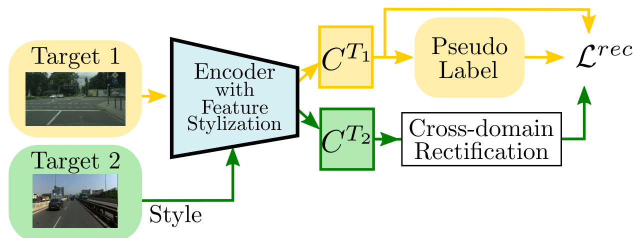
Cooperative Self-Training for Multi-Target Adaptive Semantic Segmentation

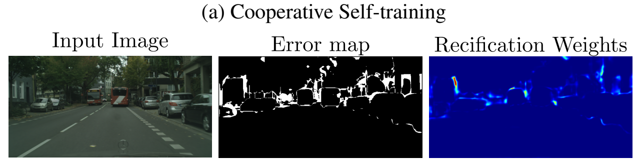
Yangsong Zhang[1,3], Subhankar Roy[2,4], Hongtao Lu[3], Elisa Ricci[2,4], Stéphane Lathuilière[1]
[1] LTCI, Télécom-Paris, Intitute Polytechnique de Paris [2] University of Trento, Trento, Italy
[3] Shanghai Jiao Tong University, Shanghai, China [4] Fondazione Bruno Kessler, Trento, Italy
yangsong.zhang.zys@gmail.com

Abstract

In this work we address multi-target domain adaptation (MTDA) in semantic segmentation, which consists in adapting a single model from an annotated source dataset to multiple unannotated target datasets that differ in their underlying data distributions. To address MTDA, we propose a self-training strategy that employs pseudo-labels to induce cooperation among multiple domain-specific classifiers. We employ feature stylization as an efficient way to generate image views that forms an integral part of self-training. Additionally, to prevent the network from overfitting to noisy pseudo-labels, we devise a rectification strategy that leverages the predictions from different classifiers to estimate the quality of pseudo-labels. Our extensive experiments on numerous settings, based on four different semantic segmentation datasets, validates the effectiveness of the proposed self-training strategy and shows that our method outperforms state-of-the-art MTDA approaches. https://github.com/Mael-zys/CoaST.



(a) Cooperative Self-training



(b) Cooperative Rectification

Figure 1: (a) Proposed method for Multi-Target Domain Adaptation (MTDA). Feature stylization is performed to favor consistency across classifiers via pseudo-labelling. Classifier consistency is used to estimate pseudo-label quality and rectify the training loss. (b) We show the uncertainty map estimated from an input image and used for loss rectification (dark blue for high confidence). We observe that low confidence regions often correspond to errors.

1. Introduction

Semantic segmentation is a key task in computer vision that consists in learning to predict semantic labels for image pixels. Given its importance in many real world applications, segmentation is widely studied and significant progress has been made [1, 3, 4] in the supervised regime. Much of the recent success can be attributed to the availability of large, curated, and annotated datasets [7, 21, 43]. As obtaining labeled data in semantic segmentation is costly and tedious, pre-trained models are often deployed in test environments without fine-tuning. Unfortunately, these models fail when the test samples are drawn from a distribution which is different from the training distribution. This phenomenon is known as the *domain shift* [30] problem. To mitigate the domain-shift between the training (*source*) and test (*target*) distributions, *Unsupervised Domain Adaptation* (UDA) methods [8] have been proposed.

Although a vast majority of UDA methods have been proposed for semantic segmentation in the single source and single target setting, in practical applications the assumption of a single target domain easily becomes vacuous. It is because the real world is more complex and target data can come from varying and different data distributions. For *e.g.*, in autonomous driving applications, the vehicle might encounter cloudy, rainy, and sunny weather conditions in a span of a very short journey. In such cases, it would require to switch among various adapted models specialized for a certain weather condition. To prevent cumbersome deployment operations one can instead train and deploy a single model for all the target environments, which is otherwise known as *Multi-Target Domain Adaptation* (MTDA). While in the context of object recognition MTDA has been explored in several works [6, 11, 22, 24, 37], it is heavily understudied for semantic segmentation, with just a handful of existing works [14, 16, 25]. The prior works are either sub-optimal at fully addressing the target-target alignment [25] or tackle it at a high computation overhead of explicit style-transfer [14, 16]. We argue that explicit interactions between a pair of target domains are essential in MTDA for minimizing the domain gap across target domains.

To this end, in this paper we present a novel MTDA framework for semantic segmentation that employs a *self-training* strategy based on pseudo-labeling to induce better synergy between different domains. Self-training is a widely used technique consisting in comparing different predictions obtained from a single image to impose consistency in network's predictions. In our proposed method, illustrated in Fig. 1 (a), we use an original image from one target domain (in yellow box) as the *view* that generates the pseudo-label; while the second prediction is obtained with the very same target image but stylized with an image coming from a different target domain (in green box). Given this stylized feature, the network is then asked to predict the pseudo-label obtained from the original view. Unlike [14] we use implicit stylization that does not need any externally trained style-transfer network, making our self-training end-to-end. Self-training not only helps the network to improve the quality of representations but also helps in *implicit alignment* between target-target pairs due to cross-domain interactions.

While our proposed self-training is well-suited for MTDA, it can still be susceptible to noisy pseudo-labels. To prevent the network from overfitting to noisy pseudo-labels when the domain-shift is large, we devise a *cross-domain cooperative rectification* strategy that captures the disagreement in predictions from different classifiers. Specifically, our proposed method uses the predictions from multiple domain-specific classifiers to estimate the quality of pseudo-labels (see Fig. 1 (b)), which are then weighted accordingly during self-training. Thus, interactions between all the target domains are further leveraged with our proposed framework, which we call **Co**-oper**a**tive **S**elf-**T**raining (*CoaST*) for MTDA.

Contributions. In summary, our contributions are three fold: *(i)* We propose a *self-training* approach for MTDA that synergistically combines pseudo-labeling and feature stylization to induce better cooperation between domains; *(ii)* To reduce the impact of noisy pseudo-labels in self-training, we propose cross-domain *cooperative objective rectification* that uses predictions from multiple domain-specific classifiers for better estimating the quality of pseudo-labels; and *(iii)* We conduct experiments on several standard MTDA benchmarks and advance the state-of-the-art performance by non-trivial margins.

2. Related Works

Our proposed method is most related to self-training and style-transfer, which we discuss in the following section.

Self-training for Domain Adaptation. Self-training in single-target domain adaptation (STDA) is a popular technique that involves generating pseudo-labels for the unlabeled target data and then iteratively training the model on the most confident labels. To that end, a plethora of UDA methods for semantic segmentation has been proposed [15, 17, 19, 35, 41, 42, 46] that use self-training due to its efficiency and simplicity. However, due to the characteristic error-prone nature of the pseudo-labeling strategy, the pseudo-labels cannot always be trusted and need a selection or correction mechanism. Most self-training methods differ in the manner in which the pseudo-labels are generated and selected. For instance, Zou *et al*. [46] proposed a class-balanced self-training strategy and used spatial priors, whereas in [39, 40] class-dependent centroids are employed to generate pseudo-labels. Most relevant to our approach are self-training methods [26, 41, 42] that rectify the pseudo-labels by measuring the uncertainty in predictions. Our proposed *CoaST* also derives inspirations from the STDA method [42], but instead of ad-hoc auxiliary classifiers, we use different stylized versions of the same image and different target domain-specific classifiers, to compute the rectification weights. The majority of the STDA self-training methods do not trivially allow target-target interactions, which is very crucial for MTDA.

Style-Transfer for Domain Adaptation. Yet another popular technique in STDA that essentially relies on transferring *style* (appearance) to make a source domain image look like a target image or vice versa. Assuming the semantic content in the image remains unchanged in the stylization process, and hence the pixel labels, *target-like* source images can be used to train a model for the target domain. Thus, the main task becomes modeling the style and content in an image through an encoder-decoder-like network. In the context of STDA in semantic segmentation, Hoffman *et al*. [13] proposed *CyCADA*, that incorporates cyclic reconstruction and semantic consistency to learn a classifier for the target data. Inspired by *CyCADA* a multitude of STDA methods [2, 5, 18, 20, 29, 36, 38, 45] have been proposed which use style-transfer in conjunction with other techniques. Learning a good encoder-decoder style-transfer network introduces additional training overheads and the success is greatly limited by the reconstruction quality. Alternatively, style-transfer can be performed in the feature space of the encoder without explicitly generating the stylized image [28, 44]. *CrossNorm* [28] explores this solution in the context of domain generalization to learn robust features. In *CoaST*, we adapt *CrossNorm* to our self-training mechanism by transferring style across target domains to induce better synergy.

Multi-target Domain Adaptation. MTDA for semantic segmentation is an under-explored field with just a handful of existing works [14, 16, 25]. For instance, Saporta *et al*. [25] proposed an adversarial framework where source-target and target-target alignment is achieved through dedicated discriminators. They also introduced a multi-target knowledge transfer (*MTKT*) approach where knowledge distillation (*KD*) [12] is used to learn a domain-agnostic classifier from multiple domain-specific experts. On the other hand, the CCL [14] and ADAS [16] rely on explicit style-transfer to tackle MTDA in semantic segmentation. Much like other style-transfer based STDA methods, [16] uses an external network for explicitly transferring styles between domains. Instead, we rely on implicit style-transfer making our proposed *CoaST* easy to implement and end-to-end trainable. Additionally, we introduce a cooperative rectification technique which prevents over-fitting on imperfect pseudo-labels, making our method more robust. We empirically prove this effectiveness over [14, 16, 25] through numerous experiments.

3. Methods

In this section we formally define the MTDA task and then we present the details of our proposed Cooperative Self-Training (*CoaST*) framework.

3.1. Preliminaries

Problem Definition and Notations. In the multi-target domain adaptation (MTDA) task, we assume that we have at our disposal $N^{\mathtt{S}}$ labeled instances from a source domain data set $\mathcal{D}^{\mathtt{S}} = \{(\mathbf{x}_n^{\mathtt{S}}, \mathbf{y}_n^{\mathtt{S}})\}_{n=1}^{N^{\mathtt{S}}}$ where $\mathbf{x}^{\mathtt{S}} \in \mathbb{R}^{H\times W\times 3}$ are input images with their corresponding one-hot ground truth labels $\mathbf{y}^{\mathtt{S}} \in \mathbb{R}^{H\times W\times K}$, assigned to each pixel in the $H\times W$ spatial grid belonging to one of the K semantic classes. Moreover, there are a total of M unlabeled target domains $\{\mathtt{T}_1, \dots, \mathtt{T}_M\}$ where each target domain $\mathtt{T}_i$ comprises of an unlabeled data set $\mathcal{D}^{\mathtt{T}_i} = \{\mathbf{x}_n^{\mathtt{T}_i}\}_{n=1}^{N^{\mathtt{T}_i}}$, with $\mathbf{x}^{\mathtt{T}_i} \in \mathbb{R}^{H\times W\times 3}$ representing the target images and $N^{\mathtt{T}_i}$ being the number of unlabeled instances. Following standard MTDA protocols, we assume that the marginal distributions between every pair of available domains differ, under the constraint of underlying semantic concept remaining the same. The goal of MTDA is to learn a single network $f = C \circ \Phi$ using $\{\bigcup_{i=1}^{M} \mathcal{D}^{\mathtt{T}_i}\} \cup \mathcal{D}^{\mathtt{S}}$ that can segment samples from any target domain, where C and ϕ are the classifier and the backbone encoder networks, respectively. While we consider that the domain information is known at training time, the domains labels of the images during inference are unknown.

Overall Framework. To address the MTDA, we operate in two stages. In the first stage we aim to learn target domain-specific classifiers with *adversarial adaptation* [31, 34] that aligns features between a given source-target domain pair. The first stage results in the network parameters that enable even better alignment in the subsequent stage. In this second stage, we adopt a pseudo-label based *cooperative self-training* strategy to further align the target domains. In particular, our proposed self-training strategy enforces consistency among the target domain-specific classifiers, allowing maximal interaction among the different target domains. Importantly, our cooperative training also incorporates a *threshold-free rectification* term that prevents overfitting to noisy pseudo-labels. Finally, we use *knowledge distillation* to distill all the learned information from domain-specific classifiers to a domain-agnostic classifier that can be used to segment a test image from any target domain, thereby alleviating the need for domain-id during inference.

Adversarial Warm-up. This marks the first stage, where we follow [25] for initializing our framework in order to obtain an encoder network Φ that is shared among all the target domains, and M distinct target domain-specific classifiers $\{C^{\mathtt{T}_i} | \forall i \in \{1, \dots, M\}\}$. Concurrently, we also initialize M target domain-specific discriminators $\{D^{\mathtt{T}_i} | \forall i \in \{1, \dots, M\}\}$ to learn a classifier that is invariant for a specific source-target pair. To recap, in adversarial warm-up stage the discriminator $D^{\mathtt{T}_i}$ is trained to distinguish between the source and target $\mathtt{T}_i$ predictions whereas the network $f^{\mathtt{T}_i} = C^{\mathtt{T}_i} \circ \Phi$ is trained to fool the $D^{\mathtt{T}_i}$. Note that unlike the original work in [10], the output from the classifier is given as an input to the domain discriminator [25, 31]. Additionally, for the source samples we employ the standard supervised cross-entropy loss, which is used to train every $f^{\mathtt{T}_i}$. Overall, for a given source-target pair $(\mathtt{S}, \mathtt{T}_i)$ the discriminator $D^{\mathtt{T}_i}$ is trained with the objective:

$$\begin{aligned} \mathcal{L}_{D^{\mathtt{T}_i}} = \mathcal{L}_{\text{bce}}\big(D^{\mathtt{T}_i}(C^{\mathtt{T}_i}(\Phi(\mathbf{x}^{\mathtt{S}}))), 1\big) + \\ \mathcal{L}_{\text{bce}}\big(D^{\mathtt{T}_i}(C^{\mathtt{T}_i}(\Phi(\mathbf{x}^{\mathtt{T}_i}))), 0\big) \end{aligned} \tag{1}$$

where $\mathcal{L}_{\text{bce}}$ stands for the *binary cross-entropy* loss. Simultaneously, the network $f^{\mathtt{T}_i}$ is trained along with the source segmentation loss and adversarial loss as:

$$\begin{aligned} \mathcal{L}_{f^{\mathtt{T}_i}} = \mathcal{L}_{\text{ce}}\big(C^{\mathtt{T}_i}(\Phi(\mathbf{x}^{\mathtt{S}})), \mathbf{y}^{\mathtt{S}}\big) + \\ \lambda_{adv}\mathcal{L}_{\text{bce}}\big(D^{\mathtt{T}_i}(C^{\mathtt{T}_i}(\Phi(\mathbf{x}^{\mathtt{T}_i}))), 1\big) \end{aligned} \tag{2}$$

where $\mathcal{L}_{\text{ce}}$ is the supervised cross-entropy loss for the source data and λ_{adv} is a hyperparameter to balance the losses. In the adversarial warm-up stage we alternatively minimize $\mathcal{L}_{\mathrm{D^{T_i}}}$ and $\mathcal{L}_{\mathrm{f^{T_i}}}$ for every source-target pairs.

3.2. Cooperative Self-Training (*CoaST*)

The goal of the second stage is to refine the image representation learned in the adversarial warm-up stage. We devise a self-training approach with the usage of pseudo-labels that iteratively improves the predictions of the model on the unlabeled data.

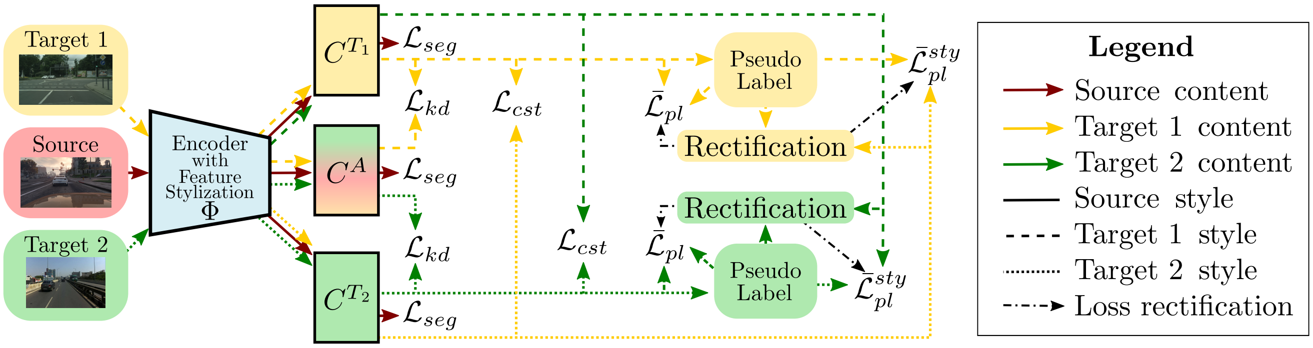


Figure 2: Illustration of the proposed *CoaST* approach in the case of two target domains. Domain-specific classifiers are distilled to learn a domain-agnostic classifier. Style-transfer is used in the encoder network to induce cooperation between the different classifiers and rectify the pseudo-labeling losses.

Pseudo-labelling. In our framework for the MTDA, we have m specialized target domain-specific classifiers, with each classifier $C^{\mathtt{T}_i}$ trained to handle data coming from the corresponding domain $\mathtt{T}_i$. We exploit these specialized classifiers to generate pseudo-labels (PLs) for the target samples in their respective target domains. Specifically, given the n^{th} image $\mathbf{x}_n^{\mathtt{T}_i}$ from the target domain i, we use the network $f^{\mathtt{T}_i}$ to predict the segmentation map $[\hat{\mathbf{p}}_n^{\mathtt{T}_i}(\mathbf{k})]_{\mathbf{k}\in[H]\times[W]\times[K]} = C^{\mathtt{T}_i}(\Phi(\mathbf{x}_n^{\mathtt{T}_i}))$ and compute the pseudo-label as:

$$\hat{\mathbf{y}}_n^{\mathtt{T}_i} = \mathbf{e}_k\big(\operatorname*{argmax}_k [\hat{\mathbf{p}}_n^{\mathtt{T}_i}(\mathbf{k})]_{\mathbf{k}\in[H]\times[W]\times[K]}\big), \tag{3}$$

where $\mathbf{e}_k(.)$ denotes the one-hot encoding operator and $\hat{\mathbf{y}}_n^{\mathtt{T}_i} \in \mathbb{R}^{H\times W\times K}$. The PL is computed at the beginning of the second stage and is updated every n_b iterations. This PL is then used to self-supervise the corresponding $f^{\mathtt{T}_i}$ network with a *cross entropy* loss:

$$\mathcal{L}_{\text{pl}} = \mathcal{L}_{\text{ce}}(\hat{\mathbf{p}}_n^{\mathtt{T}_i}, \hat{\mathbf{y}}_n^{\mathtt{T}_i}), \tag{4}$$

However, this formulation suffers from two main issues. First, the PLs act only on the same domain-specific classifier $C^{\mathtt{T}_i}$ corresponding to the domain of input images. Hence, it does not induce any synergy between the different classifiers. Second, since the PLs can be noisy, using the pseudo-labeling objective in Eq. (4) can lead to detrimental behaviour. To address these two issues and further benefit from our PLs, we introduce a self-training technique that is realized by leveraging feature stylization [28].

Style-Transfer for Cooperative Self-Training. To benefit from the self-training objective in Eq. (4), one requires to obtain the predictions from a view $t(\mathbf{x}_n^{\mathtt{T}_i})$ and enforce its predictions to match with that of $\hat{\mathbf{y}}_n^{\mathtt{T}_i}$, where $t(.)$ is any stochastic transformation. Indeed, such a consistency-based training strategy has successfully been applied in the semi-supervised learning literature [27]. However, finding optimal transformations is not trivial and varies between data sets and even tasks. In this work, we resort to a data-driven transformation policy that is based on style-transfer [32]. Style-transfer consists in transferring the"style" (appearance) from one image to another. Concretely, in our case, the transformation $t(.)$ is a style-transfer operation that essentially applies the style of an image $\mathbf{x}^{\mathtt{T}_j}$ to the image $\mathbf{x}^{\mathtt{T}_i}$, where $i \neq j$. The style transformed image $\mathbf{x}^{\mathtt{T}_{i\to j}}$ can in essence be regarded as a *virtual* image that *appears* to come from $\mathtt{T}_j$ but having the content structure of $\mathtt{T}_i$. Therefore, for the n^{th} sample $\mathbf{x}^{\mathtt{T}_{i\to j}}$ we obtain the prediction from $f^{\mathtt{T}_j}$ and optimize it to be close to $\hat{\mathbf{y}}_n^{\mathtt{T}_i}$. In this way our PL from a given target domain-specific classifier can be used to supervise another domain-specific classifier, enforcing better consistency between pairs of target domains. Moreover, we thereafter show how style-transfer is instrumental in rectifying the objective in Eq. (4) according to an estimated confidence score. We now describe how we use style-transfer to improve self-training in the MTDA setting.

Style-transfer in the pixel space, with separately trained encoder-decoder network, has very recently been used for the MTDA work [16]. To avoid such costly, and often sub-optimal, image generation with the pixel-space style-transfer methods, we perform style-transfer in the intermediate feature space of the encoder network. In particular, we adapt cross normalization (*CrossNorm*) [28] in our MTDA setting and use it as a means of exchanging feature statistics, and hence style, across different domains. More precisely, our Cross-Domain Normalization (*CrossDoNorm*) performs style-transfer by exchanging *style vectors* between two target domain images, which are computed from the channel-wise mean and standard deviation of the features maps. Exchange of style vectors is deemed sufficient for style-transfer by prior works [32] who show that these statistics encode the image style and that style-transfer can be obtained through a simple re-normalization.

Given a pair of images $(\mathbf{x}^{\mathtt{T}_i}, \mathbf{x}^{\mathtt{T}_j})$ from the target domains $\mathtt{T}_i$ and $\mathtt{T}_j$, we extract their corresponding features from the l^{th} layer of the encoder as $\mathbf{z}_l^{\mathtt{T}_i} = \Phi_l(\mathbf{x}^{\mathtt{T}_i})$ and $\mathbf{z}_l^{\mathtt{T}_j} = \Phi_l(\mathbf{x}^{\mathtt{T}_j})$. From these intermediate feature maps, we compute the corresponding channel-wise means $(\boldsymbol{\mu}_l^{\mathtt{T}_i}, \boldsymbol{\mu}_l^{\mathtt{T}_j})$ and standard deviations $(\boldsymbol{\sigma}_l^{\mathtt{T}_i}, \boldsymbol{\sigma}_l^{\mathtt{T}_j})$, such that $\boldsymbol{\mu}_l \in \mathbb{R}^k$ and $\boldsymbol{\sigma}_l \in \mathbb{R}^k$ with k being the number of channels in the layer l. For instance, *CrossDoNorm* first standardizes the features with its own channel-wise statistics, *e.g.*, $(\boldsymbol{\mu}_l^{\mathtt{T}_i}, \boldsymbol{\sigma}_l^{\mathtt{T}_i})$ for $\mathbf{z}_l^{\mathtt{T}_i}$, and then re-normalizes with the statistics from the other domain $(\boldsymbol{\mu}_l^{\mathtt{T}_j}, \boldsymbol{\sigma}_l^{\mathtt{T}_j})$ to obtain stylized features $\mathbf{z}_l^{\mathtt{T}_{i\to j}}$. The *CrossDoNorm* can be done symmetrically resulting into stylized features that are computed as:

$$\begin{aligned} \mathbf{z}_l^{\mathtt{T}_{i\to j}} &= \boldsymbol{\sigma}_l^{\mathtt{T}_j} \frac{\mathbf{z}_l^{\mathtt{T}_i} - \boldsymbol{\mu}_l^{\mathtt{T}_i}}{\boldsymbol{\sigma}_l^{\mathtt{T}_i}} + \boldsymbol{\mu}_l^{\mathtt{T}_j} \\ \mathbf{z}_l^{\mathtt{T}_{j\to i}} &= \boldsymbol{\sigma}_l^{\mathtt{T}_i} \frac{\mathbf{z}_l^{\mathtt{T}_j} - \boldsymbol{\mu}_l^{\mathtt{T}_j}}{\boldsymbol{\sigma}_l^{\mathtt{T}_j}} + \boldsymbol{\mu}_l^{\mathtt{T}_i} \end{aligned} \tag{5}$$

Our *CrossDoNorm* can ideally perform feature stylization at multiple layers in the encoder network. Next, the above computed stylized feature $\mathbf{z}_l^{\mathtt{T}_{i\to j}}$ is then given as input to the subsequent layers of the network, with the final prediction map $\hat{\mathbf{p}}^{\mathtt{T}_{i\to j}}$ obtained from the classifier $C^{\mathtt{T}_j}$. With the PL $\hat{\mathbf{y}}^{\mathtt{T}_i}$ generated from $\mathbf{z}_l^{\mathtt{T}_i}$ by the original domain classifier $C^{\mathtt{T}_i}$, the other classifier $C^{\mathtt{T}_j}$, along with the encoder Φ, is then trained in a supervised manner:

$$\begin{aligned} \mathcal{L}_{\text{pl}}^{\text{sty}}(\mathbf{x}^{\mathtt{T}_i}, \mathbf{x}^{\mathtt{T}_j}) &= \mathcal{L}_{\text{ce}}(\hat{\mathbf{p}}^{\mathtt{T}_{i\to j}}, \hat{\mathbf{y}}^{\mathtt{T}_i}) \\ \mathcal{L}_{\text{pl}}^{\text{sty}}(\mathbf{x}^{\mathtt{T}_j}, \mathbf{x}^{\mathtt{T}_i}) &= \mathcal{L}_{\text{ce}}(\hat{\mathbf{p}}^{\mathtt{T}_{j\to i}}, \hat{\mathbf{y}}^{\mathtt{T}_j}) \end{aligned} \tag{6}$$

Since it has been shown in the literature that training with soft-labels improves the learning ability of the network [12], we use a soft-version of the $\mathcal{L}_{\text{pl}}^{\text{sty}}$ loss described in Eq. (6). In other words, we further enforce consistency in predictions between two domain-specific classifiers by optimizing the KL-divergence objective between the cross-domain stylized prediction $\hat{\mathbf{p}}^{\mathtt{T}_{i\to j}}$ and the original target domain prediction $\hat{\mathbf{p}}^{\mathtt{T}_i}$ instead of PLs as:

$$\mathcal{L}_{\text{cst}}(\mathbf{x}^{\mathtt{T}_i}, \mathbf{x}^{\mathtt{T}_j}) = \mathcal{L}_{\text{kl}}(\hat{\mathbf{p}}^{\mathtt{T}_{i\to j}}, \hat{\mathbf{p}}^{\mathtt{T}_i}) \tag{7}$$

Additionally, our *CrossDoNorm* also acts as an implicit data augmentation method in the feature space. As the style information is mainly manifested in the low level features of the encoder, to prevent over-regularization we only apply the *CrossDoNorm* in the initial layers of the encoder.

Cooperative Objective Rectification. The PLs generated during the refinement process can be very noisy due to domain-shift, leading to degradation of representations. To tackle this shortcoming of self-training, we propose our cooperative objective rectification method that takes into account the uncertainty in the model predictions. This uncertainty in predictions for a given sample $\mathbf{x}_n^{\mathtt{T}_i}$ is measured by combining the predictions obtained from all the target domain classifiers. More precisely, considering $\mathbf{x}_n^{\mathtt{T}_i}$, we compute the consistency scores between the prediction from the $C^{\mathtt{T}_i}$ and the predictions from the other domain-specific classifiers on the stylized features of $\mathbf{x}_n^{\mathtt{T}_i}$. Following [42], we use the KL-divergence between a pair of predictions as a measure of consistency. Lower the consistency, less reliable is the corresponding PL. Finally, this consistency score is then used as a weight to re-weight the self-training loss introduced in the Eq. (4). The rectified self-training loss corresponding to Eq. (4) is given as:

$$\bar{\mathcal{L}}_{\text{pl}}(\mathbf{x}^{\mathtt{T}_i}) = w_i \mathcal{L}_{\text{ce}}(\hat{\mathbf{p}}^{\mathtt{T}_i}, \hat{\mathbf{y}}^{\mathtt{T}_i}) \tag{8}$$

where the weight value w_i is the averaged consistency scores obtained with the predictions between $C^{\mathtt{T}_i}$ and the rest of the classifiers $\{C^{\mathtt{T}_1}, \dots, C^{\mathtt{T}_M}\} \backslash C^{\mathtt{T}_i}$ as:

$$w_i = \frac{1}{M-1} \sum_{j=1, j\neq i}^{M} \exp\left(-\mathcal{L}_{\text{kl}}(\hat{\mathbf{p}}^{\mathtt{T}_i}, \hat{\mathbf{p}}^{\mathtt{T}_{i\to j}})\right) \tag{9}$$

where the exponential function $\exp(\cdot)$ is used here to map the KL divergence that range in $[0, +\infty[$ to weights values in]0,1]. Contrary to [42] our uncertainty score is obtained by considering the predictions from all classifier pairs, against just using a single pair of classifiers. Also, different from many pseudo-labeling approaches described in Sec. 2, our re-weighting formulation is not based on thresholding and therefore avoids manual hyperparameter tuning. Similarly, the cross-domain pseudo labeling losses introduced in Eq. (6) are rectified as:

$$\begin{aligned} \bar{\mathcal{L}}_{\text{pl}}^{\text{sty}}(\mathbf{x}^{\mathtt{T}_i}, \mathbf{x}^{\mathtt{T}_j}) &= w_i \mathcal{L}_{\text{ce}}(\hat{\mathbf{p}}^{\mathtt{T}_{i\to j}}, \hat{\mathbf{y}}^{\mathtt{T}_i}) \\ \bar{\mathcal{L}}_{\text{pl}}^{\text{sty}}(\mathbf{x}^{\mathtt{T}_j}, \mathbf{x}^{\mathtt{T}_i}) &= w_j \mathcal{L}_{\text{ce}}(\hat{\mathbf{p}}^{\mathtt{T}_{j\to i}}, \hat{\mathbf{y}}^{\mathtt{T}_j}) \end{aligned} \tag{10}$$

Knowledge Distillation. As our end goal is to be able to predict test samples coming from any target domain, we also learn an additional domain-agnostic classifier C^{A}. We use the source samples to train C^{A} in addition to the supervised segmentation objective given in Eq. (2) as:

$$\begin{aligned} \mathcal{L}_{\text{seg}} = \sum_{i=1}^{M} & \mathcal{L}_{\text{ce}}(C^{\mathtt{T}_i}(\Phi(\mathbf{x}^{\text{S}})), \mathbf{y}^{\text{S}}) \\ & + \mathcal{L}_{\text{ce}}(C^{\text{A}}(\Phi(\mathbf{x}^{\text{S}})), \mathbf{y}^{\text{S}}) \end{aligned} \tag{11}$$

In order to distill the information learned by the domain-specific classifiers $C^{\mathtt{T}_i}$ into the domain-agnostic classifier C^{A} we use knowledge distillation (KD) as in [14, 25]. For every target domain sample, we enforce consistency between the prediction from the corresponding domain-specific classifier and the domain-agnostic one using the KL divergence. The KD loss for a given $\mathtt{T}_i$ domain is given as:

$$\mathcal{L}_{\text{kd}} = \mathcal{L}_{\text{kl}}(C^{\text{A}}(\Phi(\mathbf{x}^{\mathtt{T}_i})), \hat{\mathbf{y}}^{\mathtt{T}_i}) \tag{12}$$

where only the weights of the C^{A} is only updated during the optimization of Eqn. 12. We use the domain-agnostic classifier C^{A} during inference.

Overall Training. The final objective to train our proposed *CoaST* is given by summing all the unary and pairwise losses previously described:

$$\mathcal{L}_{\textit{CoaST}} = \sum_{(\mathbf{x}^{\mathrm{S}},\mathbf{y}^{\mathrm{S}})\in\mathcal{D}^{\mathrm{S}}} \mathcal{L}_{\mathrm{seg}}(\mathbf{x}^{\mathrm{S}},\mathbf{y}^{\mathrm{S}}) +$$
$$\sum_{i=1}^{M} \sum_{\mathbf{x}^{\mathrm{T}_i}\in\mathcal{D}^{\mathrm{T}_i}} \Big[\frac{1}{M}\mathcal{L}_{\mathrm{kd}}(\mathbf{x}^{\mathrm{T}_i}) + \bar{\mathcal{L}}_{\mathrm{pl}}(\mathbf{x}^{\mathrm{T}_i}) +$$
$$\frac{1}{M-1}\sum_{\substack{j=1\\ j\neq i}}^{M} \sum_{\mathbf{x}^{\mathrm{T}_j}\in\mathcal{D}^{\mathrm{T}_j}} \bar{\mathcal{L}}_{\mathrm{pl}}^{\mathrm{sty}}(\mathbf{x}^{\mathrm{T}_i},\mathbf{x}^{\mathrm{T}_j}) + \mathcal{L}_{\mathrm{cst}}(\mathbf{x}^{\mathrm{T}_i},\mathbf{x}^{\mathrm{T}_j})\Big] \tag{13}$$

Note that, the KD loss and pairwise losses are normalized by M and $M-1$ to preserve the source-target balance when varying the number of target domains.

4. Experiments

4.1. Experimental set-up

Datasets. We conduct experiments on two standard benchmarks for MTDA in semantic segmentation. These two benchmarks have been derived from four semantic segmentation datasets, namely the synthetic *GTA5* [23] and the real world *Cityscapes* [7], *Mapillary* [21] and *IDD* [33]. Note that the datasets are varying in size as in the Mapillary is six times bigger than the Cityscapes, and thrice as big as the IDD. More details can be found in the supplement.

Benchmarks. The benchmarks for MTDA in semantic segmentation differ in the way the class labels are mapped across the datasets. They are: (i) the *7-classes* benchmark, introduced in [25], which considers 7 classes and downsamples the images to a resolution of 640×320 both for training and evaluation; and (ii) the *19-classes* benchmark, introduced in [14], which operates at higher resolution of 1024×512. Both benchmarks use several combinations of the four datasets to create four *Synthetic to Real* scenarios and one *Real to Real* scenario.

Metrics. We report the standard intersection over union (IoU) for every class and the mean-IoU (mIoU) for each target domain. Whereas, to obtain a single overall score in the MTDA, we average the mIoU across all the target domains.

Baselines. In our experiments, we compare with the state-of-the-art methods: Multi-Target Knowledge Transfer (*MTKT*) [25], Collaborative Consistency Learning (*CCL*) [14] and A Direct Adaptation Strategy (*ADAS*) [16]. We compare with these methods on the settings adopted in the corresponding papers: *7-classes* for *MTKT* and *19-classes* for *CCL*. We also include an approach, introduced in [25] and referred to as *Multi-Discriminator*, where a single classifier is trained using multiple domain-specific discriminator. In addition, we follow [14, 25] and include two baselines based on a single-target domain adaptation method. In *Individual*, an adversarial approach [34] is trained separately on every target dataset. At inference time, the target images are tested by the corresponding domain-specific model. In *Data combination*, we treat the union of all the target domains as a single target domain. For these two baselines, we report the results provided in [14, 25].

Implementation Details. To be fairly comparable, we adopt the very same network architecture as in the baseline [25], except we use the modified version of ResNet101 based DeepLab-V2 [3] that contains dropout layers [39, 42]. Due to lack of space we report the rest of the implementation details in the supplement.

4.2. Comparison with State-of-the-art: Syn to Real

Quantitative Comparison. We provide a detailed comparison with state-of-the-art on the *7-classes* benchmark using the GTA5 to Cityscapes and IDD setting. Results are reported in Tab. 1. Overall, we can observe that our method outperforms all the other baselines. In terms of average mIoU, *CoaST* outperforms *MTKT* with a 3.1% margin. This gain is remarkable considering that *MTKT* improves the *Individual* baseline trained on Cityscapes by only 0.7%. Besides, *CoaST* outperforms *ADAS* by 0.1% even though *ADAS* use a higer image resolution than *CoaST* and the other baselines. We can observe gains with respect to *MTKT* in both small objects such as *human* (54.4% vs 51.0% on Cityscapes) and background classes such as sky (88.2% vs 84.0% on Cityscapes). One noticeable point is that the IDD dataset seems more challenging since *Individual* obtains lower performance on this dataset. Similarly, the *Multi-Dis*, *MTKT* and *ADAS* obtain mIoUs of 65.7%, 65.9% and 66.9% respectively which are much lower than on Cityscapes (68.9%, 70.4% and 75.4%). However, with *CoaST*, which uses consistency training and cooperative objective rectification, we improve *MTKT* and *ADAS* performance by 4.1% and 3.1% obtaining a mIoU score of 70.0%.

We now provide experimental results on the *19-classes* benchmark using the same GTA5 to Cityscapes and IDD setting. Results are reported in Tab. 2. First, we observe that all methods have lower scores compared to the *7-classes* since the high number of classes makes the task more difficult. Nevertheless, we observe that *CoaST* outperforms all the other approaches on almost all the classes and domains. Compared to *CCL* and *ADAS*, we observe that *CoaST* obtains better average mIoU (+2.7% and +2.1% respectively) and that the gain is mostly explained by better performances on difficult classes such as *fence* or *sign* and *bus* that largely compensate the drop on the road class.

GTA5 → Cityscapes + IDD										
Method	**Target**	**flat**	**constr**	**object**	**nature**	**sky**	**human**	**vehicle**	**mIoU**	**Avg.**
Individual [34]	C	93.5	80.5	26.0	78.5	78.5	55.1	76.4	69.8	67.5
	I	91.2	53.1	16.0	78.2	90.7	47.9	78.9	65.1	
Data Comb. [34]	C	93.9	80.2	26.2	79.0	80.5	52.5	78.0	70.0	67.4
	I	91.8	54.5	14.4	76.8	90.3	47.5	78.3	64.8	
Multi-Dis [25]	C	94.3	80.7	20.9	79.3	82.6	48.5	76.2	68.9	67.3
	I	92.3	55.0	12.2	77.7	92.4	51.0	80.2	65.7	
MTKT [25]	C	94.5	82.0	23.7	80.1	84.0	51.0	77.6	70.4	68.2
	I	91.4	56.6	13.2	77.3	91.4	51.4	79.9	65.9	
ADAS [16](1024×512)	C	**95.1**	82.6	**39.8**	**84.6**	81.2	**63.6**	**80.7**	**75.4**	71.2
	I	90.5	**63.0**	**22.2**	73.7	87.9	54.3	76.9	66.9	
CoaST (Ours)	C	94.7	**82.9**	25.4	82.2	**88.2**	54.4	80.5	72.6	**71.3**
	I	**94.2**	61.5	20.0	**82.7**	**93.4**	**55.5**	**82.6**	**70.0**	

Table 1: Comparison with State-of-the-art on the *7-classes* benchmark using the GTA5 → Cityscapes + IDD configuration.

GTA5 → Cityscapes + IDD																						
Method	**Target**	**road**	**sidewalk**	**building**	**walk**	**fence**	**pole**	**light**	**sign**	**veg**	**terrain**	**sky**	**person**	**rider**	**car**	**truck**	**bus**	**train**	**motor**	**bike**	**mIoU**	**Avg.**
Individual [34]	C	88.8	23.8	81.5	27.7	27.3	31.7	33.2	22.9	83.1	27.0	76.4	58.5	**28.9**	84.3	30.0	36.8	0.3	27.7	**33.1**	43.3	43.5
	I	94.1	24.4	**66.1**	31.3	22.0	25.4	9.3	26.7	80.0	**31.4**	**93.5**	48.7	43.8	71.4	49.4	28.5	**0**	48.7	**34.3**	43.6	
Data Comb.[34]	C	86.1	32.0	79.8	24.3	22.3	28.5	27.9	14.3	85.1	29.8	79.9	56.1	20.5	77.7	**34.4**	35.2	**0.7**	18.2	13.1	40.3	41.2
	I	92.8	23.4	60.9	25.8	23.4	24.1	8.6	32.2	77.5	26.8	92.3	48.0	41.0	74.4	48.4	17.7	**0**	52.5	28.2	42.0	
CCL [14]	C	**90.3**	34.0	**82.5**	26.2	26.6	33.6	35.4	21.5	84.7	39.8	81.1	58.4	25.8	**84.5**	31.4	45.4	0	29.9	24.7	45.0	45.5
	I	**95.0**	30.5	65.6	29.4	23.4	29.2	12.0	37.8	77.3	31.3	91.9	**52.4**	48.3	74.9	50.1	36.6	**0**	56.1	32.4	46.0	
ADAS [16]	C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	45.8	46.1
	I	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	46.3	
CoaST (Ours)	C	81.7	**38.3**	71.0	**33.3**	**30.7**	**35.1**	**38.2**	**37.6**	**86.4**	**46.9**	**81.9**	**63.4**	27.4	**84.5**	29.4	**45.6**	0.3	**32.6**	31.3	**47.1**	**48.2**
	I	85.7	**36.1**	65.1	**33.2**	**23.7**	**32.8**	**19.0**	**62.9**	**82.5**	29.5	91.8	52.1	**55.3**	**83.4**	**62.9**	**46.1**	**0**	**55.5**	18.5	**49.3**	

Table 2: Comparison with State-of-the-art on the *19-classes* benchmark using the GTA5 → Cityscapes + IDD configuration.

Qualitative Comparison. Fig. 3 shows a qualitative comparison with *MTKT* on the *7-classes* benchmark when adapting from GTA5 to Cityscapes and IDD. From these visualizations, we can see that *CoaST* segment better small objects such as 'human' or 'object' classes. This difference is especially clear on the IDD dataset.

Summary of all the Settings. To complete this evaluation in the *Synthetic to Real* scenario, we report in Tab. 3 the average mIoU considering all the possible target configurations on the *19-classes* benchmark. Results on the *7-classes* benchmark are reported in supplement. For the *19-classes* benchmarks, the proposed method is compared with the best respective competitor. In short, we observe that *CoaST* obtains the best performance in all configurations and on all the domains. These experiments demonstrate the robustness of our approach.

Target			**method**	**mIoU**			**mIoU**
C	I	M		C	I	M	Avg.
√	√	-	*CCL* [14]	45.0	46.0	-	45.5
			ADAS [16]	45.8	46.3	-	46.1
			CoaST (Ours)	**47.1**	**49.3**	-	**48.2**
√	-	√	*CCL* [14]	45.1	-	48.8	47.0
			ADAS [16]	45.8	-	49.2	47.5
			CoaST (Ours)	**47.9**	-	**51.8**	**49.9**
-	√	√	*CCL* [14]	-	44.5	46.4	45.5
			ADAS [16]	-	46.1	47.6	46.9
			CoaST (Ours)	-	**49.5**	**51.6**	**50.6**
√	√	√	*CCL* [14]	46.7	47.0	49.9	47.9
			ADAS [16]	46.9	47.7	51.1	48.6
			CoaST (Ours)	**47.2**	**48.7**	**51.4**	**49.1**

Table 3: Summary of performances obtained on the *19-classes* benchmark. Cityscapes, IDD and Mapillary are referred to as C, I and M respectively. We report the mIoU averaged over the target domains.

4.3. Ablation Study

To illustrate the impact of the proposed cooperative self-training and rectification, we present a detailed ablation study. We present several variants of *CoaST*. First, we employ our architecture but with the adversarial training scheme of *MTKT* [25]. The goal of this variant is to show that our performance gain is not due to our slight architec-

Model	Adv.	Self-Tr.	*CrossDoNorm*	$\mathcal{L}_{cst}$	Rec.	C	I	Avg.
*MTKT** [25]	√					67.3	64.3	65.8
(i)		√				65.6	63.2	64.4
(ii)		√	√			69.2	67.4	68.4
(iii)		√	√	√		70.2	67.5	68.9
(iv)		√	√		√	72.1	69.9	71.0
(v)		√	√	√	√	72.6	70.0	71.3

Table 4: Ablation study of the proposed method on *7-classes* benchmark, in GTA5 → Cityscapes + IDD configuration.

Rectification	C	I	Avg.
Without	70.2	67.5	68.9
Drop-Out-based [9]	70.7	68.7	69.7
Auxiliary network [42]	69.8	68.7	69.3
Cooperative (ours)	**72.6**	**70.0**	**71.3**

Table 5: Ablation study on the GTA5 → Cityscapes + IDD configuration: Rectification strategy

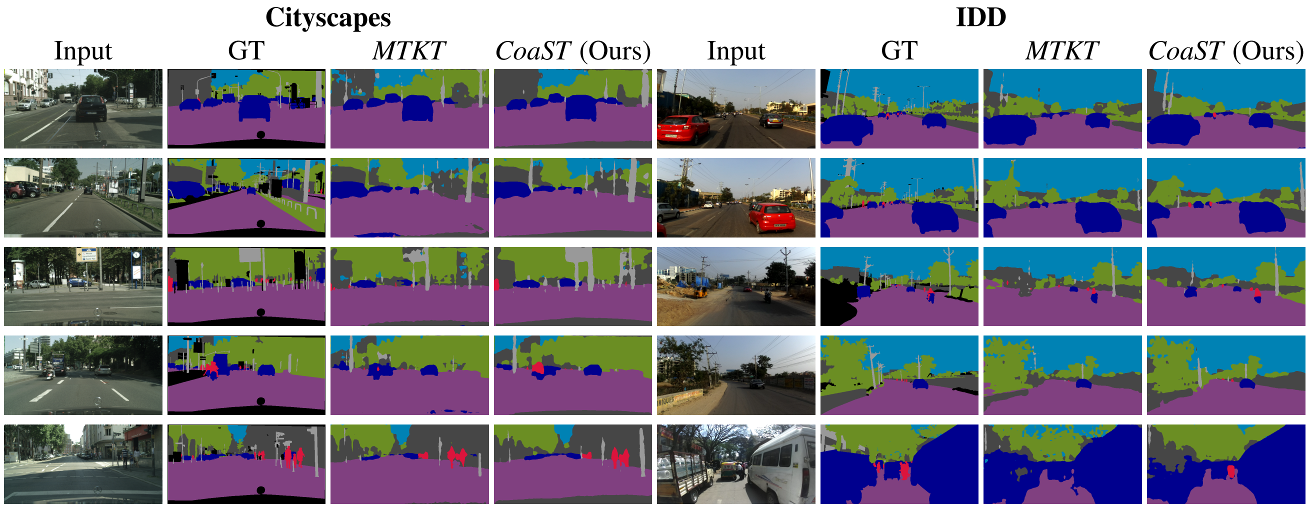


Figure 3: Qualitative comparison with *MTKT* on the *7-classes* benchmark and in the GTA5 to Cityscapes and IDD setting.

tural change in the classifiers. This variant is referred to as *MTKT**. Then we ablate different parts of our model: (i) uses a simple Self-training with pseudo-labeling without cross domain interactions. (ii) performs style-transfer and employs the cross-domain pseudo-label loss $\mathcal{L}_{pl}^{sty}$ in Eq. (10). (iii) adds the consistency loss given in Eq. (7). (iv) employs our rectified loss but does not use the consistency loss. Finally, (v) denotes our full models.

The lower performance of *MTKT** demonstrates that the higher performance of *CoaST* is not due to the use of a different classifier. Then, we can observe that (i) underperforms *MTKT** showing that naively replacing adversarial training by self-training does not work well. Adding *CrossDoNorm* in (ii) results in a 4% gain that is further increased when a consistency loss is added (see (iii)). Cooperation between domains can be also obtained by introducing cross-domain rectification (see (iv)) but the experiments show that combining both consistency and pseudo label rectification leads to the best performance.

To complete this ablation study, we evaluate different solutions to assess the rectification weights w_i in the same setting as Tab. 4. We consider different possibilities. Cooperative cross domain rectification can be replaced by the consistency between predictions obtained with multiple dropout sampling [9]. An auxiliary network can also be employed as in [42] to estimate the uncertainty. Average mIoUs are reported in Tab. 5 with these different rectification approaches. We observe our approach which benefits from the multiple target domains, achieves a 2.4% gain, which demonstrates that leveraging the multiple target domains is essential to achieve robust pseudo-labeling.

5. Conclusion

We presented *CoaST*, a new method for multi-target domain adaptation in semantic segmentation. We introduced a self-training strategy that uses pseudo-labels in conjunction with style-transfer to favor consistency between classifiers. Besides, we employed consistency between the predictions from the different classifiers as an uncertainty measure allowing better use of the pseudo-labels. We conducted experiments in two benchmarks and several settings and demonstrated that the proposed method outperforms state-of-the-art approaches.

Acknowledgements. This paper has been partially supported by the French National Research Agency (ANR) in the framework of its Technological Research JCJC pro- gram (Odace, project ANR-20-CE23-0027), NSFC (62176155), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), the EU ISFP PROTECTOR (101034216) project and the EU H2020 MARVEL (957337) project.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12):2481–2495, 2017.

[2] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1900–1909, 2019.

[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)*, 40(4):834–848, 2017.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1791–1800, 2019.

[6] Ziliang Chen, Jingyu Zhuang, Xiaodan Liang, and Liang Lin. Blending-target domain adaptation by adversarial meta-adaptation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2248–2257, 2019.

[7] 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)*, pages 3213–3223, 2016.

[8] Gabriela Csurka, Riccardo Volpi, and Boris Chidlovskii. Unsupervised domain adaptation for semantic image segmentation: a comprehensive survey. *arXiv preprint arXiv:2112.03241*, 2021.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016.

[10] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015.

[11] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing (TIP)*, 29:3993–4002, 2020.

[12] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. In *Workshop at Advances in Neural Information Processing Systems (NIPS)*, 2014.

[13] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 1989–1998, 2018.

[14] Takashi Isobe, Xu Jia, Shuaijun Chen, Jianzhong He, Yongjie Shi, Jianzhuang Liu, Huchuan Lu, and Shengjin Wang. Multi-target domain adaptation with collaborative consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8187–8196, 2021.

[15] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12975–12984, 2020.

[16] Seunghun Lee, Wonhyeok Choi, Changjae Kim, Minwoo Choi, and Sunghoon Im. Adas: A direct adaptation strategy for multi-target domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19196–19206, 2022.

[17] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 440–456, 2020.

[18] Peilun Li, Xiaodan Liang, Daoyuan Jia, and Eric P Xing. Semantic-aware grad-gan for virtual-to-real urban scene adaption. In *British Machine Vision Conference (BMVC)*, page 73, 2018.

[19] 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)*, pages 6936–6945, 2019.

[20] 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 (CVPR)*, pages 4500–4509, 2018.

[21] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4990–4999, 2017.

[22] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning (ICML)*, pages 5102–5112, 2019.

[23] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 102–118, 2016.

[24] Subhankar Roy, Evgeny Krivosheev, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Curriculum graph co-teaching for multi-target domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5351–5360, 2021.

[25] Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Multi-target adversarial frameworks for domain adaptation in semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9072–9081, 2021.

[26] Tong Shen, Dong Gong, Wei Zhang, Chunhua Shen, and Tao Mei. Regularizing proxies with multi-adversarial training for unsupervised domain-adaptive semantic segmentation. *arXiv preprint arXiv:1907.12282*, 2019.

[27] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems (NIPS)*, 33:596–608, 2020.

[28] Zhiqiang Tang, Yunhe Gao, Yi Zhu, Zhi Zhang, Mu Li, and Dimitris N Metaxas. Crossnorm and selfnorm for generalization under distribution shifts. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 52–61, 2021.

[29] Marco Toldo, Umberto Michieli, Gianluca Agresti, and Pietro Zanuttigh. Unsupervised domain adaptation for mobile semantic segmentation based on cycle consistency and feature alignment. *Image and Vision Computing (Image Vis Comput)*, 95:103889, 2020.

[30] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528, 2011.

[31] 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)*, pages 7472–7481, 2018.

[32] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[33] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1743–1751, 2019.

[34] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Mathieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2517–2526, 2019.

[35] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12635–12644, 2020.

[36] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 518–534, 2018.

[37] Xu Yang, Cheng Deng, Tongliang Liu, and Dacheng Tao. Heterogeneous graph attention network for unsupervised multiple-target domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[38] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4085–4095, 2020.

[39] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12414–12424, 2021.

[40] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *Advances in Neural Information Processing Systems (NIPS)*, 32, 2019.

[41] Zhedong Zheng and Yi Yang. Unsupervised scene adaptation with memory regularization in vivo. In *AAAI International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1076–1082, 2020.

[42] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision (IJCV)*, 2021. doi:`10.1007/s11263-020-01395-y`.

[43] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 633–641, 2017.

[44] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[45] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 568–583, 2018.

[46] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 289–305, 2018.