**, *33*, 013052. [CrossRef]
29. Ding, W.; Wei, X.; Ji, R.; Hong, X.; Tian, Q.; Gong, Y. Beyond universal person re-identification attack. *IEEE Trans. Inform. Forensics Secur.* **2021**, *16*, 3442–3455. [CrossRef]
30. Zhanhao, H.; Huang, S.; Zhu, X.; Sun, F.; Zhang, B.; Hu, X. Adversarial Texture for Fooling Person Detectors in the Physical World. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13297–13306. [CrossRef]
31. Chen, H.; Zhang, H.; Chen, P.Y.; Yi, J.; Hsieh, C.J. Attacking Visual Language Grounding with Adversarial Examples: A Case Study on Neural Image Captioning. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 2587–2597. [CrossRef]
32. Wu, H.; Liu, Y.; Cai, H.; He, S. Learning Transferable Perturbations for Image Captioning. *ACM Trans. Multimed. Comput. Commun. Appl.* **2022**, *18*, 1–18. [CrossRef]
33. Zhang, Y.; Chen, J.; Liu, L.; Chen, K.; Shi, Z.; Zou, Z. Generating Imperceptible and Cross-Resolution Remote Sensing Adversarial Examples Based on Implicit Neural Representations. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–15. [CrossRef]
34. Chen, Z.; Chen, F.; Sun, Y.; Wang, M.; Liu, S.; Ji, Y. Local aggressive and physically realizable adversarial attacks on 3D point cloud. *Comput. Secur.* **2024**, *139*, 103539. [CrossRef]
35. Zhang, J.; Chen, L.; Liu, B.; Ouyang, B.; Xie, Q.; Zhu, J.; Li, W.; Meng, Y. 3D adversarial attacks beyond point cloud. *Inform. Sci.* **2023**, *633*, 491–503. [CrossRef]
36. Zhou, W.; Hou, X.; Chen, Y.; Tang, M.; Huang, X.; Gan, X.; Yang, Y. Transferable adversarial perturbations. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 452–467.
37. Wu, L.; Zhu, Z.; Tai, C. Understanding and enhancing the transferability of adversarial examples. *arXiv* **2018**, arXiv:1802.09707.
38. Wang, X.; He, K. Enhancing the transferability of adversarial attacks through variance tuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1924–1933.
39. Dong, Y.; Pang, T.; Su, H.; Zhu, J. Evading defenses to transferable adversarial examples by translation-invariant attacks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4312–4321.
40. Lin, J.; Song, C.; He, K.; Wang, L.; Hopcroft, J.E. Nesterov accelerated gradient and scale invariance for adversarial attacks. *arXiv* **2019**, arXiv:1908.06281.
41. Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; Yuille, A.L. Improving transferability of adversarial examples with input diversity. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2730–2739.
42. Wu, W.; Su, Y.; Lyu, M.R.; King, I. Improving the transferability of adversarial samples with adversarial transformations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9024–9033.
43. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G.E. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv* 2020, arXiv:2002.05709.
44. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9726–9735. [CrossRef]
45. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked Autoencoders Are Scalable Vision Learners. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LO, USA, 19–24 June 2022; pp. 15979–15988. [CrossRef]
46. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv* **2022**, arXiv:2204.06125.
47. Shafahi, A.; Saadatpanah, P.; Zhu, C.; Ghiasi, A.; Studer, C.; Jacobs, D.; Goldstein, T. Adversarially robust transfer learning. In Proceedings of the International Conference on Learning Representations, Virtual, 26 April–1 May 2020.

48. Schmidt, L.; Santurkar, S.; Tsipras, D.; Talwar, K.; Madry, A. Adversarially Robust Generalization Requires More Data. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31.
49. Bubeck, S.; Sellke, M. A Universal Law of Robustness via Isoperimetry. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021; Volume 34, pp. 28811–28822.
50. Fort, S. Adversarial Examples for the OpenAI CLIP in Its Zero-Shot Classification Regime and Their Semantic Generalization, 2021. Available online: https://stanislavfort.com/blog/OpenAI_CLIP_adversarial_examples/ (accessed on 12 January 2021).
51. DARPA. CBCL StreetScenes Database Download Page. 2007. Available online: http://cbcl.mit.edu/software-datasets/streetscenes/ (accessed on 27 March 2007).
52. Shao, S.; Zhao, Z.; Li, B.; Xiao, T.; Yu, G.; Zhang, X.; Sun, J. CrowdHuman: A Benchmark for Detecting Human in a Crowd. *arXiv* **2018**, arXiv:1805.00123.
53. Ciocca, G.; Napoletano, P.; Schettini, R. Food recognition: A new dataset, experiments and results. *IEEE J. Biomed. Health Inform.* **2017**, *21*, 588–598. [CrossRef]
54. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]