Weakly-Supervised Domain Adaptive Semantic Segmentation with Prototypical Contrastive Learning

Anurag Das[1], Yongqin Xian[2]*, Dengxin Dai[1], Bernt Schiele[1]
[1]MPI for Informatics, Saarland Informatics Campus, [2]ETH Zurich
{andas,ddai,schiele}@mpi-inf.mpg.de, yongqin.xian@gmail.com

Abstract

There has been a lot of effort in improving the performance of unsupervised domain adaptation for semantic segmentation task, however, there is still a huge gap in performance when compared with supervised learning. In this work, we propose a common framework to use different weak labels, e.g., image, point and coarse labels from the target domain to reduce this performance gap. Specifically, we propose to learn better prototypes that are representative class features by exploiting these weak labels. We use these improved prototypes for the contrastive alignment of class features. In particular, we perform two different feature alignments: first, we align pixel features with prototypes within each domain and second, we align pixel features from the source to prototype of target domain in an asymmetric way. This asymmetric alignment is beneficial as it preserves the target features during training, which is essential when weak labels are available from the target domain. Our experiments on various benchmarks show that our framework achieves significant improvement compared to existing works and can reduce the performance gap with supervised learning. Code will be available at https://github.com/anurag-198/WDASS.

1. Introduction

Semantic segmentation requires pixel level annotation, which is expensive and time consuming. For real world urban scenes [5, 9, 32, 42], this becomes more challenging as there are far too many objects to annotate in the scene. For example, it takes around 90 min to annotate an image for Cityscapes [5]. To reduce this annotation effort, the task of Unsupervised Domain Adaptative Semantic Segmentation (UDASS) [31, 44, 47, 48] proposes to learn from photorealistic synthetic images [26, 28, 38] with relatively cheap labels. However, due to the domain gap between the real (target domain) and synthetic (source domain) images, this problem becomes more challenging. There have been many efforts [16, 29, 31, 41, 43, 44, 44, 48] to improve the performance on the UDASS task, yet there is a big performance gap compared to supervised learning. In this work, we propose to exploit additional weak labels (image [13, 22], point and [1] coarse labels [5]) for the real images to improve over the UDASS performance and reduce the performance gap with supervised learning.

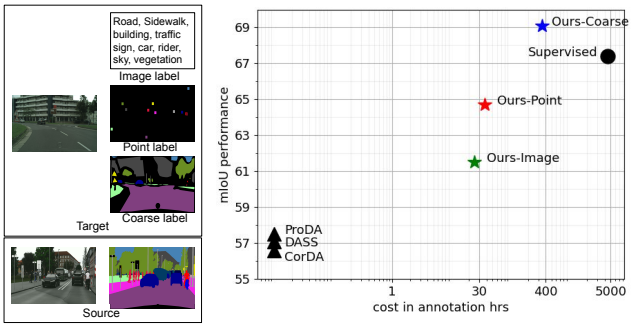

Figure 1. We propose a common framework for different weak label (image, point and coarse labels) for the task of Weakly Supervised Domain Adaptative Semantic Segmentation(WDASS). Our proposed framework, utilising cheaper weak labels bridges the gap between UDA (CorDA [35], ProDA [43], DASS [16]) and supervised learning. Notably, for coarse annotation, our method outperforms supervised learning, exhibiting the potential of weak labels for domain adaptation task.

Weakly supervised Domain Adaptive Semantic Segmentation (WDASS) relaxes the problem of UDASS by allowing weak labels from the target domain. However, it is non trivial to optimally use the weak labels. [25] works with image and point labels and focuses on pixel level adversarial alignment between source and target domains. [6] works with coarse labels and uses self-training for feature alignment. Both these methods use the weak labels only as additional supervision signal from the target domain and do not use them for aligning features between the source and target domain. Moreover, these works do not have a common framework that works with different weak labels. We pro-

*Currently with Google. This work was done at ETH Zürich.

pose a common framework that works with different weak labels (e.g., image, point and coarse labels) and use these weak labels for feature alignment between source and target domains, improving domain adaptation for semantic segmentation task.

Inspired by the recent success of prototype based learning for semantic segmentation [8, 45], few shot learning [7,34,40] and UDASS [16,20,43], we propose to extend prototype learning for the WDASS task. For the UDASS task, prototypes are constructed on the target domain by averaging features from noisy pseudo labels [16,43], resulting in noisy prototypes. With the guidance of additional weak labels from the target domain, we improve the quality of the prototypes. Specifically, we use the pixel labeled weak labels (point or coarse labels) to correct the prototypes and image labels to further improve the features by penalising the category features not present in an image. Next, we perform contrastive alignment of features using the prototypes. We propose two alignments, namely intra domain alignment and inter domain alignment. Intra domain alignment aligns pixel features with prototypes within individual domains (source and target). This helps in learning compact and better features. On the other hand, inter domain alignment aligns features from the source to prototypes from the target domain in an asymmetric manner, reducing the domain gap between source and target domains. This asymmetric alignment only changes the source features and preserves the target features during training. This type of alignment is essential when we have weak labels from the target domain. Overall our proposed framework uses weak labels and improves the performance of UDASS task substantially. We summarise our main contributions as:

- We propose a new framework for WDASS task that works seamlessly with image, point and coarse labels from the target domain. Our method constructs better prototypes using different weak labels. Further, we introduce intra and inter domain contrastive alignment of features with prototypes for source and target domains for WDASS task.
- Our framework using different weak labels (image, point and coarse labels) is able to bridge the gap between UDASS and supervised learning, showing the effectiveness of the weak labels. Distinctly, with coarse labels, our framework even outperforms supervised learning.
- We show the tradeoff between annotation cost vs. semantic segmentation performance for different weak labels. Notably, point annotation achieves better performance in lower annotation budget scenarios than coarse and image label.
- Our framework sets a new state of the art for WDASS on standard benchmarks for different weak labels, with significant improvement over prior works.

2. Related Work

Unsupervised Domain Adaptative Semantic Segmentation The task of UDASS aims to learn from a labeled source domain and unlabeled target domain and improve performance on the target domain [31, 47]. The inherent domain gap between the source and target domains makes this challenging. Prior works use adversarial training for distribution alignment [25, 31], contrastive alignment of source-target features [16, 19, 33] and self-training with pseudo labels [29, 41, 44, 47, 48]. Recently [16, 43] used prototype based self-training to further improve the performance on UDASS task. However, even with various works, the performance gap between UDASS and supervised learning remains to be high [16,43]. In this work, we propose to utilize cheaper weak labels from the target domain to improve on UDASS and reduce the gap with supervised learning.

Weakly Supervised Domain Adaptative Semantic Segmentation Weakly Supervised Domain Adaptive Semantic Segmentation (WDASS) eases the task of UDASS by allowing weak labels from target data. [25] makes use of image and point label and proposes adversarial alignment of features at pixel level to solve WDASS. [36] uses bounding boxes as weak labels and uses adversarial learning for domain-invariant features. [6] uses self training and a boundary loss for improving performance on WDASS with coarse labels. Despite the benefits of weak labels, WDASS task has not been properly explored by the community. They use the weak labels only as additional supervision signal and not for aligning the source and target features.In this work, we propose a common framework for 3 weak labels (image, point and coarse labels) that utilizes weak label for aligning the source-target features and outperforms previous methods achieving competitive performance compared to supervised learning.

Learning from prototypes Prototype learning has become popular for few shot learning [7, 17, 21, 34, 40] where prototypes are used as representative features for classes. Classification is performed by matching query features with learned prototypes from a support set. Recently, prototype based learning also improved performance on semantic segmentation task [46] and UDASS task [16,43]. [46] proposes to use a set of non-learnable prototypes for a class and perform dense prediction by nearest neighbor matching with these prototypes. [43] proposes to use prototypes to tackle the noisy pseudo labels during self-training. Different from these works, we extend prototypes for WDASS task and utilize weak labels to construct better prototypes.

Contrastive Learning Contrastive learning has become popular for learning representations in an unsupervised [3, 4, 11, 18, 24, 39] way, which performs well for downstream tasks such as object detection and semantic segmentation. The key idea is to select similar (positive)

pairs to pull together in feature space, and dissimilar (negative) pairs to contrast in feature space. Contrastive learning also helps in supervised learning. [37] showed that supervised contrastive learning improves the categorical representations and improves semantic segmentation performance. Contrastive learning has also been used for UDASS task [16, 19]. [19] proposes patch contrastive learning for aligning source and target domain for UDASS task. For our work, we use contrastive learning for WDASS task and propose inter and intra domain contrastive alignment for aligning source and target domain features using prototypes.

3. Method

In this section, we first describe the preliminaries (Sec. 3.1), where we present the problem of WDASS and prior information required for our method. We then discuss the key components, Weak Label Guided Prototype Learning (Sec. 3.2), where we discuss constructing prototypes from the weak labels. Next, we present Prototype based Contrastive Learning (Sec. 3.3), where we discuss our contrastive learning approach for reducing the domain gap between the source and target domain. In the end, we present the training details (Sec. 3.4) for our framework.

3.1. Preliminaries

Problem For the task of WDASS, we have a source dataset, $\mathcal{X}_s = \{x_s\}_{j=1}^{n_s}$ with labels $\mathcal{Y}_s = \{y_s\}_{j=1}^{n_s}$ and target dataset $\mathcal{X}_t = \{x_t\}_{j=1}^{n_t}$ with labels $\mathcal{Y}_t = \{y_t\}_{j=1}^{n_t}$, where y_t is a weak label, ie. $y_t \in \{\text{image, point, coarse}\}$ label. The goal is to train a segmentation network using source labels ($\mathcal{Y}_s$) and target weak labels ($\mathcal{Y}_t$), that adapts well to the target dataset $\mathcal{X}_t$.

Self-training with pseudo labels For the unlabeled regions in the target domain, we generate pseudo labels, based on the prediction probability, following UDASS methods [16, 29,41,43,44,47,48]. For a given pixel i and class k, pseudo label $\hat{y}_t^{(i,k)}$ is defined as:

$$\hat{y}_t^{(i,k)} = \begin{cases} 1, & \text{if } k = \arg\max_c p_t^{(i,c)} \text{ and } p_t^{(i,c)} > \tau \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Where τ is a fixed confidence thresholding parameter ($\tau = 0.96$) and $\hat{y}_t^{(i,k)}$ is the one-hot mask for each pixel. We use these pseudo labels for training a segmentation network (g_θ) on the unlabeled regions of the target domain images. For better results, following prior works [12,16,30], we employ a teacher network (h_ϕ), which is an exponential moving average of the segmentation network (also called student network,g_θ) during training. Specifically, the weights (ϕ) of the teacher network are updated using the student weights (θ) following, $\phi = \lambda\phi + (1-\lambda)\theta$, where λ is the momentum parameter. Following previous works [12, 16], we set $\lambda = 0.999$.

Boundary loss Since the weak labels do not have boundary information, we learn this boundary information from the source domain. [6] showed that using a boundary loss on synthetic data improves performance on training with coarse labels. We additionally show that this loss also improves performance with other weaker labels, namely image and point labels. The boundary loss ($\mathcal{L}_{boundary}$) computes the similarity between the ground truth boundary and prediction boundary, computed by taking the gradient over the ground truth mask and prediction mask. We present more details in the supplement.

3.2. Weak Label Guided Prototype Learning

Prototype based learning has been shown to be effective for unsupervised domain adaptation [16, 43]. These methods compute prototypes by taking the average of features from the pseudo labels in the target data. However, the generated pseudo labels for self training can be noisy due to inherent imbalance in training data and domain biases in training. Since the pseudo labels can be noisy, one can end up with sub-optimal prototypes. We propose to improve the prototypes from the guidance of the weak labels.

Point and Coarse Label In the scenario where we have point or coarse annotations for target-domain images, we first create an anchor feature for class k by averaging the features f_t^i corresponding to its labeled pixels y_t^i.

$$a_t^k = \frac{\sum_i \mathbb{1}[y_t^{(i,k)} = 1] * f_t^i}{\sum_i \mathbb{1}[y_t^{(i,k)} = 1]} \quad (2)$$

We then obtain a weight for each feature by computing its similarity w.r.t the anchor feature, $w^{(i,k)} = f_t^i * a_t^k$. We use this weight as contribution of the feature for computing the final prototype for class k as

$$\eta_t^k = \frac{\sum_i f_t^i * \mathbb{1}[\bar{y}_t^{(i,k)} = 1] * w^{(i,k)}}{\sum_i \mathbb{1}[\bar{y}_t^{(i,k)} = 1] * w^{(i,k)}} \quad (3)$$

Where $\bar{y}_t = y_t \cup \hat{y}_t$, i.e. both labeled y_t and pseudo labeled $\hat{y}_t$ pixels from the target domain. We compute this class prototype from images in a training batch. Our prototype computation assigns a higher weight to labeled pixel features as the representative class prototype should be closer to the labeled features than pseudo labeled features that can be noisy.

Image Label For image label, where we do not have any labeled pixels, we simply construct the anchor feature (a_t^k) by averaging all features that are pseudo labeled. The prototypes are weighted averaged over all pseudo labeled features. Further, we also utilize the weak image label to improve the learned features. Specifically, we obtain pixel-wise class logit scores ($m^{(i,k)}$) as similarity of the feature, f_t^i with class prototypes η_t^k.

$$m^{(i,k)} = f_t^i . \eta_t^k \quad (4)$$

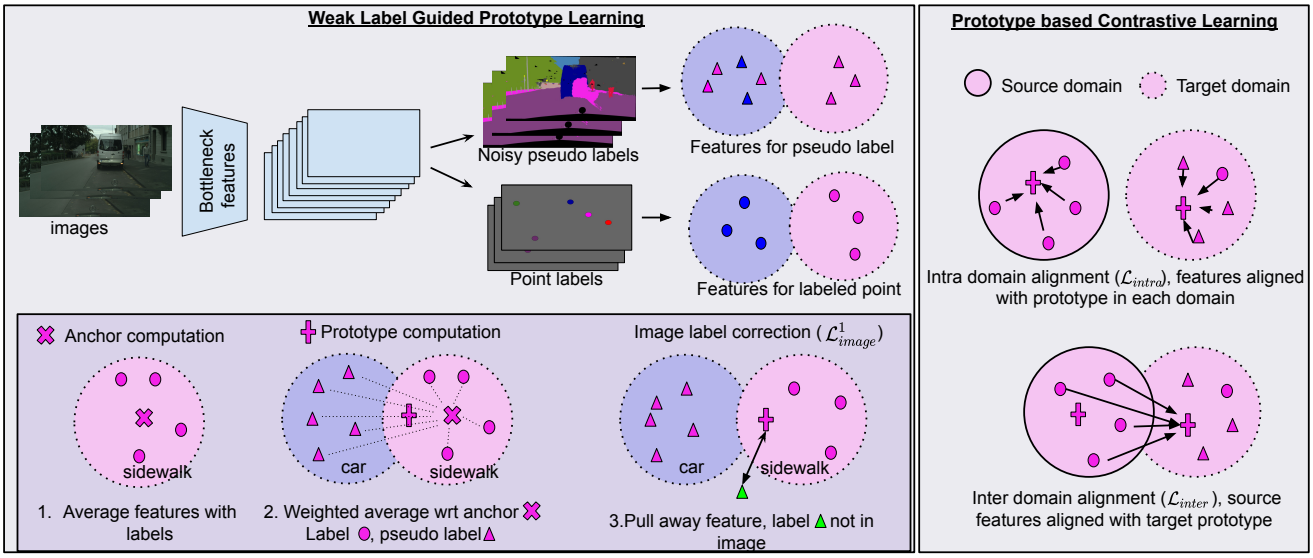

Figure 2. Overview of our framework. We illustrate our approach for point weak label. For Weak Label Guided Prototype Learning, we first compute the anchor for a class by averaging the features from labeled point pixels in a training batch, followed by computing weight of all features as similarity with the anchor. To obtain a class prototype, we perform the weighted averaging of all class features with weight computed previously. Next, we also use image label for correcting feature not present in an image by pulling it away from the prototypes of classes present in the image. Prototype Based Contrastive Alignment proposes inter domain alignment, a novel asymmetric source to target contrastive alignment along with intra domain alignment within individual domains for better feature learning for WDASS task.

We use the pixelwise logit score $m^{(i,k)}$, to obtain an image-level prediction probability, p_t^k that is used for computing the image loss, $\mathcal{L}^1_{image}$,

$$p_t^k = \sigma(log\frac{1}{N}\sum_{i}^{N}\exp m^{(i,k)}) \tag{5}$$

$$\mathcal{L}^1_{image} = \sum_{k=1}^{K} -y_t^k log(p_t^k) - (1-y_t^k)log(1-p_t^k) \tag{6}$$

Here, y_t^k is an image label, implying class k is present in an image for $y_t^k = 1$. K is the total number of classes. The LogSumExp expression in Eq. (5) is used to estimate the smooth maximum from the logits [25], representing the most activated pixel in the class map. Using the logits of the most activate pixel, we obtain the image level prediction (p_t^k). This formulation is beneficial as it penalizes 'maximum' logit features and doesn't penalize all features from the image. This image loss based on prototypes helps in correcting features during training by pulling away features from prototypes for classes not present in an image and pulling closer together features present in an image, thus overall improving the feature quality. Please note that this image loss is also applied for coarse and point labels, as image labels can be obtained from point and coarse labels.

Prototype for source domain We also obtain class prototypes for the source domain in a similar way as for weak labels. However, fully labeled masks are available for source domain and we use it to construct prototype instead of weak labels.

Teacher network for better prototypes We use a teacher network, which is an exponential moving average of the student network to obtain features for prototype computation. We use a momentum parameter of $\lambda = 0.999$ as used in previous works [16, 46].

3.3. Prototype based Contrastive Learning

Once we have obtained prototypes following Sec. 3.2, we use the prototypes for contrastive alignment of features. We align the features from the student network with prototypes from the teacher network using both Intra Domain and Inter Domain Alignment.

Intra Domain Alignment As shown in Fig. 2, we perform intra domain alignment of features for both source and target domains individually. Specifically, for a given pixel feature, we construct a positive pair of the pixel features with its corresponding class prototype while a negative pair with different class prototypes. We use the standard infoNCE [10, 24] loss function for contrastive learning. For the target domain, this loss for a given pixel feature f^i belonging to class k with prototype η_t^k is defined as:

$$\mathcal{L}^t_{intra} = -log\frac{exp(f_t^i.\eta_t^k/\tau)}{exp(f_t^i.\eta_t^k/\tau) + \sum_{c\neq k} exp(f_t^i.\eta_t^c/\tau)} \tag{7}$$

Similarly for source domain, we define $\mathcal{L}^s_{intra}$. We sum

both these losses to get the intra domain alignment loss as:

$$\mathcal{L}_{intra} = \mathcal{L}^t_{intra} + \mathcal{L}^s_{intra} \tag{8}$$

This individual domain alignment is useful in learning better compact feature representations and helps improve the performance (see Tab. 4).

Inter Domain Alignment We also align features across source and target domains. Specifically, we align features from the source domain with prototypes from the target domain in an asymmetric manner. Given a feature f_i from the source domain, we construct a positive pair with the prototype from the same class from the target domain and a negative pair with the prototype from a different class in the target domain. The inter domain alignment loss for source pixel feature f^i_s belonging to class k with target prototype η^k_t is defined as,

$$\mathcal{L}_{inter} = -log \frac{exp(f^i_s.\eta^k_t/\tau)}{exp(f^i_s.\eta^k_t/\tau) + \sum_{c \neq k} exp(f^i_s.\eta^c_t/\tau)} \tag{9}$$

We ensure the asymmetric alignment by having no gradient flow through prototype features. This is implemented by default via the construction of prototypes by the teacher network, as its parameters don't have gradients. Traditionally, adversarial or contrastive alignment [16, 25] of source and target features do not have this asymmetricity as they mostly deal with unsupervised domain adaptation where no weak labels from the target domain are present. With asymmetric alignment, we preserve the features from the target domain during training. This alignment is important for improving performance for weakly supervised domain adaptation where we already have weak labels from the target domain.

We combine the losses to get the prototypical contrastive loss as:

$$\mathcal{L}_{contrast} = \lambda_1 \mathcal{L}_{inter} + \lambda_2 \mathcal{L}_{intra} \tag{10}$$

Where we use λ_1, λ_2 as 0.5.

Selection of samples for Contrastive training [14, 15, 27, 37] shows that the amount of training samples for contrastive learning is important. Following [37], we select a subset of pixels for contrastive training. [37] samples 'hard samples' as samples with incorrect predictions as ground truth. We follow the same strategy for selecting samples from the source domain where we have the ground truth labels available. Similarly, for point and coarse labels from the target domain, where we have few labeled samples, we sample hard samples as the ones with incorrect predictions. For image label, where we do not have any pixel labels, and also for unlabeled regions of point and coarse annotation, we select 'easy samples' as ones with higher prediction confidence (threshold=0.95). We finally use a mix of hard and easy samples as in [37] for contrastive training.

3.4. Training

As an initial baseline, one can train a segmentation network with source labels (y_s), pseudo labels ($\hat{y}_t$) for unlabelled target pixels and target weak labels (y_t) with classification loss. This is straightforward for point and coarse labels,

$$\mathcal{L}_{base} = \mathcal{L}^s_{ce}(x_s, y_s) + \mathcal{L}^t_{ce}(x_t, y_t) + \mathcal{L}^t_{ce}(x_t, \hat{y}_t) \tag{11}$$

However, for image label, where we only know the class labels, we follow [25], and penalise classification logits for classes not present in an image to get the image loss, $\mathcal{L}^2_{image}$. We present more details in the supplement.

We combine the image loss from classification layer with prototype based image loss from Eq. (6) to get the final image loss as

$$\mathcal{L}_{image} = \mathcal{L}^1_{image} + \mathcal{L}^2_{image} \tag{12}$$

Finally combining the losses, we train with the following loss,

$$\mathcal{L} = \mathcal{L}_{base} + \lambda_1 \mathcal{L}_{image} + \lambda_2 \mathcal{L}_{contrast} + \lambda_3 \mathcal{L}_{boundary} \tag{13}$$

We select the hyperparameters $\lambda_1, \lambda_2, \lambda_3$ as 1.0, 1.0 and 10.0 for our experiments.

4. Experiments

Datasets and metric used. We evaluate our method on two standard domain adaptation benchmarks i.e., GTA-5 $\rightarrow$ Cityscapes and Synthia $\rightarrow$ Cityscapes. For Cityscapes, we use 2975 images from the train split and report results on the val split with 500 images. We follow previous work [25] and compare our method with point and image weak labels. We obtain point label annotation by randomly sampling one pixel per class in an image following [25]. Further, image labels are obtained from the available class labels from the ground truth labels. We also compare with an additional coarse weak label, which is provided in the Cityscapes dataset (gtCoarse). For GTA-5 and Synthia, we use the available 24966 and 9400 training images. We use the mean Intersection over Union score (mIoU) for evaluation.

Implementation details. Following prior works [25], we use a DeepLabv2 segmentation model with ImageNet pretrained ResNet-101 as backbone and also resize the Cityscapes images to 1024x512 and GTA-5 images to 1280x760 resolution. We employ SGD optimiser with poly learning rate scheduler having initial learning rate of $2.5\text{x}10e^{-4}$. We set a momentum of 0.9 and weight decay rate of 0.0001. We use a crop size of 512x512 and train for 150000 iterations. For GTA-5$\rightarrow$Cityscapes setting, we initialise the network with weights from DeepLabv2

		GTA5 → Cityscapes		Synthia → Cityscapes		
	Method	mIoU	gap	mIoU†	mIoU*	gap
	Source	36.6	+30.8	34.9	40.3	+33.6
UDA	CorDA [35]	56.6	+10.8	55.0	62.8	+11.1
UDA	ProDA [43]	57.5	+9.9	55.5	62.0	+11.9
UDA	DASS [16]	57.1	+10.3	55.6	62.9	+11.0
image	baseline	51.4	+16.0	36.1	39.1	+34.8
image	WeakSegDA [25]	53.0	+14.4	50.6	58.5	+15.4
image	Ours	**61.5**	**+5.9**	**61.3**	**63.9**	**+10.0**
point	baseline	54.9	+12.9	48.5	53.3	+20.6
point	WeakSegDA [25]	56.4	+11.0	57.2	63.7	+10.2
point	Ours	**64.7**	**+2.7**	**62.8**	**68.7**	**+5.2**
coarse	baseline	60.8	+6.6	54.6	59.1	+14.8
coarse	Coarse-to-fine [6]	66.7	+0.7	61.6	67.2	+6.7
coarse	Ours	**69.1**	**-1.7**	**66.0**	**71.0**	**+2.9**
	Supervised	67.4	0.0	68.8	73.9	0.0

Table 1. Comparison results on GTA→Cityscapes and Synthia→Cityscapes. We show two comparisons, first, domain adaptation with no labels from the target domain (UDA) vs. domain adaptation with weak labels (image, point, coarse) from the target domain. Second, Comparison of our method (ours) with baseline and existing methods(WeakSegDA [25], Coarse-to-fine [6]) for each weak label. gap: performance gap for mIoU scores wrt to the supervised setting. Lower value of gap is better. Results reported in mean IoU for 19 Cityscapes classes for GTA5→Cityscapes. For Synthia→Cityscapes, results are reported for common 16 (mIoU†) and 13 (mIoU*) Cityscapes classes. Classwise result reported in supplement.

network trained on GTA-5 dataset, and similarly for Synthia→Cityscapes, we initialise with Synthia pretrained weights. Following works on UDA [12, 30] we also employ classmix [23] based augmentation in our training. We use a bottleneck projection head (single 1x1 Conv layer to reduce dimension from 2048 to 256) on top of the segmentation backbone to obtain features for contrastive alignment.

Annotation costs. Labeling task for semantic segmentation is quite expensive. Cityscapes fine annotation takes around 90 min [5] (including quality control) for annotation. On the contrary, weak annotations are much cheaper. Coarse annotation takes only 7 min [5] per image for annotating. It sacrifices the fine boundary details to achieve this low cost. Further, point label and image label take around 45 and 30 seconds [25] respectively for annotation.

4.1. Comparison with State-of-the-Art

We make the comparisons on GTA5→Cityscapes and Synthia→Cityscapes settings in Tab. 1. We perform two key comparisons. First, we compare our method performance for each weak label (image, point and coarse labels) with prior works (WeakSegDA [25] for image and point label, Coarse-to-fine [6] for the coarse label) and also with an intuitive baseline of training segmentation network with source labels and target weak labels. Second, we compare our method's performance with UDA SoTA methods (CorDA [35], ProDA [43], DASS [16]) and with the supervised baseline to show our method is able to bridge the performance gap between them.

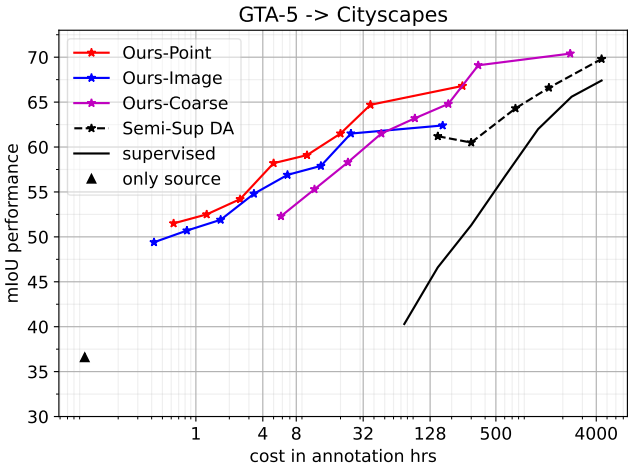

Figure 3. Annotation cost vs. performance. 1. Comparison of WDASS (point, image, coarse) vs. SoTA semi-supervised domain adaptation [2] (Semi-Sup DA) vs fully supervised training on the source (only source) and target (supervised). 2. Comparison within weakly supervised domain adaptation for various weak labels (Point, Image, Coarse labels).

For the first comparison, our method outperforms both the baseline and previous works for all three weak labels (image, point and coarse) for both GTA5→Cityscapes and Synthia→Cityscapes, see Tab. 1. Notably, we outperform the SoTA methods by a margin of 8.5, 8.3 and 2.4 mIoU for image, point and coarse weak labels, respectively for GTA5→Cityscapes setting. Similarly, for Synthia→Cityscapes setting, we outperform the SoTA methods for all three weak labels by a margin of 10.7, 5.6 and 4.4 mIoU for 16 classes (mIoU†in Tab. 1) and by 5.4, 5 and 3.8 for 13 classes (mIoU* in Tab. 1) with image, point and coarse labels respectively.

For the second comparison, we observe that using weak labels indeed bridges the gap between UDA and supervised learning. Remarkably, our method with coarse annotation for GTA5→Cityscapes setting outperforms supervised setting by a difference of 1.7mIoU. This is promising as our method used only 8% of the annotation budget compared to supervised learning (see Fig. 3).The performance gap with supervised learning for point and image labels are 2.7 and 5.9 compared to 9.9 for UDA setting for GTA5→Cityscapes (see Tab. 1), suggesting weak labels are essential for im-

proving performance over UDA. We see similar findings on the difficult Synthia→Cityscapes setting in Tab. 1.

Overall, our method outperforms previous works in weakly supervised domain adaptation by a significant margin and can also to push the performance closer to supervised learning.

Qualitative Comparison We show the qualitative comparison of our framework with baselines in Fig. 4. For all three weak labels, e.g., image, point and coarse labels. In particular, we point out the performance of the class 'train'. Our framework segments train better than the baselines for all three weak labels. Further, within different annotation types, coarse annotation performs best due to more labeled pixel supervision from the target domain. We provide more qualitative results in the supplement.

4.2. Annotation cost vs performance comparison

In this experiment, we show the cost-effectiveness of the weak labels for WDASS task. For this comparison, we sample 50, 100, 200, 400, 800, 1600, 2975 images from Cityscapes train set and additional 19998 images from the Cityscapes coarse set. The cost of annotation for weak labels is computed following costs mentioned in Sec. 4. For example, annotating 2975 images of the Cityscapes with image label costs 24.8hrs, point label costs 37.2 hrs, coarse label costs 347 hrs and fine label costs 4463 hrs.

We make two comparisons (see Fig. 3). First, we compare performances under different annotation budgets within weakly supervised domain adaptation for different weak labels (image vs. point vs. coarse) and second, we compare weakly supervised domain adaption with supervised learning (on the source and target separately) and SoTA semi-supervised domain adaptation method [2].

For the first comparison within different weak labels, we observe that Ours-Point outperforms other weak labels in lower budgets suggesting it to be better suited for low-budget settings. For the second comparison, we observe our framework in general, outperforms Semi-supervised Domain adaptation SoTA [2] and the supervised baselines (only source, supervised). Further, with only 8% of fully supervised budget (347 vs. 4463 hrs), coarse annotation outperforms supervised learning by 1.7% mIoU (69.1% for coarse vs 67.4% for supervised learning), suggesting the importance of coarse labels. Overall, weak labels are better and cost-effective alternatives compared to fine labels.

4.3. Ablation analysis

Ablation with network components We evaluate the contribution of the components of our framework in Tab. 2. 'base' refers to baseline as in Eq. (11), trained on source labels and target weak labels. We have 3 components, namely contrastive learning for prototypes ($\mathcal{L}_{contrast}$), image loss ($\mathcal{L}_{image}$) and boundary loss ($\mathcal{L}_{boundary}$) as from Eq. (13).

base	$\mathcal{L}_{boundary}$	$\mathcal{L}_{contrast}$	$\mathcal{L}_{image}$	image	point	coarse
✓				36.6	54.9	60.8
✓	✓			42.5	58.5	62.9
✓		✓		42.1	60.0	58.9
✓			✓	56.8	58.9	64.4
✓	✓	✓		43.3	63.3	67.3
✓		✓	✓	60.1	63.0	66.8
✓	✓		✓	59.7	60.3	67.0
✓	✓	✓	✓	**61.5**	**64.7**	**69.1**

Table 2. Ablation result wrt to components on GTA5→Cityscapes setting, measured in mIoU performance. Base: Supervised training on source and target weak labels (image, point and coarse labels), $\mathcal{L}_{boundary}$: Boundary loss, $\mathcal{L}_{contrast}$: contrastive loss, $\mathcal{L}_{image}$: Image loss.

	image	point	coarse
Averaging	59.0	63.1	66.8
ours	**61.5**	**64.7**	**69.1**

	image	point	coarse
Weak Sup.	39.0	49.9	65.0
ours	**61.5**	**64.7**	**69.1**

Table 3. Left: Comparison for prototype computation approach. Averaging: Prototype computation by averaging the features for a class following [16, 43], ours: Prototype computation by weak label guidance as discussed in Sec. 3.3. Right: Comparison with weakly supervised learning. We compare our method with weakly supervised method based on classmix augmentation and self-training, trained only on weak labels from target domain.

We observe that each of component, when added on base, improves the performance. Particularly, for image label, the addition of the image loss significantly improves performance (36.6 vs 56.8mIoU). This shows the importance of our image loss that corrects the features by penalising features of class not present in the image. We also observe similar improvement when we add two components two base. Our framework performs best with all three components together with the significant improvement compared to baseline, suggesting that the components are complementary to each other and important for better performance.

Constructing prototypes We compare our weak label guided prototype computation (see Sec. 3.3) to a popular baseline of prototype computation by averaging of features for a class as done by prior works [16, 43]. We show this result in Tab. 3. We observe clear improvement in performance when we construct prototypes with weak label guidance. We outperform the baseline by 2.5, 1.6 and 2.3 mIoU, clearly showing the importance of weak label guided prototype construction for WDASS task.

Comparison with weakly supervised learning We perform this comparison with weakly supervised learning to show how much gain one can get with synthetic data by WDASS (see Tab. 3) compared to weakly supervised semantic segmentation. The weakly supervised baseline is trained only on weak labels (image, point and coarse labels) from the target domain and uses classmix augmenta-

	image	point	coarse
pixel-pixel	56.4	62.7	67.0
pixel-prototype	**61.5**	**64.7**	**69.1**

	image	point	coarse
symmetric	58.6	63.5	67.4
assymmetric	**61.5**	**64.7**	**69.1**

base	$+\mathcal{L}_{inter}$	$+\mathcal{L}_{intra}$	image	point	coarse
✓			59.7	60.3	67.0
✓	✓		60.4	62.6	67.1
✓		✓	60.2	63.9	67.7
✓	✓	✓	**61.5**	**64.7**	**69.1**

Table 4. Ablation wrt to contrastive components, left: Comparison wrt to pixel-pixel contrastive learning vs pixel-prototype contrastive learning (ours), middle: comparison wrt symmetric vs assymetric contrastive learning (ours) for inter domain feature alignment, right: ablation wrt contrastive loss components, base: our method without contrastive component, $\mathcal{L}_{intra}$: contrastive alignment loss within each domain individually, $\mathcal{L}_{inter}$: asymmetric contrastive alignment between source and target domain.

tion [23] and self training, making it a stronger baseline. We observe a significant gain of 22.5% for image label, 14.8% for point label and 4.1% for the coarse weak label. This suggests that WDASS with additional relatively free synthetic images can improve over weakly supervised semantic segmentation with significant performance gain.

4.4. Analysis for contrastive feature alignment

Pixel-Pixel vs Pixel-Prototype Contrastive Alignment In this experiment, we make a comparison between pixel-pixel and pixel-prototype (ours) feature alignment (Tab. 4 left). For the pixel-pixel alignment baseline, we perform intra domain alignment with pixel features from the same domain for both the source and target domain. We perform asymmetric alignment with pixel features from source to target domain for inter domain alignment. This comparison shows the importance of our prototypes for feature alignment between source and target domains. We achieve an increment of 5.1, 2.0 and 2.1 mIoU for image, point and coarse labels respectively when using prototype based alignment, suggesting importance of our prototypes for feature alignments.

Symmetric vs Asymmetric Inter Domain Feature Alignment In this experiment, we show the importance of asymmetric alignment for inter domain alignment between source and target domains (Tab. 4 middle). We compare our method's performance with a baseline with symmetric alignment for inter domain feature alignment. For symmetric alignment, we align features from the target domain with prototypes of the source domain, along with the asymmetric alignment of source features to target prototypes. We observe that with asymmetric alignment, our method outperforms the baseline with symmetric alignment by a margin of 2.9, 1.2 and 1.7 mIoU for image, point and coarse labels, respectively. This shows the importance of preserving the target features during alignment for WDASS task.

Ablation with contrastive loss components In this experiment, we ablate the contrastive loss components (Tab. 4 right). We observe that both inter ($\mathcal{L}_{inter}$) and intra domain alignment ($\mathcal{L}_{intra}$) losses individually improve performance over baseline with no contrastive learning. Particularly for point labels, the addtion of $\mathcal{L}_{inter}$ improves performance by 2.3 mIoU, and addtion of $\mathcal{L}_{intra}$ improves performance by 3.6 mIoU. Further, both these losses complement each other and give best performance together.

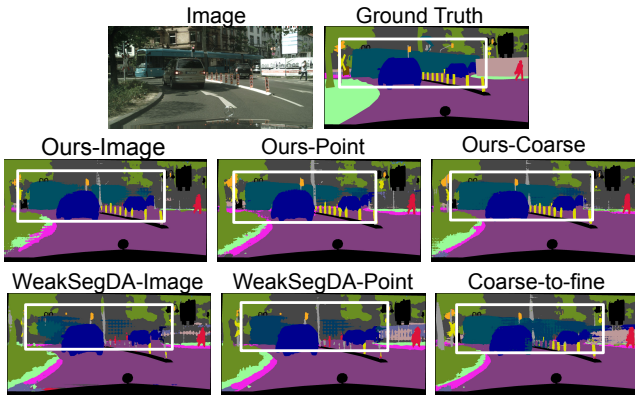

Figure 4. Qualitative comparison: We compare our framework's performance using different weak labels with previous works. In this figure, class 'train' is segmented better for our framework than previous works.

5. Conclusion

In this work, we present a common framework that utilizes different weak labels(image, point and coarse) for the WDASS task. Our framework constructs better prototypes by exploiting the weak labels. Further, we use these prototypes for intra-domain alignment of features within individual domains and inter-domain alignment of features between the source and target domain, thus effectively reducing the domain gap. Our framework effectively reduces the performance gap between UDASS and supervised learning. Notably, for coarse annotation, it even outperforms supervised learning, suggesting the importance of cost-effective weak labels. We also perform annotation cost vs. performance comparison within different weak labels, which shows point labels have the best cost vs. performance tradeoff. More importantly, our framework outperforms the previous state of the art methods in WDASS task in the standard benchmarks. We hope our work inspires more innovations in this challenging and promising direction.

References

[1] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 1

[2] Shuaijun Chen, Xu Jia, Jianzhong He, Yongjie Shi, and Jianzhuang Liu. Semi-supervised domain adaptation based on dual-level domain mixing for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11018–11027, 2021. 6, 7

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[5] 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*, pages 3213–3223, 2016. 1, 6

[6] Anurag Das, Yongqin Xian, Yang He, Zeynep Akata, and Bernt Schiele. Urban scene semantic segmentation with low-cost coarse annotation. In *2023 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2023. 1, 2, 3, 6

[7] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018. 2

[8] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4320–4329, 2022. 2

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1

[10] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 4

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[12] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9924–9935, 2022. 3, 6

[13] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, pages 7014–7023, 2018. 1

[14] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020. 5

[15] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 5

[16] Geon Lee, Chanho Eom, Wonkyung Lee, Hyekang Park, and Bumsub Ham. Bi-directional contrastive learning for domain adaptive semantic segmentation. *arXiv preprint arXiv:2207.10892*, 2022. 1, 2, 3, 4, 5, 6, 7

[17] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8334–8343, 2021. 2

[18] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 2

[19] Weizhe Liu, David Ferstl, Samuel Schulter, Lukas Zebedin, Pascal Fua, and Christian Leistner. Domain adaptation for semantic segmentation via patch-wise contrastive learning. *arXiv preprint arXiv:2104.11056*, 2021. 2, 3

[20] Yahao Liu, Jinhong Deng, Xinchen Gao, Wen Li, and Lixin Duan. Bapa-net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8801–8811, 2021. 2

[21] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 142–158. Springer, 2020. 2

[22] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, Bernt Schiele, et al. Exploiting saliency for object segmentation from image level labels. In *CVPR*, 2017. 1

[23] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1369–1378, 2021. 6, 8

[24] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 4

[25] Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K Roy-Chowdhury, and Manmohan Chandraker. Domain adaptive semantic segmentation using weak labels. In *European conference on computer vision*, pages 571–587. Springer, 2020. 1, 2, 4, 5, 6

[26] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer

games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 1

[27] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020. 5

[28] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 1

[29] Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In *European conference on computer vision*, pages 532–548. Springer, 2020. 1, 2, 3

[30] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1379–1389, 2021. 3, 6

[31] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018. 1, 2

[32] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1743–1751. IEEE, 2019. 1

[33] Midhun Vayyat, Jaswin Kasi, Anuraag Bhattacharya, Shuaib Ahmed, and Rahul Tallamraju. Cluda: Contrastive learning in unsupervised domain adaptation for semantic segmentation. *arXiv preprint arXiv:2208.14227*, 2022. 2

[34] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9197–9206, 2019. 2

[35] Qin Wang, Dengxin Dai, Lukas Hoyer, Luc Van Gool, and Olga Fink. Domain adaptive semantic segmentation with self-supervised depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8515–8525, 2021. 1, 6

[36] Qi Wang, Junyu Gao, and Xuelong Li. Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes. *IEEE Transactions on Image Processing*, 28(9):4376–4386, 2019. 2

[37] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7303–7313, 2021. 3, 5

[38] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *arXiv preprint arXiv:1810.08705*, 2018. 1

[39] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 2

[40] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 763–778. Springer, 2020. 2

[41] 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*, pages 4085–4095, 2020. 1, 2, 3

[42] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018. 1

[43] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12414–12424, 2021. 1, 2, 3, 6, 7

[44] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2, 3

[45] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2582–2593, June 2022. 2

[46] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2582–2593, 2022. 2, 4

[47] Yang Zou, Zhiding Yu, BVK 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)*, pages 289–305, 2018. 1, 2, 3

[48] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019. 1, 2, 3