Classes Matter: A Fine-grained Adversarial Approach to Cross-domain Semantic Segmentation

Haoran Wang[1]*, Tong Shen[2]*, Wei Zhang[2], Ling-Yu Duan[3], and Tao Mei[2]

[1]ETH Zürich   [2]JD AI Research   [3]Peking University

Abstract. Despite great progress in supervised semantic segmentation, a large performance drop is usually observed when deploying the model in the wild. Domain adaptation methods tackle the issue by aligning the source domain and the target domain. However, most existing methods attempt to perform the alignment from a holistic view, ignoring the underlying class-level data structure in the target domain. To fully exploit the supervision in the source domain, we propose a fine-grained adversarial learning strategy for class-level feature alignment while preserving the internal structure of semantics across domains. We adopt a fine-grained domain discriminator that not only plays as a domain distinguisher, but also differentiates domains at class level. The traditional binary domain labels are also generalized to domain encodings as the supervision signal to guide the fine-grained feature alignment. An analysis with Class Center Distance (CCD) validates that our fine-grained adversarial strategy achieves better class-level alignment compared to other state-of-the-art methods. Our method is easy to implement and its effectiveness is evaluated on three classical domain adaptation tasks, i.e., GTA5→Cityscapes, SYNTHIA→Cityscapes and Cityscapes→Cross-City. Large performance gains show that our method outperforms other global feature alignment based and class-wise alignment based counterparts. The code is publicly available at https://github.com/JDAI-CV/FADA.

1   Introduction

The success of semantic segmentation [26] in recent years is mostly driven by a large amount of accessible labeled data. However, collecting massive densely annotated data for training is usually a labor-intensive task [9]. Recent advances in computer graphics provide an alternative for replacing expensive human labor. Through physically based rendering, we can obtain photo-realistic images with the pixel-level ground-truth readily available in an effortless way [23,24].

However, performance drop is observed when the model trained with synthetic data (a source domain) is applied in realistic scenarios (a target domain), because the data from different domains usually share different distributions.

---

* These authors contributed equally. This work was performed when Haoran Wang was visiting JD AI research as a research intern.
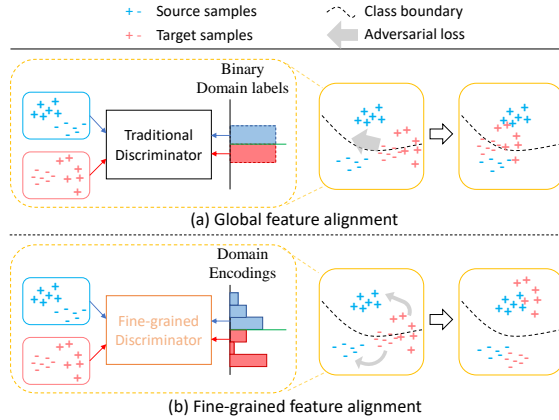
Fig. 1: Illustration of traditional and our fine-grained adversarial learning. Traditional adversarial learning pursues the marginal distribution alignment while ignoring the semantic structure inconsistency between domains. We propose to use a fine-grained discriminator to enable class-level alignment.

This phenomenon is known as domain shift problem [27], which poses a challenge to cross-domain tasks [16].

Domain adaptation aims to alleviate the domain shift problem by aligning the feature distributions of the source and the target domain. A group of works focus on adopting an adversarial framework, where a domain discriminator is trained to distinguish the target samples from the source ones, while the feature network tries to fool the discriminator by generating domain-invariant features [8, 15, 16, 20, 25, 30, 34, 35, 38].

Although impressive progress has been achieved in domain adaptive semantic segmentation, most of prior works strive to align global feature distributions without paying much attention to the underlying structures among classes. However, as discussed in recent works [3, 17], matching the marginal feature distributions does not guarantee small expected error on the target domain. The class conditional distributions should also be aligned, meaning that class-level alignment also plays an important role. As illustrated in Figure 1, the upper part shows the result of global feature alignment where the two domains are well-aligned but some samples are falsely mixed up. This motivates us to incorporate class information into the adversarial framework to enable fine-grained feature alignment. As illustrated in the bottom of Figure 1, features are expected to be aligned according to specific classes.

There have been some pioneering works [7, 20] trying to address this problem. Chen et al. [7] propose to use several independent discriminators to perform class-wise alignment, but independent discriminators might fail to capture the

relationships between classes. Luo et al. [20] introduce an self-adaptive adversarial loss to apply different weights to each region. However, in fact, they do not explicitly incorporate class information in their methods, which might fail to promote class-level alignment.

Our motivation is to directly incorporate class information into the discriminator and encourage it to align features at a fine-grained level. Traditional adversarial training has been proven effective for aligning features by using a binary domain discriminator to model distribution $P(d|f)$ (d refers to domain and f is the feature extracted from input data). By confusing such a discriminator, expecting $P(d=0|f) \approx P(d=1|f)$ where 0 stands for the source domain and 1 for the target domain, the features become domain invariant and well aligned. To further take classes into account, we split the output into multiple channels according to $P(d|f) = \sum_{c=1}^{K} P(d,c|f)$ (where c refers to classes $\{1, \ldots, K\}$). We directly model the discriminator as $P(d,c|f)$ to formulate a fine-grained domain alignment task. Although in the setting of domain adaptation the category-level labels for target domain are inaccessible, we find that the model predictions on target domain also contain class information and prove that it is possible to supervise the discriminator with the predictions on both domains. In the adversarial learning process, class information is incorporated and the features are expected to be aligned according to specific classes.

In this paper, we propose such a fine-grained adversarial learning framework for domain adaptive semantic segmentation (FADA). As illustrated in Figure 1, we represent the supervision of traditional discriminator at a fine-grained semantic level, which enables our fine-grained discriminator to capture rich class-level information. The adversarial learning process is performed at fine-grained level, so the features are expected to be adaptively aligned according to their corresponding semantic categories. The class mismatch problem, which broadly exists in the global feature alignment, is expected to be further suppressed. Correspondingly, by incorporating class information, the binary domain labels are also generalized to a more complex form, called "domain encodings" to serve as the new supervision signal. Domain encodings could be extracted from the network's predictions on both domains. Different strategies of constructing domain encodings will be discussed. We conduct an analysis with Class Center Distance to demonstrate the effectiveness of our method regarding class-level alignment. Our method is also evaluated on three popular cross-domain benchmarks and presents new state-of-the-art results.

The main contributions of this paper are summarized below.

- We propose a fine-grained adversarial learning framework for cross-domain semantic segmentation that explicitly incorporates class-level information.
- The fine-grained learning framework enables class-level feature alignment, which is further verified by analysis using Class Center Distance.
- We evaluate our methods with comprehensive experiments. Significant improvements compared to other state-of-the-art methods are achieved on popular domain adaptive segmentation tasks including GTA5 $\rightarrow$ Cityscapes, SYNTHIA $\rightarrow$ Cityscapes and Cityscapes $\rightarrow$ Cross-City.

2   Related Work

2.1   Semantic Segmentation

Semantic segmentation is a task of predicting unique semantic label for each pixel of the input image. With the advent of deep convolutional neural networks, the academia of computer vision witnesses a huge progress in this field. FCN [26] triggered the interests in introducing deep learning for this task. Many follow-up methods are proposed to enlarge the receptive fields to cover more context information [4–6,36]. Among all these works, the family of Deeplab [4–6] attracts a lot of attention and has been widely applied in many works for their simplicity and effectiveness.

2.2   Domain Adaptation

Domain adaptation strives to address the performance drop caused by the different distributions of training data and testing data. In the recent years, several works are proposed to approach this problem in image classification [3,25]. Inspired by the theoretical upper bound of risk in target domain [2], some pioneering works suggest to optimize some distance measurements between the two domains to align the features [18,29]. Recently, motivated by GAN [13], adversarial training becomes popular for its power to align features globally [7,25,30].

2.3   Domain Adaptive Semantic Segmentation

Unlike domain adaptation for image classification task, domain adaptive semantic segmentation receives less attention for its difficulty even though it supports many important applications including autonomous driving in the wild [8,16]. Based on the theoretical insight [2] on domain adaptive classification, most works follow the path of shortening the domain discrepancy between the two domains. Large progress is achieved through optimization by adversarial training or explicit domain discrepancy measures [15,16,30]. In the context of domain adaptive semantic segmentation task, AdaptSegnet [30] attempts to align the distribution in the output space. Inspired by CycleGAN [37], CyCADA [15] suggests to adapt the representation in pixel-level and feature-level. There are also many works focusing on aligning different properties between two domains such as entropy [32] and information [19].

Although huge progress has been made in this field, most of existing methods share a common limitation: Enforcing global feature alignment would inevitably mix samples with different semantic labels together when drawing two domains closer, which usually results in a mismatch of classes from different domains. CLAN [20] is a pioneer work to address category-level alignment. It suggests applying different adversarial weight to different regions, but it does not directly and explicitly incorporate the classes into the model.
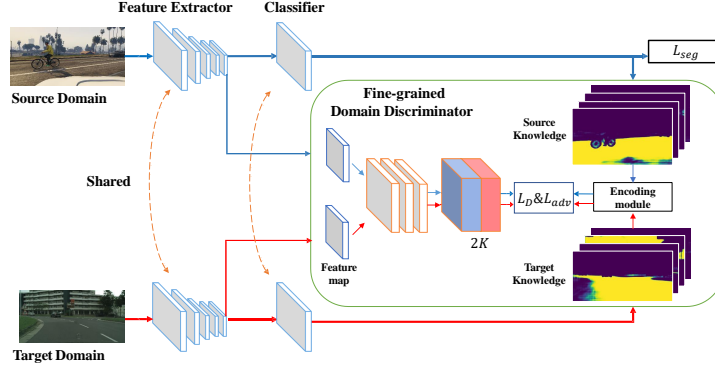
Fig. 2: Overview of the proposed fine-grained adversarial framework. Images from the source domain and target domain are randomly picked and fed to the feature extractor and the classifier. A segmentation loss is computed with the source predictions and the source annotations to help the segmentation network to generate discriminative features and learn task specific knowledge. The semantic features from both domains are fed to the convolutional fine-grained domain discriminator. The discriminator strives to distinguish the feature's domain information at a fine-grained class level using the domain encodings processed from the sample predictions.

3 Method

3.1 Revisit Traditional Feature Alignment

Semantic segmentation aims to predict per-pixel unique label for the input image [26]. In an unsupervised domain adaptation setting for semantic segmentation, we have access to a collection of labeled data $X_{\mathcal{S}} = \{(x_i^{(s)}, y_i^{(s)})\}_{i=1}^{n_s}$ in a source domain $\mathcal{S}$, and unlabeled data $X_{\mathcal{T}} = \{x_j^{(t)}\}_{j=1}^{n_t}$ in a target domain $\mathcal{T}$ where n_s and n_t are the numbers of samples from different domains. Domain $\mathcal{S}$ and domain $\mathcal{T}$ share the same K semantic class labels $\{1, \ldots, K\}$. The goal is to learn a segmentation model G which could achieve a low expected risk on the target domain. Generally, segmentation network G could be divided into a feature extractor F and a multi-class classifier C, where $G = C \circ F$.

Traditional feature-level adversarial training relies on a binary domain discriminator D to align the features extracted by F on both domains. Domain adaptation is tackled by alternatively optimizing G and D with two steps:

(1) D is trained to distinguish features from different domains. This process is usually achieved by fixing F and C and solving:

$$\min_D \mathcal{L}_D = -\sum_{i=1}^{n_s}(1-d)\log P(d=0|f_i) - \sum_{j=1}^{n_t} d \log P(d=1|f_j) \qquad (1)$$

where f_i and f_j are the features extracted by F on source sample $x_i^{(s)}$ and target sample $x_j^{(t)}$; d refers to the domain variable where 0 refers to the source domain and 1 refers to the target domain. $P(d|f)$ is the probability output from the discriminator.

(2) G is trained with the task loss $\mathcal{L}_{seg}$ on the source domain and the adversarial loss $\mathcal{L}_{adv}$ on the target domain. This process requires fixing D and updating F and C:

$$\min_{F,C} \mathcal{L}_{seg} + \lambda_{adv}\mathcal{L}_{adv} \tag{2}$$

The cross-entropy loss $\mathcal{L}_{seg}$ on source domain minimizes the difference between the prediction and the ground truth, which helps G to learn the task specific knowledge.

$$\mathcal{L}_{seg} = -\sum_{i=1}^{n_s}\sum_{k=1}^{K} y_{ik}^{(s)} \log p_{ik}^{(s)}, \tag{3}$$

where $p_{ik}^{(s)}$ is the probability confidence of source sample $x_i^{(s)}$ belonging to semantic class k predicted by C, $y_{ik}^{(s)}$ is the entry for the one-hot label.

The adversarial loss $\mathcal{L}_{adv}$ is used to confuse the discriminator to encourage F to generate domain invariant features.

$$\mathcal{L}_{adv} = -\sum_{j=1}^{n_t} \log P(d = 0|f_j) \tag{4}$$

3.2   Fine-grained Adversarial Learning

To incorporate the class information into the adversarial learning framework, we propose a novel discriminator and enable a fine-grained adversarial learning process. The whole pipeline is illustrated in Figure 2.

The traditional adversarial training strives to align the marginal distribution by confusing a binary discriminator. To make the discriminator not merely focus on distinguishing domains, we split each of the two output channels of the binary discriminator into K channels and encourage a fine-grained level adversarial learning. With this design, the predicted confidence for domains is represented as a confidence distribution over different classes, which enables the new fine-grained discriminator to model more complex underlying structures between classes, thus encouraging class-level alignment.

Correspondingly, the binary domain labels are also converted to a general form, namely domain encodings, to incorporate class information. Traditionally, the domain labels used for training the binary discriminator are $[1, 0]$ and $[0, 1]$ for the source and target domains respectively. The domain encodings are represented as a vector $[\mathbf{a}; \mathbf{0}]$ and $[\mathbf{0}; \mathbf{a}]$ for the two domains respectively, where $\mathbf{a}$ is the knowledge extracted from the classifier C represented by a K-dimensional
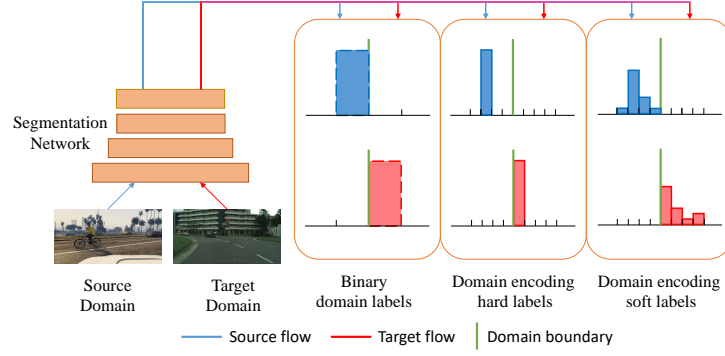
Fig. 3: Illustration of different strategies to generate domain encodings. Here we compare three different strategies to extract knowledge from segmentation network for constructing domain encodings: binary domain labels, one-hot hard labels and multi channel soft labels.

vector; $\mathbf{0}$ is an all-zero K-dimensional vector. The choices of how to generate domain knowledge $\mathbf{a}$ will be discussed in Section 3.3.

During the training process, the discriminator not only tries to distinguish domains, but also learns to model class structures. The $\mathcal{L}_D$ in Equation 1 becomes:

$$
\begin{aligned}
\mathcal{L}_D = & -\sum_{i=1}^{n_s}\sum_{k=1}^{K} a_{ik}^{(s)} \log P(d=0, c=k|f_i) \\
& -\sum_{j=1}^{n_t}\sum_{k=1}^{K} a_{jk}^{(t)} \log P(d=1, c=k|f_j)
\end{aligned}
\tag{5}
$$

where $a_{ik}^{(s)}$ and $a_{jk}^{(t)}$ are the kth entries of the class knowledge for the source sample i and target sample j. The adversarial loss $\mathcal{L}_{adv}$ used to confuse the discriminator and guide the generation of domain-invariant features in Equation 4 becomes:

$$
\mathcal{L}_{adv} = -\sum_{j=1}^{n_t}\sum_{k=1}^{K} a_{jk}^{(t)} \log P(d=0, c=k|f_j),
\tag{6}
$$

$\mathcal{L}_{adv}$ is designed to maximize the probability of features from target domain being considered as the source features without hurting the relationship between features and classes.

The overall network in Figure 2 is used in the training stage. During inference, the domain adaptation component is removed and one only needs to use the original segmentation network with the adapted weights.

3.3   Extracting class knowledge for domain encodings

Now that we have a fine-grained domain discriminator, which could adaptively align features according to the class-level information contained in domain encodings, another challenge raises: how to get the class knowledge $a_{ik}^{(s)}$ and $a_{ik}^{(t)}$ in Equations 5 and 6 to construct domain encoding for each sample? Considering that in the unsupervised domain adaptive semantic segmentation task none of annotations in target domain is accessible, it seems contradictory to use the class knowledge on the target domain for guiding class-level alignment. However, during training, with ground-truth annotations from the source domain, the classifier C learns to map features into the semantic classes. Considering that the source domain and the target domain share the same semantic classes, it would be a natural choice to use the predictions of C as knowledge to supervise the discriminator.

As illustrated in equations 5 and 6, the class knowledge for optimizing the fine-grained discriminator works as the supervision signal. The choices of $a_{ik}^{(s)}$ and $a_{jk}^{(t)}$ are open to many possibilities. For specific tasks, people could design different forms to produce class knowledge with prior knowledge. Here we discuss two general solutions to extract class knowledge from network predictions for constructing domain encodings. Because the class-level knowledge for different domains could be extracted in the same way, in the following discussion we would use a_k to represent kth entry for a single sample without differentiating the domain.

The one-hot hard labels could be a straightforward solution for generating knowledge, which could be denoted as:

$$a_k = \begin{cases} 1 \text{ if } k = \arg\max_k p_k \\ 0 \text{ otherwise} \end{cases} \tag{7}$$

where p_k is the softmax probability output of C for class k. In this way, only the most confident class is selected. In practice, in order to remove the impact of noisy samples, we can select samples whose confidence is higher than a certain threshold and ignore those with low confidence.

Another alternative is multi-channel soft labels, which has the following definition:

$$a_k = \frac{\exp\left(\frac{z_k}{T}\right)}{\sum_{j=1}^{K} \exp\left(\frac{z_j}{T}\right)} \tag{8}$$

where z_k is kth entry of logits and T is a temperature to encourage soft probability distribution over classes. Note that during training, an additional regularization could also be applied. For example, we practically find that clipping the values of the soft labels by a given threshold achieves more stable performance because it prevents from overfitting to certain classes.

An illustrative comparison of these two strategies with the traditional binary domain labels is presented in Figure 3. We also conduct experiments in section 4.6 to demonstrate the performance of different strategies.

Table 1: Experimental results for Cityscapes → Cross-City.

City	Method	road	sidewalk	building	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU
		Cityscapes → Cross-City													
Rome	Source Dilation-Frontend	77.7	21.9	83.5	0.1	10.7	78.9	88.1	21.6	10.0	67.2	30.4	6.1	0.6	38.2
	Cross-City [7]	79.5	29.3	84.5	0.0	22.2	80.6	82.8	29.5	13.0	71.7	37.5	25.9	1.0	42.9
	Source DeepLab-v2	83.9	34.3	87.7	13.0	41.9	84.6	92.5	37.7	22.4	80.8	38.1	39.1	5.3	50.9
	AdaptSegNet [30]	83.9	34.2	**88.3**	18.8	**40.2**	**86.2**	**93.1**	47.8	21.7	80.9	**47.8**	48.3	8.6	53.8
	FADA	**84.9**	**35.8**	**88.3**	**20.5**	40.1	85.9	92.8	**56.2**	**23.2**	**83.6**	31.8	**53.2**	**14.6**	**54.7**
Rio	Source Dilation-Frontend	69.0	31.8	77.0	4.7	3.7	71.8	80.8	38.2	8.0	61.2	38.9	11.5	3.4	38.5
	Cross-City [7]	74.2	43.9	79.0	2.4	7.5	77.8	69.5	39.3	10.3	67.9	**41.2**	27.9	10.9	42.5
	Source DeepLab-v2	76.6	47.3	82.5	**12.6**	22.5	77.9	86.5	43.0	19.8	74.5	36.8	29.4	16.7	48.2
	AdaptSegNet [30]	76.2	44.7	**84.6**	9.3	**25.5**	**81.8**	87.3	55.3	**32.7**	74.3	28.9	43.0	27.6	51.6
	FADA	**80.6**	**53.4**	84.2	5.8	23.0	78.4	**87.7**	**60.2**	26.4	**77.1**	37.6	**53.7**	**42.3**	**54.7**
Tokyo	Source Dilation-Frontend	81.2	26.7	71.7	8.7	5.6	73.2	75.7	39.3	14.9	57.6	19.0	1.6	33.8	39.2
	Cross-City [7]	83.4	35.4	72.8	12.3	12.7	77.4	64.3	42.7	21.5	64.1	**20.8**	8.9	40.3	42.8
	Source DeepLab-v2	83.4	35.4	72.8	12.3	12.7	77.4	64.3	42.7	21.5	64.1	**20.8**	8.9	40.3	42.8
	AdaptSegNet [30]	81.5	26.0	77.8	**17.8**	**26.8**	82.7	90.9	55.8	**38.0**	**72.1**	4.2	24.5	50.8	49.9
	FADA	**85.8**	**39.5**	**76.0**	14.7	24.9	**84.6**	**91.7**	**62.2**	27.7	71.4	3.0	**29.3**	**56.3**	**51.3**
Taipei	Source Dilation-Frontend	77.2	20.9	76.0	5.9	4.3	60.3	81.4	10.9	11.0	54.9	32.6	15.3	5.2	35.1
	Cross-City [7]	78.6	28.6	80.0	13.1	7.6	68.2	82.1	16.8	9.4	60.4	34.0	26.5	9.9	39.6
	Source DeepLab-V2	78.6	28.6	80.0	13.1	7.6	68.2	82.1	16.8	9.4	60.4	34.0	26.5	9.9	39.6
	AdaptSegNet [30]	81.7	29.5	85.2	**26.4**	15.6	76.7	91.7	31.0	12.5	71.5	**41.1**	47.3	27.7	49.1
	FADA	**86.0**	**42.3**	**86.1**	6.2	**20.5**	**78.3**	**92.7**	47.2	**17.7**	**72.2**	37.2	**54.3**	**44.0**	**52.7**

4 Experiments

4.1 Datasets

We present a comprehensive evaluation of our proposed method on three popular unsupervised domain adaptive semantic segmentation benchmarks, e.g., Cityscapes → Cross-City, SYNTHIA → Cityscapes, and GTA5 → Cityscapes.

Cityscapes   Cityscapes [9] is a real-world urban scene dataset consisting of a training set with 2,975 images, a validation set with 500 images and a testing set with 1,525 images. Following the standard protocols [15,16,30], we use the 2,975 images from Cityscapes training set as the unlabeled target domain training set and evaluate our adapted model on the 500 images from the validation set.

Cross-City   Cross-City [7] is an urban scene dataset collected with Google Street View. It contains 3,200 unlabeled images and 100 annotated images of four different cities respectively. The annotations of Cross-City share 13 classes with Cityscapes.

SYNTHIA   SYNTHIA [24] is a synthetic urban scene dataset. We pick its subset SYNTHIA-RAND-CITYSCAPES, which shares 16 semantic classes with Cityscapes, as the source domain. In total, 9,400 images from SYNTHIA dataset are used as source domain training data for the task.

GTA5   GTA5 dataset [23] is another synthetic dataset sharing 19 semantic classes with Cityscapes. 24,966 urban scene images are collected from a physically-based rendered video game Grand Theft Auto V (GTAV) and are used as source training data.

Table 2: Experimental results for SYNTHIA → Cityscapes.

		SYNTHIA → Cityscapes																
Backbone	Method	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Sky	PR	Rider	Car	Bus	Motor	Bike	mIoU	mIoU*
VGG-16	FCNs in the wild [16]	11.5	19.6	30.8	4.4	0.0	20.3	0.1	11.7	42.3	68.7	51.2	3.8	54.0	3.2	0.2	0.6	20.2	22.9
	CDA [34]	65.2	26.1	74.9	0.1	**0.5**	10.7	3.5	3.0	76.1	70.6	47.1	8.2	43.2	20.7	0.7	13.1	29.0	34.8
	ST [38]	0.2	14.5	53.8	1.6	0.0	18.9	0.9	7.8	72.2	80.3	48.1	6.3	67.7	4.7	0.2	4.5	23.9	27.8
	CBST [38]	69.6	28.7	69.5	**12.1**	0.1	25.4	**11.9**	13.6	**82.0**	81.9	**49.1**	14.5	66.0	6.6	3.7	**32.4**	35.4	36.1
	AdaptSegNet [30]	78.9	29.2	75.5	-	-	-	0.1	4.8	72.6	76.7	43.4	8.8	71.1	16.0	3.6	8.4	-	37.6
	SIBAN [19]	70.1	25.7	80.9	-	-	-	3.8	7.2	72.3	80.5	43.3	5.0	73.3	16.0	1.7	3.6	-	37.2
	CLAN [20]	80.4	30.7	74.7	-	-	-	1.4	8.0	77.1	79.0	46.5	8.9	73.8	18.2	2.2	9.9	-	39.3
	AdaptPatch [31]	72.6	29.5	77.2	3.5	0.4	21.0	1.4	7.9	73.3	79.0	45.7	14.5	69.4	19.6	7.4	16.5	33.7	39.6
	ADVENT [32]	67.9	29.4	71.9	6.3	0.3	19.9	0.6	2.6	74.9	74.9	35.4	9.6	67.8	21.4	4.1	15.5	31.4	36.6
	Source only	10.0	14.7	52.4	4.2	0.1	20.9	3.5	6.5	74.3	77.5	44.9	4.9	64.0	21.6	4.2	6.4	25.6	29.6
	Baseline (feat. only) [30]	63.6	26.8	67.3	3.8	0.3	21.5	1.0	7.4	76.1	76.5	40.5	11.2	62.1	19.4	5.3	13.2	31.0	36.2
	FADA	**80.4**	**35.9**	**80.9**	2.5	0.3	**30.4**	7.9	**22.3**	81.8	**83.6**	48.9	**16.8**	**77.7**	**31.1**	**13.5**	17.9	**39.5**	**46.0**
ResNet-101	SIBAN [19]	82.5	24.0	79.4	-	-	-	16.5	12.7	79.2	82.8	**58.3**	18.0	79.3	25.3	17.6	25.9	-	46.3
	AdaptSegNet [30]	84.3	42.7	77.5	-	-	-	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	-	46.7
	CLAN [20]	81.3	37.0	80.1	-	-	-	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	-	47.8
	AdaptPatch [31]	82.4	38.0	78.6	8.7	**0.6**	26.0	3.9	11.1	75.5	**84.6**	53.5	21.6	71.4	32.6	19.3	31.7	40.0	46.5
	ADVENT [32]	**85.6**	**42.2**	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	**23.8**	73.3	36.4	14.2	**33.0**	41.2	48.0
	Source only	55.6	23.8	74.6	9.2	0.2	24.4	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	33.5	38.6
	Baseline (feat. only) [30]	62.4	21.9	76.3	**11.5**	0.1	24.9	11.7	11.4	75.3	80.9	53.7	18.5	59.7	13.7	20.6	24.0	35.4	40.8
	FADA	84.5	40.1	**83.1**	4.8	0.0	**34.3**	**20.1**	**27.2**	84.8	84.0	53.5	22.6	**85.4**	**43.7**	**26.8**	27.8	**45.2**	**52.5**

4.2 Evaluation Metrics

The metrics for evaluating our algorithm is consistent with the common semantic segmentation task. Specifically, we compute PSACAL VOC intersection-over-union (**IoU**) [11] of our prediction and the ground truth label. We have $\mathbf{IoU} = \frac{\mathsf{TP}}{\mathsf{TP+FP+FN}}$, where TP, FP and FN are the numbers of true positive, false positive and false negative pixels respectively. In addition to the **IoU** for each class, a **mIoU** is also reported as the mean of **IoU**s over all classes.

4.3 Implementation Details

Our pipeline is implemented by PyTorch [22]. For fair comparison, we employ DeeplabV2 [4] with VGG-16 [28] and ResNet-101 [14] as the segmentation base networks. All models are pre-trained on ImageNet [10]. For the fine-grained discriminator, we adopt a simple structure consisting of 3 convolution layers with channel numbers $\{256, 128, 2K\}$, 3×3 kernels, and stride of 1. Each convolution layer is followed by a Leaky-ReLU [21] parameterized by 0.2 except for the last layer.

To train the segmentation network, we use the Stochastic Gradient Descent (SGD) optimizer where the momentum is 0.9 and the weight decay is 10^{-4}. The learning rate is initially set to 2.5×10^{-4} and is decreased following a 'poly' learning rate policy with power of 0.9. For training the discriminator, we adopt the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and the initial learning rate as 10^{-4}. The same 'poly' learning rate policy is used. λ_{adv} is constantly set to 0.001. Temperature T is set as 1.8 for all experiments.

Table 3: Experimental results for GTA5 → Cityscapes.

		GTA5 → Cityscapes																			
Backbone	Method	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Terrain	Sky	PR	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
VGG-16	FCNs in the wild [16]	70.4	32.4	62.1	14.9	5.4	10.9	14.2	2.7	79.2	21.3	64.6	44.1	4.2	70.4	8.0	7.3	0.0	3.5	0.0	27.1
	CDA [34]	74.9	22.0	71.7	6.0	11.9	8.4	16.3	11.1	75.7	13.3	66.5	38.0	9.3	55.2	18.8	18.9	0.0	16.8	14.6	28.9
	ST [38]	83.8	17.4	72.1	14.6	2.9	16.5	16.0	6.8	81.4	24.2	47.2	40.7	7.6	71.7	10.2	7.6	0.5	11.1	0.9	28.1
	CBST [38]	90.4	50.8	72.0	18.3	9.5	27.2	**28.6**	14.1	82.4	25.1	70.8	42.6	14.5	76.9	5.9	12.5	1.2	14.0	**28.6**	36.1
	CyCADA [15]	85.2	37.2	76.5	21.8	15.0	23.8	22.9	21.5	80.5	31.3	60.7	50.5	9.0	76.9	17.1	28.2	4.5	9.8	0.0	35.4
	AdaptSegNet [30]	87.3	29.8	78.6	21.1	18.2	22.5	21.5	11.0	79.7	29.6	71.3	46.8	6.5	80.1	23.0	26.9	0.0	10.6	0.3	35.0
	SIBAN [19]	83.4	13.0	77.8	20.4	17.5	24.6	22.8	9.6	81.3	29.6	77.3	42.7	10.9	76.0	22.8	17.9	5.7	14.2	2.0	34.2
	CLAN [20]	88.0	30.6	79.2	23.4	20.5	26.1	23.0	14.8	81.6	**34.5**	72.0	45.8	7.9	80.5	**26.6**	29.9	0.0	10.7	0.0	36.6
	AdaptPatch [31]	87.3	35.7	79.5	32.0	14.5	21.5	24.8	13.7	80.4	32.0	70.5	50.5	16.9	81.0	20.8	28.1	4.1	15.5	4.1	37.5
	ADVENT [32]	86.9	28.7	78.7	28.5	25.2	17.1	20.3	10.9	80.0	26.4	70.2	47.1	8.4	81.5	26.0	17.2	**18.9**	11.7	1.6	36.1
	Source only	35.4	13.2	72.1	16.7	11.6	20.7	22.5	13.1	76.0	7.6	66.1	41.1	19.0	69.8	15.2	16.3	0.0	16.2	4.7	28.3
	Baseline (feat. only) [30]	85.7	22.8	77.6	24.8	10.6	22.2	19.7	10.8	79.7	27.8	64.8	41.5	18.4	79.7	19.9	21.8	0.5	16.2	4.2	34.1
	FADA	**92.3**	**51.1**	**83.7**	**33.1**	**29.1**	**28.5**	28.0	**21.0**	**82.6**	32.6	**85.3**	**55.2**	**28.8**	**83.5**	24.4	**37.4**	0.0	**21.1**	15.2	**43.8**
ResNet-101	AdaptSegNet [30]	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
	SIBAN [19]	88.5	35.4	79.5	26.3	24.3	28.5	32.5	18.3	81.2	**40.0**	76.5	58.1	25.8	82.6	30.3	34.4	3.4	21.6	21.5	42.6
	CLAN [20]	87.0	27.1	79.6	27.3	23.3	28.3	**35.5**	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
	AdaptPatch [31]	92.3	**51.9**	82.1	29.2	25.1	24.5	33.8	**33.0**	82.4	32.8	82.2	58.6	27.2	84.3	33.4	46.3	2.2	29.5	32.3	46.5
	ADVENT [32]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	**38.5**	44.5	1.7	31.6	32.4	45.5
	Source only	65.0	16.1	68.7	18.6	16.8	21.3	31.4	11.2	83.0	22.0	78.0	54.4	33.8	73.9	12.7	30.7	**13.7**	28.1	19.7	36.8
	Baseline (feat. only) [30]	83.7	27.6	75.5	20.3	19.9	27.4	28.3	27.4	79.0	28.4	70.1	55.1	20.2	72.9	22.5	35.7	8.3	20.6	23.0	39.3
	FADA	**92.5**	47.5	85.1	37.6	**32.8**	33.4	33.8	18.4	85.3	37.7	83.5	63.2	39.7	**87.5**	32.9	**47.8**	1.6	**34.9**	**39.5**	49.2
	FADA-MST	91.0	50.6	**86.0**	**43.4**	29.8	**36.8**	43.4	25.0	**86.8**	38.3	**87.4**	**64.0**	**38.0**	85.2	31.6	46.1	6.5	25.4	37.1	**50.1**

Regarding the training procedure, the network is first trained on source data for 20k iterations and then fine-tuned using our framework for 40k iterations. The batch size is eight. Four are source images and the other four are target images. Some data augmentations are used including random flip and color jittering to prevent overfitting.

Although our model is already able to achieve new state-of-the-art results, we further boost the performance by using self distillation [1,12,33] and multi-scale testing. A detailed ablation study is conducted in Section 4.5 to reveal the effect of each component, which, we hope, could provide more insights into the topic.

4.4   Comparison with State-of-the-art Methods

Small shift: Cross city adaptation.   Adaptation between real images from different cities is a scenario with great potential for practical applications. Table 1 shows the results of domain adaptation on Cityscapes → Cross-City dataset. Our method has different performance gains for the four cities. On average over four cities, our FADA achieves 8.5% improvement compared with the source-only baselines, and 2.25% gain compared with the previous best method.

Large shift: Synthetic to real adaptation.   Table 2 and 3 demonstrate the semantic segmentation performance on SYNTHIA → Cityscapes and GTA5 → Cityscapes tasks in comparison with existing state-of-the-art domain adaptation methods. We could observe that our FADA outperforms the existing methods by a large margin and obtain new state-of-the-art performance in terms of mIoU. Compared to the source model without any adaptation, a gain of 16.4% and 13.9% are achieved for VGG16 and ResNet101 respectively on SYNTHIA →
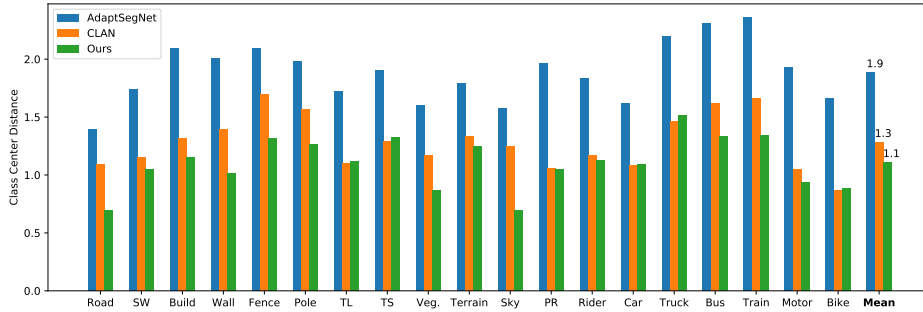
Fig. 4: Quantitative analysis of the feature joint distributions. For each class, we show the Class Center Distance as defined in Equation 9. Our FADA shows a better aligned structure in class-level compared with other state-of-the-art methods.

Cityscapes. FADA also obtains 15.5% and 12.4% improvement on different baelines for GTA5 $\rightarrow$ Cityscapes task. Besides, compared to the state-of-the-art feature-level methods, a general improvement of over 4% is witnessed. Note that as mentioned in [34], the "train" images in Cityscapes are more visually similar to the "bus" in GTA5 instead of the "train" in GTA5, which is also a challenge to other methods. Qualitative results for GTA5 $\rightarrow$ Cityscapes task are presented at Figure 5, reflecting that FADA also brings a significant visual improvement.

4.5   Feature distribution

To verify whether our fine-grained adversarial framework aligns features on a class-level, we design an experiment to investigate to what degree the class-level features are aligned. Considering different networks map features to different feature spaces, it's necessarily to find a stable metric. CLAN [20] suggests to use a Cluster Center Distance, which is defined as the ratio of intra-class distance between the trained model and the initial model, to measure class-level alignment degree. To better evaluate the effectiveness of class-level feature alignment on the same scale, we propose to modify the Cluster Center Distance to the Class Center Distance (CCD) by taking inter-class distance into account. The CCD for class i is defined as follows:

$$CCD(i) = \frac{1}{K-1} \sum_{j=1, j \neq i}^{K} \frac{\frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mu_i\|^2}{\|\mu_i - \mu_j\|^2} \tag{9}$$

where μ_i is the class center for class i, S_i is the set of all features belonging to class i. With CCD, we could measure the ratio of intra-class compactness over inter-class distance. A low CCD suggests the features of same class are clustered densely while the distance between different classes is relatively large. We randomly pick 2,000 source samples and 2,000 target samples respectively, and

Table 4: Ablation studies of each component. F-Adv refers to fine-grained adversarial training; SD refers to self distillation; MST refers to multi-scale testing.

F-Adv	SD	MST	mIoU
			36.8
✓			46.9
✓	✓		49.2
✓	✓	✓	**50.1**

compare the CCD values with other state-of-the-art methods: AdaptSegNet for global alignment and CLAN for class-wise alignment without explicitly modeling the class relationship. As shown in the Figure 4, FADA achieves a much lower CCD on most classes and get the lowest mean CCD value 1.1 compared to other algorithms. With FADA, we can achieve better class-level alignment and preserve consistent class structures between domains.

4.6   Ablation studies

Analysis of different components.   Table 4 presents the impact of each component on DeeplabV2 with ResNet-101 on GTA5 → Cityscapes task. The fine-grained adversarial training brings an improvement of 10.1%, which already makes it the new state of the art. To further explore the potential of the model, the self distillation strategy leads to an improvement of 2.3% and multi-scale testing further boosts the performance by 0.7%.

Hard labels vs. Soft labels.   As discussed in Section 3.3, the knowledge extracted from the classifier C could be produced from hard labels or soft labels. Here we compare these two forms of label on GTA5 → Cityscapes and SYN-THIA → Cityscapes tasks with DeeplabV2 ResNet-101. For soft labels, we use "confidence clipping" with threhold 0.9 as regularization. For hard labels, we only keep high-confidence samples, while ignoring the samples with confidence lower than 0.9. The results are reported in Table 5. Both choices give great boost to the baseline global feature alignment model. We observe that soft label is a more flexible choice and present more superior performance.

Impact of Confidence Clipping.   In our experiments, we use "confidence clipping" as a regularizer to prevent overfitting on noisy soft labels. The values of the confidence are truncated by a given threshold, therefore the values are not encouraged to heavily fit to a certain class. We test several thresholds and the results are shown in Table 6. Note that when the threshold is 1.0, it means no regularization is used. We observe constant performance gain using the confidence clipping. The best result is found when the threshold is 0.9.

5   Conclusion

In this paper, we address the problem of domain adaptive semantic segmentation by proposing a fine-grained adversarial training framework. A novel fine-

Table 5: Comparison of different strategies for extracting class-level knowledge on GTA5 → Cityscapes and SYNTHIA → Cityscapes tasks.

	GTA5	SYNTHIA
baseline [30]	39.4	35.4
hard labels	45.7	40.8
soft labels	**46.9**	**41.5**

Table 6: Influence of threshold for confidence clipping.

GTA5 → Cityscapes				
threshold	0.7	0.8	0.9	1.0
mIoU	46.2	46.3	**46.9**	45.7

grained discriminator is designed to not only distinguish domains, but also capture category-level information to guide a fine-grained feature alignment. The binary domain labels used to supervise the discriminator are generalized to domain encodings correspondingly to incorporate class information. Comprehensive experiments and analysis validate the effectiveness of our method. Our method achieves new state-of-the-art results on three popular tasks, outperforming other methods by a large margin.

Image   Before adaptation After adaptation   Ground-truth
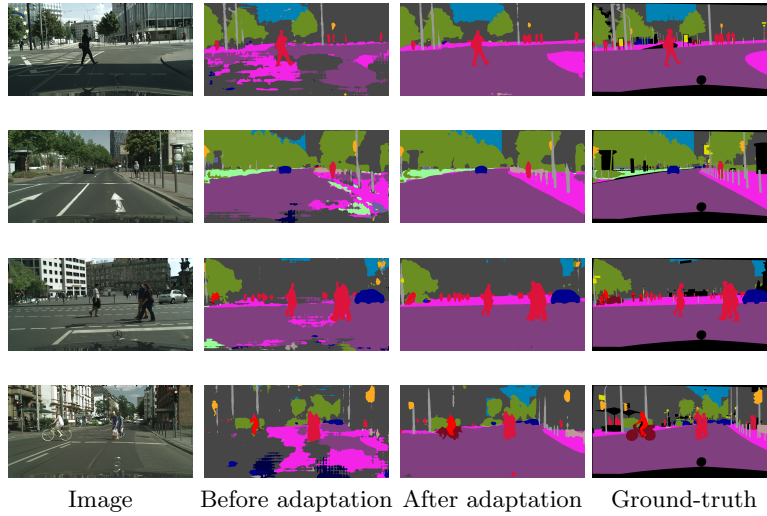
Fig. 5: Qualitative segmentation results for GTA5 → Cityscapes.

References

1. Bagherinezhad, H., Horton, M., Rastegari, M., Farhadi, A.: Label refinery: Improving imagenet classification through label progression. CoRR **abs/1805.02641** (2018), `http://arxiv.org/abs/1805.02641`
2. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine Learning **79**(1), 151–175 (May 2010). https://doi.org/10.1007/s10994-009-5152-4, `https://doi.org/10.1007/s10994-009-5152-4`
3. Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., Xu, T., Huang, J.: Progressive feature alignment for unsupervised domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
4. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (April 2018). https://doi.org/10.1109/TPAMI.2017.2699184
5. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), `http://arxiv.org/abs/1412.7062`
6. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
7. Chen, Y., Chen, W., Chen, Y., Tsai, B., Wang, Y.F., Sun, M.: No more discrimination: Cross city adaptation of road scene segmenters. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 2011–2020 (2017)
8. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: Computer Vision and Pattern Recognition (CVPR) (2018)
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: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
11. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision **111**(1), 98–136 (Jan 2015)
12. Furlanello, T., Lipton, Z.C., Tschannen, M., Itti, L., Anandkumar, A.: Born-again neural networks. In: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. pp. 1602–1611 (2018)
13. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. pp. 2672–2680. NIPS'14, MIT Press, Cambridge, MA, USA (2014), `http://dl.acm.org/citation.cfm?id=2969033.2969125`
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)

15. Hoffman, J., Tzeng, E., Park, T., Jun-Yan Zhu, a.P.I., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle consistent adversarial domain adaptation. In: International Conference on Machine Learning (ICML) (2018)

16. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation (2016)

17. Kumar, A., Sattigeri, P., Wadhawan, K., Karlinsky, L., Feris, R., Freeman, B., Wornell, G.: Co-regularized alignment for unsupervised domain adaptation. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31, pp. 9345–9356. Curran Associates, Inc. (2018), http://papers.nips.cc/paper/8146-co-regularized-alignment-for-unsupervised-domain-adaptation.pdf

18. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37. pp. 97–105. ICML'15, JMLR.org (2015), http://dl.acm.org/citation.cfm?id=3045118.3045130

19. Luo, Y., Liu, P., Guan, T., Yu, J., Yang, Y.: Significance-aware information bottleneck for domain adaptive semantic segmentation. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)

20. Luo, Y., Zheng, L., Guan, T., Yu, J., Yang, Y.: Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

21. Maas, A., Hannun, A., Ng, A.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the International Conference on Machine Learning. Atlanta, Georgia (2013)

22. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

23. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) European Conference on Computer Vision (ECCV). LNCS, vol. 9906, pp. 102–118. Springer International Publishing (2016)

24. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)

25. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. arXiv preprint arXiv:1712.02560 (2017)

26. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (Apr 2017). https://doi.org/10.1109/TPAMI.2016.2572683, https://doi.org/10.1109/TPAMI.2016.2572683

27. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference **90**(2), 227–244 (2000)

28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (May 2015)
29. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: Hua, G., Jégou, H. (eds.) Computer Vision – ECCV 2016 Workshops. pp. 443–450. Springer International Publishing, Cham (2016)
30. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
31. Tsai, Y.H., Sohn, K., Schulter, S., Chandraker, M.: Domain adaptation for structured output via discriminative patch representations. In: IEEE International Conference on Computer Vision (ICCV) (2019)
32. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: CVPR (2019)
33. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7130–7138 (July 2017). https://doi.org/10.1109/CVPR.2017.754
34. Zhang, Y., David, P., Gong, B.: Curriculum domain adaptation for semantic segmentation of urban scenes. In: The IEEE International Conference on Computer Vision (ICCV). vol. 2, p. 6 (Oct 2017)
35. Zhang, Y., Qiu, Z., Yao, T., Liu, D., Mei, T.: Fully convolutional adaptation networks for semantic segmentation. CoRR **abs/1804.08286** (2018)
36. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
37. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networkss. In: 2017 IEEE International Conference on Computer Vision (ICCV) (2017)
38. Zou, Y., Yu, Z., Kumar, B.V., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 289–305 (2018)