Exploring Robustness of Unsupervised Domain Adaptation in Semantic Segmentation

Jinyu Yang[1], Chunyuan Li[1], Weizhi An[1], Hehuan Ma[1], Yuzhi Guo[1], Yu Rong[2], Peilin Zhao[2], and Junzhou Huang[1]
[1]University Of Texas at Arlington, [2]Tencent AI Lab
{jinyu.yang, chunyuan.li, weizhi.an, hehuan.ma, yuzhi.guo}@mavs.uta.edu
yu.rong@hotmail.com, masonzhao@tencent.com, jzhuang@uta.edu

Abstract

Recent studies imply that deep neural networks are vulnerable to adversarial examples, i.e., inputs with a slight but intentional perturbation are incorrectly classified by the network. Such vulnerability makes it risky for some security-related applications (e.g., semantic segmentation in autonomous cars) and triggers tremendous concerns on the model reliability. For the first time, we comprehensively evaluate the robustness of existing UDA methods and propose a robust UDA approach. It is rooted in two observations: i) the robustness of UDA methods in semantic segmentation remains unexplored, which poses a security concern in this field; and ii) although commonly used self-supervision (e.g., rotation and jigsaw) benefits model robustness in classification and recognition tasks, they fail to provide the critical supervision signals that are essential in semantic segmentation. These observations motivate us to propose adversarial self-supervision UDA (or ASSUDA) that maximizes the agreement between clean images and their adversarial examples by a contrastive loss in the output space. Extensive empirical studies on commonly used benchmarks demonstrate that ASSUDA is resistant to adversarial attacks.

1. Introduction

Semantic segmentation aims to predict semantic labels of each pixel in the given images, which plays an important role in autonomous driving [19] and medical diagnosis [28]. However, pixel-wise labeling is extremely time-consuming and labor-intensive. For instance, 90 minutes are required to annotate a single image for the Cityscapes dataset [6]. Although synthetic datasets [29, 30] with freely available labels provide an opportunity for model training, the model trained on synthetic data suffers from dramatic performance degradation when applying it directly to the real data of interest.

Motivated by the success of unsupervised domain adaptation (UDA) in image classification, various UDA methods for semantic segmentation are recently proposed. The key idea of these methods is to learn domain-invariant representations by minimizing marginal distribution distance between the source and target domains [15], adapting structured output space [38, 5], or reducing appearance discrepancy through image-to-image translation [1, 51, 18]. Another alternative is to explicitly explore the supervision signals from the target domain through self-training. The key idea is to alternatively generate pseudo labels on target data and re-train the model with these labels. Most of the existing state-of-the-art UDA methods in semantic segmentation rely on this strategy and demonstrate significant performance improvement. [54, 18, 48, 44, 31].

However, one of the critical issues of the aforementioned UDA methods is that they are possibly vulnerable to adversarial attacks. In other words, the performance of a UDA model may dramatically degrade under an unnoticeable perturbation. Unfortunately, the robustness of UDA methods remains largely unexplored in the literature. With the increasing applications of UDA methods in security-related areas, the lack of robustness of these methods leads to massive safety concerns. For instance, even small-magnitude perturbations on traffic signs can potentially cause disastrous consequences to autonomous cars [9, 33], such as life-threatening accidents.

Self-supervised learning (SSL) aims to learn more transferable and generalizable features for vision tasks (*e.g.*, classification and recognition) [8, 10, 12, 4]. Key to SSL is the design of pretext tasks, such as rotation prediction, selfie, and jigsaw, to obtain self-derived supervisory signals on unlabeled data. Recent studies reveal that SSL is effective in improving model robustness and uncertainty [13]. However, commonly used pretext tasks are designed to capture the global representation of a given image or an image patch. Such pretext tasks fail to provide critical supervision sig-

nals for segmentation tasks where fine-grained or pixel-level representations are required [49].

In this paper, we first perform a comprehensive study to evaluate the robustness of existing UDA methods in semantic segmentation. Our results reveal that these methods can be easily fooled by small perturbations and show dramatic performance degradation. To remedy this problem, we introduce a new UDA method known as ASSUDA to robustly adapt domain knowledge in urban-scene semantic segmentation. The key insight of our method is to leverage the regularization power of adversarial examples. Specifically, we propose the adversarial self-supervision that maximizes the agreement between clean images and their adversarial examples by a contrastive loss in the output space. The adversarial examples aim to i) provide fine-grained supervision signals for unlabeled target data, so that more transferable and generalizable features can be learned and ii) improve the robustness of our model against adversarial attacks by taking advantage of both adversarial training and self-supervision.

Our main contributions can be summarized as i) To the best of our knowledge, this paper presents the first systematic study on how existing UDA methods in semantic segmentation are vulnerable to adversarial attacks. We believe this investigation provides new insight into this area; ii) We propose a new UDA method that takes advantage of adversarial training and self-supervision to improve the model robustness; iii) Comprehensive empirical studies demonstrate the robustness of our method against adversarial attacks on two benchmark settings, *i.e.*, "GTA5 to Cityscapes" and "SYNTHIA to Cityscapes".

2. Related Work

Unsupervised Domain Adaptation Unsupervised domain adaptation (UDA) refers to the scenario where no labels are available for the target domain. In the past few years, various UDA methods are proposed for semantic segmentation, which can be mainly summarized as three streams: i) adapt domain-invariant features by directly minimizing the representation distance between two domains [15, 53]; ii) align pixel space through translating images from the source domain to the target domain [1, 25]; iii) align structured output space, which is inspired by the fact that source output and target output share substantial similarities in terms of structure layout [38]. However, simply aligning cross-domain distribution has limited capability in transferring pixel-level domain knowledge for semantic segmentation. To address this problem, the most recent studies integrate self-training into existing UDA frameworks and demonstrate the state-of-the-art performance [54, 18, 48, 44].

Our method instead resorts to self-supervision by integrating contrastive learning into existing UDA methods. This strategy demonstrates two advantages: i) provides supervision for the target domain, which is proved to be robust to the label corruption; ii) encourages the model to learn more transferable and robust features. Another major difference is that our method mainly focuses on improving model robustness against adversarial attacks, which is overlooked by existing UDA methods.

Self-supervised Learning Self-supervision aims to make use of massive amounts of unlabeled data through getting free supervision from the data itself. This is typically achieved by training self-supervised tasks (a.k.a., pretext tasks) through two paradigms, *i.e.*, pre-training & fine-tuning and multi-task learning. Specifically, the pre-training & fine-tuning first performs pre-training on the pretext task, then fine-tunes on the downstream task. In contrast, multi-task learning optimizes the pretext task and the downstream task simultaneously. Our method falls into the latter, where the downstream task is to predict the segmentation labels of the target domain. To learn transferable and generalizable features through self-supervision, it is essential to design pretext tasks that are tailored to the downstream task. Commonly used pretext tasks include exemplar [8], rotation [10], predicting the relative position between two random patches [7], and jigsaw [26]. Motivated by this, recent UDA methods introduce self-supervision into segmentation adaptation to learn domain invariant feature representations [43, 35]. Although these commonly used pretext tasks contribute to cross-domain feature alignment, they are mainly designed to capture the global feature, and therefore have limited capability in learning fine-grained representations that are essential in semantic segmentation.

By contrast, this paper proposes to use adversarial examples to build pretext tasks. Specifically, we maximize agreement between each image and its adversarial example via a contrastive loss in the output space. This is different from [4] that performs contrastive learning in the latent space. Furthermore, rather than focus on single-domain tasks [14, 16], our method is tailored to UDA environments to adapt domain knowledge and improve robustness simultaneously. Therefore, i) our method is encouraged to learn more transferable features which are domain-invariant and fine-grained; ii) the trained model is more robust to label corruption and adversarial attacks. Another closely related work is [46] which shares a similar spirit with us but with clear differences: i) rather than perturb the intermediate feature maps, we perform the perturbation to the input images; ii) we target on improving model robustness, instead of the segmentation accuracy on clean images.

Adversarial Attacks Previous studies reveal that adversarial attacks are commonly observed in machine learning methods such as SVMs [2] and logistic regression [22]. Recent publications suggest that neural networks are also highly vulnerable to adversarial perturbations [36, 11]. Even worse, adversarial attacks are proven to be transferable across different models [37], *i.e.*, the adversarial examples generated to

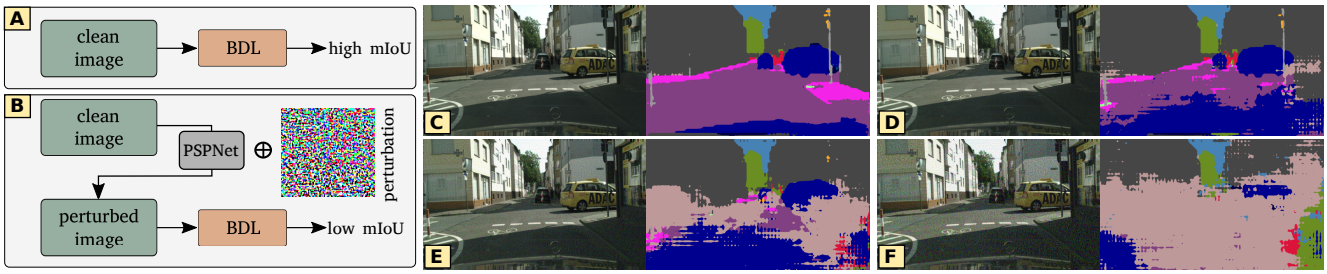

Figure 1. Robustness study of BDL [18] on "GTA5 to Cityscapes" with backbone ResNet101. (A) the traditional paradigm uses clean test data to evaluate the performance of BDL; (B) we use PSPNet as the surrogate model to generate perturbed test data which are then used to evaluate BDL; (C) a clean image and its segmentation output predicted by BDL; (D), (E), and (F) indicate the perturbed images of (C) with $\epsilon = 0.1$, $\epsilon = 0.25$, and $\epsilon = 0.5$, respectively, along with their BDL predictions. Although the perturbations are unnoticeable, they can easily deceive BDL, resulting in dramatic performance degradation.

Base	ϵ	GTA5 to City	SYNTHIA to City
VGG16	0.1	41.3 → 30.5	39.0 → 29.3
	0.25	41.3 → 14.6	39.0 → 13.6
	0.5	41.3 → 7.10	39.0 → 5.90
ResNet101	0.1	48.5 → 36.2	51.4 → 41.2
	0.25	48.5 → 19.9	51.4 → 26.6
	0.5	48.5 → 6.50	51.4 → 11.0

Table 1. Performance of pre-trained BDL on clean test data vs perturbed test data. Three sets of perturbed data are generated with $\epsilon = 0.1$, $\epsilon = 0.25$, and $\epsilon = 0.5$, respectively.

attack a specific model are also harmful to other models. To fully understand adversarial attacks in deep neural networks (DNNs), considerable attention is received in the past few years. Specifically, [11] proposes a fast gradient sign method (FGSM) to efficiently generate adversarial examples with only one gradient step. DeepFool [24] generates minimal perturbations by iteratively linearizing the image classifier. By utilizing the differential evolution, [34] enables us to generate one-pixel adversarial perturbations to accurately attack DNNs.

Unlike the aforementioned studies that focus on effectively creating adversarial attacks, our method uses adversarial examples to build pretext tasks for UDA models, and in turn to improve the model robustness. This is motivated by the fact that a clean image and its adversarial example should have the same segmentation output. Therefore, we can get supervision for free and encourage our method to learn discriminative representation for segmentation tasks.

3. Methodology

We first briefly recall the preliminary of UDA, adversarial training, and self-supervision. We then perform the first-of-its-kind empirical study to show that existing UDA methods are vulnerable to adversarial attacks, which arises tremendous concerns for the application of these methods in safety-critical areas. To address this problem, we propose a new domain adaptation method known as ASSUDA to improve the model robustness without satisfying much predictive accuracy. Specifically, our method takes advantage of adversarial training and self-supervision and thus enabling us to generate more robust and generalizable features.

3.1. Preliminary

UDA in Semantic Segmentation Consider the problem of UDA in semantic segmentation, where a labeled source domain $\mathcal{X}_s\{(x_s^{(i)}, y_s^{(i)})\}_{i=1}^{n_s}$ and an unlabeled target domain $\mathcal{X}_t\{x_t^{(j)}\}_{j=1}^{n_t}$ are given. Our goal is to learn a segmentation model $f_{\theta_C}(\cdot)$ which guarantees accurate prediction on the target domain. Formally, the loss function of a typical UDA model is defined as:

$$\mathcal{L}_{seg}(x_s, y_s; \theta_C) + \alpha \mathcal{L}_{dis}(x_s, x_t), \tag{1}$$

where $\mathcal{L}_{seg}$ is the typical segmentation objective, $\mathcal{L}_{dis}$ measures the domain distance. The most commonly used $\mathcal{L}_{dis}$ is the adversarial loss $\mathcal{L}_{adv}$ that encourages a discriminative and domain-invariant feature representation through a domain discriminator $D_{\theta_D}(\cdot)$ [15, 1, 38], which is formalized as:

$$\begin{aligned}\mathcal{L}_{adv}(x_s, x_t; \theta_C, \theta_D) = \ & \mathbb{E}[log D_{\theta_D}(f_{\theta_C}(x_s))] + \\ & \mathbb{E}[log(1 - D_{\theta_D}(f_{\theta_C}(x_t)))]\end{aligned} \tag{2}$$

Adversarial Training Recall that the objective of the vanilla adversarial training is:

$$\arg\min_x \mathbb{E}_{(x,y)\sim\mathbb{D}}[\max_{\eta\in\mathbb{S}} \mathcal{L}(f_\theta(x+\eta), y)] \tag{3}$$

where $\mathbb{S}$ are allowed perturbations, $\tilde{x} \leftarrow x + \eta$ is an adversarial example of x with the perturbation η. To obtain η, the most commonly used attack method is FGSM [11]:

$$\eta = \epsilon \, \text{sign}(\bigtriangledown_x \mathcal{L}(f_\theta(x), y)), \tag{4}$$

where ϵ is the magnitude of the perturbation. The generated adversarial examples $\tilde{x}$ are imperceptible to human but can easily fool deep neural networks. Recent studies further prove that training models exclusively on adversarial examples can improve the model robustness [21].

3.2. Robustness of UDA Methods

Although existing UDA methods achieve record-breaking predictive accuracy, their robustness against adversarial attacks remains unexplored. We hypothesis that they are also vulnerable to adversarial attacks, which makes it risky to apply them in safety-critical scenarios. To fill this gap and to validate our hypothesis, we perform black-box attacks on BDL [18] by conducting the following two steps: 1) for each clean image in the test data, we first generate its adversarial example by attacking PSPNet [52] with $\epsilon = 0.1$, $\epsilon = 0.25$ and $\epsilon = 0.5$, respectively; 2) we then evaluate the pre-trained BDL model on the generated adversarial examples (or perturbed test data) (Figure 1). The rationale behind this setting is that i) recent state-of-the-art UDA methods in semantic segmentation [42, 17, 48, 44, 31] share similar spirits with BDL, so conducting pilot studies on this method would be representative; ii) a black-box attack assumes that the attacker can only access very limited information of the victim model, which is a common case in the real world. Therefore, a black-box attack would be very dangerous if it can work; iii) adversarial attacks are transferable across different models [11], *i.e.*, the adversarial examples generated to attack a surrogate model are also harmful to other models. We hereby perform the black-box attack to examine the transferability of adversarial examples on UDA models.

As shown in Table 1, despite the remarkable performance of BDL on the clean test data, even slight and unnoticeable perturbations can result in dramatic performance degradation. For instance, BDL (with VGG16 backbone) only achieves a mean IoU (mIoU) of 30.5% on the perturbed test data generated by $\epsilon = 0.1$, compared to 41.3% on the clean data. By increasing the perturbation ratio ϵ, the performance can drop even further (Figure 1), indicating that BDL can be easily fooled by slight perturbations on the test data, even though the perturbation is generated by a surrogate model. This empirical study suggests that existing UDA methods are also possibly vulnerable to adversarial perturbations, which can make them especially risky for some security-related areas.

3.3. Adversarial Self-Supervision UDA

To address this problem, the most straightforward approach is adversarial training (equation 3) which requires class labels to generate adversarial examples. However, we are unable to access the labels of target data under the scenario of UDA (equation 1). The success of existing UDA methods heavily relies on the self-training strategy that alternatively generates highly confident pseudo labels for the target domain and re-trains the model using these labels [18, 17, 44, 31, 45]. Although pseudo labels provide an opportunity to generate adversarial examples for the target data, these labels are usually noisy and less accurate. Hendrycks *et al.* prove that self-supervision improves the robustness

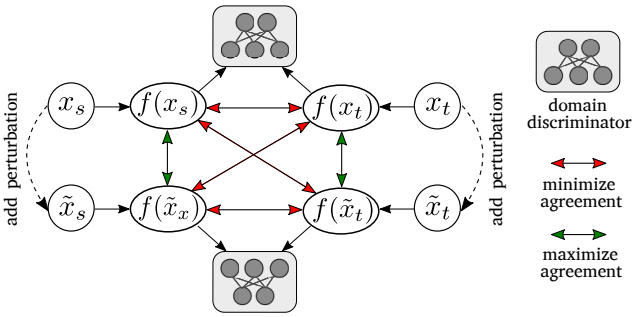

Figure 2. An overview of the proposed method. For each sampled pair of source image x_s and target image x_t, we generate their adversarial example $\tilde{x}_s$ and $\tilde{x}_t$, respectively. A segmentation model $f(\cdot)$ and a domain discriminator are trained to maximize/minimize agreement and align cross-domain representations.

of deep neural networks for vision tasks [13]. Nevertheless, commonly used pretext tasks (*e.g.*, rotation prediction and jigsaw) model global representation and fail to provide the critical supervision signals in learning discriminative features for semantic segmentation.

These challenges raise the question: *can we take advantage of both adversarial training and self-supervision in improving the robustness of UDA methods in semantic segmentation?* To answer this question, we propose to build a pretext task by using adversarial examples (Figure 2). Specifically, we consider a clean image and its adversarial example as a positive pair and maximize agreement on their segmentation outputs by a contrastive loss. This is motivated by the fact that a clean image and its adversarial example should share the same segmentation map. Different from [4] that uses a contrastive loss in the latent space, our pretext task is performed in the output space to learn discriminative representations for semantic segmentation. To adapt knowledge from the source domain to the target domain, a domain discriminator is applied to the source and target outputs. It is worth mentioning that the domain discriminator minimizes the domain-level difference, while the contrastive loss is performed on the pixel level.

Our model is built upon BDL [18] that generates the transformed source images $\mathcal{X}_{s\to t}$ and pseudo labels $Y_{t'}$ of $\mathcal{X}_t$. For simplicity, we use $\mathcal{X}_s$ to represent $\mathcal{X}_{s\to t}$ in the remaining of this paper, unless otherwise specified. At each training iteration r, a minibatch of N source-target pairs are randomly sampled from $\mathcal{X}_s$ and $\mathcal{X}_t$, resulting in $2N$ examples: $\{x_s^{(i)}, x_t^{(i)}\}_{i=1}^N$. Their adversarial examples $\{\tilde{x}_s^{(i)}, \tilde{x}_t^{(i)}\}_{i=1}^N$ are generated by:

$$\begin{aligned}\tilde{x}_s^{(i)} &= x_s^{(i)} + \epsilon_m \text{sign}(\bigtriangledown_x[\mathcal{L}_{seg}(x_s^{(i)}, y_s^{(i)}; \theta_C)]) \\ \tilde{x}_t^{(i)} &= x_t^{(i)} + \epsilon_m \text{sign}(\bigtriangledown_x[\mathcal{L}_{seg}(x_t^{(i)}, y_{t'}^{(i)}; \theta_C)])\end{aligned} \tag{5}$$

where ϵ_m is the training perturbation magnitude.

Given these $4N$ data points $\{x_s^{(i)}, x_t^{(i)}, \tilde{x}_s^{(i)}, \tilde{x}_t^{(i)}\}_{i=1}^N$,

each pair of examples $\{x_\alpha^{(i)}, \tilde{x}_\alpha^{(i)}\}$ is considered as a positive pair (α can be either s or t to denote a source or a target domain), while the other $4N-2$ examples are considered as negative examples. We define the contrastive loss for a positive pair (i,j) as

$$\ell_{i,j} = -log\frac{exp(sim(f_{\theta_C}(x^{(i)}), f_{\theta_C}(x^{(j)})))}{\sum_{k=1}^{4N} \mathbb{1}_{[k\neq i]} exp(sim(f_{\theta_C}(x^{(i)}), f_{\theta_C}(x^{(k)})))}, \tag{6}$$

where $sim(\mathbf{U},\mathbf{V}) = exp(-dist(\mathbf{U},\mathbf{V})/(2\sigma^2))$ is Gaussian kernel that is used to measure the similarity between two segmentation output tensors $\mathbf{U}$ and $\mathbf{V}$, $dist(;)$ is the Euclidean distance. The contrastive loss $\mathcal{L}_{con}(x_s, \tilde{x}_s, x_t, \tilde{x}_t; \theta_C))$ is computed across all positive pairs (see Algorithm 1). Taken together, the training objective of our goal is $\min_{\theta_C}\max_{\theta_D} \mathcal{L}_{total}$, where $\mathcal{L}_{total}$ is:

$$\begin{aligned}\mathcal{L}_{total} = &\mathcal{L}_{seg}(x_s, y_s; \theta_C) + \mathcal{L}_{seg}(\tilde{x}_s, y_s; \theta_C) + \\ &\mathcal{L}_{seg}(x_t, y_{t'}; \theta_C) + \mathcal{L}_{seg}(\tilde{x}_t, y_{t'}; \theta_C) + \\ &\gamma\mathcal{L}_{adv}(x_s, x_t; \theta_C, \theta_D) + \\ &\gamma\mathcal{L}_{adv}(\tilde{x}_s, \tilde{x}_t; \theta_C, \theta_D) + \\ &\delta\mathcal{L}_{con}(x_s, \tilde{x}_s, x_t, \tilde{x}_t; \theta_C),\end{aligned} \tag{7}$$

where δ and γ are two hyper-parameters. Therefore, our model can leverage the regularization power of adversarial examples through a self-supervision manner, and in turn, improve the model robustness against adversarial attacks. The whole training process is detailed in Algorithm 1.

Algorithm 1: The whole training process.

Input: Source data $\{\mathcal{X}_s, Y_s\}$ and target data $\{\mathcal{X}_t\}$,
segmentation model initialized as θ_C,
domain discriminator initialized as θ_D,
batch size N, number of training iteration R
Result: θ_C and θ_D
for $r \leftarrow 1$ **to** R **do**
  Sample a batch of source-target pairs $\{x_s^{(k)}, x_t^{(k)}\}_{k=1}^N$
  # adversarial attack
  for $k \in \{1, ..., N\}$ **do**
    Generate adversarial examples: $\{\tilde{x}_s^{(k)}, \tilde{x}_t^{(k)}\}_{k=1}^N$
    Define $x^{(4k-3)} = x_s^{(k)}, x^{(4k-2)} = x_t^{(k)}, x^{(4k-1)} = \tilde{x}_s^{(k)}, x^{(4k)} = \tilde{x}_t^{(k)}$
  end
  # adversarial self-supervision
  for $i \in \{1, ..., 4N\}$ *and* $j \in \{1, ..., 4N\}$ **do**
    $s_{i,j} = exp(\frac{-dist\left(f_{\theta_C}(x^{(i)}), f_{\theta_C}(x^{(j)})\right)}{2\sigma^2})$
  end
  Define $\ell_{i,j} = -log\frac{exp(s_{i,j})}{\sum_{k=1}^{4N}\mathbb{1}_{[k\neq i]}exp(s_{i,k})}$
  # contrastive loss
  $\mathcal{L}_{con} = \frac{1}{4N}\sum_{k=1}^{N}[\ell_{4k-3,4k-1} + \ell_{4k-1,4k-3} + \ell_{4k-2,4k} + \ell_{4k,4k-2}]$
  # update model parameters
  $\theta_C \leftarrow \theta_C - \beta \bigtriangledown_{\theta_C} \mathcal{L}_{total}$
  $\theta_D \leftarrow \theta_D - \lambda \bigtriangledown_{\theta_D} \mathcal{L}_{total}$
end
return θ_C and θ_D

4. Experiments

Datasets Following the same setting as previous studies, we use GTA5 [29] and SYNTHIA-RAND-CITYSCAPES [30] as the source domain, and use Cityscapes [6] as the target domain. GTA5 is composed of 24,966 images (resolution: 1914 × 1052) with pixel-level semantic labels, which are collected from a photo-realistic open-world game known as Grand Theft Auto V. SYNTHIA-RAND-CITYSCAPES dataset is generated from a virtual city, including 9,400 images (resolution: 1280 × 760) with precise pixel-level semantic annotations. Cityscapes is a large-scale street scene dataset collected from 50 cities. A total of 5,000 images (resolution: 2048 × 1024) are contained in Cityscapes, with 2,975 training images, 500 validation images, and 1,525 test images. We follow the tradition to use the training images from Cityscapes as the target domain and use the validation images as the clean test data.

Implementation Details Following the same experimental protocol in this area, we use two network architectures: DeepLab-v2 [3] with VGG16 [32] backbone, and DeepLab-v2 with ResNet101 backbone. The domain discriminator has 5 convolution layers with kernel 4×4 and stride of 2, each of which is followed by a leaky ReLU parameterized by 0.2 except the last one. The channel number of each layer is {64, 128, 256, 512, 1}. The Adam optimizer with initial learning rate 1e-4 and momentum (0.9, 0.99) is used in DeepLab-VGG16. We apply step decay to the learning rate with step size 30000 and drop factor 0.1. Stochastic Gradient Descent optimizer with momentum 0.9 and weight decay 5e-4 is used in DeepLab-ResNet101. The learning rate of DeepLab-ResNet101 is initialized as 1e-4 and is decreased by the polynomial policy with a power of 0.9. Adam optimizer with momentum (0.9, 0.99) and initial learning rate 1e-6 is used in the domain discriminator. We set $\epsilon_m = 1.0$ in equation 5. Code and data are available at https://github.com/uta-smile/ASSUDA.

Perturbed Test Data To evaluate model robustness, we first generate the perturbed test data. Specifically, PSPNet [52] is used as the surrogate model owing to its popularity. We generate three sets of perturbed test data using FGSM with $\epsilon = 0.1$, $\epsilon = 0.25$, and $\epsilon = 0.5$. The generated perturbed data sets are then used for performance assessment. For a fair comparison with existing UDA methods, we download the pre-trained models from the original papers and perform the evaluation.

4.1. Experimental Results

Since the robustness of existing UDA methods remains unexplored, we first comprehensively evaluate their robustness against adversarial attacks in this section (Table 2 and

GTA5 to Cityscapes																							
	ϵ	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bicycle	**mIoU**	**mIoU drop**	**mIoU***
FDA [48]		73.9	18.5	69.7	7.5	6.4	**18.7**	23.9	21.5	76.7	12.2	66.3	45.2	18.4	70.2	18.9	13.9	**14.6**	9.3	22.0	32.0	10.2	42.4
AdaptSegNet [38]		71.9	22.7	70.8	7.6	7.9	16.5	15.4	8.3	71.8	12.2	52.6	33.8	0.6	65.8	15.8	7.6	0.0	0.7	0.1	25.4	9.6	35.0
PCEDA [47]	0.1	**90.9**	25.0	73.5	6.3	7.2	14.2	**24.0**	**27.4**	76.2	23.4	70.3	45.0	19.9	70.0	16.3	20.3	0.0	9.8	**25.1**	33.4	11.2	44.6
BDL [18]		64.0	21.9	70.0	10.0	3.9	8.4	20.5	12.8	77.4	22.3	79.2	49.8	13.8	73.2	17.8	12.1	0.0	7.8	15.2	30.5	10.8	41.3
Ours		90.6	**41.5**	**80.1**	**22.6**	**10.4**	15.4	23.0	16.0	**82.7**	**34.9**	**81.6**	**52.5**	**23.9**	**82.2**	**22.5**	**21.9**	7.0	**15.4**	21.4	**39.3**	**0.4**	39.7
FDA		25.4	3.4	24.5	0.5	1.6	2.4	7.7	6.4	58.6	1.2	44.8	6.5	1.4	14.6	4.9	0.4	0.1	0.1	1.3	10.8	31.4	42.4
AdaptSegNet		5.4	5.0	43.8	1.2	2.2	3.7	6.3	2.5	31.3	3.9	22.8	6.2	0.0	11.9	4.3	0.1	0.0	0.0	0.0	7.9	27.1	35.0
PCEDA	0.25	34.6	1.5	40.9	0.6	1.6	2.2	9.6	11.1	56.4	0.5	43.8	12.7	2.0	28.0	7.0	3.7	0.0	1.0	5.0	13.8	30.8	44.6
BDL		25.4	4.7	55.1	2.8	1.5	1.3	9.1	4.3	61.3	1.5	54.1	26.7	0.1	20.7	6.5	1.5	0.0	0.7	1.0	14.6	26.7	41.3
Ours		**89.7**	**30.4**	**78.2**	**13.4**	**11.4**	**11.1**	**19.4**	**14.5**	**79.2**	**27.0**	**84.8**	**49.7**	**19.0**	**78.6**	**17.1**	**18.1**	**3.0**	**7.2**	**17.2**	**35.2**	**4.5**	39.7
FDA		22.0	0.4	3.2	0.0	1.3	0.1	1.9	0.6	33.8	1.1	22.6	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	4.6	37.6	42.4
AdaptSegNet		0.1	0.0	14.4	0.0	2.1	0.7	2.9	0.4	23.3	0.0	8.4	0.2	0.0	0.1	0.0	0.0	0.0	0.0	0.0	2.8	32.2	35.0
PCEDA	0.5	26.8	0.1	15.0	0.1	1.3	0.1	2.5	2.3	18.1	0.0	15.4	0.1	0.0	2.0	0.2	0.0	0.0	0.0	0.0	4.4	40.2	44.6
BDL		27.8	0.9	36.8	0.5	1.2	0.1	2.7	0.9	34.1	0.0	25.1	5.4	0.0	0.7	0.0	0.0	0.0	0.0	0.0	7.1	34.2	41.3
Ours		**75.7**	**11.7**	**66.1**	**2.7**	**6.0**	**3.7**	**13.6**	**8.6**	**66.8**	**14.0**	**79.1**	**37.2**	**4.0**	**59.0**	**7.2**	**9.6**	**0.4**	**0.1**	**6.0**	**24.8**	**14.9**	39.7
FDA [48]		85.8	27.8	70.2	8.6	7.4	17.9	30.7	23.4	70.8	22.4	59.7	53.8	26.5	71.6	29.2	26.8	**6.3**	23.1	**38.3**	36.9	13.5	50.4
FADA [41]		53.2	19.7	65.2	6.3	14.1	21.3	19.0	8.2	74.4	21.6	55.7	50.3	14.8	73.2	13.4	9.1	1.0	9.6	20.5	29.0	20.2	49.2
IntraDA [27]		89.1	31.1	76.6	11.3	16.4	14.9	25.3	15.8	80.8	29.4	74.9	54.3	23.3	78.7	32.1	**39.2**	0.0	21.5	30.8	39.2	7.1	46.3
CLAN [20]		75.8	21.3	69.8	11.9	7.3	12.7	24.6	8.8	77.1	20.4	66.9	51.0	19.6	65.4	28.7	31.3	2.5	15.2	24.8	33.4	9.8	43.2
MaxSquare [23]	0.1	28.6	9.3	52.0	3.9	3.1	9.7	29.1	10.3	73.6	10.2	41.7	46.1	19.1	36.1	26.5	10.7	0.2	17.2	28.0	24.0	22.4	46.4
AdaptSegNet [38]		80.9	21.2	66.3	7.4	5.7	7.4	25.2	6.5	76.2	12.5	69.9	45.6	11.7	71.3	21.8	8.0	1.6	6.5	14.3	29.5	12.9	42.4
PCEDA [47]		**89.8**	31.8	75.8	17.4	9.2	26.9	**31.1**	30.0	80.0	19.3	**85.6**	**55.2**	**27.5**	79.4	30.2	34.4	0.0	20.3	**38.3**	41.2	9.3	50.5
BDL [18]		75.5	31.3	75.3	8.8	8.5	17.1	29.3	23.0	76.9	22.4	80.5	51.2	25.8	51.9	24.0	33.3	1.6	20.3	31.3	36.2	12.3	48.5
Ours		89.3	**37.7**	**81.3**	**21.0**	**18.3**	**28.6**	29.0	**31.4**	**81.8**	**33.9**	82.2	51.9	25.9	**80.4**	**34.9**	31.3	0.0	**30.4**	33.1	**43.3**	**0.6**	43.9
FDA		50.8	6.7	51.0	1.6	3.7	3.5	17.2	6.3	49.5	1.5	60.9	28.3	12.8	49.1	14.5	4.6	1.2	2.6	25.0	20.6	29.8	50.4
FADA		54.1	14.8	50.4	2.2	8.2	6.8	4.7	0.9	59.4	7.4	32.8	29.9	3.0	53.6	4.1	0.3	1.2	0.7	5.9	17.9	31.3	49.2
IntraDA		26.4	3.0	46.3	0.4	4.5	0.7	8.6	0.5	30.9	0.4	43.9	21.3	1.2	47.5	8.33	7.5	0.0	0.2	6.5	13.6	32.7	46.3
CLAN		58.3	9.4	52.7	5.0	2.7	1.3	14.7	2.1	58.5	3.0	64.5	37.6	14.0	46.1	20.0	13.6	**1.8**	3.6	17.3	22.4	20.8	43.2
MaxSquare	0.25	15.2	2.3	37.9	2.7	1.5	1.0	15.8	1.8	54.1	1.5	30.6	14.3	7.2	31.5	11.8	1.6	0.0	0.7	13.8	12.9	33.5	46.4
AdaptSegNet		66.9	4.8	32.8	1.3	2.4	0.7	13.2	1.2	60.6	2.4	65.3	19.6	1.5	49.0	8.2	1.2	0.0	0.1	0.8	17.5	24.9	42.4
PCEDA		76.4	3.0	50.9	1.5	3.3	11.5	18.1	10.0	59.3	0.6	59.4	37.0	16.1	49.6	11.6	5.6	0.0	2.6	25.2	23.3	27.2	50.5
BDL		40.7	7.2	56.6	3.1	2.0	4.0	20.3	5.5	62.7	1.5	65.8	19.4	15.3	30.2	8.0	8.4	0.0	6.4	21.2	19.9	28.6	48.5
Ours		**87.9**	**26.6**	**75.0**	**11.1**	**12.5**	**24.4**	**26.0**	**28.3**	**74.2**	**19.5**	**81.8**	**48.7**	**22.9**	**78.5**	**31.8**	**34.2**	0.0	**27.2**	**30.2**	**39.0**	**4.9**	43.9
FDA		14.5	0.9	23.2	1.0	**5.3**	1.1	7.6	0.9	28.4	0.0	57.9	3.0	0.2	8.2	3.8	0.0	0.0	0.0	1.6	8.3	42.1	50.4
FADA		17.4	7.6	18.1	1.2	2.1	0.4	0.5	0.1	29.2	0.0	11.8	3.8	0.2	18.5	0.0	0.1	0.0	0.0	0.0	5.8	43.4	49.2
IntraDA		26.4	3.0	46.3	0.4	4.5	0.7	8.6	0.5	30.9	0.4	43.9	21.3	1.2	47.5	8.3	7.5	0.0	0.2	6.5	13.6	32.7	46.3
CLAN		33.0	0.6	39.2	2.3	1.8	0.1	8.4	0.2	36.2	0.3	38.1	21.5	3.4	38.0	9.4	3.4	0.0	0.1	4.3	12.6	30.6	43.2
MaxSquare	0.5	17.0	0.3	33.6	0.6	2.2	0.4	9.9	0.4	29.5	0.0	31.2	3.5	0.4	28.8	5.7	0.4	0.0	0.0	1.3	8.7	37.7	46.4
AdaptSegNet		43.0	0.2	10.1	0.7	2.8	0.2	7.3	0.1	34.8	0.0	58.1	4.9	0.0	18.6	0.8	0.3	0.0	0.0	0.0	9.6	32.8	42.4
PCEDA		30.4	0.0	36.6	0.2	1.7	1.5	4.0	1.2	27.1	0.0	8.1	9.7	0.4	7.4	1.2	0.0	0.0	0.0	5.3	7.1	43.4	50.5
BDL		9.7	0.1	25.9	0.0	0.8	0.2	8.1	0.6	43.5	0.0	13.7	4.8	4.3	7.6	2.6	0.0	0.0	0.2	1.9	6.5	42.0	48.5
Ours		**82.9**	**10.0**	**49.8**	**3.4**	4.5	**12.7**	**20.7**	**19.9**	**59.9**	**5.8**	**78.6**	**35.9**	**12.6**	**60.2**	**18.9**	**18.2**	0.0	**10.8**	**15.5**	**27.4**	**16.5**	43.9

Table 2. Quantitative study of "GTA5 to Cityscapes". VGG16 (upper part) and ResNet101 (lower part) are used as backbones in this experiment. The performance is measured on 19 common classes with criteria: per-class IoU, mean IoU (mIoU), mIoU drop (performance degradation of the model after being attacked), and mIoU*. The higher the mIoU and the lower the mIoU drop, the more robust the model is. The best result in each column is highlighted in bold.

Table 3). We then perform a comparison of our method on two widely used benchmark settings, *i.e.*, "GTA5 to Cityscapes" and "SYNTHIA to Cityscapes". Three criteria, *i.e.*, mIoU, mIoU drop, and mIoU* are used for performance assessment. Specifically, mIoU and mIoU* indicate the mean IoU on the perturbed test data and the clean test data, respectively, while mIoU drop indicates the performance degradation (*i.e.*, the difference between mIoU and mIoU*). Therefore, the higher the mIoU and the lower the mIoU drop, the more robust the model is.

GTA5 to Cityscapes As shown in Table 2, we achieve the best performance on all three adversarial attacks. In particular, even slight adversarial perturbations can mislead AdaptSegNet [38] and BDL [18] and dramatically degrade their

SYNTHIA to Cityscapes																				
	ϵ	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	sky	person	rider	car	bus	motorbike	bicycle	**mIoU**	**mIoU drop**	**mIoU***
FDA [48]	0.1	68.5	28.4	72.7	0.4	0.3	22.2	5.1	19.1	57.6	75.7	45.8	18.8	55.6	18.5	5.1	31.5	32.8	7.7	40.5
PCEDA [47]		80.9	25.0	**73.5**	**6.3**	**7.1**	14.2	**24.0**	**27.4**	76.2	70.3	45.0	19.9	70.0	20.3	**9.8**	25.1	37.2	3.9	41.1
BDL [18]		34.9	21.2	47.8	0.0	0.2	20.5	9.2	20.2	67.2	74.3	49.0	17.5	57.2	11.9	2.5	34.6	29.3	9.7	39.0
Ours		**88.2**	**46.5**	46.5	0.0	0.1	**24.6**	8.4	23.8	**79.3**	**81.2**	**54.4**	**24.5**	**78.2**	**22.4**	9.2	**44.4**	**41.3**	**-2.2**	39.1
FDA	0.25	46.3	16.0	38.7	0.0	0.2	4.9	2.5	8.9	31.3	38.9	8.6	5.3	17.7	6.0	1.3	5.4	14.5	26.0	40.5
PCEDA		75.6	11.4	59.1	0.0	**0.4**	9.6	5.5	12.9	63.1	45.0	30.7	13.4	34.9	8.6	2.5	24.5	24.8	16.3	41.1
BDL		8.0	8.9	31.1	0.0	0.1	8.7	6.9	9.8	52.0	54.1	22.9	4.9	25.6	2.5	0.8	13.3	13.6	25.4	39.0
Ours		**87.4**	**41.6**	**73.7**	0.0	0.1	**23.2**	**8.7**	**23.0**	**75.7**	**78.8**	**49.7**	**21.1**	**72.5**	**20.3**	**7.5**	**39.5**	**38.9**	**0.2**	39.1
FDA	0.5	42.2	4.9	14.2	0.0	0.1	0.6	1.0	1.7	26.2	1.9	0.5	0.4	1.5	0.1	0.1	0.1	6.0	34.5	40.5
PCEDA		66.2	1.1	47.9	0.0	**0.4**	3.1	2.5	5.0	47.8	18.8	10.0	1.9	8.3	3.2	1.1	10.2	14.2	26.9	41.1
BDL		0.6	1.0	24.8	0.0	0.0	1.6	1.9	2.3	35.8	18.6	2.2	0.1	4.1	0.1	0.0	0.5	5.9	33.1	39.0
Ours		**68.8**	**21.8**	**57.1**	0.0	0.1	**17.9**	**6.8**	**15.6**	**65.9**	**54.2**	**30.4**	**12.8**	**43.1**	**5.9**	**4.1**	**25.3**	**26.9**	**12.2**	39.1
FDA [48]	0.1	83.4	32.4	73.5	✗	✗	✗	13.1	**18.9**	71.6	79.5	**56.1**	24.9	77.5	27.6	18.2	**42.8**	47.7	4.8	52.5
FADA [41]		74.0	32.5	69.8	✗	✗	✗	6.8	15.8	57.0	58.3	46.7	8.6	55.1	18.0	4.5	9.8	35.1	17.4	52.5
DADA [40]		80.0	33.8	75.0	✗	✗	✗	8.0	9.4	62.1	76.3	49.7	14.3	76.3	27.8	5.2	31.7	42.3	7.5	49.8
MaxSquare [23]		70.1	23.3	72.8	✗	✗	✗	6.7	7.2	60.2	77.6	48.7	13.8	63.7	17.4	3.1	20.1	37.3	10.9	48.2
AdaptSegNet [38]		79.5	34.7	76.6	✗	✗	✗	4.1	5.4	61.0	80.8	49.3	18.3	72.1	26.1	7.5	29.8	41.9	4.8	46.7
PCEDA [47]		64.5	33.4	77.1	✗	✗	✗	**17.6**	16.5	50.1	81.3	48.9	24.8	71.9	25.7	13.3	41.0	43.6	10.0	53.6
BDL [18]		79.2	33.7	75.3	✗	✗	✗	5.6	8.7	61.1	80.6	45.0	21.7	65.7	26.7	8.5	24.5	41.2	10.2	51.4
Ours		**89.1**	**46.6**	**78.2**	✗	✗	✗	11.4	16.9	**76.1**	**81.5**	52.6	**26.7**	**79.9**	**35.3**	**25.0**	37.5	**50.5**	**-1.1**	49.4
FDA	0.25	8.6	9.0	40.8	✗	✗	✗	3.9	7.1	21.5	51.3	14.5	6.9	35.3	5.4	0.0	14.4	16.8	35.7	52.5
FADA		80.8	23.5	59.3	✗	✗	✗	1.7	3.7	50.6	15.6	26.2	0.8	21.2	6.2	0.3	2.1	22.5	30.0	52.5
DADA		58.0	11.5	42.7	✗	✗	✗	4.5	4.2	31.9	41.2	23.4	6.0	53.9	8.3	0.4	14.0	23.1	26.7	49.8
MaxSquare		70.3	4.6	53.1	✗	✗	✗	8.1	6.0	37.2	61.0	11.2	3.9	42.3	6.9	0.4	3.4	23.7	24.5	48.2
AdaptSegNet		28.4	7.6	56.8	✗	✗	✗	4.4	2.6	26.4	62.8	22.5	9.8	44.2	8.3	1.1	10.2	21.9	24.8	46.7
PCEDA		15.4	7.2	64.9	✗	✗	✗	9.3	9.8	27.0	71.4	35.3	13.9	52.0	12.3	2.2	25.4	26.7	26.9	53.6
BDL		46.9	9.1	65.5	✗	✗	✗	4.0	5.9	34.7	68.5	22.7	12.5	50.7	10.8	1.2	12.8	26.6	21.3	51.4
Ours		**87.4**	**25.0**	**70.7**	✗	✗	✗	**10.9**	**18.2**	**60.0**	**74.9**	**43.8**	**20.7**	**64.8**	**17.7**	**4.5**	**29.9**	**40.7**	**8.7**	49.4
FDA	0.5	0.0	0.0	7.2	✗	✗	✗	1.3	0.7	17.8	13.7	0.0	0.0	2.5	0.2	0.0	0.0	3.3	49.2	52.5
FADA		**76.0**	**15.9**	**56.3**	✗	✗	✗	0.2	0.6	**45.0**	0.2	7.6	0.0	5.2	0.9	0.0	0.1	16.0	36.5	52.5
DADA		42.9	2.3	16.3	✗	✗	✗	1.8	0.7	24.1	12.5	2.5	0.8	23.5	2.1	0.0	4.8	10.3	39.5	49.8
MaxSquare		42.7	0.2	25.3	✗	✗	✗	5.0	2.7	24.5	18.0	0.8	0.1	15.0	1.5	0.0	0.2	10.5	37.7	48.2
AdaptSegNet		2.1	0.4	24.5	✗	✗	✗	2.1	0.5	19.2	21.4	1.4	2.2	11.7	1.7	0.1	2.5	6.9	39.8	46.7
PCEDA		0.1	0.1	40.0	✗	✗	✗	2.4	1.8	21.0	37.2	**13.1**	1.3	9.3	2.5	0.7	1.6	10.1	43.5	53.6
BDL		2.8	0.7	32.1	✗	✗	✗	2.0	1.8	20.3	53.7	2.7	1.3	22.3	1.4	0.4	1.7	11.0	40.4	51.4
Ours		65.5	4.3	44.0	✗	✗	✗	**6.6**	**13.7**	31.9	**60.8**	12.6	**7.8**	**24.8**	**3.4**	**1.2**	**14.4**	**22.4**	**27.0**	49.4

Table 3. Quantitative study of "SYNTHIA to Cityscapes". VGG16 (upper part) and ResNet101 (lower part) are used as backbones in this experiment. The comparison is performed on 16 common classes for VGG16 and 13 common classes for ResNet101.

performance. For instance, when evaluated with VGG16 backbone on perturbed test data from $\epsilon = 0.25$, they only achieve mIoU 7.9 and mIoU 14.6, with mIoU drop 27.1 and 26.7, respectively. Similarly, two recently proposed UDA methods, *i.e.*, FDA [48] and PCEDA [47] suffer from mIoU drop of 31.4 and 30.8, respectively. By contrast, our method still gets mIoU 35.2 and only has a performance drop of mIoU 4.5. The results suggest that existing UDA methods in semantic segmentation are broadly vulnerable to adversarial attacks. The reason is that although these methods demonstrate remarkable performance on the clean test data (as indicated by mIoU*), none of them, however, take the adversarial attack into account during learning transferable representations. Instead, we innovatively propose adversarial self-supervision to improve the robustness of UDA models by taking advantage of both adversarial training and self-supervision. This is evidenced by the qualitative study in Figure 3, where our method demonstrates accurate predictions on the perturbed test data.

In terms of the clean performance (or mIoU*), our method usually lags behind the existing state of the arts. This is consistent with recent studies that clean performance and adversarial robustness might be at odds [39, 50].

SYNTHIA to Cityscapes Table 3 shows the performance comparison on "SYNTHIA to Cityscapes", where our method again demonstrates significant robustness improvement. In contrast, other UDA methods can be easily fooled by small perturbations in the test data. Interestingly, our

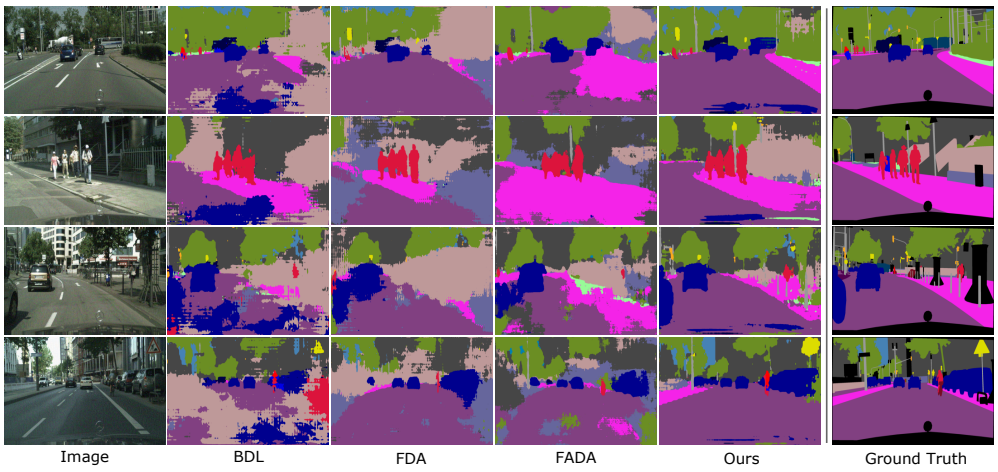

Figure 3. Qualitative comparison of our method against BDL [18], FDA [48], and FADA [41] on the perturbed test data ($\epsilon = 0.25$). All of these models are trained on "GTA5 to Cityscapes" with ResNet101. The first column indicates perturbed test images.

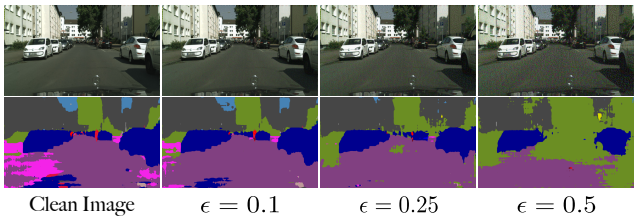

Figure 4. Qualitative study of our method under three adversarial attacks, *i.e.*, $\epsilon = 0.1$, $\epsilon = 0.25$, and $\epsilon = 0.5$. All of these models are trained on "SYNTHIA to Cityscapes" with ResNet101.

method achieves better performance on the perturbed test data ($\epsilon = 0.1$) than on the clean test data. This can be explained by the fact that training on adversarial examples can regularize the model somewhat, as reported in [11, 36]. We further perform a qualitative study of our method when evaluated on the test data with different magnitudes of the perturbation. As shown in Figure 4, although large ϵ usually results in worse performance, our method still demonstrates robust predictions.

Ablation Study To learn the contribution of the self-supervision, we conduct the ablation study in Table 4. Compared to $\delta = 0$ which only contains self-training, incorporating self-supervision consistently improves the performance. We further investigate the training perturbation magnitude ϵ_m in equation 5. Table 5 reveals that $\epsilon_m = 1.0$ (Ours) results in more robust UDA model than $\epsilon_m = 0.1$. The reason is that the adversarial examples generated by $\epsilon_m = 1.0$ are highly perturbed compared to the adversarial examples from $\epsilon_m = 0.1$, which in turn encourages our model to be more robust against perturbations.

5. Conclusion

In this paper, we introduce a new unsupervised domain adaptation framework for semantic segmentation. This is motivated by the observation that the robustness of semantic adaptation methods against adversarial attacks has not been investigated. Our pilot studies reveal that existing UDA methods can be easily deceived by unnoticeable perturbations. We therefore propose adversarial self-supervision by maximizing agreement between clean samples and their adversarial examples to improve model robustness. Extensive empirical studies are performed to explore the benefits of our method in improving the model robustness against adversarial attacks. The effectiveness of our method is thoroughly proved on commonly used benchmarks.

	GTA5 to Cityscapes		**SYNTHIA to Cityscapes**	
ϵ	$\delta = 0$	Ours	$\delta = 0$	Ours
0.1	39.2	39.3	41.5	41.3
0.25	33.8	35.2	36.7	38.9
0.5	21.8	24.8	23.6	26.9
0.1	43.3	43.3	49.7	50.5
0.25	37.8	39.0	37.8	40.7
0.5	24.3	27.4	15.7	22.4

Table 4. Ablation study of δ with backbone VGG16 (upper part) and ResNet101 (lower part).

	VGG16		**ResNet101**	
ϵ	$\epsilon_m = 0.1$	$\epsilon_m = 1.0$	$\epsilon_m = 0.1$	$\epsilon_m = 1.0$
0.1	36.4	39.3	44.9	43.3
0.25	17.8	35.2	34.3	39.0
0.5	7.4	24.8	15.7	27.4

Table 5. Ablation study of ϵ_m on "GTA5 to Cityscapes".

6. Acknowledgments

This work was partially supported by US National Science Foundation IIS-1718853, the CAREER grant IIS-1553687 and Cancer Prevention and Research Institute of Texas (CPRIT) award (RP190107).

References

[1] Cycada: Cycle consistent adversarial domain adaptation. In *International Conference on Machine Learning*, 2018. 1, 2, 3

[2] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *International conference on machine learning (ICML)*, 2012. 2

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2018. 5

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning (ICML)*, 2020. 1, 2, 4

[5] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 5

[7] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2015. 2

[8] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 2015. 1, 2

[9] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[10] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *International Conference on Learning Representations (ICLR)*, 2018. 1, 2

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)*, 2015. 2, 3, 4, 8

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1

[13] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 4

[14] Chih-Hui Ho and Nuno Vasconcelos. Contrastive learning with adversarial examples. *Advances in neural information processing systems (NeurIPS)*, 2020. 2

[15] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 1, 2, 3

[16] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. In *NeurIPS*, 2020. 2

[17] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[18] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 4, 6, 7, 8

[19] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1

[20] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations (ICLR)*, 2018. 3

[22] Shike Mei and Xiaojin Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015. 2

[23] Hongyang Xue Minghao Chen and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2019. 6, 7

[24] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016. 3

[25] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[26] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[27] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[28] Ashwin Raju, Chi-Tung Cheng, Yunakai Huo, Jinzheng Cai, Junzhou Huang, Jing Xiao, Le Lu, ChienHuang Liao, and Adam P Harrison. Co-heterogeneous and adaptive segmentation from multi-source and multi-phase ct imaging data: A

study on pathological liver and lesion segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

[29] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 1, 5

[30] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 5

[31] Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. *Proceedings of the European Conference on Computer Vision*, 2020. 1, 4

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 5

[33] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430*, 2018. 1

[34] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019. 3

[35] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019. 2

[36] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations (ICLR)*, 2014. 2, 8

[37] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017. 2

[38] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3, 6, 7

[39] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *International Conference on Learning Representations (ICLR)*, 2019. 7

[40] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2019. 7

[41] Haoran Wang, Tong Shen, Wei Zhang, Lingyu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 6, 7, 8

[42] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-mei Hwu, Thomas S Huang, and Honghui Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[43] Jiaolong Xu, Liang Xiao, and Antonio M López. Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 2019. 2

[44] Jinyu Yang, Weizhi An, Sheng Wang, Xinliang Zhu, Chaochao Yan, and Junzhou Huang. Label-driven reconstruction for domain adaptation in semantic segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 4

[45] Jinyu Yang, Weizhi An, Chaochao Yan, Peilin Zhao, and Junzhou Huang. Context-aware domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021. 4

[46] Jihan Yang, Ruijia Xu, Ruiyu Li, Xiaojuan Qi, Xiaoyong Shen, Guanbin Li, and Liang Lin. An adversarial perturbation oriented domain adaptation approach for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2

[47] Yanchao Yang, Dong Lao, Ganesh Sundaramoorthi, and Stefano Soatto. Phase consistent ecological domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6, 7

[48] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 4, 6, 7, 8

[49] Xiaohang Zhan, Ziwei Liu, Ping Luo, Xiaoou Tang, and Chen Change Loy. Mix-and-match tuning for self-supervised semantic segmentation. *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2

[50] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*. PMLR, 2019. 7

[51] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[52] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 5

[53] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[54] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2018. 1, 2