This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Multi-Target Adversarial Frameworks for Domain Adaptation in Semantic Segmentation

Antoine Saporta[1,2] Tuan-Hung Vu[2] Matthieu Cord[1,2] Patrick Pérez[2]
[1]Sorbonne University [2]Valeo.ai
{antoine.saporta, tuan-hung.vu, matthieu.cord, patrick.perez}@valeo.com

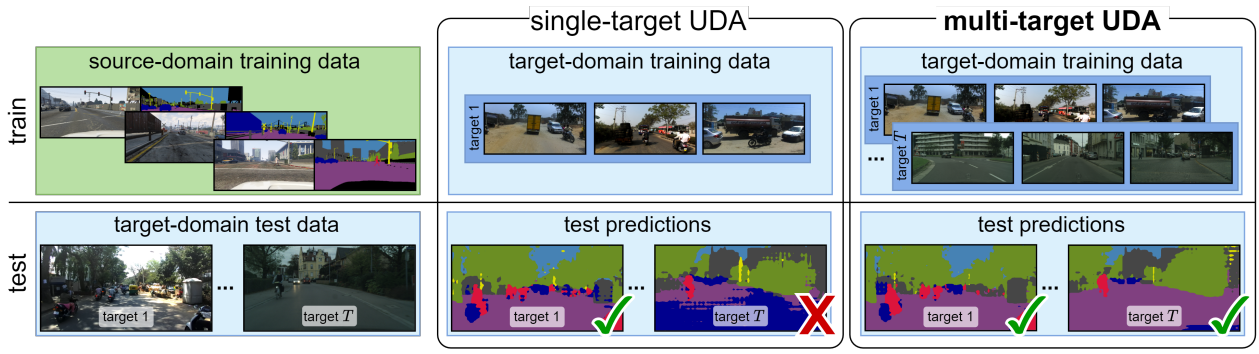

Figure 1: **Multi-target unsupervised domain adaptation (UDA) for semantic segmentation.** In the standard single-target setting, UDA methods produce good segmentation in the target domain they are trained on, but generalize poorly to other unseen domains. Multi-target UDA aims at excelling in the multiple domains the model is trained on. (*Top*) The information available during training is composed of source-domain RGB images with ground-truth semantic maps (green), here from GTA5, and unannotated RGB images from target domain(s) (blue), here from IDD ('target 1') and Cityscapes ('target T'). (*Bottom*) Test-time segmentation is on new images from the target domains, without knowing which domain they stem from.

Abstract

In this work, we address the task of unsupervised domain adaptation (UDA) for semantic segmentation in presence of multiple target domains: The objective is to train a single model that can handle all these domains at test time. Such a multi-target adaptation is crucial for a variety of scenarios that real-world autonomous systems must handle. It is a challenging setup since one faces not only the domain gap between the labeled source set and the unlabeled target set, but also the distribution shifts existing within the latter among the different target domains. To this end, we introduce two adversarial frameworks: (i) multi-discriminator*, which explicitly aligns each target domain to its counterparts, and (ii)* multi-target knowledge transfer*, which learns a target-agnostic model thanks to a multi-teacher/single-student distillation mechanism. The evaluation is done on four newly-proposed multi-target benchmarks for UDA in semantic segmentation. In all tested scenarios, our approaches consistently outperform baselines, setting competitive standards for the novel task.*

1. Introduction

Recent advances in domain adaptation help alleviate the labeling efforts required for training fully-supervised models, which is especially helpful for tasks like semantic segmentation. Most previous works address the single-target setting whose goal is to adapt from source to a particular target domain of interest, e.g. a specific urban area. However in practice, the perception system is often put to test in various scenarios including different cities, weathers or lighting conditions. To deal with multiple test distributions, one can straight-forwardly adopt single-target techniques by either (i) training multiple models for all target domains and adaptively activating one at test time or (ii) merging all target data and treat them as being drawn from a single target distribution. While the former strategy raises storage issues for embedded platforms and is difficult to scale up, the latter overlooks distribution shifts across different target domains.

In this work, we address multi-target unsupervised domain adaptation (UDA) in semantic segmentation. We aim to learn a single segmenter that achieves equally good performance in all target domains, simultaneously closing dis-

tribution gaps between labeled-unlabeled data (source *vs*. target) and among target domains (target *vs*. target). Our work is inline with recent efforts [3, 7, 15] toward more practical domain adaption settings for real-life applications. Different from most existing multi-target works that specifically consider image classification, we study here the more complex task of semantic segmentation.

We propose two adversarial UDA frameworks with architectures and learning schemes designed for the multi-target setup. The *multi-discriminator* model explicitly reduces both source-target and target-target domain gaps via adversarial learning – each target domain is aligned to its counterparts. Our second framework, called *multi-target knowledge transfer* (MTKT) relaxes the multi-target optimization complexity by adopting a multi-teacher/single-student mechanism. Each *target-specific* teacher handles a specific source-target domain gap via adversarial training; The *target-agnostic* student is learned from all teachers to achieve target-target alignment and to perform equally well in all target domains.

Our contributions can be summarized as follows:

- We propose two multi-target UDA frameworks for semantic segmentation.
- We define four different evaluation benchmarks for the task making use of existing semantic segmentation datasets, i.e. GTA5 [20], Cityscapes [4], Mapillary Vistas [17] and India Driving Dataset [24].
- We conduct extensive experiments of these two models against state-of-the-art baselines on the proposed benchmarks. Our approaches report consistent improvements over addressed baselines.

2. Related Works

Unsupervised Domain Adaptation for Semantic Segmentation. UDA is a setting that has received a lot of attention recently [10, 16, 22, 23, 25, 27]. The objective is to train a model on an unlabeled *target* domain by leveraging information from a labeled *source* domain, which is usually performed by aligning in some way the distributions between source and target domains. Some strategies include constraining the training with regularization such as maximum mean discrepancy (MMD) [16] or correlation alignment [22]. Most recent works, in particular in UDA for semantic segmentation, adopt an adversarial training strategy either at feature level [11] or output level [23, 25]. Some works also include a form of style transfer or image translation [10, 27, 28] to obtain target-looking source images while keeping source annotation. Additionally, a few works resort to "pseudo-labeling" [14, 21, 31] to refine their model with the help of automatically produced annotation in the target domain.

While these methods are really effective to adapt from one domain to another, their UDA setting is limited. In real-world scenarios, data may come from various domains: In urban scenes for instance, such domain variations may stem from different sensors, weather conditions or cities. While the underlying distribution is similar across domains, traditional UDA models are not robust to changes of target domains. Moreover, since they are specifically designed for single-source to single-target alignment, they fail to leverage information across more source or target domains.

Some recent works extend the standard UDA setting in semantic segmentation to more source or target domains. MADAN [30] tackles the task of multi-source domain adaptation for semantic segmentation where a model is trained using multiple labeled source domains and adapted on a single target domain. The authors first transform source images into adapted domains, similar to the target domain, then bring these new domains closer together with a sub-domain aggregation discriminator. They finally train the segmentation network by performing adversarial feature-level alignment between adapted and target domains. Closer to our setting, OCDA [15] addresses UDA with an *open compound* target domain: In this task, the target domain may be considered as a combination of multiple homogeneous target domains – for instance, similar weather conditions such as 'sunny', 'foggy', etc. – where the domain labels are not known during training. Moreover, previously unseen target domains may be encountered during inference. Unlike OCDA, our multi-target setting assumes that the domain of origin is known at training time and that no new domains are faced at test time (except in additional generalization experiments).

Multi-Target Domain Adaptation for Classification. Multi-target domain adaptation is still a fairly recent setting in the literature and mostly tackles classification tasks. Two main scenarios emerge in the works on this task. In the first one, even though the target is considered composed of multiple domains with gaps and misalignments, the domain labels are unknown during training and test. [19] proposes an architecture that extracts domain-invariant features by performing source-target domain disentanglement. Moreover, it also removes class-irrelevant features by adding a class disentanglement loss. In a similar setting, [3] presents an adversarial meta-adaptation network that both aligns source with mixed-target features and uses an unsupervised meta-learner to cluster the target inputs into k clusters, which are adversarially aligned. In the second scenario, the target identities are labeled on the training samples but remain unknown during inference. To handle it, [29] learns a common parameter dictionary from the different target domains and extracts the target model parameters by sparse representation; [7] adopts a disentanglement strategy by capturing separately both domain-specific private features and feature representations

by learning a domain classifier and a class label predictor, and trains a shared decoder to reconstruct the input sample from those disentangled representations.

In the present work, we adopt the second multi-target hypothesis: The target identities are known for the training samples but not for test ones. In fact, assuming that this information is available at test time is incompatible with some practical scenarios. More importantly, it would hinder generalization to previously-unseen domains, an important issue for autonomous systems in the wild. To the best of our knowledge, tackling semantic segmentation in this multi-target UDA scenario has only been proposed in a recently published concurrent work [12]. This work proposes to train a fully-fledged segmentation network for each domain and to ensure consistency among these multiple networks with image stylization between domains.

3. Adversarial Adaptation to Multiple Targets

3.1. Problem Formulation

Standard Unsupervised Domain Adaptation. The standard setting that is addressed in most UDA works is single source and single target. For adaptation, the model is trained on both a source-domain set $\mathcal{X}_\text{s}$ with the associated ground-truth set $\mathcal{Y}_\text{s}$ and an unlabeled target-domain set $\mathcal{X}_\text{t}$.

For semantic segmentation in C classes, sets $\mathcal{X}_\text{s}$ and $\mathcal{X}_\text{t}$ contain training images $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$, while the annotation set $\mathcal{Y}_\text{s} \subset [0,1]^{H\times W\times C}$ contains for each $\mathbf{x} \in \mathcal{X}_\text{s}$ a map $\mathbf{y}$ of $H \times W$ one-hot vectors indicating the ground-truth semantic classes for all pixels.

A segmentation network F takes an image $\mathbf{x}$ as input and predicts a soft-segmentation map $[\mathbf{P}_\mathbf{x}(\boldsymbol{k})]_{\boldsymbol{k}\in[H]\times[W]\times[C]}$.[1] The final segmentation map, $F(\mathbf{x})$, is given by max-score class, $\arg\max_{c\in[C]} \mathbf{P}_\mathbf{x}(i,j,c)$, at each pixel. UDA methods aim at aligning the distributions of the source-domain and target-domain training data such that, at test time, the segmenter F produces satisfactory predictions for target-domain inputs, without having been trained on labeled images from this domain.

Multi-Target UDA. In this work, we consider a different UDA scenario where $T \geq 2$ distinct target domains must be jointly handled. These target domains are represented by unlabeled training sets $\mathcal{X}_{\text{t},n} \subset \mathbb{R}^{H\times W\times 3}$, $n \in [T]$. Similar to the standard setting, we assume that the annotated training examples $(\mathbf{x},\mathbf{y}) \in \mathcal{X}_\text{s} \times \mathcal{Y}_\text{s}$ stem from a single source domain, a specific synthetic environment for instance. The main goal is to train a single segmenter F that achieves equally good results on all target-domain test sets. While the target domain of origin is known for all unlabeled training examples, we assume as in classification approaches in [7, 29] that this information is not accessible at test time.

[1] We use notation $[A] = \{1,\dots,A\}$ for $A \in \mathbb{N}^*$.

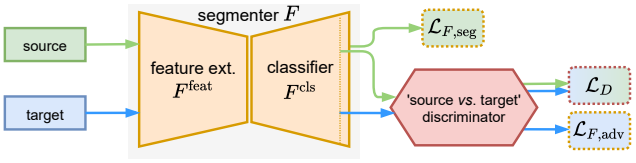

Figure 2: **Training in adversarial UDA.** The segmentation model under training ingests source-domain (green) and target-domain (blue) data. The former contribute to the segmentation loss, the latter to the adversarial loss, and both to the discriminator's loss. The three losses (dotted boxes) are defined in Eqs. (1) and (2).

3.2. Revisiting Adversarial UDA Approach

Recent state-of-the-art single-target UDA approaches are based on adversarial training to align source-target distributions. In such approaches, besides the segmenter F with parameters θ, an additional network D with parameters ϕ, called discriminator, is trained to play the segmenter's "adversary": D is learned to predict the domain of an input from suitable representations extracted by F such as intermediate or close-to-output features. Concurrently, F tries to produce results that can fool D into wrong discrimination. In semantic segmentation, adversarial approaches operating on close-to-prediction representations have the most success. AdaptSegnet [23] proposes to have adversarial learning on top of the soft-segmentation predictions $\mathbf{P}_\mathbf{x}$. AdvEnt [25] improves AdaptSegnet by using instead the "weighted self-information" maps $\mathbf{I}_\mathbf{x}$,[2] which brings additional entropy-minimization effect through adversarial alignment. Such single-target adversarial frameworks serve as the building block on top of which we develop our multi-target strategies. Hereafter, we denote $\mathbf{Q}_\mathbf{x}$ the used representation, which stands for either $\mathbf{P}_\mathbf{x}$ in [23] or $\mathbf{I}_\mathbf{x}$ in [25].

In practice, D is a fully-convolutional binary classifier with parameters ϕ. It classifies segmenter's output $\mathbf{Q}_\mathbf{x}$ into either class 1 (source) or 0 (target). To train the discriminator, we minimize the classification loss:

$$\mathcal{L}_D(\phi) = \big\langle \mathcal{L}_\text{BCE}(D(\mathbf{Q}_\mathbf{x}),1)\big\rangle_{\mathcal{X}_\text{s}} + \big\langle \mathcal{L}_\text{BCE}(D(\mathbf{Q}_\mathbf{x}),0)\big\rangle_{\mathcal{X}_\text{t}}, \tag{1}$$

where $\mathcal{L}_\text{BCE}$ stands for the binary cross-entropy loss and $\langle . \rangle$ denotes averaging over the set in subscript.

Concurrently, the segmenter F is trained over its parameters θ not only to minimize the supervised segmentation loss $\mathcal{L}_{F,\text{seg}}$ on source-domain data, but also to fool the discriminator D via minimizing an adversarial loss $\mathcal{L}_{F,\text{adv}}$. The final objective reads:

$$\mathcal{L}_F(\theta) = \underbrace{\big\langle \mathcal{L}_\text{CE}(\mathbf{x},\mathbf{y})\big\rangle_{\mathcal{X}_\text{s}}}_{\mathcal{L}_{F,\text{seg}}(\theta)} + \lambda_\text{adv} \underbrace{\big\langle \mathcal{L}_\text{BCE}(D(\mathbf{Q}_\mathbf{x}),1)\big\rangle_{\mathcal{X}_\text{t}}}_{\mathcal{L}_{F,\text{adv}}(\theta)}, \tag{2}$$

[2] Defined as $\mathbf{I}_\mathbf{x} = -\mathbf{P}_\mathbf{x} \log \mathbf{P}_\mathbf{x}$, with entry-wise operations.

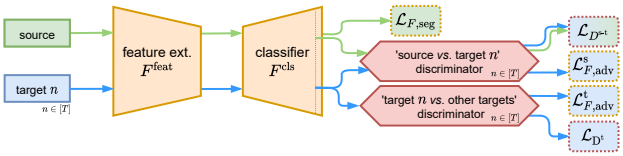

Figure 3: **Multi-discriminator approach to multi-target UDA**. With Multi-Dis., the segmenter is trained against two types of adversaries that discriminate respectively source *vs.* one target and one target *vs.* all other targets. The four types of adversarial losses are defined in Eqs. (3), (4), (6) and (7). Symbols and colors follow those in Figure 2.

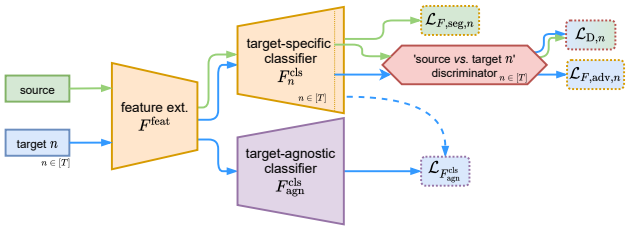

Figure 4: **Multi-target knowledge transfer approach to multi-target UDA.** With MTKT, a set of target-specific segmenters is first trained adversarially. Their knowledge is then jointly distilled to the target-agnostic segmenter whose loss (10) is not back-propagated into the target-specific branches (as indicated by the dotted arrow). Symbols and colors follow those in Figure 2.

with a weight λ_{adv} balancing the two terms; $\mathcal{L}_{\text{CE}}$ is the common cross-entropy loss. During training, one alternately minimizes the two losses $\mathcal{L}_D$ and $\mathcal{L}_F$.

Figure 2 provides a high-level view of the training flow in recent adversarial UDA approaches. For more details, we refer the readers to [23, 25] for instance. To later facilitate the presentation of our proposed strategies, the segmenter F is decoupled into a feature extractor, F^{feat}, followed by a pixel-wise classifier, F^{cls}.

Discussion. Approaches like [23, 25] handle only one source domain and one target domain. In our setting with multiple target domains, a simple strategy is to merge all target datasets into a single one and then to utilize an existing single-source single-target UDA framework. Such a strategy however disregards the inherent discrepancy among target domains. As we show in the experiments, this multi-target baseline is less effective than the proposed strategies which explicitly handle inter-target domain shifts. In what follows, we describe these two novel frameworks.

3.3. Multi-Target Frameworks

Multi-Discriminator. Our first strategy for multi-target UDA, called *multi-discriminator* ('Multi-Dis.' in short), relies on two types of discriminators to align each target domain with the source (source-target discriminators) and with other targets (target-target discriminators). Figure 3 illustrates this first approach.

Source-target adversarial alignment. We introduce a discriminator $D^{\text{s-t}}_n$ with parameters $\phi^{\text{s-t}}_n$ for each target domain n. It is learned to discriminate $\mathcal{X}_{\text{t},n}$ from the source set $\mathcal{X}_{\text{s}}$. By denoting $\mathcal{L}_{D^{\text{s-t}}_n}$ the minimization objective of this discriminator, defined as in (1) on domain n, we train these T source-target discriminators with the mean objective:

$$\mathcal{L}_{D^{\text{s-t}}}\big(\phi^{\text{s-t}}_{1:T}\big) = \frac{1}{T}\sum_{n\in[T]} \mathcal{L}_{D^{\text{s-t}}_n}\big(\phi^{\text{s-t}}_n\big). \tag{3}$$

Concurrently, the segmenter F is trained to fool these T discriminators with the adversarial objective:

$$\mathcal{L}^{\text{s}}_{F,\text{adv}}(\theta) = \frac{1}{T}\sum_{n\in[T]} \big\langle \mathcal{L}_{\text{BCE}}(D^{\text{s-t}}_n(\mathbf{Q}_{\mathbf{x}}), 1)\big\rangle_{\mathcal{X}_{\text{t},n}}. \tag{4}$$

Target-target adversarial alignment. In the above source-target alignment, the source acts as an anchor for each target to "pull" closer the other targets. However, as this alignment is imperfect, there remain gaps across targets, which we propose to reduce further by additional target-target alignments. To this end, we introduce for each target domain n a discriminator D^{t}_n with parameters ϕ^{t}_n that classifies $\mathcal{X}_{\text{t},n}$ (class 1) *vs.* all other target domains $\mathcal{X}_{\text{t},k}$, $k \neq n$ (class 0), resulting in T 1-*vs.*-all discriminators. The target-target discriminator D^{t}_n is trained by minimizing the loss

$$\mathcal{L}_{D^{\text{t}}_n}\big(\phi^{\text{t}}_n\big) = \big\langle \mathcal{L}_{\text{BCE}}(D^{\text{t}}_n(\mathbf{Q}_{\mathbf{x}}), 1)\big\rangle_{\mathcal{X}_{\text{t},n}} + \big\langle \mathcal{L}_{\text{BCE}}(D^{\text{t}}_n(\mathbf{Q}_{\mathbf{x}}), 0)\big\rangle_{\bigcup_{k\neq n}\mathcal{X}_{\text{t},k}}. \tag{5}$$

The collective objective of all target-target discriminators now reads:

$$\mathcal{L}_{D^{\text{t}}}\big(\phi^{\text{t}}_{1:T}\big) = \frac{1}{T}\sum_{n\in[T]} \mathcal{L}_{D^{\text{t}}_n}\big(\phi^{\text{t}}_n\big). \tag{6}$$

The segmenter F tries to fool all the target-target discriminators by minimizing the adversarial loss:

$$\mathcal{L}^{\text{t}}_{F,\text{adv}}(\theta) = \frac{1}{T}\sum_{n\in[T]} \big\langle \mathcal{L}_{\text{BCE}}(D^{\text{t}}_n(\mathbf{Q}_{\mathbf{x}}), 1)\big\rangle_{\bigcup_{k\neq n}\mathcal{X}_{\text{t},k}}. \tag{7}$$

To sum up, the segmenter F is trained by minimizing over θ the objective:

$$\mathcal{L}_F = \mathcal{L}_{F,\text{seg}} + \lambda^{\text{s}}_{\text{adv}}\mathcal{L}^{\text{s}}_{F,\text{adv}} + \lambda^{\text{t}}_{\text{adv}}\mathcal{L}^{\text{t}}_{F,\text{adv}}, \tag{8}$$

with weights $\lambda^{\text{s}}_{\text{adv}}$ and $\lambda^{\text{t}}_{\text{adv}}$ to balance the adversarial terms.

Multi-Target Knowledge Transfer. The main driving force in prediction-level adversarial approaches [23, 25] is the adjustment of the decision boundaries. Alignment in feature space then follows to comply with adjusted boundaries. We thus stress the importance of classifier design in the multi-target UDA scenario. In our multi-discriminator

approach, one classifier simultaneously handles multiple domain shifts, either source-target or target-target. The main challenge is the instability of adversarial training, which is amplified if several adversarial losses are jointly minimized. Such an issue is particularly problematic in the early training phase when most target predictions are very noisy. To address this challenge, we propose the *multi-target knowledge transfer* (MTKT) framework, with novel network design and learning scheme which do not rely on the joint minimization of multiple adversarial losses over the same classifier module, hopefully reducing the instability of the training. Figure 4 shows the MTKT architecture.

The classification part of the network is first re-designed with T *target-specific* instrumental classifiers, F_n^{cls}, $n \in [T]$, based on the same feature extractor F^{feat}, each handling one specific source-target domain shift. Such an architecture allows separate output-space adversarial alignment for each specific source-target pair, alleviating the instability problem. For each target-specific classifier F_n^{cls}, we introduce a domain discriminator D_n^{t} as to classify source *vs.* target n. The training objectives are similar to those used in single-target models (Eqs. 1 and 2).

We then introduce a *target-agnostic* classification branch $F_{\text{agn}}^{\text{cls}}$ that fuses all the knowledge transferred from the T target-specific classifiers. This target-agnostic classifier is the final product of the approach, *i.e.*, the one used at test time when domain knowledge is not available.

The knowledge from the T "teachers" is transferred to the target-agnostic "student" via minimizing the Kullback-Leibler divergence [9] between teachers' and student's predictions on target domains. In details, for a given sample $\mathbf{x} \in \mathcal{X}_{\text{t},n}$, we compute the KL loss

$$\mathcal{L}_{\text{KL},n}(\mathbf{x}) = \sum_{\boldsymbol{k} \in [H] \times [W] \times [C]} \mathbf{P}_{n,\mathbf{x}}(\boldsymbol{k}) \log \frac{\mathbf{P}_{n,\mathbf{x}}(\boldsymbol{k})}{\mathbf{P}_{\mathbf{x}}(\boldsymbol{k})}, \quad (9)$$

where $\mathbf{P}_{n,\mathbf{x}}$ and $\mathbf{P}_{\mathbf{x}}$ are soft-segmentation predictions coming from the target-specific F_n^{cls} and the target-agnostic $F_{\text{agn}}^{\text{cls}}$ respectively. The minimization objective of the target-agnostic classifier $F_{\text{agn}}^{\text{cls}}$ over the segmenter's parameters (including feature extractor's) then reads:

$$\mathcal{L}_{F_{\text{agn}}^{\text{cls}}}(\theta) = \frac{1}{T} \sum_{n \in [T]} \left\langle \mathcal{L}_{\text{KL},n}(\mathbf{x}) \right\rangle_{\mathcal{X}_{\text{t},n}}. \quad (10)$$

Minimizing KL losses helps $F_{\text{agn}}^{\text{cls}}$ adjust its decision boundaries toward good behavior in all T target domains. As the KL loss is back-propagated through the feature extractor, such an adjustment results in implicit alignment in target feature space, which overall mitigates the distribution shifts between the T domains.

Discussion. Unlike Multi-Dis., the multi-teacher/single-student mechanism in MTKT avoids direct alignment between unlabeled parts. The target-agnostic classifier is encouraged to adjust its decision boundaries to favor all the target-specific teachers, thus helping cross-target alignment.

Although we build our frameworks over output-space alignment [25, 23], note that they could be adapted to other adversarial feature-alignment methods [11]. Moreover, orthogonal approaches like pseudo-labeling can also be included in our frameworks and we show some experiments with such an addition in Section 4.3.

4. Experiments

4.1. Experimental Details

Datasets. We build our experiments on four urban driving datasets, one being synthetic and the three others being recorded in various geographic locations:

- GTA5 [20] is a dataset of 24,966 labeled synthetic images generated from the eponymous video game;
- Cityscapes [4] contains labeled urban scenes from cities around Germany, split in training and validation sets of 2,975 and 500 samples respectively;
- IDD [24] is an Indian urban dataset having 6,993 training and 981 validation labeled scenes;
- Mapillary Vistas [17] is a dataset collected in multiple cities around the world, which is composed of 18,000 training and 2,000 validation labeled scenes.

Though all containing urban scenes, the four datasets have different labeling policies and semantic granularity. We follow the protocol used in [13, 26] and standardize the label set with 7 super classes, common to all four datasets: *flat*, *construction*, *object*, *nature*, *sky*, *human* and *vehicle*. The mapping from original classes to these super classes is given in the Supplementary Material.

When Cityscapes, IDD or Mapillary are used as target domain, only unlabeled images from them are used for training, by definition of the UDA problem.

Implementation Details. Our experiments are conducted with PyTorch [18]. The adversarial framework is based on AdvEnt's published code.[3] We adopt DeepLab-V2 [2] as the semantic segmentation model, built upon the ResNet-101 [8] backbone initialized with ImageNet [5] pre-trained weights. The segmenters are trained by Stochastic Gradient Descent [1] with learning rate 2.5×10^{-4}, momentum 0.9 and weight decay 10^{-4}. We train the discriminators using an Adam optimizer [6] with learning rate 10^{-4}. All experiments were conducted at the 640×320 resolution.

For MTKT, we "warm up" the target-specific branches for 20,000 iterations before training the target-agnostic branch. The warm-up step avoids distillation of noisy target predictions in the early phase, which helps stabilize target-agnostic training.

[3] https://github.com/valeoai/ADVENT

GTA5 → Cityscapes + Mapillary											
Method	Target	Train	flat	constr.	object	nature	sky	human	vehicle	mIoU	mIoU Avg.
Single-Target Baselines [25]	Cityscapes	✓	93.5	80.5	26.0	78.5	78.5	55.1	76.4	69.8 (*)	66.6
	Mapillary	-	86.8	69.0	30.2	71.2	91.5	35.3	59.5	63.4$_{\downarrow 6.2}$	
	Cityscapes	-	89.3	79.3	19.5	76.9	84.6	47.7	63.0	65.8$_{\downarrow 4.0}$	67.7
	Mapillary	✓	89.5	72.6	31.0	75.3	94.1	50.7	73.8	69.6 (*)	
Multi-Target Baseline [25]	Cityscapes	✓	93.1	80.5	24.0	77.9	81.0	52.5	75.0	69.1$_{\downarrow 0.7}$	68.9
	Mapillary	✓	90.0	71.3	31.1	73.0	92.6	46.6	76.6	68.7$_{\downarrow 0.9}$	
Multi-Dis.	Cityscapes	✓	94.5	80.8	22.2	79.2	82.1	47.0	79.0	69.3$_{\downarrow 0.5}$	69.5
	Mapillary	✓	89.4	71.2	29.5	76.2	93.6	50.4	78.3	69.8$_{\uparrow 0.2}$	
MTKT	Cityscapes	✓	95.0	81.6	23.6	80.1	83.6	53.7	79.8	**71.1**$_{\uparrow 1.3}$	**70.9**
	Mapillary	✓	90.6	73.3	31.0	75.3	94.5	52.2	79.8	**70.8**$_{\uparrow 1.2}$	

Table 1: **Semantic segmentation performance on GTA5 → Cityscapes + Mapillary.** Per-class IoU (%), per-domain mean IoU ('mIoU') and mIoU averaged over domains ('mIoU Avg.'); mIoU gain (green) or loss (red) w.r.t. corresponding per-target baselines (marked as '*'); 'train': indication of the unlabeled target data used for training.

GTA5 → Cityscapes + IDD											
Method	Target	Train	flat	constr.	object	nature	sky	human	vehicle	mIoU	mIoU Avg.
Single-Target Baselines [25]	Cityscapes	✓	93.5	80.5	26.0	78.5	78.5	55.1	76.4	69.8 (*)	66.5
	IDD	-	91.3	52.3	13.3	76.1	88.7	46.7	74.8	63.3$_{\downarrow 1.8}$	
	Cityscapes	-	78.6	79.2	24.8	77.6	83.6	48.7	44.8	62.5$_{\downarrow 7.3}$	63.8
	IDD	✓	91.2	53.1	16.0	78.2	90.7	47.9	78.9	65.1 (*)	
Multi-Target Baseline [25]	Cityscapes	✓	93.9	80.2	26.2	79.0	80.5	52.5	78.0	70.0$_{\uparrow 0.2}$	67.4
	IDD	✓	91.8	54.5	14.4	76.8	90.3	47.5	78.3	64.8$_{\downarrow 0.3}$	
Multi-Dis.	Cityscapes	✓	94.3	80.7	20.9	79.3	82.6	48.5	76.2	68.9$_{\downarrow 0.9}$	67.3
	IDD	✓	92.3	55.0	12.2	77.7	92.4	51.0	80.2	65.7$_{\uparrow 0.6}$	
MTKT	Cityscapes	✓	94.5	82.0	23.7	80.1	84.0	51.0	77.6	**70.4**$_{\uparrow 0.5}$	**68.2**
	IDD	✓	91.4	56.6	13.2	77.3	91.4	51.4	79.9	**65.9**$_{\uparrow 0.8}$	

Table 2: **Semantic segmentation performance on GTA5 → Cityscapes + IDD.** Organization as in Tab. 1.

GTA5 → Cityscapes + Mapillary + IDD											
Method	Target	Train	flat	constr.	object	nature	sky	human	vehicle	mIoU	mIoU Avg.
Single-Target Baselines [25]	Cityscapes	✓	93.5	80.5	26.0	78.5	78.5	55.1	76.4	69.8 (*)	65.5
	Mapillary	-	86.8	69.0	30.2	71.2	91.5	35.3	59.5	63.3$_{\downarrow 6.3}$	
	IDD	-	91.3	52.3	13.3	76.1	88.7	46.7	74.8	63.3$_{\downarrow 1.8}$	
	Cityscapes	-	89.3	79.3	19.5	76.9	84.6	47.7	63.0	65.8$_{\downarrow 4.0}$	66.7
	Mapillary	✓	89.5	72.6	31.0	75.3	94.1	50.7	73.8	69.6 (*)	
	IDD	-	91.7	54.3	13.0	77.3	92.3	47.4	76.8	64.7$_{\downarrow 0.4}$	
	Cityscapes	-	78.6	79.2	24.8	77.6	83.6	48.7	44.8	62.5$_{\downarrow 7.3}$	65.5
	Mapillary	-	88.5	71.2	32.4	72.8	92.8	51.3	73.7	69.0$_{\downarrow 0.6}$	
	IDD	✓	91.2	53.1	16.0	78.2	90.7	47.9	78.9	65.1 (*)	
Multi-Target Baseline [25]	Cityscapes	✓	93.6	80.6	26.4	78.1	81.5	51.9	76.4	69.8	67.8
	Mapillary	✓	89.2	72.4	32.4	73.0	92.7	41.6	74.9	68.0$_{\downarrow 1.6}$	
	IDD	✓	92.0	54.6	15.7	77.2	90.5	50.8	78.6	65.6$_{\uparrow 0.5}$	
Multi-Dis.	Cityscapes	✓	94.6	80.0	20.6	79.3	84.1	44.6	78.2	68.8$_{\downarrow 1.0}$	68.2
	Mapillary	✓	89.0	72.5	29.3	75.5	94.7	50.3	78.9	70.0$_{\uparrow 0.4}$	
	IDD	✓	91.6	54.2	13.1	78.4	93.1	49.6	80.3	65.8$_{\uparrow 0.7}$	
MTKT	Cityscapes	✓	94.6	80.7	23.8	79.0	84.5	51.0	79.2	**70.4**$_{\uparrow 0.6}$	**69.1**
	Mapillary	✓	90.5	73.7	32.5	75.5	94.3	51.2	80.2	**71.1**$_{\uparrow 1.5}$	
	IDD	✓	91.7	55.6	14.5	78.0	92.6	49.8	79.4	**65.9**$_{\uparrow 0.8}$	

Table 3: **Results on GTA5 → Cityscapes + Mapillary + IDD ($T = 3$).** Organization as in Tab. 1.

4.2. Main results

We consider four setups, varying the type of domain-shift ('syn-2-real' or 'city-2-city') or the number T of targets (two to three domains). To measure per-target segmentation performance, we use the standard mean Intersection-over-Union (mIoU) metric. For multi-target performance, we report the mIoU averaged over the target domains; Using the average helps mitigate the potential bias caused by target evaluation sets with substantially different sizes.

GTA5 → Cityscapes + Mapillary. Table 1 reports segmentation results on the two target validation sets of Cityscapes and Mapillary; GTA5 is the source domain in this setup. For comparison, we consider the single-target AdvEnt models, i.e. trained on either Cityscapes or Mapillary unlabeled images. We have also the multi-target AdvEnt model, denoted as 'Multi-Target Baseline' in Table 1, which is trained on the merging of the two targets. For all models, including the single-target ones, we report both per-target and average mIoUs. The two rows marked with '(*)' indicate results of the single-target models on the same domains used for training, regarded as per-target baselines.

Single-target baselines achieve worse average mIoU than those trained on both domains, which indicates the benefit of having access to diverse data from multiple domains during training. Our proposed approaches outperform the multi-target baseline with mIoU gains of $+0.6\%$ for multi-discriminator and $+2.0\%$ for MTKT. Looking closer at the per-target results, we observe unfavorable performance if one directly transfers single-target models to a new domain. Indeed, testing the Cityscapes-only model on Mapillary results in a drop of -6.2% mIoU compared to the reference performance and a similar drastic drop is seen for Mapillary-only model on Cityscapes. Especially we notice important degradation on safety-critical classes like *human* or *vehicle* using those single-target models. The multi-discriminator model achieves comparable mIoUs as the per-target baselines. The MTKT model improves over the per-target baselines by significant margin, i.e. $+1.3\%$ on Cityscapes and $+1.2\%$ on Mapillary. Such results highlight the merit of the proposed strategies, especially MTKT. Note that adding adversarial training on the target-agnostic branch of MTKT hinders the alignment effect, reducing the performance by 0.9% mIoU Avg.

GTA5 → Cityscapes +IDD. We experiment with another syn-2-real setup in which the two target datasets have noticeably different landscapes, i.e. European cities in Cityscapes and Indian ones in IDD. Results are reported in Table 2. Here also, multi-target models outperform the single-target ones. In this setup, the performance of Multi-Dis. is comparable to the multi-target baseline's. We conjecture that the complex and unstable optimization problem in the multi-discriminator framework makes it difficult to achieve good alignment across targets, especially when the two targets are more noticeably different. With a dedicated architecture and learning scheme that alleviate such an optimization issue, the MTKT model achieves the best results, in terms of both per-target and average mIoUs.

We visualize some qualitative results in Figure 5.

GTA5 → Cityscapes + Mapillary + IDD. We consider a more challenging setup involving three target domains – Cityscapes, Mapillary and IDD – and show results in Table 3. With more target domains, the same conclusions hold. In terms of average mIoU, the multi-discriminator

Cityscapes → Mapillary + IDD											
Method	Target	Train	flat	constr.	object	nature	sky	human	vehicle	mIoU	mIoU Avg.
Single-Target Baselines [25]	Mapillary	✓	87.4	65.9	28.2	72.8	92.1	46.9	72.7	66.6 (*)	65.8
	IDD	-	91.8	52.2	15.9	80.2	91.1	45.7	77.6	65.0$_{\downarrow 2.3}$	
	Mapillary	-	88.2	70.0	28.5	75.4	93.6	49.1	76.7	68.8$_{\uparrow 2.2}$	68.0
	IDD	✓	93.2	53.4	16.5	83.4	93.4	51.4	79.5	67.3 (*)	
Multi-Target Baseline [25]	Mapillary	✓	87.7	65.9	29.0	73.2	91.5	47.9	75.7	67.3$_{\uparrow 0.7}$	67.0
	IDD	✓	93.3	53.0	17.2	82.8	92.2	49.3	79.6	66.8$_{\downarrow 0.5}$	
Multi-Dis.	Mapillary	✓	88.6	70.9	29.6	75.8	94.7	49.2	76.1	69.3$_{\uparrow 2.7}$	67.9
	IDD	✓	92.8	52.8	17.0	83.1	94.2	48.5	77.4	66.5$_{\downarrow 0.8}$	
MTKT	Mapillary	✓	88.3	70.4	31.6	75.9	94.4	50.9	77.0	**69.8**$_{\uparrow 3.2}$	**69.0**
	IDD	✓	93.6	54.9	18.6	84.0	94.5	53.4	79.2	**68.3**$_{\uparrow 1.0}$	

Table 4: **Results of city-2-city multi-target UDA on Cityscapes → Mapillary + IDD.** Organization as in Tab. 1.

model marginally improves over the multi-target baseline. The MTKT model significantly outperforms all other models with 69.1% mIoU Avg. Moreover, when compared to the per-target baselines, MTKT is the only model to show improvement on every target domain.

Cityscapes → Mapillary + IDD. Finally, we experiment on a realistic city-2-city setup with Cityscapes as source and Mapillary and IDD as target domains. The results are shown in Table 4. Interestingly, on Mapillary, the single-target model trained on IDD achieves better results than the one trained only on Mapillary. We conjecture that the domain gap between Cityscapes and Mapillary is less than the one between Cityscapes and IDD; The extra data diversity coming from IDD improves the single-target IDD-only model generalization and helps mitigate the small Cityscapes-Mapillary domain gap. Another observation is that the IDD-only model outperforms the multi-target baseline. This indicates the disadvantage of the naive dataset merging strategy: Not only complementary signals but also conflicting/negative ones get transferred. The two proposed models outperform the multi-target baseline; MTKT obtains the best performance overall. Again in this realistic setup, we showcase the advantages of our methods, especially the multi-target knowledge transfer model.

Conclusions. These four sets of experiments demonstrate that the proposed multi-target frameworks consistently deliver competitive performance on the multiple target domains they are trained for. MTKT always gives the best performance, both in per-target and average mIoUs, compared to the baselines and to the multi-discriminator model. Note that our models are compatible with techniques such as image translation [10, 27, 28] or pseudo-labeling self-training [14, 21, 31], from which they could benefit. In particular, we show next with additional experiments how to use pseudo-labeling [21] with MTKT.

4.3. Further Experiments

Additional Impact of Pseudo-Labeling. Pseudo-labeling (PL) is a strategy that has become quite popular in UDA for semantic segmentation [14, 21, 31]. It can be easily combined with our multi-target frameworks. Taking for instance the recently-proposed ESL [21], we consider three ways to adapt its pseudo-labeling strategy to the MTKT architecture. In all of them, we collect pseudo-labels in each target domain using the corresponding target-specific classifier and use them as additional self-supervision for these target-specific heads; In the second method we also use these pseudo-labels to restrict the back-propagation of the KL losses to pixels that are correctly classified according to these pseudo-labels; In the third method, they are also used to refine the target-agnostic classifier. We report in Table 5 the results of the models trained with these three PL-based refinement strategies on GTA5 → Cityscapes + IDD and compare them to the baseline trained with ESL. The three ways of extending MTKT with PL result in similar performance gains of at least +1.6% mIoU Avg. This demonstrates that knowledge transfer is complementary to pseudo-labeling. Moreover, MTKT with ESL outperforms the baseline with ESL by +1.7% mIoU Avg.

GTA5 → Cityscapes + IDD					
Method	M-T base.	M-T base. + PL	MTKT	MTKT + PL (1)	MTKT + PL (2)	MTKT + PL (3)
mIoU Avg.	67.4	68.9	68.2	69.8	69.7	**69.9**

Table 5: **Additional impact of pseudo-labeling (PL).** Trained models are refined with one step of ESL [21] (pseudo-labeling with predictive entropy as selection criteria). For MTKT, pseudo-labels are extracted for each target domain with the associated teacher head, and used either (1) to refine this head only, (2) to refine this head and to back-propagate KL-loss only on the pixels with predictions compliant with pseudo-labels or (3) to refine both this head and the target-agnostic model.

setup	Method	Test set	flat	constr.	object	nature	sky	human	vehicle	mIoU
G→C+I	M-T Baseline	Mapillary	88.4	71.0	**31.0**	72.4	92.0	37.4	74.7	66.7
	Multi-Dis.		89.2	72.1	21.7	73.8	**94.0**	34.8	75.9	65.9
	MTKT		**89.8**	**74.0**	30.4	**74.1**	93.6	**52.6**	**79.4**	**70.6**
G→C+M	M-T Baseline	IDD	**91.6**	54.7	**13.9**	76.5	90.9	48.3	77.5	64.8
	Multi-Dis.		91.2	54.6	12.9	**77.7**	**92.5**	50.3	78.6	**65.4**
	MTKT		91.5	**56.1**	12.3	76.1	90.9	**51.4**	**79.2**	**65.4**

Table 6: **Direct transfer to new target.** Multi-target models are tested on a new unseen domain: (Top) GTA5 → Cityscapes + IDD, tested on Mapillary; (Bottom) GTA5 → Cityscapes + Mapillary, tested on IDD.

Direct Transfer to a New Dataset. We consider a direct transfer setup in which the models see no images from the test domain during training: This experiment highlights how well the models can generalize to new previously-unseen domains. We report in Table 6 the results of such a direct transfer to a new dataset in different setups. The models are trained on GTA5 → Cityscapes + IDD (resp. on GTA5 → Cityscapes + Mapillary) and tested on Mapillary (resp. IDD). On both setups, MTKT shows better performance in terms of mIoU compared to the baselines on the

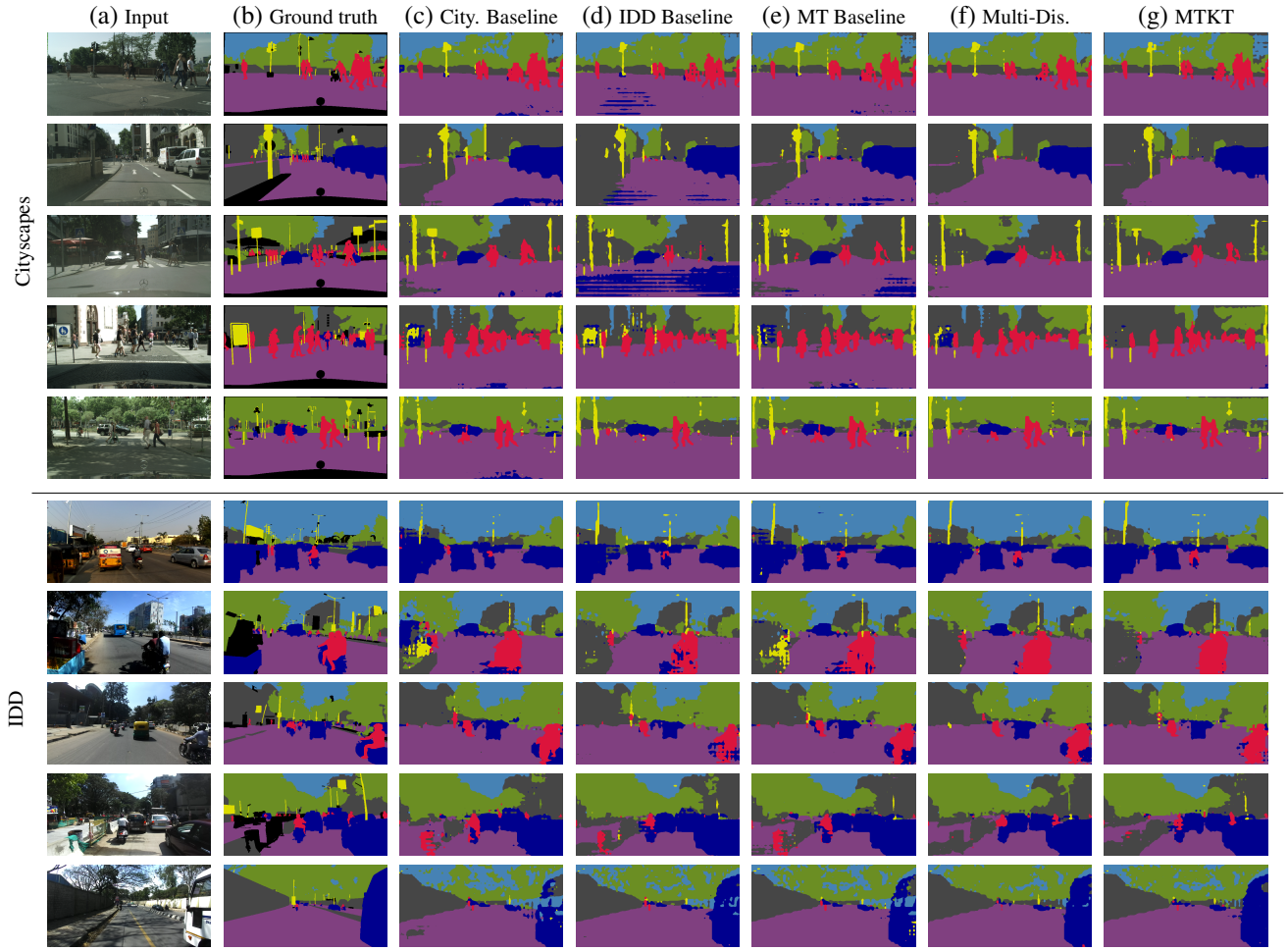

Figure 5: **Qualitative results in the GTA5 → Cityscapes + IDD setup.** (a) Test images from Cityscapes and IDD; (b) Ground-truth segmentation maps; Results of (c) single-target baseline trained on Cityscapes target, (d) single-target baseline trained on IDD target, (e) multi-target baseline, (f) proposed Multi-Dis. and (g) proposed MTKT. Both proposed multi-target frameworks give overall cleaner segmentation maps compared to the baselines.

new domain. In the first one in particular, with Mapillary as the new test domain, MTKT outperforms the multi-target baseline by $+3.9\%$. What is particularly noticeable in this setup is the performance on the *human* class: While we observe an IoU of around 50% in the main results on domain adaptation to Mapillary (e.g. in Tab. 1), the direct transfer results of the multi-target baseline and of Multi-Dis. drop under 38% on this class; Differently, MTKT manages to get similar performance with 52.8% IoU on *human*. This experiment hints at the ability of MTKT to better generalize to new unseen domains.

5. Conclusion

This work addresses the new problem of unsupervised adaptation to multiple target domains in semantic segmentation. We discuss the challenges that this UDA setup raises in terms of distribution alignment and of joint learning. That leads to two novel frameworks: The multi-discriminator approach extends single-target UDA to handle pair-wise domain alignment; The multi-target knowledge transfer approach alleviates the instability of multi-domain adversarial learning with a multi-teacher/single-student distillation mechanism. In the context of driving scenes, we propose four experimental setups, varying the type of source-target gaps and the number of target domains. Our approaches outperform all baselines on these four setups, which are representative of real-world applications. Further experiments additionally show that our frameworks can be combined to state-of-the-art pseudo-labeling strategies and that the proposed learning schemes help to generalize to previously-unseen datasets. This work thus contributes to the recent research line in domain adaptation toward more practical use cases. With the same goal, future research directions may consider more complex mixes of source and target domains, making use of several labeled and unlabeled datasets.

References

[1] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*. 2010. 5

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 5

[3] Ziliang Chen, Jingyu Zhuang, Xiaodan Liang, and Liang Lin. Blending-target domain adaptation by adversarial meta-adaptation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 5

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 5

[6] Jimmy Ba Diederik P. Kingma. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 5

[7] 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)*, 2020. 2, 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5

[10] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. 2018. 2, 7

[11] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv:1612.02649*, 2016. 2, 5

[12] 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)*, 2021. 3

[13] Kuan-Hui Lee, German Ros, Jie Li, and Adrien Gaidon. SPIGAN: Privileged adversarial learning from simulation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 5

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

[15] Ziwei Liu, Zhongqi Miao, Xingang Pan, Xiaohang Zhan, Dahua Lin, Stella X. Yu, and Boqing Gong. Open compound domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[16] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015. 2

[17] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 5

[18] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Workshop at Advances in Neural Information Processing Systems (NIPS)*, 2017. 5

[19] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning (ICML)*, 2019. 2

[20] 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)*, 2016. 2, 5

[21] Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Esl: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation. In *Workshop on Scalability in Autonomous Driving of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 7

[22] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2016. 2

[23] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 4, 5

[24] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and C V Jawahar. Dd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 2, 5

[25] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu 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)*, 2019. 2, 3, 4, 5, 6, 7

[26] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 5

[27] 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 European Conference on Computer Vision (ECCV)*, 2018. 2, 7

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

[29] Huanhuan Yu, Menglei Hu, and Songcan Chen. Multi-target unsupervised domain adaptation without exactly shared categories. *arXiv preprint arXiv:1809.00852*, 2018. 2, 3

[30] Sicheng Zhao, Bo Li, Xiangyu Yue, Yang Gu, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source domain adaptation for semantic segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[31] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2, 7