SegPGD: An Effective and Efficient Adversarial Attack for Evaluating and Boosting Segmentation Robustness

Jindong Gu[1,3], Hengshuang Zhao[2,3], Volker Tresp[1], and Philip Torr[3]

[1] University of Munich [2] The University of Hong Kong
[3] Torr Vision Group, University of Oxford

Abstract. Deep neural network-based image classifications are vulnerable to adversarial perturbations. The image classifications can be easily fooled by adding artificial small and imperceptible perturbations to input images. As one of the most effective defense strategies, adversarial training was proposed to address the vulnerability of classification models, where the adversarial examples are created and injected into training data during training. The attack and defense of classification models have been intensively studied in past years. Semantic segmentation, as an extension of classifications, has also received great attention recently. Recent work shows a large number of attack iterations are required to create effective adversarial examples to fool segmentation models. The observation makes both robustness evaluation and adversarial training on segmentation models challenging. In this work, we propose an effective and efficient segmentation attack method, dubbed SegPGD. Besides, we provide a convergence analysis to show the proposed SegPGD can create more effective adversarial examples than PGD under the same number of attack iterations. Furthermore, we propose to apply our SegPGD as the underlying attack method for segmentation adversarial training. Since SegPGD can create more effective adversarial examples, the adversarial training with our SegPGD can boost the robustness of segmentation models. Our proposals are also verified with experiments on popular Segmentation model architectures and standard segmentation datasets.

Keywords: Adversarial Robustness, Semantic Segmentation

1 Introduction

Due to their vulnerability to artificial small perturbations, the adversarial robustness of deep neural networks has received great attention [40,13]. A large amount of attack and defense strategies have been proposed for classification in past years [5,37,46,47,54,57,43,1,52,14,39,34]. As an extension of classification, semantic segmentation also suffers from adversarial examples [50,2]. Segmentation models applied in real-world safety-critical applications also face potential threats, *e.g.,* in self-driving systems [32,25,19,33,4,36] and in medical image analysis [10,35,12,30]. Hence, the adversarial robustness of segmentation has also raised great attention recently [50,2,51,49,20,42,27,24,49,8,38,53].

In terms of the attack methods, different from classification, the attack goal in segmentation is to fool all pixel classifications at the same time. An effective adversarial example of a segmentation model are expected to fool as many pixel classifications as possible, which requires the larger number of attack iterations [50,15]. The observation makes both robustness evaluation and adversarial training on segmentation models challenging. In this work, we propose an effective and efficient segmentation attack method, dubbed SegPGD. Besides, we provide a convergence analysis to show why the proposed SegPGD can create more effective adversarial examples than PGD under the same number of attack iterations.

The right evaluation of model robustness is an important step to building robust models. Evaluation with weak or inappropriate attack methods can give a false sense of robustness [3]. Recent work [51] evaluates the robustness of segmentation models under a similar setting to the one used in classification. This could be problematic given the fact that a large number of attack iterations are required to create effective adversarial examples of segmentation [50]. We evaluate the adversarially trained segmentation models in previous work with a strong attack setting, namely with a large number of attack iterations. We found the robustness can be significantly reduced. Our SegPGD can reduce the mIoU score further. For example, the mIoU of adversarially trained PSPNet [56] on Cityscapes dataset [9] can be reduced to near zero under 100 attack iterations.

As one of the most effective defense strategies, adversarial training was proposed to address the vulnerability of classification models, where the adversarial examples are created and injected into training data during training [13,29]. One promising way to boost segmentation robustness is to apply adversarial training to segmentation models. However, the creation of effective segmentation adversarial examples during training can be time-consuming. In this work, we demonstrate that our effective and efficient SegPGD can mitigate this challenge. Since it can create effective adversarial examples, the application of SegPGD as the underlying attack method of adversarial training can effectively boost the robustness of segmentation models. It is worth noting that many adversarial training strategies with single-step attacks have been proposed to address the efficiency of adversarial training in classification [37,47,57,43,1]. However, they do not work well on segmentation models since the adversarial examples created by single-step attacks are not effective enough to fool segmentation models.
The contributions of our work can be summarised as follows:

- Based on the difference between classification and segmentation, we propose an effective and efficient segmentation attack method, dubbed SegPGD. Especially, we show its generalization to single-step attack SegFGSM.
- We provide a convergence analysis to show the proposed SegPGD can create more effective adversarial examples than PGD under the same number of attack iterations.
- We apply SegPGD as the underlying attack method for segmentation adversarial training. The adversarial training with our SegPGD achieves state-of-the-art performance on the benchmark.

– We conduct experiments with popular segmentation model structures (*i.e.,* PSPNet and DeepLabV3) on standard segmentation datasets (*i.e.,* PASCAL VOC and Cityscapes) to demonstrate the effectiveness of our proposals.

2 Related Work

Adversarial Robustness of Segmentation Models. The work [2] makes an extensive study on the adversarial robustness of segmentation models and demonstrates the inherent robustness of standard segmentation models. Especially, they find that adversarial examples in segmentation do not transfer well across different scales and transformations. Another work [50] also found that the adversarial examples created by their attack method do not transfer well across different network structures. The observations in the two works [2,50] indicate the standard segmentation models are inherently robust to transfer-based black-box method. The belief is broken by the work [15] where they propose a method to improve the transferability of adversarial examples and show the feasibility of transfer-based black-box method. In addition, the adversarial robustness of segmentation models has also been studied from other perspectives, such as universal adversarial perturbation [20,23], adversarial example detection [49], and backdoor attack [28]. Theses works also imply the necessity of building robust segmentation models to defend against potential threats. Along this direction, the work [25] shows self-supervised learning with more data can improve the robustness of standard models. However, the obtained model can be easily completely fooled with a strong attack [25]. A recent work [51] makes the first exploration to apply adversarial training to segmentation models. We find that the adversarially trained models is still vulnerable under strong attacks. The robust accuracy of their adversarial trained models can be significantly reduced under PGD with a large number of attack iterations. In this work, we propose an effective and efficient segmentation attack method, which be used in adversarial training to build robust segmentation models against strong attacks.

Adversarial Training of Classification Models. Adversarial training has been intensively studied on classification models [13,29]. When a multi-step attack is applied to create adversarial examples for adversarial training, the obtained model is indeed robust against various attack to some extent, as shown in [29,46,5,52]. However, adversarial training with multi-step attack can be very time consuming due to the adversarial example creation, which is N times longer than standard natural training [29,37]. To accelerate the adversarial training, single-step attack has also been explored therein. When standard single-step attack is applied during training, the obtained model is only robust to single-step attack [41]. One reason behind is that the gradient masking phenomenon of the model can be observed on the adversarial examples created by single-step attack. Besides, another challenge to apply single-step attack in adversarial training is the label leaking problem where the model show higher robust accuracy against single-step attack than clean accuracy [26]. The low defensive effectiveness of single-step attack and the low efficiency of multi-step attack pose a dilemma.

One way to address the dilemma is to overcome the challenges using advanced single-step attacks [41,44,46,55,47,21,22], which can address label leaking problem and avoid gradient masking phenomenon. Though it boosts the robustness of the classification models, however, single-step attack based adversarial training does work well on segmentation model due to the challenge to create effective segmentation adversarial examples with a single-step attack. Another way to address the dilemma is to simulate the robustness performance of multi-step attack-based adversarial training in an efficient way [37,57,5]. However, it is not clear how well the generalization of the methods above to segmentation is.

3 SegPGD for Evaluating and Boosting Segmentation

In semantic segmentation, given the segmentation model $f_{seg}(\cdot)$, the clean image $\boldsymbol{X}^{clean} \in \mathbb{R}^{H\times W\times C}$ and its segmentation label $\boldsymbol{Y} \in \mathbb{R}^{H\times W\times M}$, the segmentation model classifies all individual pixels of the input image $f_{seg}(\boldsymbol{X}^{clean}) \in \mathbb{R}^{H\times W\times M}$. The notation (H, W) corresponds to the size of input image, C is the number of image channels, and M stands for the number of output classes. The goal of the attack is to create an adversarial example to mislead classifications of all pixels of an input image.

3.1 SegPGD: An Effective and Efficient Segmentation Attack

Formally, the goal of attack is defined to create the adversarial example $\boldsymbol{X}^{adv}$ to mislead all the pixel classifications of an image $\boldsymbol{X}^{clean}$, *i.e.,* $argmax(f_{seg}(\boldsymbol{X}^{adv})_i) \neq argmax(\boldsymbol{Y}_i)$ where $i \in [1, H \times W]$ corresponds to the index of a input pixel. One of the most popular attack method PGD [29] creates adversarial examples via multiple iterations in Equation 1.

$$\boldsymbol{X}^{adv_{t+1}} = \phi^{\epsilon}(\boldsymbol{X}^{adv_t} + \alpha * sign(\nabla_{\boldsymbol{X}^{adv_t}} L(f(\boldsymbol{X}^{adv_t}), \boldsymbol{Y}))), \tag{1}$$

where α, ϵ are the step size and the perturbation range, respectively. $\boldsymbol{X}^{adv_t}$ is the adversarial example after the t-th attack step, and the initial value is set to $\boldsymbol{X}^{adv_0} = \boldsymbol{X}^{clean} + \mathcal{U}(-\epsilon, +\epsilon)$, which corresponds to the random initialization of perturbations. The $\phi^{\epsilon}(\cdot)$ function clips its output into the range $[\boldsymbol{X}^{clean} - \epsilon, \boldsymbol{X}^{clean} + \epsilon]$. Besides, $\boldsymbol{X}^{adv_t}$ is always clipped into a valid image space. $sign(\cdot)$ is the sign function and $\nabla_a(b)$ is the matrix derivative of b with respect to a. $L(\cdot)$ stands for the cross-entropy loss function. In segmentation, the loss is

$$L(f_{seg}(\boldsymbol{X}^{adv_t}), \boldsymbol{Y}) = \frac{1}{H \times W} \sum_{i=1}^{H\times W} CE(f_{seg}(\boldsymbol{X}^{adv_t})_i, \boldsymbol{Y}_i) = \frac{1}{H \times W} \sum_{i=1}^{H\times W} L_i. \tag{2}$$

We reformulate the loss function into two parts in Equation 3. The first term therein is the loss of the correctly classified pixels, while the second one is formed by the wrongly classified pixels.

$$L(f_{seg}(\boldsymbol{X}^{adv_t}), \boldsymbol{Y}) = \frac{1}{H \times W} \sum_{j\in P^T} L_j + \frac{1}{H \times W} \sum_{k\in P^F} L_k, \tag{3}$$

Algorithm 1 SegPGD: An Efficient and Effective Segmentation Attack

Require: segmentation model $f_{seg}(\cdot)$, clean samples $\boldsymbol{X}^{clean}$, perturbation range ϵ, step size α, attack iterations T

$\boldsymbol{X}^{adv_0} = \boldsymbol{X}^{clean} + \mathcal{U}(-\epsilon, +\epsilon)$ ▷ initialize adversarial example

for t ← 1 to T **do** ▷ loop over attack iterations

$P = f_{seg}(\boldsymbol{X}^{adv_{t-1}})$ ▷ make predictions

$P^T, P^F \leftarrow P$ ▷ split predictions

$\lambda(t) \leftarrow (t-1)/2T$ ▷ compute weight

$L \leftarrow (1-\lambda(t)) * L(P^T, \boldsymbol{Y}) + \lambda(t) * L(P^F, \boldsymbol{Y})$ ▷ loss for example updates

$\boldsymbol{X}^{adv_t} \leftarrow \boldsymbol{X}^{adv_{t-1}} + \alpha * sign(\nabla_{\boldsymbol{X}^{adv_{t-1}}} L)$ ▷ update adversarial examples

$\boldsymbol{X}^{adv_t} \leftarrow \phi^{\epsilon}(\boldsymbol{X}^{adv_t})$ ▷ clip into ϵ-ball of clean image

end for

where P^T is the set of correctly classified pixels, P^F corresponds to wrongly classified ones. The two sets make up all pixels, *i.e.,* $\#P^T + \#P^F = H \times W$.

The loss of the second term is often large since the wrongly classified pixels lead to large cross-entropy loss. When creating adversarial examples, the gradient of the second loss term can dominate. However, the increase of the second-term loss does not lead to better adversarial effect since the involved pixels have already been wrongly classified. To achieve highly effective adversarial examples on segmentation, a large number of attack iterations are required so that the update towards increasing the first-term loss can be accumulated to mislead correctly classified pixels.

To tackle the issue above, considering the dense pixel classifications in segmentation, we propose the **Seg**mentation-specific **PGD**, dubbed **SegPGD**, which can create more effective adversarial examples with the same number of attack iterations in Equation 4.

$$L(f_{seg}(\boldsymbol{X}^{adv_t}), \boldsymbol{Y}) = \frac{1-\lambda}{H \times W} \sum_{j \in P^T} L_j + \frac{\lambda}{H \times W} \sum_{k \in P^F} L_k, \tag{4}$$

where two loss terms are weighted with $1-\lambda$ and λ, respectively. Note that the selection of λ is non-trivial. It does not work well by simply setting $\lambda = 0$ where only correctly classified pixels are considered. In such a case, the previous wrongly classified pixels can become benign again after a few attack iterations since they are ignored when updating perturbations. The claim is also consistent with the previous observation [48,45] that adversarial perturbation is also sensitive to small noise. Furthermore, setting λ to a fixed value in [0, 0.5] does not always lead to better attack performance due to a similar reason. When most of pixel classifications are fooled after a few attack iterations, less weight on the wrongly classified pixels can make some of them benign again.

In this work, instead of manually specifying a fixed value to λ, we propose to set λ dynamically with the number of attack iterations. The intuition behind the dynamic schedule is that we mainly focus on fooling correct pixel classifications in the first a few attack iterations and then treat the wrong pixel classifications

quasi equally in the last few iterations. By doing this, our SegPGD can achieve similar attack effectiveness with less iterations. We list some instances of our dynamic schedule as follows

$$\lambda(t) = \frac{t-1}{2T}, \qquad \lambda(t) = \frac{1}{2} * \log_2(1 + \frac{t-1}{T}), \qquad \lambda(t) = \frac{1}{2} * (2^{(t-1)/T} - 1), \tag{5}$$

where t is the index of current attack iteration and T are the number of all attack iterations. Our experiments show that all the proposed instances are similarly effective. In this work, we mainly use the first simple linear schedule. The pseudo code of our SegPGD with the proposed schedule is shown in Algorithm 1. Further discussion on the schedules to dynamically set λ are in Sec. 4.2.

Similarly, the loss function in Equation 4 can also be applied in single-step adversarial attack, *e.g.,* FGSM [13]. In the resulted SegFGSM, only correctly classified pixels are considered in case of the proposed λ schedule. Since it only takes one-step update, the wrongly classified pixels is less likely to become benign. Hence, SegFGSM with the proposed λ schedule (*i.e.,* $\lambda = 1$) also shows superior attack performance than FGSM.

In this subsection, we propose a fast segmentation attack method, *i.e.,* SegPGD. It can be applied to evaluate the adversarial robustness of segmentation models in an efficient way. Besides, SegPGD can also be applied to accelerate the adversarial training on segmentation models.

3.2 Convergence Analysis of SegPGD

Problem Formulation. The goal of the attack is to create an adversarial example $\boldsymbol{X}^{adv}$ to maximize cross-entropy loss of all the pixel classifications. The adversarial example is constrained into ϵ-ball of the clean example $\boldsymbol{X}^{clean}$. The cross-entropy loss of i-th pixel is

$$L(\boldsymbol{X}, \boldsymbol{Y}_i) = CE(f_{seg}(\boldsymbol{X}^{adv_t})_i, \boldsymbol{Y}_i). \tag{6}$$

The process to create adversarial example for segmentation can be formulated into a constrained minimization problem

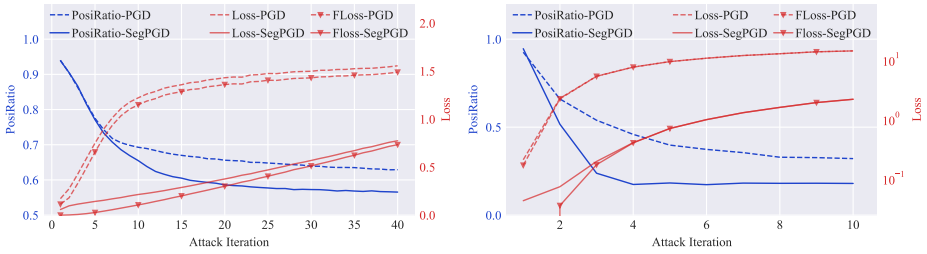
$$\min_{\boldsymbol{X}} \frac{1}{H \times W} \sum_{i=1}^{H \times W} g_i(\boldsymbol{X}) \quad s.t. \ \|\boldsymbol{X} - \boldsymbol{X}^{clean}\|_\infty < \epsilon \text{ and } \boldsymbol{X} \in [0, 1], \tag{7}$$

where $g_i(\boldsymbol{X}) = -L(\boldsymbol{X}, \boldsymbol{Y}_i)$. The variable is constrained into concave region since both constraints are linear.

Projected Gradient Descent-based optimization method is often applied to solve the constrained minimization problem above [29]. The method first takes a step towards the negative gradient direction to get a new point while ignoring the constraint, and then correct the new point by projecting it back into the constraint set.

The gradient-descent step of PGD attack is

$$\boldsymbol{X}^{t+1} = \boldsymbol{X}^t - \alpha * sign(\nabla \sum_{i=1}^{H \times W} g_i(\boldsymbol{X}^t)), \tag{8}$$



(a) PGD3 AT-PSPNet on VOC (b) Standard PSPNet on VOC

Fig. 1: Convergence Analysis. SegPGD marked with blue solid lines achieve higher MisRatio than PGD under the same number of attack iterations. The loss of false classified pixels (**FLoss**) marked with triangle down dominate the overall loss (i.e. red lines without markers) during attacks. Compared to PGD, the FLoss in SegPDG makes up a smaller portion of the overall loss since SegPGD main focuses on correctly classified pixels in the first a few attack iterations.

In contrast, the gradient-descent step of our SegPGD attack is

$$\boldsymbol{X}^{t+1} = \boldsymbol{X}^t - \alpha * sign(\nabla(\sum_{j \in P^T}(1-\lambda(t))g_j(\boldsymbol{X}^t) + \sum_{k \in P^F}\lambda(t)g_k(\boldsymbol{X}^t))), \quad (9)$$

where α is the step size. The initial point is the original clean example $\boldsymbol{X}^{clean}$ or a random initialization $\boldsymbol{X}^{clean} + \mathcal{U}(-\epsilon, +\epsilon)$.

Convergence Criterion. In classification task, the loss is directly correlated with attack goal. The larger the loss is, the more likely the input is to be misclassified. However, it does not hold in segmentation task. The large loss of segmentation not necessarily leads to more pixel misclassifications since the loss consists of losses of all pixel classifications. Once a pixel is misclassified, the increase of the loss on the pixel does not bring more adversarial effect. Hence, we propose a new convergence criterion for segmentation, dubbed MisRatio, which is defined as the ratio of misclassified pixels to all input pixels.

Convergence Analysis. In the first step to update adversarial examples, the update rule of our SegPGD can be simplified as

$$\boldsymbol{X}^1 = \boldsymbol{X}^0 + \alpha * sign(\sum_{j \in P^T}\nabla g_j(\boldsymbol{X}^t)), \quad (10)$$

For almost all misclassified pixels $k \in P^F$ of $\boldsymbol{X}^0$, the k-th pixel of $\boldsymbol{X}^1$ is still misclassified since natural misclassifications are not sensitive to small adversarial noise in general. The claim is also true with PGD update rule. Besides, our SegPGD can turn part of the pixels $k \in P^T$ of $\boldsymbol{X}^0$ into misclassified ones of $\boldsymbol{X}^1$. However, PGD is less effective to do so since the update direction also takes the misclassified pixels of $\boldsymbol{X}^0$ into consideration. Therefore, our SegPGD can achieve higher MisRatio than PGD in the first step.

Algorithm 2 Segmentation Adversarial Training with SegPGD

Require: segmentation model $f_{seg}(\cdot)$, training iterations $\mathcal{N}$, perturbation range ϵ, step size α, attack iterations T

for i $\leftarrow$ 1 to $\mathcal{N}$ **do**

$\boldsymbol{X}_1^{clean}$, $\boldsymbol{X}_2^{clean} \leftarrow \boldsymbol{X}^{clean}$ ▷ split mini-batch

$\boldsymbol{X}_2^{adv} \leftarrow$ SegPGD($f_{seg}(\cdot)$, $\boldsymbol{X}_2^{clean}$, ϵ, α, i) ▷ create adversarial examples

$L \leftarrow L(f_{seg}(\boldsymbol{X}_1^{clean}),\boldsymbol{Y}_1) + L(f_{seg}(\boldsymbol{X}_2^{adv}),\boldsymbol{Y}_2)$ ▷ loss for network updates

end for

In all intermediate steps, both SegPGD and PGD leverage gradients of all pixels classification loss to update adversarial examples. The difference is that our SegPGD assign more weight to loss of correctly classified pixel classifications. The assigned value depends on the update iteration t. Our SegPGD focuses more on fooling correctly classified pixels at first a few iterations and then treat both quasi equally. By doing this, our SegPGD can achieve higher MisRatio than PGD under the same attack iterations.

In Fig. 1, we show the pixel classification loss and PosiRatio (=1 - MisRatio) in each attack iteration. Fig. 1a shows the case to attack adversarially trained PSPNet on VOC (see more details in experimental section). SegPGD marked with blue solid lines achieve higher MissRatio than PGD under the same number of attack iterations. The loss of False classified pixels (FLoss) marked with triangle down dominate the overall loss (i.e. red lines without markers) during attacks. Compared to PGD, the FLoss in SegPDG makes up a smaller portion of the overall loss since SegPGD main focuses on correctly classified pixels in the first a few attack iterations. Note that the scale of loss does not matter since only the signs of input gradients are leveraged to create adversarial examples.

3.3 Segmentation Adversarial Training with SegPGD

Adversarial training, as one of the most effective defense methods, has been well studied in the classification task. In classification, the main challenge of applying adversarial training is computational cost. It requires multiple gradient propagation to produce adversarial images, which makes adversarial training slow. In fact, it can take 3-30 times longer to train a robust network with adversarial training than training a non-robust equivalent [37]. The segmentation task makes the adversarial training more challenging. More attack iterations are required to create effective adversarial examples for boosting segmentation robustness. *E.g.,* more than 100 attack iterations are required to fool segmentation [50].

In this work, we improve segmentation adversarial training by applying SegPGD as the underlying attack. As an effective and efficient segmentation attack method, SegPGD can create more effective adversarial examples than the popular PGD. By injecting the created adversarial examples into the training data, adversarial training with SegPGD can achieve a more robust segmentation model with the same computational cost. Following the previous work, the adversarial training procedure on segmentation is shown in Algorithm 2.

4 Experiment

In this section, we first introduce the experimental setting. Then, we show the effectiveness of SegPGD. Specifically, we show SegPGD can achieve similar attack effect with less attack iterations than PGD on both standard models and adversarially trained models. In the last part, we show that adversarial training with SegPGD can achieve more adversarially robust segmentation models.

4.1 Experimental Setting

Datasets. The popular semantic segmentation datasets, PASCAL VOC 2012 (VOC) [11] and Cityscapes (CS) [9], are adopted in experiments. VOC dataset contains 20 object classes and one class for background, with 1,464, 1,499, and 1,456 images for training, validation, and testing, respectively. Following the popular protocol [17], the training set is augmented to 10,582 images. Cityscapes dataset contains urban scene understanding images with 19 categories, which contains high-quality pixel-level annotations with 2,975, 500, and 1,525 images for training, validation, and testing, respectively.

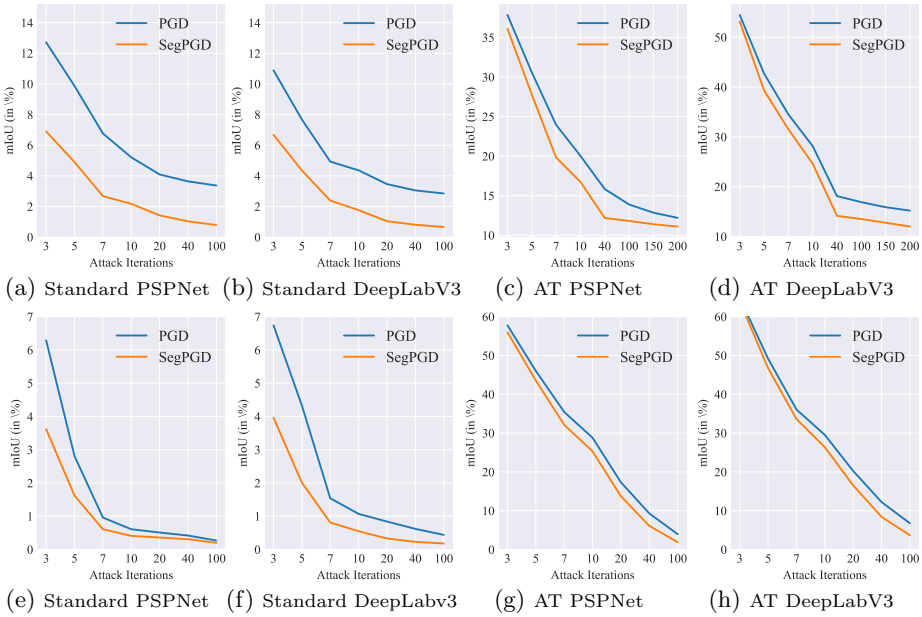
Models. We choose popular semantic segmentation architectures PSPNet [56] and DeepLabv3 [7] for our experiments. The standard configuration of the model architectures is used as in [56]. By default, ResNet50 [18] is applied as a backbone for feature extraction in both segmentation models.

Adversarial Attack. We choose the popular single-step attack FGSM [13] and the popular multiple-step attack PGD [29] as our baseline attack methods. In this work, we focus on ℓ_∞-based perturbations. The maximum allowed perturbation value ϵ is set to $0.03 = 8/255$. The step size α is set to 0.03 for FGSM and 0.01 for PGD. The PGD with 3 attack iterations is denoted as PGD3. Besides, for evaluating the robustness of segmentation models, we also apply attack methods, such as CW attack [6], DeepFool [31] and ℓ_2-based BIM [26].

Metrics. The standard segmentation evaluation metric mIoU (in %) is used to evaluate the adversarial robustness of segmentation models. The mIoUs on both clean image and adversarial images are reported, respectively. The higher the mIoUs are, the more robust the model is.

4.2 Evaluating Segmentation Robustness with SegPGD

Quantitative Evaluation. We train PSPNet and DeepLabV3 on VOC and Cityscapes, respectively. Both standard training and adversarial training are considered in this experiment. PGD with 3 attack iterations is applied as the underlying attack method of adversarial training. This result in 8 models. We apply PGD and SegPGD on the 8 models. On each model, we report the final mIoU under attack with different attack iterations, *e.g.,* 20, 40 and 100. As shown in Fig. 2, the segmentation models show low mIoU on the adversarial examples created by our SegPGD. SegPGD achieve can converge faster to a better minima than PGD, which shows the high effectiveness and efficiency of SegPGD.



(a) Standard PSPNet (b) Standard DeepLabV3 (c) AT PSPNet (d) AT DeepLabV3

(e) Standard PSPNet (f) Standard DeepLabv3 (g) AT PSPNet (h) AT DeepLabV3

Fig. 2: SegPGD is more effective and efficient than PGD. SegPGD creates more effective adversarial examples with the same number of attack iterations and converges to a better minima than PGD. The subfigures (a-d) show the segmentation mIoUs on VOC, while the scores on Cityscapes are reported in the subfigures (e-h). AT PSPNet stands for the adversarially trained PSPNet.

Qualitative Evaluation. For qualitative evaluations, we visualize the created adversarial examples and model's predictions on them. We take the adversarial examples created on standard PSPNet on VOC with 20 attack iterations as examples. As shown in Fig. 3, the adversarial perturbations created by both PGD and SegPGD are imperceptible to human vision. In other words, the created adversarial examples in Fig. 3b and 3b are not distinguishable from the counterpart clean images in Fig. 3a. The predicted masks on the adversarial examples by SegPGD have deviated more from the ground truth than the ones corresponding to PGD. The visualization in Fig. 3 shows SegPGD creates more effective adversarial examples than PGD under the same number of attack iterations.

Comparison with other Segmentation Attack Methods. The segmentation attack methods have also been explored in related work. The work [20] aim to create adversarial perturbations that are always deceptive when added to any sample. Similarly, The work [23] creates universal perturbations to attack multiple segmentation models. Since more constraints are applied to universal perturbations, both types of universal adversarial perturbations are supposed to be less effective than the sample-specific ones. Another work [20] related to us proposes Dense Adversary Generation (DAG), which can be seen as a special

(a) Clean Images and Ground-truth Masks

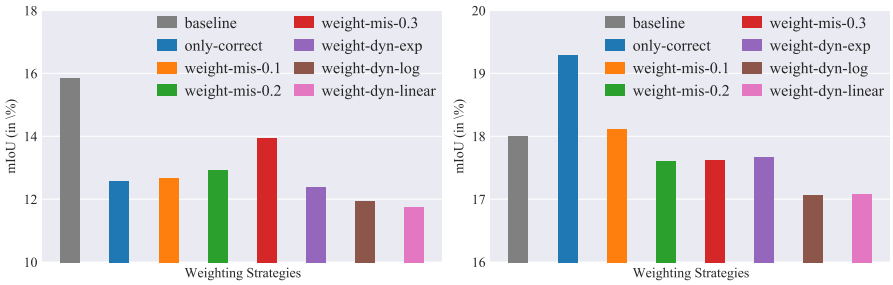
(b) Adversarial Images Created by PGD and Model Predictions

(c) Adversarial Images Created by SegPGD and Model Predictions

Fig. 3: Visualizing of Adversarial Examples and Predictions on them. SegPGD create more effective adversarial examples than PGD.

case of our SegPGD along with other minor differences. DAG only considers the correctly classified pixels in each attack iteration, which is equivalent to set $\lambda = 0$ in our SegPGD. To further improve the attack effectiveness, the work [16] proposes multiple-layer attack (MLAttack) where the losses in feature spaces of multiple intermediate layers and the one in the final output layer are combined to create adversarial examples. SegPGD outperforms both DAG and MLAttack in terms of both efficiency and effectiveness, as shown in Appendix A.

Single-Step Attack. When a single attack iteration is applied, SegPGD is degraded to SegFGSM. In SegFGSM, only the loss of correctly classified pixels are considered in the case of the proposed λ schedule. We compare FGSM and SegFGSM and report the mIOU. Our SegFGSM outperforms FGSM on both standard models and adversarially trained models. See Appendix B for details.



(a) Standard PSPNet on VOC (b) PGD3 AT-PSPNet on VOC

Fig. 4: Schedules for weighting misclassified pixels. SegPGD with our *weight-dyn-linear* weighting schedules can better reduce the mIoU and achieve better attack effectiveness than the ones with baseline schedules.

Ablation on Weighting Schedules. In this work, we argue that the weight should be changed dynamically with the attack iterations. At the beginning of the attack, the update of adversarial example should focus more on fooling correctly classified pixels. In Equation 5, we list three schedule instances, i.e., the *weight-dyn-linear*, *weight-dyn-exp*, and *weight-dyn-log* schedule respectively. We denote the case as *baseline* where the losses of all the pixels are equally treated. Another choice to weigh the loss of misclassified pixels is to use a constant λ, *e.g.,* 0.1, 0.2 or 0.3, which is denoted as *weight-mis-*λ. When the constant is set to zero, only correctly classified pixels are considered to compute the loss in all attack iterations, which is denoted as *only-correct*. We report the mIoU of segmentation under different weighting schedules in Fig. 4. As shown in the figure, SegPGD with our *weight-dyn-linear* weighting schedules can better reduce the mIoU and achieve better attack effectiveness than baselines. Given its simplicity, we apply the linear schedule rule in our SegPGD. We leave the exploration of more dedicated weighting schedules in future work.

4.3 Boosting Segmentation Robustness with SegPGD-AT

The setting of adversarial training in previous work [51] is adopted in this work. In the baseline, PGD is applied as the underlying attack method for adversarial training. In our approach, We apply SegPGD to create adversarial examples for adversarial training. For both standard training and adversarial training, we train for one more time and report the average results.

White-Box Attack. We evaluate the segmentation models with popular white-box attacks. The results are reported in Tab. 1 and Tab. 2. The mIoU of the standard segmentation model can be reduced to near zero. As expected, they are not robust at all to strong attack methods. Adversarial training methods boost the robustness of segmentation models to different degrees. Under the evaluation of all attack methods, adversarial training with our SegPGD achieves more robust segmentation performance than the one with PGD. Besides the

Table 1: Adversarial Training on VOC Dataset. We evaluate the robustness of adversarially trained models with various attacks, especially under strong attacks (*e.g.,* PGD with 100 attack iterations). We report mIoU scores on different segmentation architectures and different adversarial training settings. Adversarial training with our SegPGD can boost the robustness of segmentation models.

PSPNet	Clean	CW	DeepFool	BIMl2	PGD10	PGD20	PGD40	PGD100
Standard	76.64	4.72	14.2	15.32	5.21	4.09	3.64	3.37
PGD3-AT	74.51	52.23	55.46	51.56	20.04	17.34	15.84	13.89
SegPGD3-AT	75.38	**56.52**	**59.47**	50.17	**26.6**	**20.69**	**17.19**	**14.49**
PGD7-AT	74.99	42.30	45.05	47.21	21.79	19.39	17.99	16.97
SegPGD7-AT	74.45	**48.79**	**51.44**	45.15	**25.73**	**22.05**	**20.61**	**19.23**
DeepLabv3	Clean	CW	DeepFool	BIMl2	PGD10	PGD20	PGD40	PGD100
Standard	77.36	5.24	13.57	14.76	4.36	3.46	3.05	2.85
PGD3-AT	75.03	57.10	60.23	36.83	28.16	20.77	18.12	16.91
SegPGD3-AT	75.01	**59.55**	**62.12**	**39.46**	26.29	**20.92**	**19.1**	**18.24**
PGD7-AT	73.45	48.51	48.87	43.13	26.23	21.15	20.06	19.10
SegPGD7-AT	74.46	**51.42**	**51.47**	42.91	**30.95**	**26.68**	**24.32**	**23.09**

popular segmentation attack methods, we also evaluate the adversarially-trained models with our SegPGD. The evaluation results also support our conclusion, which can be found in Appendix C.

We also compare our SegPGD-AT with the recently proposed segmentation adversarial training method DDCAT [51]. We load the pre-trained DDCAT models from their released the codebase and evaluate the model with strong attacks. We found that their models are vulnerable to strong attacks, *e.g.,* PGD100. For fair comparison, we compare the scores on our SegPGD3-AT with the ones on their models since three steps are applied to generate adversarial examples in both cases. Our model trained with SegPGD3-AT outperform the DDCAT by a large margin under strong attacks, *e.g.,* 10.98 (DDCAT) vs. 18.24 (ours) with DeepLabv3 architecture on VOC dataset under PGD100. More results can be found in Appendix D.

In our experiments, PGD-AT PSPNet on Cityscapes can be almost completely fooled under strong attack where the mIoU is 3.95 under PGD100 attack. Adversarial training with SegPGD boosts the robustness to 13.04. Although the improvement is large, there is still much space to improve.

Black-Box attack. We also evaluate the segmentation robustness with black-box attacks. Different from white-box attacks, black-box attackers are supposed to have no access to the gradient of the target model. Following the previous work [51], we conduct experiments with transfer-based black-box attacks. We train PSPNet and DeepLabV3 on the same dataset. Then, we create adversarial examples on PSPNet with PGD100 or SegPGD100 and test the robustness of DeepLabV3 on these adversarial examples. The detailed results are reported in Appendix E. The DeepLabV3 models trained with different adversarial training methods are tested. The model trained with our SegPGD-AT shows the best

Table 2: Adversarial Training on Cityscapes Dataset. This table show that the boosting effect of adversarial training with our SegPGD still clearly holds on a different dataset. Besides, we show the previous adversarially trained baseline model can be reduced to near zero under strong attack, *i.e.,* PGD3-AT PSPNet under PGD100 attack. Our segPGD improves the robustness significantly.

PSPNet	Clean	CW	DeepFool	BIMl2	PGD10	PGD20	PGD40	PGD100
Standard	73.98	5.94	12.68	12.36	0.96	0.61	0.42	0.27
DDCAT [51]	76.64	4.72	14.2	15.32	5.21	4.09	3.64	3.37
PGD3-AT	71.28	35.21	36.84	32.22	28.79	17.3	9.29	3.95
SegPGD3-AT	71.01	**36.30**	**38.27**	**35.34**	**33.52**	**25.23**	**19.22**	**13.04**
PGD7-AT	69.85	27.78	28.44	27.87	26.00	24.75	23.86	22.8
SegPGD7-AT	70.21	**29.59**	**30.68**	**32.55**	**27.13**	**25.56**	**24.29**	**23.13**
DeepLabv3	Clean	CW	DeepFool	BIMl2	PGD10	PGD20	PGD40	PGD100
Standard	73.82	8.24	14.26	13.86	1.07	0.84	0.62	0.44
DDCAT [51]	76.64	4.72	14.2	15.32	5.21	4.09	3.64	3.37
PGD3-AT	71.45	36.72	38.98	36.78	29.52	20.23	12.22	6.74
SegPGD3-AT	71.04	**37.93**	37.63	34.54	**32.11**	**25.49**	**17.67**	**15.23**
PGD7-AT	69.91	28.87	29.63	30.58	25.64	24.48	22.87	21.24
SegPGD7-AT	69.93	**29.73**	**31.30**	**32.35**	**30.43**	**28.78**	**26.73**	**25.31**

performance against the transfer-based black-box attacks. The claim is also true when different attack methods are applied to create adversarial examples.

5 Conclusions

A large number of attack iterations are required to create effective segmentation adversarial examples. The requirement makes both robustness evaluation and adversarial training on segmentation challenging. In this work, we propose an effective and efficient segmentation-specific attack method, dubbed SegPGD. We first show SegPGD can converge better and faster than the baseline PGD. The effectiveness and efficiency of SegPGD is verified with comprehensive experiments on different segmentation architectures and popular datasets. Besides the evaluation, we also demonstrate how to boost the robustness of segmentation models with SegPGD. Specifically, we apply SegPGD to create segmentation adversarial examples for adversarial training. Given the high effectiveness of the created adversarial examples, the adversarial training with SegPGD improves the segmentation robustness significantly and achieves the state of the art. However, there is still much space to improve in terms of the effectiveness and efficiency of segmentation adversarial training. We hope this work can serve as a solid baseline and inspire more work to improve segmentation robustness.

Acknowledgement This work is supported by the UKRI grant: Turing AI Fellowship EP/W002981/1, EPSRC/MURI grant: EP/N019474/1, HKU Startup Fund, and HKU Seed Fund for Basic Research. We would also like to thank the Royal Academy of Engineering and FiveAI.

References

1. Andriushchenko, M., Flammarion, N.: Understanding and improving fast adversarial training. NeurIPS (2020)
2. Arnab, A., Miksik, O., Torr, P.H.: On the robustness of semantic segmentation models to adversarial attacks. In: CVPR (2018)
3. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: ICML (2018)
4. Bar, A., Lohdefink, J., Kapoor, N., Varghese, S.J., Huger, F., Schlicht, P., Fingscheidt, T.: The vulnerability of semantic segmentation networks to adversarial attacks in autonomous driving: Enhancing extensive environment sensing. IEEE Signal Processing Magazine **38**(1), 42–52 (2020)
5. Cai, Q.Z., Du, M., Liu, C., Song, D.: Curriculum adversarial training. IJCAI (2018)
6. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)
7. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. In: arXiv:1706.05587 (2017)
8. Cho, S., Jun, T.J., Oh, B., Kim, D.: Dapas: Denoising autoencoder to prevent adversarial attack in semantic segmentation. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2020)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
10. Daza, L., Pérez, J.C., Arbeláez, P.: Towards robust general medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 3–13. Springer (2021)
11. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision (IJCV) (2010)
12. Full, P.M., Isensee, F., Jäger, P.F., Maier-Hein, K.: Studying robustness of semantic segmentation under domain shift in cardiac mri. In: International Workshop on Statistical Atlases and Computational Models of the Heart. pp. 238–249. Springer (2020)
13. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
14. Gu, J., Wu, B., Tresp, V.: Effective and efficient vote attack on capsule networks. arXiv preprint arXiv:2102.10055 (2021)
15. Gu, J., Zhao, H., Tresp, V., Torr, P.: Adversarial examples on segmentation models can be easy to transfer. arXiv preprint arXiv:2111.11368 (2021)
16. Gupta, P., Rahtu, E.: Mlattack: Fooling semantic segmentation networks by multi-layer attacks. In: German Conference on Pattern Recognition. pp. 401–413. Springer (2019)
17. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 447–456 (2015)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
19. He, X., Yang, S., Li, G., Li, H., Chang, H., Yu, Y.: Non-local context encoder: Robust biomedical image segmentation against adversarial attacks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8417–8424 (2019)

20. Hendrik Metzen, J., Chaithanya Kumar, M., Brox, T., Fischer, V.: Universal adversarial perturbations against semantic image segmentation. In: ICCV (2017)
21. Jia, X., Zhang, Y., Wu, B., Ma, K., Wang, J., Cao, X.: Las-at: Adversarial training with learnable attack strategy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13398–13408 (2022)
22. Jia, X., Zhang, Y., Wu, B., Wang, J., Cao, X.: Boosting fast adversarial training with learnable adversarial initialization. IEEE Transactions on Image Processing (2022)
23. Kang, X., Song, B., Du, X., Guizani, M.: Adversarial attacks for image segmentation on multiple lightweight models. IEEE Access **8**, 31359–31370 (2020)
24. Kapoor, N., Bär, A., Varghese, S., Schneider, J.D., Hüger, F., Schlicht, P., Fingscheidt, T.: From a fourier-domain perspective on adversarial examples to a wiener filter defense for semantic segmentation. In: 2021 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2021)
25. Klingner, M., Bar, A., Fingscheidt, T.: Improved noise and attack robustness for semantic segmentation by using multi-task training with self-supervised depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 320–321 (2020)
26. Kurakin, A., Goodfellow, I., Bengio, S., et al.: Adversarial examples in the physical world. In: ICLR (2016)
27. Lee, H.J., Ro, Y.M.: Adversarially robust multi-sensor fusion model training via random feature fusion for semantic segmentation. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 339–343. IEEE (2021)
28. Li, Y., Li, Y., Lv, Y., Jiang, Y., Xia, S.T.: Hidden backdoor attack against semantic segmentation models. arXiv preprint arXiv:2103.04038 (2021)
29. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
30. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3DV (2016)
31. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
32. Nakka, K.K., Salzmann, M.: Indirect local attacks for context-aware semantic segmentation networks. In: European Conference on Computer Vision. pp. 611–628. Springer (2020)
33. Nesti, F., Rossolini, G., Nair, S., Biondi, A., Buttazzo, G.: Evaluating the robustness of semantic segmentation for autonomous driving against real-world adversarial patch attacks. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2280–2289 (2022)
34. Park, G.Y., Lee, S.W.: Reliably fast adversarial training via latent adversarial perturbation. ICCV (2021)
35. Paschali, M., Conjeti, S., Navarro, F., Navab, N.: Generalizability vs. robustness: investigating medical imaging networks using adversarial examples. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 493–501. Springer (2018)
36. Rossolini, G., Nesti, F., D'Amico, G., Nair, S., Biondi, A., Buttazzo, G.: On the real-world adversarial robustness of real-time semantic segmentation models for autonomous driving. arXiv preprint arXiv:2201.01850 (2022)
37. Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! NeurIPS (2019)

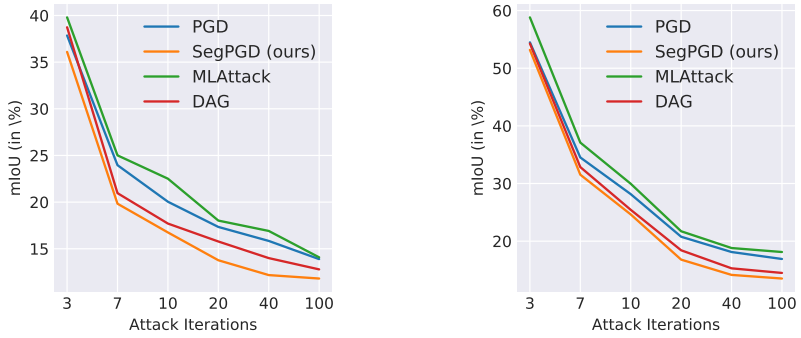
38. Shen, G., Mao, C., Yang, J., Ray, B.: Advspade: Realistic unrestricted attacks for semantic segmentation. arXiv preprint arXiv:1910.02354 (2019)
39. Sriramanan, G., Addepalli, S., Baburaj, A., et al.: Towards efficient and effective adversarial training. NeurIPS (2021)
40. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: ICLR (2014)
41. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. ICLR (2018)
42. Tran, H.D., Pal, N., Musau, P., Lopez, D.M., Hamilton, N., Yang, X., Bak, S., Johnson, T.T.: Robustness verification of semantic segmentation neural networks using relaxed reachability. In: International Conference on Computer Aided Verification. pp. 263–286. Springer (2021)
43. Vivek, B., Babu, R.V.: Single-step adversarial training with dropout scheduling. In: CVPR (2020)
44. Vivek, B., Mopuri, K.R., Babu, R.V.: Gray-box adversarial training. In: ECCV. pp. 203–218 (2018)
45. Wang, D., Ju, A., Shelhamer, E., Wagner, D., Darrell, T.: Fighting gradients with gradients: Dynamic defenses against adversarial attacks. arXiv preprint arXiv:2105.08714 (2021)
46. Wang, J., Zhang, H.: Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In: ICCV (2019)
47. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. ICLR (2020)
48. Wu, B., Pan, H., Shen, L., Gu, J., Zhao, S., Li, Z., Cai, D., He, X., Liu, W.: Attacking adversarial attacks as a defense. arXiv preprint arXiv:2106.04938 (2021)
49. Xiao, C., Deng, R., Li, B., Yu, F., Liu, M., Song, D.: Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 217–234 (2018)
50. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.: Adversarial examples for semantic segmentation and object detection. In: ICCV (2017)
51. Xu, X., Zhao, H., Jia, J.: Dynamic divide-and-conquer adversarial training for robust semantic segmentation. In: ICCV (2021)
52. Ye, N., Li, Q., Zhou, X.Y., Zhu, Z.: Amata: An annealing mechanism for adversarial training acceleration. AAAI (2021)
53. Yu, Y., Lee, H.J., Kim, B.C., Kim, J.U., Ro, Y.M.: Towards robust training of multi-sensor data fusion network against adversarial examples in semantic segmentation. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4710–4714. IEEE (2021)
54. Zhang, D., Zhang, T., Lu, Y., Zhu, Z., Dong, B.: You only propagate once: Accelerating adversarial training via maximal principle. NeurIPS (2019)
55. Zhang, H., Wang, J.: Defense against adversarial attacks using feature scattering-based adversarial training. NeurIPS (2019)
56. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
57. Zheng, H., Zhang, Z., Gu, J., Lee, H., Prakash, A.: Efficient adversarial training with transferable adversarial examples. In: CVPR (2020)

SegPGD: An Effective and Efficient Adversarial Attack for Evaluating and Boosting Segmentation Robustness

Supplementary Material

A Comparison of SegPGD with other Segmentation Methods

We report the robust accuracy of adversarially trained (PGD3-AT) models under different attacks, namely, SegPGD, DAG and MLAttack. In DAG method, we apply projected gradient descent as the underlying optimization method and only focus on the correctly classified pixels. In MLAttack, three losses are considered for each input image, *i.e.,* the segmentation loss in the output layer, the of in the last layer of encoder and the MSE loss of features multiple Note that the MSE loss is computed as the MSE between the features on the clean input and the ones on current adversarial examples. For each of the three losses, the input gradients are computed to update the input examples. For fair comparison, we compare the segmentation methods with the same number of gradient propagation passes. As shown in Fig. 5, our SegPGD achieves better attack effectiveness and converges faster than other segmentation methods.



(a) PSPNet trained with PGD3-AT on VOC (b) DeepLabV3 trained with PGD3-AT on VOC

Fig. 5: Comparison of SegPGD with other Segmentation Methods. Given the same computational cost (*i.e.*,, the same number of propagation passes), our SegPGD achieves better attack effectiveness.

B Single-step Attack: SegFGSM

When a single-step attack iteration is applied, SegPGD is degraded to SegFGSM. The results under the single-step attack is shown in Tab. 3. As shown in the table, our SegFGSM outperforms FGSM on both standard models and adversarially trained models. The conclusion is true across popular segmentation model architectures on two standard segmentation datasets.

	PSPNet-VOC		DeepLabV3-VOC		PSPNet-CityScapes		DeepLabV3-CityScapes	
	Standard	AT	Standard	AT	Standard	AT	Standard	AT
Clean	76.64	74.51	77.36	75.03	73.98	71.28	73.82	71.45
FGSM	36.76	55.33	37.59	46.78	43.76	57.5	42.79	53.85
SegFGSM	**30.80**	**53.98**	**31.58**	**43.88**	**38.53**	**56.53**	**37.97**	**52.92**

Table 3: Single-step Attack. Our SegFGSM ourperforms FGSM on both standard models and adversarially trained models.

C Model Evaluation under SegPGD Attack

We evaluate adversarial trained SegPGD-AT models with our SegPGD attack method. As shown in Tab. 4, the model adversarially trained with SegPGD also outperforms the one with PGD under the SegPGD attack evaluation. In addition, the observation also echos our claim that the SegPGD can better fool segmentation models than PGD.

		AT on VOC			
		PGD3-AT	SegPGD3-AT	PGD7-AT	SegPGD7-AT
Attack Method	PGD100	13.89	14.49	16.97	19.23
	SegPGD100	9.67	10.34	16.20	17.03
		AT on Cityscapes			
		PGD3-AT	SegPGD3-AT	PGD7-AT	SegPGD7-AT
Attack Method	PGD100	3.95	13.04	22.80	23.13
	SegPGD100	1.91	8.86	17.03	22.54

Table 4: Model Evaluation under SegPGD Attack. The evaluation on SegPGD-AT PSPNet is reported with mIoU metric.

D Comparison of SegPGD-AT with DDCAT

We also compare our SegPGD-AT with the recently proposed segmentation adversarial training method DDCAT. We load the pre-trained DDCAT models from their released the codebase and evaluate the model with strong attacks. We found that their models are very weak to defend strong attacks. For fair comparison, we compare the scores on our SegPGD3-AT with the ones on their models since three steps are applied to generate adversarial examples in both case. As shown in Tab. 5, our model trained with SegPGD3-AT outperform the DDCAT by a large margin under strong attacks.

		Attack on PSPNet			Attack on DeepLabV3		
		PGD20	PGD40	PGD100	PGD20	PGD40	PGD100
AT-Models	DDCAT [51]	18.96	14.22	10.84	15.23	11.27	10.98
	SegPGD3-AT	20.69	17.19	14.49	20.92	19.10	18.24

Table 5: Comparison of SegPGD-AT with DDCAT. The SegPGD-AT model shows higher robust accuracy than DDCAT model under the same attack.

E Black-box Attack on Adversarially Trained Models

We train PSPNet and DeepLabV3 on the same dataset. Then, we create adversarial examples on PSPNet with PGD100 or SegPGD100 and test the robustness of DeepLabV3 on these adversarial examples. The results are reported in Tab. 6. We test the DeepLabV3 models trained with different methods. The model trained with our SegPGD3 shows the best performance against the transfer-based black-box attacks. The claim is also true when different attack methods are applied to create adversarial examples.

			Target Model: Deeplabv3 on VOC		
Source Model:	Training	Attack	PGD3-AT	DDCAT	**SegPGD3-AT**
PSPNet	PGD3-AT	PGD100	15.98	14.87	**16.94**
		SegPGD100	12.38	11.94	**13.43**
			Target Model: Deeplabv3 on Cityscapes		
Source Model:	Training	Attack	PGD3-AT	DDCAT	**SegPGD3-AT**
PSPNet	PGD3-AT	PGD100	14.28	15.02	**19.42**
		SegPGD100	13.32	14.26	**20.11**

Table 6: Evaluation under Black-box Attacks. The model with our SegPGD3-based adversarial training performs more robust than other methods on different datasets under different attacks.