Dynamically Instance-Guided Adaptation: A Backward-free Approach for Test-Time Domain Adaptive Semantic Segmentation

Wei Wang[1] , Zhun Zhong[2], Weijie Wang[2], Xi Chen[3] , Charles Ling[1], Boyu Wang[1]*, Nicu Sebe[2]
[1]Western University [2]University of Trento [3]Huawei Noah's Ark Lab

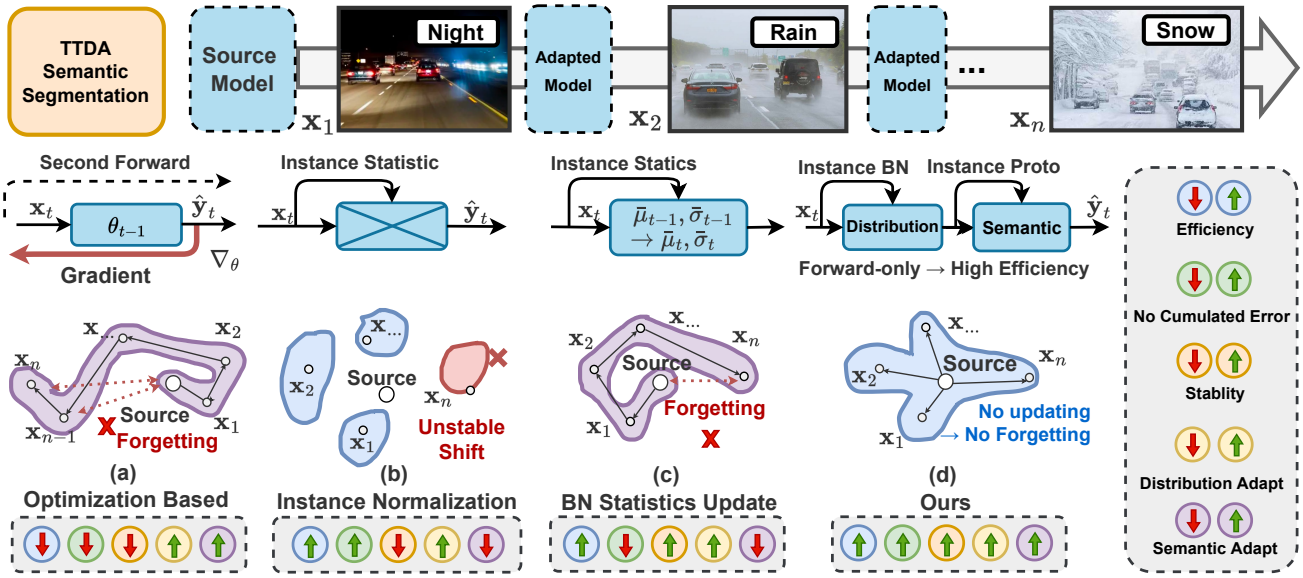

Figure 1. **Top**: Illustration of test-time domain adaptive semantic segmentation (TTDA-Seg). **Bottom**: Comparison with different TTDA methods. The proposed DIGA is a holistic method that has the properties of effectiveness (distribution&semantic adaptation and avoid unstable training&error accumulation) and efficiency (backward-free).

Abstract

In this paper, we study the application of Test-time domain adaptation in semantic segmentation (TTDA-Seg) where both efficiency and effectiveness are crucial. Existing methods either have low efficiency (e.g., backward optimization) or ignore semantic adaptation (e.g., distribution alignment). Besides, they would suffer from the accumulated errors caused by unstable optimization and abnormal distributions. To solve these problems, we propose a novel backward-free approach for TTDA-Seg, called Dynamically Instance-Guided Adaptation (DIGA). Our principle is utilizing each instance to dynamically guide its own adaptation in a non-parametric way, which avoids the error accumulation issue and expensive optimizing cost. Specifically, DIGA is composed of a distribution adaptation module (DAM) and a semantic adaptation module (SAM), enabling us to jointly adapt the model in two indispensable aspects. DAM mixes the instance and source BN statistics to encourage the model to capture robust representation. SAM combines the historical prototypes with instance-level prototypes to adjust semantic predictions, which can be associated with the parametric classifier to mutually benefit the final results. Extensive experiments evaluated on five target domains demonstrate the effectiveness and efficiency of the proposed method. Our DIGA establishes new state-of-the-art performance in TTDA-Seg. Source code is available at: https://github.com/Waybaba/DIGA.

*Corresponding author

1. Introduction

Semantic segmentation (Seg) [3, 40, 45, 46, 49] is a fundamental task in computer vision, which is an important step in the visual-based robot, autonomous driving and etc. Modern deep-learning techniques have achieved impressive success in segmentation. However, one serious drawback of them is that the segmentation models trained on one dataset (source domain) may undergo catastrophic performance degradation when applied to another dataset sam-

pled from a different distribution. This phenomenon will be even more serious under complex and ever-changing contexts, *e.g.*, autonomous driving.

To solve this well-known problem caused by domain shifts, researchers have devoted great effort to domain generalization (DG) [6,11,18,19,21,30] and domain adaptation (DA) [23,47,47,50]. Specifically, DG aims to learn generalized models with only labeled source data. Traditional DA attempts to adapt the model on the target domain by using both labeled source data and unlabeled target data. However, both learning paradigms have their own disadvantages. The performance of DG is limited especially when evaluated on a domain with a large gap from the source since it does not leverage target data [11]. DA assumes that the unlabeled target data are available in advance and can be chronically exploited to improve target performance. This assumption, however, can not always be satisfied in real-world applications. For example, when driving in a new city, the data are incoming sequentially and we expect the system to dynamically adapt to the ever-changing scenario.

To meet the real-world applications, [41] introduces the test-time domain adaptation (TTDA), which aims at adapting the model during the testing phase in an online fashion (see Fig. 1 Top). Generally, existing methods can be divided into two categories: backward-based methods [1,22,27,37,41] and backward-free methods [15,25,28,33]. The former category (see Fig. 1 (a)) focuses on optimizing the parameters of models with self-supervision losses, such as entropy loss [27,41]. In this way, both distribution adaptation and semantic adaptation can be achieved, which however has the following drawbacks. **(1) Low-Efficiency :** Due to the requirement of back-propagation, the computation cost will be multiplied, leading to low efficiency. **(2) Unstable Optimization & Error Accumulation:** Since the gradient is calculated with single sample by weak supervision, the randomness could be high thus leading to unstable optimization. Although this problem can be mitigated in some certain by increasing the testing batch size, it still cannot be solved well. In such cases, the accumulated errors may lead the model to forget the original well-learned knowledge and thus cause performance degradation.

The second category aims to adapt the model in the distribution level by updating statistics in batch normalization (BN) [25] layers, which is very efficient as it is directly implemented in forward propagation with a light computation cost. Instance normalization [28] (see Fig. 1 (b)) directly replaces the source statistics with those from each instance, which is sensitive to the target variations due to discarding the basic source knowledge and thus is unstable. Mirza et al [25] (see Fig. 1 (c)) study the impacts of updating the historical statistics by instance statistics with fixed momentum or dynamically fluctuating momentum. However, these methods also suffer from the error accumulation issue caused by abnormal target distributions as well as the neglect of semantic adaptation, both of which will result in inferior adaptation performance.

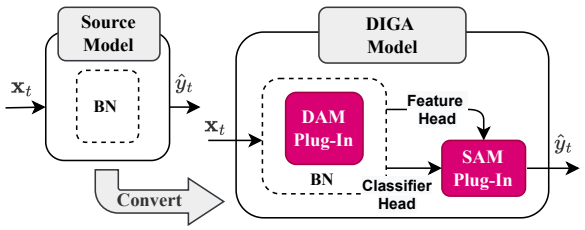

Figure 2. Illustration of the implementation of our DIGA. Given a source model, our DIGA can be readily equipped with only access to the BN layers, classifier head and feature head.

To this end, we propose a holistic approach (see Fig. 1 (d)), called Dynamically Instance-Guided Adaptation (DIGA), for TTDA-Seg, which takes into account both effectiveness and efficiency. The main idea of DIGA is leveraging each instance to dynamically its own adaptation in a non-parametric manner, which is efficient and can largely avoid the error accumulation issue. In addition, our DIGA is implemented in a considerate manner by injecting with distribution adaptation module (DAM) and semantic adaptation module (SAM). Specifically, in DAM, we compute the weighed sum of the source and current statistics in BN layers to adapt target distribution, which enables the model to obtain a more robust representation. In SAM, we build a dynamic non-parametric classifier by mixing the historical prototypes with instance-level prototypes, enabling us to adjust the semantic prediction. In addition, the non-parametric classifier can be associated with the parametric one, which can further benefit the adaptation results. Our contributions can be summarized as follows:

- **Efficiency.** We propose a backward-free approach for TTDA-Seg, which can be implemented within one forward propagation with a light computation cost.
- **Effectiveness.** We introduce a considerate approach to adapt the model in both distribution and semantic aspects. In addition, our method takes the mutual advantage of two types of classifiers to achieve further improvements.
- **Usability.** Our method is easy to implement and is model-agnostic, which can be readily injected into existing models (see Fig.2).
- **Promising Results.** We conduct experiments on three source domains and five target domains based on driving benchmarks and show that our method produces new state-of-the-art performance for TTDA-Seg. We also study the continual TTDA-Seg and verify the superiority of our method in this challenging task.

2. Related Work

Test-time Domain Adaptation (TTDA) aims to adapt models on the target domain only in test time. It is firstly proposed in TTT [37] and has been applied to many fields such as instance tracking [9], object detections [16] and reinforcement learning [12]. The early works (TTT [37] and its extensions [20, 22]) require an extra training process on source data, making it inapplicable when only the source model is available. In this paper, we focus on the more practical Fully Test-time Domain Adaptation setting proposed in [41]. Current fully TTDA methods can be categorized into two main branches: backward-based adaptation and backward-free adaptation. As the pioneer of the self-supervision adaptation methods, TENT [41] proposes to minimize the entropy by updating BN affine parameters during test time. EATA [27] shows that skipping low entropy samples would achieve higher efficiency and performance. Also, an updated regularization term is utilized to alleviate the forgetting problem. [1] introduce an efficient framework by introducing contrastive learning. The problem with these methods is that they cost a long time and large GPU memory due to backpropagation, which largely limited the application for real-time inference. As for the backward-free branch, most of the approaches focus on BN statistics adaptation. IN [28] directly uses batch statistics while Momenmtum [33] and DUA [25] use running average to update the statistics. This branch is much more efficient while they only work on distribution, ignoring the semantic adaptation, leading to discounted adaptation power. Besides the above two branches, T3A [14] proposes to denoise the classification results in the post-processing stage, where adaptation of the model itself is not well exploited.

Domain Adaptive for Semantic Segmentation (DASS) aims to bridge the domain gap between the training and testing data. The early works in DASS mainly focus on building adversarial training architectures to learn the domain-invariant features [24, 38, 43]. Complementary modules have been introduced to facilitate the training [24,38,40,43]. Another category exploits the self-training techniques such as entropy minimization [40] and pseudo-labeling [45, 46, 49, 51]. While these approaches require the co-existence of source-target data. Source-free DA (SFDA) is more practical and closer to our setting as they assume the source data is not available during adaptation. [40] proposes to recover source information by utilizing the BN statistics. MAS^3 [17] proposes to store source distribution information as prototypes and then use them during adaptation. [35] uses a multi-head structure to increase the reliability of pseudo-labeling for self-supervised training. Zhao et al. [48] present a special augmentation module to diversify samples with various patch styles at the feature level and then use them for generalization ability improvement. However, these methods can not handle TTDA well. Firstly, they do not consider the efficiency problem [17, 35, 40, 48]. Moreover, they often require to visit samples repeatedly in large batch sizes [17, 35, 48].

3. Methodology

Problem Definition. In test-time domain adaptation in semantic segmentation (TTDA-Seg), we are given a segmentation model $f_\theta : \mathbf{x} \rightarrow y$ pretrained on a source domain $\mathcal{D}_S$, which will be directly deployed to unseen domains for evaluation. Due to domain shifts, the model f_θ would normally produce a poor performance on unseen testing domains. The goal of TTDA-Seg is to adapt the model by utilizing continuously incoming testing data in an online fashion (see Fig. 1). For example, at each testing step t, the model f_θ receives an instance $\mathbf{x}_t$ and simultaneously performs adaptation as well as produces segmentation prediction $\hat{y}_t$. At the next step $t+1$, the model f_θ will perform adaptation and prediction on instance $\mathbf{x}_{t+1}$ without the access to previous data $\mathbf{x}_{1\rightarrow t}$.

3.1. Overview

In this section, we propose a Dynamically Instance-Guided Adaptation (DIGA) method for TTDA-Seg, which is backward-free and non-parametric. As shown in Fig. 3, our DIGA includes two adaptation modules, the distribution adaptation module (DAM) and the semantic adaptation module (SAM), which are both guided by instance-aware information. Specifically, given a testing sample, we first input it into the source pretrained model and perform distribution alignment by DAM in each BN layer. The distribution alignment is implemented by weighted summing of the source statistics and instance statistics. After this, we apply semantic adaptation at the last feature level by SAM, in which we build a dynamic non-parametric classifier by weighted mixing the historical prototypes with instance-aware prototypes. This allows us to adjust the semantic prediction. Lastly, we obtain the final prediction by taking the mutual advantage between the original parametric classifier and the dynamic non-parametric classifier.

In Fig. 3 (a-g), we show an illustration of how our DIGA helps to adapt the model with the guidance of instance-aware information. (a-d) Due to the large domain shifts (*e.g.*, light variations), the segmentation results on the target sample might be poor. After distribution alignment by DAM, the segmentation results could be improved, especially the instances that are similar to the source (e). However, there might still exist poorly-recognized pixels that are very different from the source. Our DAM further leverages the reliable pixels to guide the predictions of other pixels in a non-parametric way (f), enabling us to achieve more accurate results (g).

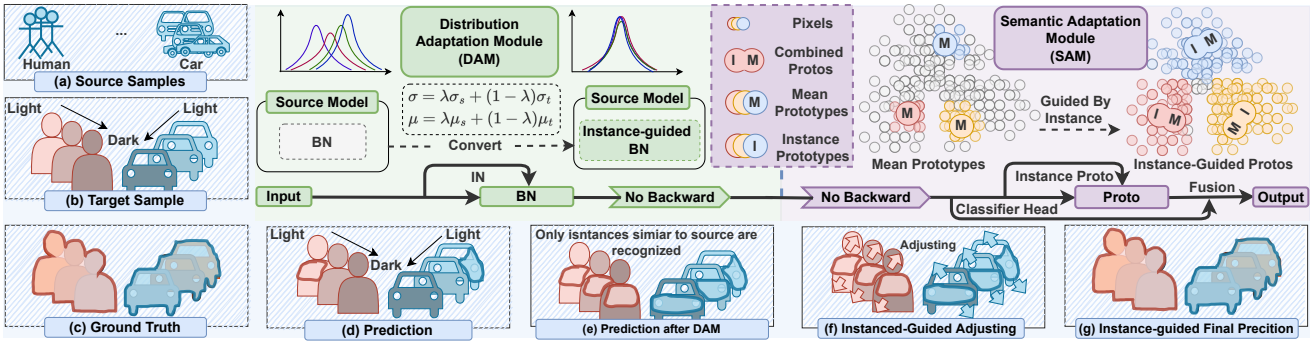

Figure 3. Illustration of the proposed DIGA. (a-d) The source model commonly produces poor results on unseen targets, due to significant domain shifts, such as light variation. To solve this problem, our DIGA is proposed to adapt the model in an online fashion, which consists of a distribution adaptation module (DAM) and a semantic adaptation module (SAM). Both modules are guided by instance-aware information (statistics for DAM and prototypes for SAM). The DAM is implemented in BN layers for distribution alignment (e). The SAM is implemented in the feature level for semantic alignment (f-g). The final prediction is obtained by the fusion of original classifier and non-parametric classifier built by SAM.

3.2. Distribution Adaptation Module (DAM)

The most common way to adapt the distribution is based on adversarial training [10, 19, 23] and minimization of distribution gap metrics [4, 36]. However, these methods are not suitable for TTDA due to limited available training data and the high-cost backpropagation. Recent works [25, 28, 33] show that the static mismatch between domains in the Batch Normalization (BN) layers is a major reason that causes the performance degradation in cross-domain testing. We thus first revisit the mechanism of BN. Specifically, for each BN layer, given the input feature representation F, the corresponding output is given by:

$$BN(F) = \gamma \frac{F - \mathbb{E}[F]}{\sqrt{Var[F]}} + \beta, \tag{1}$$

where γ and β are trainable parameters for scaling and shifting. $\mathbb{E}[F]$ and $Var[F]$ are expected value and variance of input feature F. In practice, due to the batch-wise training process, their values are calculated by running mean [42] during training as follows:

$$\begin{aligned} \bar{u}_t^S &= (1-\rho_{BN}) \cdot \bar{\mu}_{t-1}^S + \rho_{BN} \cdot \mu_{t-1}^S, \\ (\bar{\sigma}_t^S)^2 &= (1-\rho_{BN}) \cdot (\bar{\sigma}_{t-1}^S)^2 + \rho_{BN} \cdot (\sigma_{t-1}^S)^2, \end{aligned} \tag{2}$$

where $\bar{\mu}_t^s$ and $\bar{\sigma}_t^s$ are serving as estimation for $\mathbb{E}[F]$ and $Var[F]$ of source domain respectively.

Out of the motivation that the amount of training sample is usually much larger than the testing batch and thus more stable, the last value $\bar{\mu}_t^s, \bar{\sigma}_t^s$ would be frozen and serve as the estimation for $\mathbb{E}[F]$ and $Var[F]$ for the test data the during the test phase.

However, it has been shown that when applied to a different environment, the source statistics can hamper performance significantly. To solve this problem, DUA [42] proposes to adapt the statistics γ, β of BN layers to the target domain with a dynamic learning module. Despite the efficiency, its performance is still not satisfactory. One possible reason is that the updating rate is usually very small so that instance-level information is not fully considered during each instance evaluation.

Different from [25, 33], instead of updating the γ, β, the proposed Distribution Adaptation Module (DAM) dynamically merges the source and instance BN statistics to constitute the estimation $\bar{\mu}_t^T$ and $(\bar{\sigma}_t^T)^2$ for $\mathbb{E}[F], Var[F]$ as follows:

$$\begin{aligned} \bar{\mu}_t^T &= \lambda_{BN} \cdot \bar{\mu}^S + (1-\lambda_{BN}) \cdot \mu_t^T, \\ (\bar{\sigma}_t^T)^2 &= \lambda_{BN} \cdot (\bar{\sigma}^S)^2 + (1-\lambda_{BN}) \cdot (\sigma_t^T)^2, \end{aligned} \tag{3}$$

where μ_t^T and $(\sigma_t^T)^2$ are the mean and variance calculated with the t-th instance during testing.

3.3. Semantic Adaptation Module (SAM)

The proposed DAM is a category-agnostic since it only aligns the distribution of feature maps globally. However, category-specific is also important to segmentation adaptation because the distribution of each category varies a lot even in the same image. Hence, we argue that it is also important to implement semantic adaptation in TTDA-Seg. To achieve this, two straightforward methods are entropy maximization [41] and pseudo-labeling [22]. However, both of them require gradient-based backpropagation and thus limit the testing efficiency. Inspired by the *prototype-based* methods in few-shot learning [34] and domain adaptation [29,45], we introduce the semantic adaptation module (SAM) for category-specific adaptation.

As shown in Sec. 3.1, even though distribution alignment is implemented by DAM, the model still produces wrong predictions for pixels that are very different from the source. Fortunately, we could observe that pixels of the same ob-

ject share several of the same properties, *e.g.,* the appearance within a car, the same texture color within a road, the same outfit within a person, the light intensity within an image, etc. Motivated by this, we propose to leverage the similarities between pixels to further guide the recognition of wrongly recognized pixels. To this end, we propose the semantic adaptation module (SAM) to adjust the semantic predictions by dynamic instance-aware prototypes.

The segmentation model f_θ can be separated into two learnable parts: i) an encoder h_ϕ for dense visual feature extraction which maps each pixel $\mathbf{x}^{(h,w)}$ to feature $\mathbf{z}^{(h,w)} \in \mathbb{R}^D$, and ii) a classifier g_ψ for following prediction which maps $\mathbf{z}^{(h,w)}$ to a distribution $\hat{p}^{(h,w)}(c|\mathbf{x})$ over C classes. Formally, it can be denoted as:

$$\mathbf{z}_t^{(h,w)} = h_\phi^{(h,w)}(\mathbf{x}_t), \tag{4}$$

$$\hat{p}^{(h,w)}(c|\mathbf{x}_t) = g_\psi^{(h,w)}(\mathbf{z}_t). \tag{5}$$

The value of logits indicates the confidence of the corresponding classes it would belong to. Thus, the largest value $\max_c \hat{p}_{t,c}^{(h,w)}$ of the prediction distribution can be considered as the confidence of the prediction for one pixel. For one input image $\mathbf{x}_t$, we select pixels whose confidences are larger than $\mathcal{P}_0$ to calculate the centroids of each class in feature space, which are called as *instance-aware prototypes* $\mathbf{q}_t$ and can be formulated as follows:

$$\mathbf{q}_t^c = \frac{\sum^{H,W} \mathbf{z}_t^{(h,w)} \cdot \mathbb{I}(c_t^{(h,w)} = c, \max_c \hat{p}_{t,c}^{(h,w)} \geq \mathcal{P}_0)}{\sum^{H,W} \mathbb{I}(c_t^{(h,w)} = c, \max_c \hat{p}_{t,c}^{(h,w)} \geq \mathcal{P}_0)}. \tag{6}$$

Using instance-aware prototypes only may produce unstable predictions due to the instance variance. To make the prediction more stable, we additionally calculate the moving average of the prototypes of different instances for each category, which are called *historical prototypes*.

$$\bar{\mathbf{q}}_t^c = \rho_P \cdot \bar{\mathbf{q}}_{t-1}^c + (1-\rho_P)\mathbf{q}_t^c, \text{ with } \bar{\mathbf{q}}_0^c = \mathbf{q}_0^c. \tag{7}$$

Since the historical prototypes are calculated by averaging prototypes from a large number of target instances, they are more stable than instance-aware prototypes.

Given the instance-aware prototypes, we can obtain the instance-aware prediction for each class $p^{(h,w)}(c|\mathbf{x}_t, \mathbf{q})$ by:

$$p^{(h,w)}(c|\mathbf{x}_t, \mathbf{q}) = \frac{\exp(-\langle \mathbf{z}^{(h,w)}, \mathbf{q}_c \rangle)}{\sum_{c'=1}^{C} \exp(-\langle \mathbf{z}^{(h,w)}, \mathbf{q}_{c'} \rangle)}. \tag{8}$$

The historical prediction $p^{(h,w)}(c|\mathbf{x}_t, \bar{\mathbf{q}})$ could be obtained in a similar way.

By combing the predictions of the two types of prototypes, we form a dynamic non-parametric classifier and the predictions are formulated as:

$$\begin{aligned} \tilde{p}^{(h,w)}(c|\mathbf{x}_t) =& \lambda_P \cdot p^{(h,w)}(c|\mathbf{x}_t, \bar{\mathbf{q}}) \\ &+ (1-\lambda_P) \cdot p^{(h,w)}(c|\mathbf{x}_t, \mathbf{q}), \end{aligned} \tag{9}$$

Algorithm 1 DIGA (Testing Phase)

Input: Model f_θ, target testing sample $\mathbf{x}_t$.
Output: Prediction of $\mathbf{x}_t$.
1. Produce feature $\mathbf{z}_t$ and prediction $\hat{p}_t(\mathbf{x}_t)$ with distribution alignment of DAM (Eq. 3).
2. Calculate instance-aware prototypes $\mathbf{q}_t$ (Eq. 6).
3. Calculate historical prototypes $\bar{\mathbf{q}}_t$ (Eq. 7).
4. Calculate non-parametric predictions $\tilde{p}_t(\mathbf{x}_t)$ with SAM (Eq. 9).
5. Obtain final prediction $p(\mathbf{x}_t)$ by weighed fusion of $\hat{p}_t(\mathbf{x}_t)$ and $\tilde{p}_t(\mathbf{x}_t)$ (Eq. 10).
Return: $p(\mathbf{x}_t)$

where λ_P controls the importance of two types of prototypes.

3.4. Classifier Association

To this end, we could have two types of predictions: one from the original parametric classifier ($\hat{p}$) and one from the introduced non-parametric prototype classifier ($\tilde{p}$). To leverage the mutual benefit between them, we obtain the final prediction by weighted sum of them, formulated as:

$$p^{(h,w)} = \lambda_F \cdot \tilde{p}^{(h,w)}(c|\mathbf{x}_t) + (1-\lambda_F)\hat{p}^{(h,w)}(c|\mathbf{x}_t), \tag{10}$$

where λ_F balances the importance of two classifiers. The overall process of DIGA is shown in Alg. 1.

4. Experiment

4.1. Experimental Setup

Datasets. Following the previous works [3, 15, 40], we evaluate our method on sim2real scenarios. Specifically, for the source model, we pretrain it with three different source domains: GTA5 [31], Synthia [31], and GTA5+Synthia. GTA5 provides 24,971 images from video games with 19 semantic classes. Synthia includes 12,000 simulated images with 16 semantic classes. GTA5+Synthia is the combination of GTA5 and Synthia datasets. Performance are evaluated on five target domains: Cityscapes [7], BDD-100K [44], Mapillary [26], IDD [39], Cross-City [5]. We test the results on the validation sets, where the number of samples is {500, 1,000, 2,000, 100, and 400} for {Cityscapes, BDD-100K, Mapillary, IDD, Cross-City} respectively.

Evaluation. The mean intersection-over-union (mIoU) is used as the evaluation metric. As in [3,40], for source models pretrained on GTA5 and GTA5+Synthia, we report the mIoU of 19 shared semantic categories. Due to missing of annotations of some classes, we report the mIoU of 16 shared semantic classes for the model pretrained on the Synthia dataset.

Table 1. Comparison with state-of-the-art methods in terms of mIoU. The best score for each column is **highlighted**. CS: CityScapes, BDD: BDD100K, MA: Mapillary, IDD: IDD, CC: Cross-City. *: Use an extra augmented sample during adaptation. Avg.: Mean of mIoUs over five target domains.

Method	GTA5→						Synthia→						GTA5+Synthia →					
	CS	BDD	MA	IDD	CC	Avg.	CS	BDD	MA	IDD	CC	Avg.	CS	BDD	MA	IDD	CC	Avg.
Source [42]	35.87	29.89	38.67	38.05	30.03	34.50	30.87	21.01	31.12	26.23	31.96	28.24	37.00	28.85	41.56	39.88	32.93	36.04
Backward-based Methods																		
TENT [41]	37.30	31.53	38.29	38.96	30.59	35.33	34.89	16.99	33.46	26.23	31.68	28.65	39.39	25.19	37.32	39.51	32.84	34.85
EATA [27]	37.08	30.67	39.35	38.75	30.24	35.22	31.31	20.52	31.59	26.46	31.91	28.36	38.45	29.34	41.63	40.33	32.91	36.53
Backward-free Methods																		
IN [28]	34.25	29.64	35.01	29.8	23.87	30.51	29.53	19.33	21.92	22.08	28.24	24.22	37.09	28.81	36.02	30.99	28.63	32.31
Momentum [33]	38.12	32.42	40.79	38.74	30.2	36.05	32.84	22.51	31.12	27.24	32.23	29.45	39.61	31.72	41.79	39.88	33.17	36.66
DUA [25]	37.79	31.76	40.26	34.75	26.32	34.18	32.17	21.56	27.42	24.06	29.87	27.02	39.17	30.59	39.95	35.30	30.65	35.13
SITA* [15]	40.64	32.94	37.80	35.66	28.19	35.26	34.63	22.51	26.60	24.64	28.18	27.79	42.62	32.24	41.20	38.82	33.22	37.62
DIGA (Ours)	**45.81**	**35.78**	**44.25**	**42.73**	**33.72**	**40.46**	**41.85**	**29.09**	**36.54**	**38.36**	**36.78**	**36.52**	**46.43**	**33.87**	**43.51**	**42.08**	**34.41**	**40.06**

Implementation Details. Following previous works [3, 15, 25, 41], we use DeepLabV2 [2] as the segmentation model. The ResNet-101 [13] pretrained on ImageNet [8] is used as the backbone. It is worth mentioning that globally consistent parameter sets are used for our DIGA, which achieves consistently good performance in all experiments. Specifically, we set the momentum updating rate (ρ_P and ρ_{BN}) both to 0.1. The weights of DAM, SAM, and classifier association (λ_{BN}, λ_P and λ_F) are all set to 0.8. The confidence bar for prototype selection $\mathcal{P}_0$ is 0.9. All the experiments are conducted with one RTX3090 GPU.

4.2. Comparison with State of the Art

We first compare our method with the state-of-the-art approaches. Generally, the compared methods can be divided into two categories: backward-based methods and backward-free methods.

Backward-based methods: TENT [41] performs adapting by minimizing the output entropy and updating the learnable parameters of BN layers. As an extension to TENT [41], EATA [27] proposes to skip the high-entropy samples and only leverage reliable samples during model optimization, which can effectively increase testing efficiency. Both of them are initially designed for image classification. We implement them for TTDA-Seg by minimizing the entropy of pixel-level output. For EATA [27], we skip the low-entropy pixels during optimization. *Backward-free methods:* IN [28] uses instance statistics to replace source ones in BN at each testing step. Momentum [33] utilizes the instance statistics to update BN in a momentum-based manner. DUA [25] proposes a decaying strategy to adaptively control the momentum of BN updating. SITA [15] leverages extra augmented samples to obtain stable instance statistics, which are then mixed with the source statistics.

To make a fair comparison, we implement all the methods with the same source models. Note that, we report the results of the compared methods by selecting the best parameters for each source-target pair. In contrast, in our method, we only use one parameter setting for all experiments to better meet the real-world applications.

The following observations can be made from the results reported in Tab. 1. First, backward-based methods can consistently improve the performance when evaluating on CityScapes. However, the improvements on other target domains are limited or even negative. For example, when using Synthia as the source domain, TENT [41] increases the mIoU from 30.87% to 34.89% on CityScapes while largely reduces the mIoU from 21.01% to 16.99% for BDD100K. This indicates that using self-supervision only may not be a good choice for TTDA-Seg. Second, except for IN [28], the backward-free methods are generally effective on CityScapes and BDD100K while failing to achieve consistent improvements on other datasets, even though we have well-tuned them for each target domain. On the other hand, IN [28] largely reduces the average mIoU due to ignoring the source statistics. Third, the proposed DIGA consistently improves the mIoUs of the source models on all settings and outperforms all the compared methods by a large margin in most cases. Specifically, our DIGA is higher than the best competitor (Momentum [33]) by 4.41%, 7.07%, and 3.4% in average mIoU for GTA5, Synthia, and GTA5+Synthia settings, respectively. In Fig. 4, we provide the qualitative comparison of different methods. It is clear that our DIGA consistently improves the segmentation results of the source model and outperforms other state-of-the-art methods. The above observations demonstrate the effectiveness and universality of the proposed method for solving TTDA-Seg.

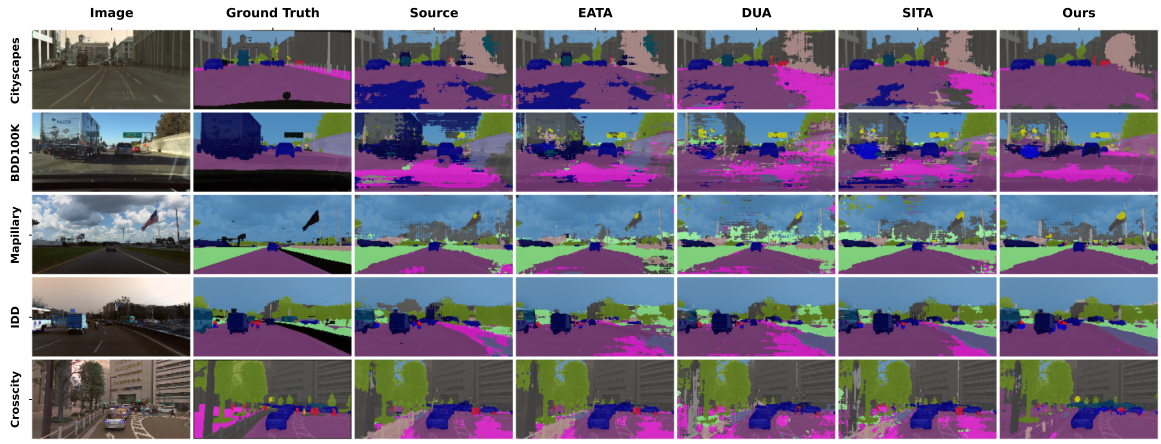

Figure 4. Qualitative comparison of segmentation results.

Table 2. Ablation study on DIGA. DAM: domain alignment module, SAM: semantic alignment module, Association: classifier association. For the BN branch and Semantic branch, we highlight the **best** and second best results, respectively. Source: GTA5.

Modules		CS	BDD	MA	IDD	CC	Avg.
BN	Historical	35.87	29.89	38.67	**38.05**	30.03	34.50
	Instance	34.25	29.64	35.01	29.80	23.87	30.51
	DAM	**39.26**	**33.39**	**40.11**	37.23	**30.12**	**36.02**
Semantic	Historical	38.63	29.25	37.64	**41.67**	32.88	36.01
	Instance	39.16	32.26	38.68	35.09	27.98	34.63
	SAM	**42.99**	**32.69**	**41.37**	41.30	**33.49**	**38.37**
Association		**45.81**	**35.78**	**44.25**	**42.73**	**33.72**	**40.46**

4.3. Ablation Study

We conduct ablative study to investigate the effectiveness of the components of the proposed DIGA, *i.e.*, domain adaptation module, semantic adaptation module, and classifier association. Experiments are evaluated on five target domains with the source model pretrained on GTA5. Results are reported in Tab. 2.

Effectiveness of DAM. In the BN branch of Tab. 2, "Historical" indicates directly using BN statistics of the source for normalization, which can be regarded as the baseline or source model. "Instance" represents using instance statistics for normalization. Two observations can be made. First, the "Instance" model produces worse performance than the "Historical" model on all target domains, especially on IDD and Cross-City. The average mIoU of "Instance" is 4.49% lower than the "Historical". This indicates that using instance statistics only is not suitable for TTDA-Seg. Second, DAM improves the results in most cases and obtains an improvement of 1.48% in average mIoU over the "Historical" model. Specifically, DAM boosts the mIoU by 3.39% and 3.5% on CityScapes and BDD100K, respectively. Even when the gap between "Historical" and "Instance" is too large (e.g., 9.25% on IDD), our DAM is not deteriorated by the negative impact of "Instance" too much and still produces competitive results to the "Historical" with a marginal gap of 0.82%. These two observations suggest that our DAM can effectively merge the guidance of instance knowledge into historical statistics to achieve an effective and stable adaptation process.

Effectiveness of SAM. In the semantic branch, "Instance" indicates the instance-specific prototypes calculated by reliable pixels in the current testing image. "Historical" represents the historical prototypes. Notice that, the semantic branch is conducted based on DAM, where the features for calculating prototypes are obtained after distribution alignment. We can make the following conclusions. First, the "Historical" classifier and the parametric classifier (DAM) achieve a very similar average mIoU. Second, the "Instance" classifier obtains lower average mIoU than the "Historical" classifier. However, by taking a close look at the results on five target domains, we can find that the "Instance" classifier outperforms the "Historical" classifier on three datasets (CityScapes, BDD and Mapillary). This indicates that these two non-parametric classifiers have particular merits in particular datasets. Third, SAM clearly outperforms both non-parametric classifiers in average mIoU. Specifically, our SAM surpasses the "Historical" classifier by 2.36% in average mIoU. Fourth, similar to the BN branch, when the gap between "Historical" and "Instance" classifiers is large, SAM may not bring improvement, *e.g.*, the IDD case. However, our SAM still remains the high performance without influencing by the inferior classifier. The above observations verify the appropriateness of using the prototype classifiers and also the effectiveness of the proposed SAM across different target domains.

Effectiveness of Classifier Association. With the association of the parametric classifier (outputs of DAM) and the non-parametric classifier (outputs of SAM), the results are consistently improved on all target domains. Specifi-

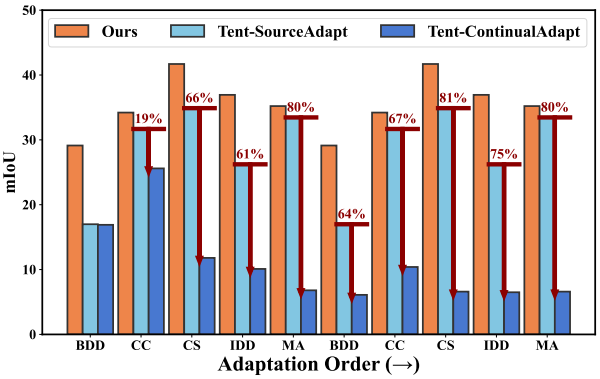

Figure 5. Experiments of continual TTDA-Seg. THe model is pretrained on Synthia [32]. The domain adaptation stream is "BDD→CC→CS→IDD→MA" with two rounds.

cally, the average mIoU is increased by 4.44% for DAM and 2.09% for SAM. This validates the effectiveness of leveraging the mutual benefit between parametric and non-parametric classifiers.

4.4. Continual TTDA-Seg

In real-world applications, such as autonomous driving, the environments are ever-changing and complex. To better simulate such practice scenarios, we design a continual TTDA-Seg experiment. Specifically, the dynamic environment is built by sequentially incoming target domains. The domain-stream is "BDD→CC→CS→IDD→MA", which is sorted by alphabetical order for simplicity and performed by two rounds. We use the Synthia-pretrained model as the source model and report the mIoU after meeting each target domain. In Fig. 5, we compare our method with TENT [41]. We implement two versions for TENT [41]. *TENT-ContinualAdapt*: continually adapt the model with the incoming target domains. *TENT-SourceAdapt*: directly adapt the source-pretrained model on the given domain.

We can observe that "TENT-ContinualAdapt" suffers from significant performance degradation when compared to "TENT-SourceAdapt". For example, when testing on the CityScapes dataset, "TENT-ContinualAdapt" is 19% and 67% lower than "TENT-SourceAdapt" in mIoU at the first round and second round, respectively. This phenomenon can also be observed in other domains. This is mainly because that TENT will accumulate the errors during adaptation and thus leads a worse model. Instead, our DIGA does not have the error accumulation problem and consistently performs well on all domains. This further validates the effectiveness of our method in real-world TTDA-Seg.

4.5. Computational Cost

In TTDA-Seg, efficiency is also very important. In Tab. 3, we investigate the computational costs of different methods. We conduct experiments on the "GTA5→CityScapes" setting. For the inference time, we report the average time (T_{Avg}/ms) and the maximize time (T_{Max}/ms) for each testing sample. In addition, the GPU memory cost is also estimated. We can find that the backward-based methods significantly improve the inference time and GPU memory cost. For example, TENT [41] increases the average time from 134ms to 411ms and the memory cost from 3.5GB to 14.5GB. Even though EATA skips the unreliable pixels during optimization and leads to a lower average inference time than TENT, it still introduces large extra computational cost over the source model. Since SITA [15] uses extra augmented image during testing, its computational cost is doubled. Our DIGA and the other two backward-free methods (Momentum [33] and DUA [25]) produce very limited extra computational cost benefiting from their lightweight designs. However, our DIGA significantly surpasses than Momentum [33] and DUA [25] in mIoU. This experiment suggests that our DIGA is an effective and efficient TTDA-Seg method.

Table 3. Time and Memory Cost. Highlights indicate the **Best**, and Second/Third Best results.

Methods	T_{Avg}	T_{Max}	GPU Mem.	mIoU
Source-Only	134ms	141ms	3.5GB	35.87
TENT [41]	411ms	425ms	14.5GB	37.30
EATA [27]	235ms	490ms	15.6GB	37.08
Momentum [33]	**144ms**	**151ms**	**3.5GB**	37.33
DUA [25]	145ms	152ms	3.6GB	37.79
SITA [15]	253ms	256ms	5.6GB	40.64
Ours	153ms	160ms	4.0GB	**45.51**

5. Conclusion

In this paper, we propose the Dynamically Instance-Guided Adaptation (DIGA) approach for solving TTDA-Seg, which jointly enjoys the effectiveness and efficiency factors. Specifically, DIGA includes two adaptation modules, the distribution adaptation module (DAM) and the semantic adaptation module (SAM), which are both guided by instance-aware information in a non-parametric way. Experiments conducted on five target domains verify that our DIGA effectively can adapt the model at both distribution and semantic levels. We also show that the proposed DIGA achieves state-of-the-art results in TTDA-Seg. In future work, we would like to investigate (1) the learning of adaptive weights in DIGA and (2) the implementation of DIGA in other tasks, *e.g.*, object detection.

Acknowledgement This work has been supported by the EU H2020 project AI4Media (No. 951911), the PRIN project CREATIVE (Prot. 2020ZSL9F9) and Natural Sciences and Engineering Research Council of Canada (NSERC), Discovery Grants program.

References

[1] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *CVPR*, 2022. 2, 3

[2] 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. *PAMI*, 2017. 6

[3] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *ICCV*, 2019. 1, 5, 6

[4] Qingchao Chen and Yang Liu. Structure-aware feature fusion for unsupervised domain adaptation. In *AAAI*, 2020. 4

[5] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *ICCV*, 2017. 5

[6] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *CVPR*, 2021. 2

[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 *CVPR*, 2016. 5

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[9] Yang Fu, Sifei Liu, Umar Iqbal, Shalini De Mello, Humphrey Shi, and Jan Kautz. Learning to track instances without video annotations. In *CVPR*, 2021. 3

[10] Mohammadreza Ghorvei, Mohammadreza Kavianpour, Mohammad TH Beheshti, and Amin Ramezani. Spatial graph convolutional neural network via structured subdomain adaptation and domain adversarial learning for bearing fault diagnosis. *Neurocomputing*, 2022. 4

[11] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2021. 2

[12] Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. In *ICLR*, 2021. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[14] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *NeurIPS*, 2021. 3

[15] Ansh Khurana, Sujoy Paul, Piyush Rai, Soma Biswas, and Gaurav Aggarwal. Sita: Single image test-time adaptation. *arXiv preprint arXiv:2112.02355*, 2021. 2, 5, 6, 8

[16] Junho Kim, Inwoo Hwang, and Young Min Kim. Ev-tta: Test-time adaptation for event-based object recognition. In *CVPR*, 2022. 3

[17] Jogendra Nath Kundu, Akshay Kulkarni, Amit Singh, Varun Jampani, and R. Venkatesh Babu. Generalize then adapt: Source-free domain adaptive semantic segmentation. In *ICCV*, 2021. 3

[18] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *CVPR*, 2021. 2

[19] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *CVPR*, 2018. 2, 4

[20] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020. 3

[21] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *CVPR*, 2021. 2

[22] Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *NeurIPS*, 2021. 2, 3, 4

[23] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *NeurIPS*, 2018. 2, 4

[24] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *CVPR*, 2019. 3

[25] M Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *CVPR*, 2022. 2, 3, 4, 6, 8

[26] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 5

[27] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, 2022. 2, 3, 6, 8

[28] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018. 2, 3, 4, 6

[29] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*, 2019. 4

[30] Duo Peng, Yinjie Lei, Munawar Hayat, Yulan Guo, and Wen Li. Semantic-aware domain generalized segmentation. In *CVPR*, 2022. 2

[31] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 5

[32] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 8

[33] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *NeurIPS*, 2020. 2, 3, 4, 6, 8

[34] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 2017. 4

[35] Serban Stan and Mohammad Rostami. Unsupervised model adaptation for continual semantic segmentation. *AAAI*, 2021. 3

[36] Petar Stojanov, Zijian Li, Mingming Gong, Ruichu Cai, Jaime Carbonell, and Kun Zhang. Domain adaptation with invariant representation learning: What transformations to learn? *NeurIPS*, 2021. 4

[37] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020. 2, 3

[38] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 3

[39] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*, 2019. 5

[40] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019. 1, 3, 5

[41] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 2, 3, 4, 6, 8

[42] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 4, 6

[43] Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *ECCV*, 2020. 3

[44] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020. 5

[45] 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 *CVPR*, 2021. 1, 3, 4

[46] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsuperv ised domain adaptation for semantic segmentation. *NeurIPS*, 2019. 1, 3

[47] Sicheng Zhao, Bo Li, Xiangyu Yue, Yang Gu, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source domain adaptation for semantic segmentation. *NeurIPS*, 2019. 2

[48] Yuyang Zhao, Zhun Zhong, Zhiming Luo, Gim Hee Lee, and Nicu Sebe. Source-free open compound domain adaptation in semantic segmentation. *TCSVT*, 2022. 3

[49] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *IJCV*, 2021. 1, 3

[50] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. 2

[51] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019. 3