

# SePiCo: Semantic-Guided Pixel Contrast for Domain Adaptive Semantic Segmentation

Binhui Xie, Shuang Li, Mingjia Li, Chi Harold Liu, *Senior Member, IEEE*, Gao Huang, and Guoren Wang

**Abstract**—Domain adaptive semantic segmentation attempts to make satisfactory dense predictions on an unlabeled target domain by utilizing the supervised model trained on a labeled source domain. One popular solution is self-training, which retrains the model with pseudo labels on target instances. Plenty of approaches tend to alleviate noisy pseudo labels, however, they ignore the intrinsic connection of the training data, i.e., intra-class compactness and inter-class dispersion between pixel representations across and within domains. In consequence, they struggle to handle cross-domain semantic variations and fail to build a well-structured embedding space, leading to less discrimination and poor generalization. In this work, we propose *Semantic-Guided Pixel Contrast (SePiCo)*, a novel one-stage adaptation framework that highlights the semantic concepts of individual pixels to promote learning of class-discriminative and class-balanced pixel representations across domains, eventually boosting the performance of self-training methods. Specifically, to explore proper semantic concepts, we first investigate a *centroid-aware pixel contrast* that employs the category centroids of the entire source domain or a single source image to guide the learning of discriminative features. Considering the possible lack of category diversity in semantic concepts, we then blaze a trail of distributional perspective to involve a sufficient quantity of instances, namely *distribution-aware pixel contrast*, in which we approximate the true distribution of each semantic category from the statistics of labeled source data. Moreover, such an optimization objective can derive a closed-form upper bound by implicitly involving an infinite number of (dis)similar pairs, making it computationally efficient. Extensive experiments show that SePiCo not only helps stabilize training but also yields discriminative representations, making significant progress on both synthetic-to-real and daytime-to-nighttime adaptation scenarios. The code and models are available at <https://github.com/BIT-DA/SePiCo>.

**Index Terms**—Domain adaptation, semantic segmentation, semantic variations, representation learning, self-training.

arXiv:2204.08808v2 [cs.CV] 20 Feb 2023

## 1 INTRODUCTION

GENERALIZING deep neural networks to an unseen domain is pivotal to a broad range of critical applications such as autonomous driving [1], [2] and medical analysis [3], [4]. For example, autonomous cars are required to operate smoothly in diverse weather and illumination conditions, e.g., foggy, rainy, snowy, dusty, and nighttime. While humans excel at such scene understanding problems, it is struggling for machines to forecast. Semantic segmentation is a fundamental task relevant that assigns a unique label to every single pixel in the image. Recently, deep Convolution Neural Networks (CNNs) have made rapid progress with remarkable generalization ability [5], [6], [7], [8]. CNNs, however, are quite data-hungry and the pixel-level labeling process is expensive and labor-intensive, thereby restricting their real-world utility. As a trade-off, training with freely-available synthetic data rendered from game engines [9], [10] turns into a promising alternative. This is not the case, unfortunately, deep models trained on simulated data often drop largely in realistic scenarios due to *domain shift* [11].

Recent trends of domain adaptation (DA) inspire the emergence of extensive works to transfer knowledge from a label-rich source (synthetic) domain to a label-scarce target (real) domain, which enjoys tremendous success [12], [13], [14], [15], [16]. Most previous works develop adversarial

- B. Xie, S. Li, M. Li, C. H. Liu and G. Wang are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. Email: {binhuixie, shuangli, mingjiali, chilliu, wanggr-bit}@bit.edu.cn.
- G. Huang is with Department of Automation, Tsinghua University, Beijing, China, Email: gaohuang@tsinghua.edu.cn.
- Corresponding author: Shuang Li.

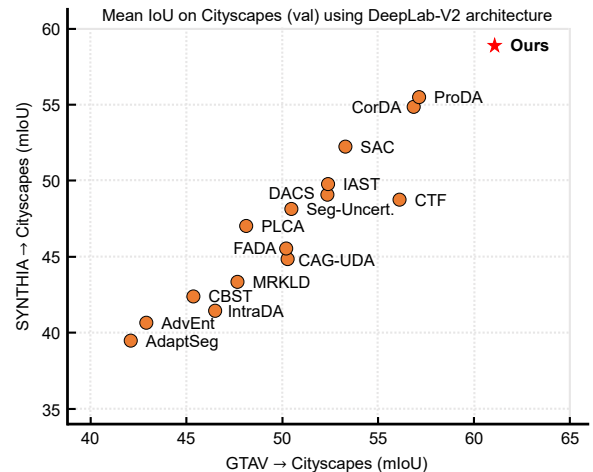


Fig. 1: Results preview on two popular synthetic-to-real semantic segmentation tasks. Our method is shown in **bold**.

training algorithms to diminish the domain shift existing in input [17], [18], feature [19], [20] or output [21], [22] space. Despite the fact that the above methods can draw two domains closer globally, it does not guarantee those feature representations from different classes in the target domain are well-separated. Utilizing category information can refine such alignment [23], [24], [25], [26], [27]. But, pixels in different images might share much similar semantics while their visual characteristics, such as color, scale, illumination, etc. could be quite different, which is deleterious to the continual learning of pixel representations across two domains.

Another line of work harnesses self-training to promote

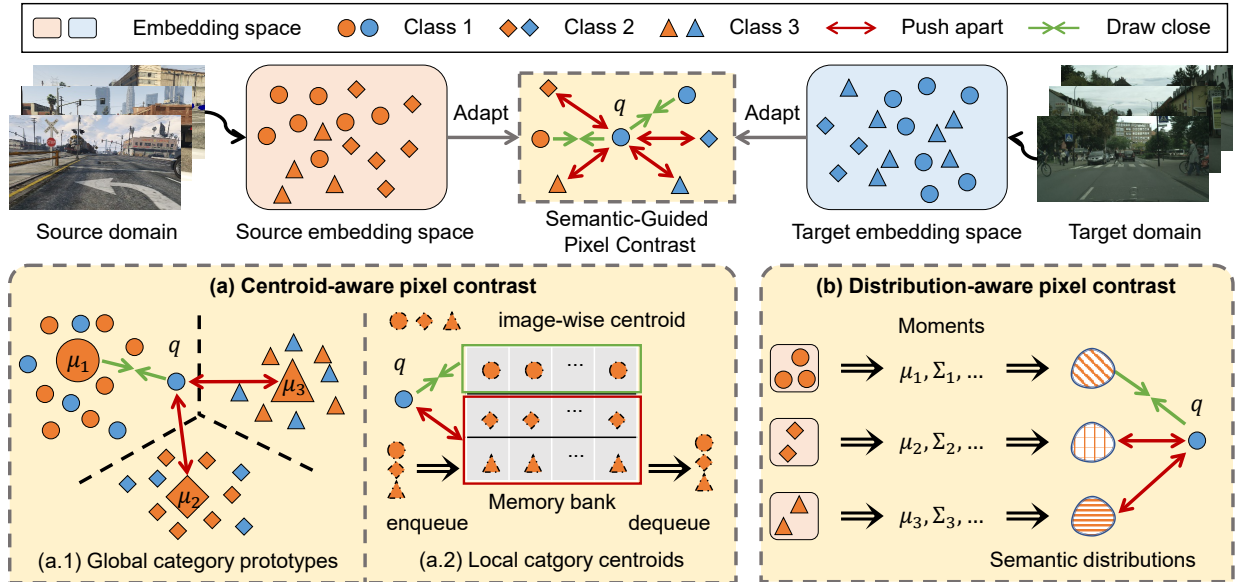


Fig. 2: **Illustration of the main idea.** By contrastively matching a pixel query  $q$  to distinct semantics, features with the same semantic concepts are drawn closer while those with different ones are pushed apart across domains. We first explore (a) **Centroid-aware pixel contrast** including (a.1) **global category prototypes** simply computed on the entire source domain, which render the overall appearance of each category and (a.2) **local category centroids** of each class in a single source image, which are stored into a memory bank. Further, we develop (b) **Distribution-aware pixel contrast**: the distributions of each category on source features are depicted as class-specific holistic concepts to guide the semantic alignment.

the segmentation performance [28], [29], [30], [31]. By adopting confidence estimation [32], [33], consistency regularization [34], [35], or label denoising [36], [37], the noisy in pseudo labels could be relieved to some extent. While many works are already capable of establishing milestone performance, there is still much room for improvement beyond the current state-of-the-art. We find that most approaches do not explicitly address the domain discrepancy, and the learned target representations are still dispersed. In addition, many works opt for a stage-wise training mechanism to avoid training error amplification in a single-stage model, which heavily relies on a well-initialized model to increase the reliability of generated pseudo labels. Hereafter, several methods combine adversarial training and self-training [35], [38] or train with auxiliary tasks [39], [40] to learn discriminative representations from unlabeled target data.

Contrastive learning is a relevant topic, which learns proper visual representations by comparing different unlabeled data [41], [42], [43], [44]. Without any supervision, models are capable of finding patterns like similarity and dissimilarity. The huge success of contrastive learning and the aforementioned drawbacks in prior arts together motivate us to rethink the current de facto training paradigm in semantic segmentation under a domain shift. Basically, the power of contrastive learning roots in instance discrimination, which takes advantage of semantic concepts within data. With this insight, we find a new path to build models that are robust to distribution shifts by exploring cross-domain pixel contrast under the guidance of proper semantic concepts, which attracts similar pixels and dispels dissimilar ones in a latent space, as illustrated in Fig. 2.

In this work, we present a novel end-to-end framework, SePiCo, for domain adaptive semantic segmentation. Not only does SePiCo outperform previous works (Fig. 1), but it

is also simple yet effective, keeping one-stage training complexity. Precisely, we build upon a self-training method [30] and introduce several dense contrastive learning mechanisms. The core is to explore suitable semantic concepts to guide the learning of a well-structured pixel embedding space across domains. Here, a plain way is to adopt the averaged feature of a category over the entire source domain as its global prototype. A prototype could render the overall appearance of a category but might omit variations in some attributes (e.g., shape, color, illumination) of the category, impairing the discriminability of the learned features. To enhance diversity, a natural extension is to enlarge the number of contrastive pairs. We then investigate a memory bank mechanism, in which the averaged features of each category in the current source image are enqueued into a dictionary and the oldest ones are dequeued. Unfortunately, class biases may exist in this mechanism since those under-represented classes (e.g., truck, bus, rider) are updated more slowly. Meanwhile, it is computationally expensive as well.

Grounded on the above discussions, we hypothesize that if every dimension in the embedding space follows a distribution, pixel representations from a similar semantic class would have a similar distribution, which is independent of the domain. Thereby, we take the distribution of each category in the source domain as a richer and more comprehensive semantic description. The real distribution can be properly estimated with sufficient supervision of source data. This formulation enables a wide variety of samples from estimated distributions, which is tailored for pixel representation learning in dense prediction tasks. Furthermore, we analyze Pixel-wise Discrimination Distance (PDD) to certify the validity of our method regarding pixel-wise category alignment. Extensive experiments demonstrate that contrastively driving the source and target pixel

representations towards proper semantic concepts can lead to more effective domain alignment and significantly improve the generalization capacity of the model. We hope this exploration will shed light on future studies.

In a nutshell, our contributions can be summarized:

- We provide a new impetus to mitigate domain shift by explicitly enhancing the similarity of pixel features with corresponding semantic concepts and increasing the discrimination power on mismatched pairs, no matter the source or target domain.
- To facilitate efficiency and effectiveness, a closed-form upper bound of the expected contrastive loss is derived with the moments of each category. SePiCo is also a one-stage adaptation framework robust to both daytime and nighttime segmentation situations.
- Extensive experiments on popular semantic segmentation benchmarks show that SePiCo achieves superior performance. Particularly, we obtain mIoUs of 61.0%, 58.1%, and 45.4% on benchmarks GTAV  $\rightarrow$  Cityscapes, SYNTHIA  $\rightarrow$  Cityscapes, and Cityscapes  $\rightarrow$  Dark Zurich respectively. Equipped with the latest Transformer, SePiCo further improves by mIoUs of 9.3%, 5.6%, and 8.0% respectively, setting the new state of the arts. Ablation study and throughout analysis verify the effectiveness of each component.
- Our SePiCo, aiming at a general framework, can further well generalize to unseen target domains and be effortlessly employed to object detection tasks.

## 2 RELATED WORK

Our work draws upon existing literature on semantic image segmentation, domain adaptation, and representation learning. For brevity, we only discuss the most relevant works.

### 2.1 Semantic Segmentation

The recent renaissance in semantic segmentation began with the fully convolutional networks [6]. Mainstream methods strive to enlarge receptive fields and capture context information [7], [8], [45]. Among them, the family of DeepLab enjoys remarkable popularity because of its effectiveness. Inspired by the success of the Transformers [46] in natural language processing, many works adopt it to visual tasks including image classification [47] and semantic segmentation [48], offering breakthrough performance. These studies, though impressive, require a large amount of labeled datasets and struggle to generalize to new domains.

In this work, we operate semantic segmentation under such a domain shift with the aim of learning an adequate model on the unlabeled target domain. Concretely, we map pixel representations in different semantic classes to a distinctive feature space via a pixel-level contrastive learning formulation. The learned pixel features are not only discriminative for segmentation within the source domain, but also, more critically, well-aligned for cross-domain segmentation.

### 2.2 Nighttime Semantic Segmentation

Nighttime Semantic Segmentation is much more challenging in safe autonomous driving due to poor illuminations

and arduous human annotations. Only a handful of works have been investigated in the past few years. Dai *et al.* [49] introduce a two-step adaptation method with the aid of an intermediate twilight domain. Sakaridis *et al.* [50] leverage geometry information to refine predictions and transfer the style of nighttime images to that of daytime images to reduce the domain gap. Recently, Wu *et al.* [51] jointly train a translation model and a segmentation model in one stage, which efficiently performs on par with prior methods.

While daytime and nighttime image segmentation tasks differ only in appearance, current works focus on designing specialized methods for each task. Different from the above methods, SePiCo is able to address both daytime and nighttime image segmentation tasks in a universal framework.

### 2.3 Domain Adaptation in Semantic Segmentation

Domain Adaptation (DA) has been investigated for decades in theory [52], [53] and in various tasks [?], [16], [54], [55], [56]. Given the power of DCNNs, deep DA methods have been gaining momentum to significantly boost the transfer performance of a segmentation model. A multitude of works generally fall into two categories: *adversarial training* [17], [20], [23], [26] and *self-training* [30], [36], [57], [58].

**Adversarial training** methods diminish the distribution shift of two domains at image [17], [18], [19], feature [22], [59], or output [20], [21], [23] level in an adversarial manner. To name a few, Hoffman *et al.* [19] bring DA to segmentation by building generative images for alignment. On the other hand, Tsai *et al.* [20] suggest that performing alignment in the output space is more practical. A few works also leverage different techniques via entropy [21], [22] and information bottleneck [23]. Other concurrent works [23], [26] incorporate category information into the adversarial loss to intensify local semantic consistency. Due to the absence of holistic information about each category, adversarial training is usually less stable. Therefore, some methods instead adopt category anchors [25], [27], [35] computed on source data to advance the alignment. A recent work [60] presents a category contrast method to learn discriminative representation. By contrast, we endeavor to explore semantic concepts from multiple perspectives. More importantly, we set forth a generic semantic-guided pixel contrast to emphasize pixel-wise discriminative learning, allowing us to minimize the intra-class discrepancy and maximize the inter-class margin of pixel representations across domains.

**Self-training** methods exploit unlabeled target data via training with pseudo labels [29], [32], [34], [61], [62]. In an example, Zou *et al.* [63] propose an iterative learning strategy with class balance and spatial prior for target instances. In [30], Tranheden *et al.* propose a domain-mixed self-training pipeline to avoid training instabilities, which mixes images from two domains along with source ground-truth labels and target pseudo labels. Later on, Wang *et al.* [40] enhance self-training via leveraging the auxiliary supervision from depth estimation to diminish the domain gap. Lately, Zhang *et al.* [36] utilize the feature distribution from prototypes to refine target pseudo labels and distill knowledge from a strongly pre-trained model. However, most existing methods always encounter an obstacle in that target representations are dispersed due to the discrepancy

across domains. In addition, most of them utilize a warm-up model to generate initial pseudo labels, which is hard to tune. Differently, our framework performs one-stage end-to-end adaptation produce without using any separate pre-processing stages. In addition, SePiCo can largely improve self-training and easily optimize pixel embedding space.

## 2.4 Representation Learning

To date, unsupervised representation learning has been extensively investigated due to its promising ability to learn representations in the absence of human supervision, especially for contrastive learning [41], [42], [44], [64], [65]. Let  $f$  be an embedding function that transforms a sample  $x$  to an embedding vector  $q = f(x), q \in \mathbb{R}^d$  and let  $(x, x^+)$  be similar pairs and  $(x, x^-)$  be dissimilar pairs. Then, normalize  $q$  onto a unit sphere and a popular contrastive loss such as InfoNCE [65] is formulated as:

$$\mathbb{E}_{q, q^+, \{q_n^-\}_{n=1}^N} \left[ -\log \frac{e^{q^\top q^+ / \tau}}{e^{q^\top q^+ / \tau} + \sum_{n=1}^N e^{q^\top q_n^- / \tau}} \right].$$

In practice, the expectation is replaced by the empirical estimate. As shown above, the contrastive loss is essentially based on the softmax formulation with a temperature  $\tau$ .

Intuitively, the above methods encourage instance discrimination. Recent works [43], [66], [67], [68] also extend contrastive learning to dense prediction tasks. These methods either engage in better visual pre-training for dense prediction tasks [43], [66] or explore dense representation learning in the fully supervised setting [67] or semi-supervised setting [68]. Thereby, they generally tend to learn the pixel correspondence on the category of objects that appear in different views of an image rather than learning the semantic concepts across datasets/domains, so the learned representations cannot directly deploy under domain shift. On the contrary, we draw inspiration from contrastive learning and construct contrastive pairs according to different ways of semantic information to bridge the domain shift, which has received limited consideration in the existing literature.

## 3 METHODOLOGY

In this section, we first briefly introduce the background and illustrate the overall idea in Section 3.1. Then the details of semantic statistics and our framework are elaborated in Section 3.2 and Section 3.3, respectively. Finally, we present the training procedure and the SePiCo algorithm in Section 3.4.

### 3.1 Background

#### 3.1.1 Problem formulation

For domain adaptive semantic segmentation, we have a collection of labeled source data  $I_s$  with pixel-level labels  $Y_s$  as well as unlabeled target data  $I_t$ . The goal is to categorize each pixel of a target image into one of the predefined  $K$  categories through learning a model consisting of a feature encoder  $\Theta_e$ , a multi-class segmentation head  $\Theta_c$ , and an auxiliary projection head  $\Theta_p$ . We adopt the teacher-student architecture [69] (Teacher networks are denoted as  $\Theta'_e$ ,  $\Theta'_c$ , and  $\Theta'_p$ .) as our basic framework, shown in Fig. 3.

During training, images from source and target domains  $I_s, I_t \in \mathbb{R}^{H \times W \times 3}$  are randomly sampled and passed

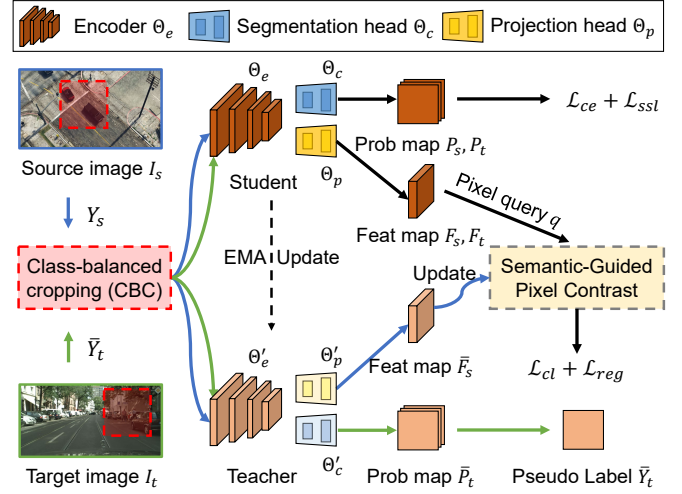


Fig. 3: **Framework overview.** First, our basic framework is based on a teacher-student architecture and the teacher model provides source feature map  $\bar{F}_s$  and target pseudo labels  $\bar{Y}_t$ . Second, we propose class-balanced cropping to frequently crop image patches with under-represented objects that balance performance across classes. And third, except for self training losses,  $\mathcal{L}_{ce}$  and  $\mathcal{L}_{ssl}$ , we contrastively enforce the pixel representations  $F_s, F_t$  towards centroid-aware or distribution-aware semantics using  $\mathcal{L}_{cl}$  and  $\mathcal{L}_{reg}$ . After training is completed, we throw away projection head  $\Theta_p$  and use encoder  $\Theta_e$  and head  $\Theta_c$  for segmentation task.

into both teacher and student networks, respectively. The hidden-layer features  $F_s, F_t \in \mathbb{R}^{H' \times W' \times A}$ , and final pixel-level predictions  $P_s, P_t \in \mathbb{R}^{H \times W \times K}$  are generated from the student, where  $A$  is the channel dimension of intermediate features and  $H' (\ll H), W' (\ll W)$  are spatial dimensions of features. Similarly, we access corresponding source features  $\bar{F}_s \in \mathbb{R}^{H' \times W' \times A}$ , and target pixel-level predictions  $\bar{P}_t$  from the momentum-updated teacher. Note that no gradients will be back-propagated into the teacher network [69].

#### 3.1.2 Self-training domain adaptation revisit

Here, we give an overview of a self-training method [30] for evaluating different semantic-guided pixel contrasts. Traditional self-training methods usually consider two aspects. On the one hand, these methods train a model  $\Theta_c \circ \Theta_e$  to minimize the categorical cross-entropy (CE) loss in the source domain, formalized as a fully supervised problem:

$$\mathcal{L}_{ce} = -\frac{1}{HW} \sum_{i \in \{1, 2, \dots, H \times W\}} \sum_k \mathbb{1}_{[Y_{s,i}=k]} \log P_{s,i}^k, \quad (1)$$

where  $Y_{s,i}$  is the one-hot label for pixel  $i$  in  $I_s$  and  $\mathbb{1}_{[\cdot]}$  is an indicator function that returns 1 if the condition holds or 0 otherwise. On the other hand, to better transfer the knowledge from the source domain to the target domain, self-training usually uses a teacher network to produce more reliable pseudo labels  $\bar{Y}_t$  for a target image,

$$\bar{Y}_{t,j} = \arg \max_k \bar{P}_{t,j}^k, \quad j \in \{1, 2, \dots, H \times W\}. \quad (2)$$

In practice, we compute the pseudo labels online during the training and avoid any additional inference step, which is simpler and more efficient. Specifically, we forward a target image and obtain the pseudo labels using Eq. (2). Besides, since the pseudo labels are usually noisy, a confidence

estimation is made for generated pseudo labels. Specifically, the number of pixels with the maximum softmax probability exceeding a threshold  $\alpha$  is calculated first:

$$NUM_{conf} = \sum_{j \in \{1, 2, \dots, H \times W\}} \mathbb{1}_{[\max_k \bar{P}_{t,j}^k > \alpha]}. \quad (3)$$

Next, the ratio of pixels exceeding the threshold over the whole image serves as confidence weights,  $w = \frac{NUM_{conf}}{HW}$  and the student network is re-trained on target data,

$$\mathcal{L}_{ssl} = -\frac{1}{HW} \sum_{j \in \{1, 2, \dots, H \times W\}} \sum_k w \cdot \mathbb{1}_{[\bar{Y}_{t,j} = k]} \log P_{t,j}^k. \quad (4)$$

Finally, let's go back to the teacher network. The weights of teacher network  $\Theta'_e, \Theta'_c, \Theta'_p$  are set as the exponential moving average (EMA) of the weights of student network  $\Theta_e, \Theta_c, \Theta_p$  in each iteration [69]. Take  $\Theta'_e$  as an example,

$$\Theta'_e \leftarrow \beta \Theta'_e + (1 - \beta) \Theta_e, \quad (5)$$

where  $\beta$  is a momentum parameter. Similarly,  $\Theta'_c, \Theta'_p$  should also be updated via Eq. (5). In this work,  $\beta$  is fixed to 0.999.

Note that incorporating data augmentation with self-training has been shown to be particularly efficient [30], [34]. Following [30], we use the teacher network to generate a set of pseudo labels  $\bar{Y}_t$  on the weakly-augmented target data. Concurrently, the student network is trained on strongly-augmented target data. We use standard resize and random flip as the weak augmentation. Strong augmentation includes color jitter, gaussian blur, and ClassMix [70].

### 3.1.3 Overall motivation

As mentioned before, however, there is a major limitation of traditional self-training methods: most of them neglect explicit domain alignment. As a result, even under perfect pseudo labeling on target samples, the negative transfer may exist, causing pixel features from different domains but of the same semantic class to be mapped farther away. To sidestep this issue, we promote semantic-guided representation learning in the embedding space. A naive way is to directly adopt global category prototypes computed on the source domain to guide the alignment between source and target domains. An obstacle to this design, however, is that prototypes could only reflect the common characteristic of each category but cannot fully unlock the potential strength of semantic information, leading to erroneous representation learning. Inspired by [41], we go a step further and store the local image centroids of each source image into a memory bank so that the semantic information exploited is roughly proportional to the size of the bank. But this mechanism will arise class bias as features of some classes, e.g., bicycles, pedestrians and poles, rarely appear. Meanwhile, this does consume a lot of computing resources.

Consequently, to promote diversity in semantic concepts, we newly introduce the distribution-aware pixel constant to contrastively strengthen the connections between each pixel representation and estimated distributions. Moreover, such a distribution-aware mechanism could be viewed as training on infinite data and is more computation efficient, which is intractable for a memory bank.

## 3.2 Semantic Statistics Calculation

Given the source feature map  $\bar{F}_s \in \mathbb{R}^{H' \times W' \times A}$  from the teacher model, for any pixel indexed  $i \in \{1, 2, \dots, H' \times$

$W'\}$  in  $\bar{F}_s$ , we first divide its feature into the set of  $k^{th}$  semantic class, i.e.,  $\Lambda^k$  according to its mask  $M_{s,i} \in \mathbb{R}^{H' \times W'}$  downsampled from ground truth label. Hereafter, the local centroid of the  $k^{th}$  category in an image is calculated by

$$\mu^{ik} = \frac{1}{|\Lambda^k|} \sum_{i \in \{1, 2, \dots, H' \times W'\}} \mathbb{1}_{[M_{s,i} = k]} \bar{F}_{s,i}, \quad (6)$$

where  $|\cdot|$  is the cardinality of the set.

For centroid-aware semantic information, we require either global category prototypes or local category centroids. On the one side, we opt for an online fashion on the entire source domain, aggregating mean statistics one by one to build global category prototypes. Mathematically, the online estimate algorithm for the mean of the  $k^{th}$  category is given by

$$\mu_{(t)}^k = \frac{n_{(t-1)}^k \mu_{(t-1)}^k + m_{(t)}^k \mu_{(t)}^{ik}}{n_{(t-1)}^k + m_{(t)}^k}, \quad (7)$$

where  $n_{(t-1)}^k$  is the total number of pixels belonging to the  $k^{th}$  category in previous  $t-1$  images, and  $m_{(t)}^k$  is the number of pixels belonging to the  $k^{th}$  category in current  $t^{th}$  image. Thereby, we are allowed to obtain  $K$  global prototypes:

$$\mathcal{P} = \{\mu^1, \mu^2, \dots, \mu^K\}. \quad (8)$$

On the other side, we maintain local centroids of each class from the latest source images to form a dynamic categorical dictionary with  $K$ -group queue,

$$\mathcal{B} = \{\mathcal{B}^1, \mathcal{B}^2, \dots, \mathcal{B}^K\}, \quad (9)$$

where  $\mathcal{B}^k = \{\mu_{(t-B)}^{ik}, \mu_{(t-B+1)}^{ik}, \dots, \mu_{(t)}^{ik}\}$ .  $B$  is the shared queue size for all queues. Note that the oldest centroids are dequeued and currently computed centroids are enqueued.

**Discussion:** *merit and demerit of centroid-aware statistics.*

Each global category prototype renders the overall appearance of one category, yet it might omit diversity and impair the discriminability of the learned representations. On the other way, the memory bank is able to expand the set of negative and positive samples, thus it can embrace more semantic information. More importantly, almost all semantic information could be covered when  $B$  is large enough, however, it is neither elegant nor efficient in presenting pixel embedding space. To capture and utilize rich semantic information as efficiently and comprehensively as possible, we try to build from the distributional perspective as follows.

We observe that pixel representations with respect to each class will have a similar distribution. With this in mind, we propose to build the distribution-aware semantic concepts with sufficient labeled source instances. Therefore, we need to acquire the covariance of the multidimensional feature vector  $\bar{F}_{s,i}$  for a better representation of the variance between any pair of elements in the feature vector. The covariance matrix  $\Sigma^k$  for category  $k$  can be updated via

$$\Sigma_{(t)}^k = \frac{n_{(t-1)}^k \Sigma_{(t-1)}^k + m_{(t)}^k \Sigma_{(t)}^{ik}}{n_{(t-1)}^k + m_{(t)}^k} + \frac{n_{(t-1)}^k m_{(t)}^k (\mu_{(t-1)}^k - \mu_{(t)}^{ik}) (\mu_{(t-1)}^k - \mu_{(t)}^{ik})^\top}{(n_{(t-1)}^k + m_{(t)}^k)^2}, \quad (10)$$

where  $\Sigma_{(t)}^k$  is the covariance matrix of the features between the  $k^{th}$  category in the  $t^{th}$  image. It is noteworthy that  $K$  mean vectors and  $K$  covariance matrices are initialized to

zeros. During training, we dynamically update these statistics using Eq. (7) and Eq. (10) with source feature map  $\bar{F}_s$  from momentum-updated teacher network. The estimated distribution-aware semantic statistics are more informative to guide the pixel representation learning between domains.

### 3.3 Semantic-Guided Pixel Contrast

In the literature, a handful of methods have leveraged categorical feature centroids [25], [27], [35], [60] as anchors to remedy domain shift, yielding promising results. However, few attempts have been made in this regime to quantify the distance between features of different categories. It is arduous to separate pixel representations with similar semantic information in target data as no supervision information is available, which severely limits their potential capability in dense prediction tasks. On the contrary, we design a unified framework to integrate three distinct contrastive losses that target learning similar/dissimilar pairs at the pixel level to mitigate the domain gap via either centroid-aware pixel contrast or distribution-aware pixel contrast.

As stated above, the pixel representation separation in the source domain is naturally guaranteed by source mask  $M_s$  from the ground truth label. Similarly, for target data, we desire to obtain satisfactory target mask  $M_t$  for each pixel via generated pseudo labels from  $\bar{Y}_t$  (Eq. (2)). Whereafter, any pixel representation either in source or target feature maps (defined as the pixel query  $q \in \mathbb{R}^A$  for simplicity) now needs to yield a low loss value when simultaneously forming multiple positive pairs ( $q, q_m^+$ ) and multiple negative pairs ( $q, q_n^{k-}$ ), where  $q_m^+$  indicates the  $m^{\text{th}}$  positive example from the same category considering  $q$  and  $q_n^{k-}$  represents  $n^{\text{th}}$  negative example from the  $k^{\text{th}}$  different class. Formally, we define a new pixel contrast loss function for  $q$ :

$$\ell_q^{cl} = -\frac{1}{M} \sum_{m=1}^M \log \frac{e^{q^\top q_m^+ / \tau}}{e^{q^\top q_m^+ / \tau} + \sum_{k \in \mathcal{K}^-} \frac{1}{N} \sum_{n=1}^N e^{q^\top q_n^{k-} / \tau}}, \quad (11)$$

where  $M$  and  $N$  are the numbers of positive and negative pairs and  $\mathcal{K}^-$  denotes the set containing all different classes from that of  $q$ . In the following sections, we will describe three pixel contrast losses  $\ell_q^{\text{protocl}}$ ,  $\ell_q^{\text{bankcl}}$  and  $\ell_q^{\text{distcl}}$  respectively to derive a better-structured embedding space, eventually boosting the performance of segmentation model.

In short, we enable learning discriminative pixel representations across domains via a unified contrastive loss

$$\mathcal{L}_{cl} = \frac{1}{|\Psi|} \sum_{q \in F_s \cup F_t} \ell_q^{cl}, \quad (12)$$

where  $|\Psi|$  is the total number of pixels in the union of  $F_s$  and  $F_t$ . Note that such contrastive loss is employed in both domains simultaneously. For one thing, when the loss is applied in the source domain, the student network is capable of yielding more discriminative representations for pixel-level predictions, which increases the robustness of the model. Another effect is that the target representations are contrastively adapted in a pixel-wise manner, which benefits minimizing the intra-category discrepancy and maximizing the inter-category margin and facilitates transferring knowledge from source to target explicitly.

Moreover, except for individual pixel representation learning, we introduce a regularization term to make the

feature representations of input images globally diverse and smooth, which is formalized as

$$\mathcal{L}_{reg} = \frac{1}{K \log K} \sum_{k=1}^K \log \frac{e^{Q^\top \mu^k / \tau}}{\sum_{l=1}^K e^{Q^\top \mu^l / \tau}}, \quad (13)$$

where  $Q = \frac{1}{H' \times W'} \sum_{i \in \{1, 2, \dots, H' \times W'\}} F_{s/t, i}$  is the mean feature representation of a source or target image. This objective is similar to the diversity-promoting objective used in prior DA methods [71], but is employed in the embedding space. It could circumvent the trivial solution where all unlabeled target data have the same feature encoding.

#### 3.3.1 Centroid-aware Pixel Contrast

Here, we introduce two variants of centroid-aware pixel contrast, namely SePiCo (ProtoCL) and SePiCo (BankCL).

CASE 1: **ProtoCL** ( $M = N = 1$ ). Naively operate  $K$  global category prototypes to establish one positive pair and  $K - 1$  negative pairs. We consider this formulation as the prototype pixel contrast loss function

$$\ell_q^{\text{protocl}} = -\log \frac{e^{q^\top \mu^+ / \tau}}{e^{q^\top \mu^+ / \tau} + \sum_{k \in \mathcal{K}^-} e^{q^\top \mu^{k-} / \tau}}, \quad (14)$$

where  $\mu^+$  is the positive prototype belonging to the same category as the specific query  $q$  and  $\mu^{k-}$  is the prototype of the  $k^{\text{th}}$  different category.

CASE 2: **BankCL** ( $M = N = B$ ). To involve more negative and positive samples for representation learning, we could access more contrastive pairs from a memory bank, in which local category centroids of a single source image are stored. We consider this formulation as the bank pixel contrast loss function

$$\ell_q^{\text{bankcl}} = -\mathbb{E}_{q^+ \in \mathcal{B}^+} \log \frac{e^{q^\top q^+ / \tau}}{e^{q^\top q^+ / \tau} + \sum_{k \in \mathcal{K}^-} \mathbb{E}_{q^{k-} \in \mathcal{B}^{k-}} e^{q^\top q^{k-} / \tau}}, \quad (15)$$

where  $\mathcal{B}^+$  is the queue comprised of positive samples and  $\mathcal{B}^{k-}$  refs to a queue containing negative ones. In Section 4.3, we will analyze the effect of bank size  $B$ .

**Discussion:** *merit and demerit of centroid-aware pixel contrast.* In a word, global prototypes or local centroids can be used as good contrastive samples to pull similar pixel representations closer and push those dissimilar pixel representations away in the embedding space. However, from Eq. (14) and Eq. (15), we can theorize that the main difference between them is the number of positive and negative pairs. Because of this, if the number of contrastive pairs does matter, it is intuitively reasonable that an infinite number of such pairs would contribute to the establishment of a more robust and discriminative embedding space. We will justify this assumption from the distributional perspective.

#### 3.3.2 Distribution-aware Pixel Contrast

In this part, we derive a particular form of contrastive loss where infinite positive/negative pixel pairs are simultaneously involved with regard to each pixel representation in the source and target domain. A naive implementation is to explicitly sample  $M$  examples from the estimated distribution that has the same latent class and  $N$  examples from each of the other distributions featuring different semantic concepts. Unfortunately, this is not computationally feasible

when  $M$  and  $N$  are large, as carrying all positive/negative pairs in an iteration would quickly drain the GPU memory.

To get around this issue, we take an infinity limit on the number of  $M$  and  $N$ , where the effect of  $M$  and  $N$  are hopefully absorbed in a probabilistic way. With this application of infinity limit, the statistics of the data are sufficient to achieve the same goal of multiple pairing. As  $M, N$  goes to infinity, it becomes the estimation of:

$$\begin{aligned} \ell_q^\infty &= \lim_{M \rightarrow \infty, N \rightarrow \infty} \ell_q^{cl} \\ &= \lim_{M \rightarrow \infty} -\frac{1}{M} \sum_{m=1}^M \log \frac{e^{q^\top q^+ / \tau}}{e^{q^\top q^+ / \tau} + \sum_{k \in \mathcal{K}^-} \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N e^{q^\top q^{k-} / \tau}} \\ &= -\mathbb{E}_{q^+ \sim p(q^+)} \log \frac{e^{q^\top q^+ / \tau}}{e^{q^\top q^+ / \tau} + \sum_{k \in \mathcal{K}^-} \mathbb{E}_{q^{k-} \sim p(q^{k-})} e^{q^\top q^{k-} / \tau}}, \end{aligned}$$

where  $p(q^+)$  is the positive semantic distribution with the same label as  $q$  and  $p(q^{k-})$  is the  $k^{th}$  negative semantic distribution with a different label from that of each query  $q$ . The analytic form of the above is intractable, but it has a rigorous closed form of upper bound, which can be derived

$$\begin{aligned} \ell_q^\infty &= -\mathbb{E}_{q^+} \log \frac{e^{q^\top q^+ / \tau}}{e^{q^\top q^+ / \tau} + \sum_{k \in \mathcal{K}^-} \mathbb{E}_{q^{k-}} e^{q^\top q^{k-} / \tau}} \\ &\leq \log \left[ \mathbb{E}_{q^+} \left[ e^{\frac{q^\top q^+}{\tau}} + \sum_{k \in \mathcal{K}^-} \mathbb{E}_{q^{k-}} e^{\frac{q^\top q^{k-}}{\tau}} \right] \right] - q^\top \mathbb{E}_{q^+} \left[ \frac{q^+}{\tau} \right] \\ &= \log \left[ \mathbb{E}_{q^+} e^{\frac{q^\top q^+}{\tau}} + \sum_{k \in \mathcal{K}^-} \mathbb{E}_{q^{k-}} e^{\frac{q^\top q^{k-}}{\tau}} \right] - q^\top \mathbb{E}_{q^+} \left[ \frac{q^+}{\tau} \right] \\ &= \ell_q^{distcl}, \end{aligned} \quad (16)$$

where the above inequality follows from Jensen's inequality on concave functions, i.e.,  $\mathbb{E} \log(X) \leq \log \mathbb{E}(X)$ . Thus, distribution-aware pixel contrast loss, i.e., SePiCo (DistCL) is yielded to implicitly explore infinite samples.

Next, to facilitate our formulation, we further need an assumption on the feature distribution. For any random variable  $x$  that follows Gaussian distribution  $x \sim \mathcal{N}(\mu, \Sigma)$ , we have the moment generation function [72] that satisfies:

$$\mathbb{E} \left[ e^{a^\top x} \right] = e^{a^\top \mu + \frac{1}{2} a^\top \Sigma a},$$

where  $\mu$  is the expectation of  $x$ ,  $\Sigma$  is the covariance matrix of  $x$ . Therefore, we assume that  $q^+ \sim \mathcal{N}(\mu^+, \Sigma^+)$  and  $q^{k-} \sim \mathcal{N}(\mu^{k-}, \Sigma^{k-})$ , where  $\mu^+$  and  $\Sigma^+$  are respectively the statistics i.e., mean and covariance matrix, of the positive semantic distribution for  $q$ ,  $\mu^{k-}$  and  $\Sigma^{k-}$  are respectively the statistics of the  $k^{th}$  negative distribution. Under this assumption, Eq. (16) for a certain pixel representation  $q$  immediately reduces to

$$\begin{aligned} \ell_q^{distcl} &= \log \left[ e^{\frac{q^\top \mu^+}{\tau} + \frac{q^\top \Sigma^+ q}{2\tau^2}} + \sum_{k \in \mathcal{K}^-} e^{\frac{q^\top \mu^{k-}}{\tau} + \frac{q^\top \Sigma^{k-} q}{2\tau^2}} \right] - \frac{q^\top \mu^+}{\tau} \\ &= -\log \frac{e^{\frac{q^\top \mu^+}{\tau} + \frac{q^\top \Sigma^+ q}{2\tau^2}}}{e^{\frac{q^\top \mu^+}{\tau} + \frac{q^\top \Sigma^+ q}{2\tau^2}} + \sum_{k \in \mathcal{K}^-} e^{\frac{q^\top \mu^{k-}}{\tau} + \frac{q^\top \Sigma^{k-} q}{2\tau^2}}} + \frac{q^\top \Sigma^+ q}{2\tau^2}. \end{aligned}$$

Eventually, the overall loss function regarding each pixel-wise representation thereby boils down to the closed form whose gradients can be analytically solved.

---

### Algorithm 1: Pseudocode of class-balanced cropping on an unlabeled target image (PyTorch-style)

---

```
# img: an unlabeled target image to be cropped
# pl: corresponding pseudo label of img
# cat_max_ratio: max ratio of a category in img

best_score = -1, best_crop_box = None # initialize

# randomly crop N_crop times and get the best crop
for _ in range(N_crop):
    score = 0 # initial score
    # get a random box crop
    box_crop = get_random_box_crop(img)
    pl_crop = crop(pl, box_crop) # crop pl by crop_box
    # unique classes with pixel count in cropped pl
    classes, cnt = unique_with_counts(pl_crop)
    # category max ratio should be satisfied
    if max(cnt) / sum(cnt) < cat_max_ratio:
        score = sum(log(cnt)) # calculate score
    # compare and get the best
    if score > best_score:
        best_score, best_box_crop = score, box_crop

# perform class-balanced cropping (CBC)
img = crop(img, best_box_crop)
pl = crop(pl, best_box_crop)
```

---

### 3.4 Training Procedure

In brief, the training procedure of SePiCo can be optimized in a one-stage manner, and we further introduce class-balanced cropping in Alg. 1 to stabilize and regularize the process. We summarize the algorithm in Alg. 2.

#### 3.4.1 Class-balanced Cropping (CBC)

As we are all aware, realistic segmentation datasets are highly imbalanced. Thus, one challenge of training a capable model under distribution shift is overfitting to the majority classes of the source domain. One can solve this during training through *class-balanced sampling over the entire dataset* like rare class sampling (RCS) in DAFormer [28]. Though this strategy is effective, it is only suited to source-domain data with ground-truth labels. Unfortunately, there are no available annotations for target-domain data. To handle this issue, we utilize generated target pseudo labels and provide an alternative strategy, *class-balanced cropping within a single image*, to crop image regions that jointly promote class balance in pixel number and diversity of internal categories (see Alg. 1<sup>1</sup>). Note that we turn to this online strategy for the fact that pseudo labels are constantly changing, thus collecting class statistics over the whole target dataset (like RCS on the whole source dataset) could be much more inefficient. On this basis, we employ RCS for the entire source domain while CBC for a single target image. Accordingly, samples with smaller class frequencies throughout the source domain will have a higher sampling probability, while regions of an unlabeled target image with multiple classes will enjoy a higher cropping probability. Experimentally, we also compare RCS and CBC in Sec. 4.3.

#### 3.4.2 Optimization Objective

The well-known self-training extensively studied in previous methods [27], [33], [36], [37], is usually achieved by iteratively generating a set of pseudo labels based on the most confident predictions on target data. Nevertheless, it primarily depends on a good initialization model and

<sup>1</sup>We fix  $N\_crop=10$  and  $cat\_max\_ratio=0.75$  for all experiments.

**Algorithm 2: SePiCo algorithm.**


---

```

1 Input: Input data  $I_s, Y_s, I_t$ , bank size  $B$ , parameters
    $\lambda_{cl}, \lambda_{reg}$  and maximum/warm-up iteration  $L/L_w$ .
2 Initialize  $\Theta_e$  with ImageNet pre-trained parameters
   and randomly initialize two heads  $\Theta_c$  and  $\Theta_p$ .
3 Initialize statistics  $\{\mu^k\}_{k=1}^K$  and  $\{\Sigma^k\}_{k=1}^K$  to zeros.
4 Teachers init:  $\Theta'_e \leftarrow \Theta_e, \Theta'_c \leftarrow \Theta_c, \Theta'_p \leftarrow \Theta_p$ .
5 for  $iter \leftarrow 0$  to  $L$  do
6   Randomly sample a source image  $I_s$  with  $Y_s$  and
   a target image  $I_t$ .
7   Apply class-balance cropping on both  $I_s$  and  $I_t$ .
8   Obtain feature maps  $F_s$  and  $F_t$  and separate
   pixel-wise representations in the embedding
   space using corresponding masks  $M_s$  and  $M_t$ .
9   Update mean  $\{\mu^k\}_{k=1}^K$  via Eq. (7) and covariance
   matrices  $\{\Sigma^k\}_{k=1}^K$  via Eq. (10) or memory bank
   with current image-wise centroids  $\{\mu^k\}_{k=1}^K$ .
   if  $iter > L_w$  then
10    Train  $\Theta_e, \Theta_c, \Theta_p$  using  $\mathcal{L}_{ce}, \mathcal{L}_{ssl}, \mathcal{L}_{cl}, \mathcal{L}_{reg}$ .
   else
11    Train  $\Theta_e, \Theta_c$  using  $\mathcal{L}_{ce}, \mathcal{L}_{ssl}$ .
12  Update  $\Theta'_e, \Theta'_c, \Theta'_p$  with  $\Theta_e, \Theta_c, \Theta_p$  via Eq. (5).
Return: Final network weights  $\Theta_c$  and  $\Theta_e$ .

```

---

is hard to tune. Our SePiCo aims to learn a discriminative embedding space and is complementary to the self-training. Therefore, we unify both into a one-stage, end-to-end pipeline to stabilize training and yield discriminative features, which promotes the generalization ability of the model. The overall training objective is formulated as:

$$\min_{\Theta_e, \Theta_c, \Theta_p} \mathcal{L}_{ce} + \mathcal{L}_{ssl} + \lambda_{cl} \mathcal{L}_{cl} + \lambda_{reg} \mathcal{L}_{reg}, \quad (17)$$

where  $\lambda_{cl}, \lambda_{reg}$  are constants controlling the strength of corresponding loss. Initial tests suggest that using equal weights to combine the  $\mathcal{L}_{cl}$  with  $\mathcal{L}_{reg}$  yields better results. For simplicity, both are set to 1.0 without any tuning. By optimizing Eq. (17), clusters of pixels belonging to the same category are pulled together in the feature space while synchronously pushed apart from other categories, which eventually establishes a discriminative embedding space. In this way, our method can simultaneously minimize the gap across domains as well as enhance the intra-class compactness and inter-class separability in a unified framework. Meanwhile, it is beneficial for the generation of reliable pseudo labels which in turn facilitates self-training.

## 4 EXPERIMENT

In this section, we validate SePiCo on two popular synthetic-to-real tasks and challenging daytime-to-nighttime tasks. First, we describe datasets and implementation details. Next, numerous experimental results are reported for comparison across diverse datasets and architectures. Finally, we conduct detailed analyses to obtain a complete picture.

### 4.1 Experimental Setups

#### 4.1.1 Datasets

GTAV [9] is a composite image dataset sharing 19 classes with Cityscapes. 24,966 city scene images are extracted from

the physically-based rendered computer game ‘‘Grand Theft Auto V’’ and are used as source domain data for training.

**SYNTHIA** [10] is a synthetic urban scene dataset. Following [20], [28], we select its subset, called SYNTHIA-RANDCITYSCAPES, that has 16 common semantic annotations with Cityscapes. In total, 9,400 images with the resolution 1280×760 from SYNTHIA dataset are used as source data.

**Cityscapes** [74] is a dataset of real urban scenes taken from 50 cities in Germany and neighboring countries. We use finely annotated images which consist of 2,975 training images, 500 validation images, and 1,525 test images, with a resolution at 2048×1024. Each pixel of the image is divided into 19 categories. For synthetic-to-real adaptation [28], [30], [36], we adopt training images as unlabeled target domain and operate evaluations on its validation set. For daytime-to-nighttime adaptation [50], [51], [75], we use all images from the training set as the source training data.

**Dark Zurich** [75] is another real-world dataset consisting of 2,416 nighttime images, 2,920 twilight images and 3,041 daytime images, with a resolution of 1920×1080. Following [51], we utilize 2,416 day-night image pairs as target training data and another 151 test images as target test data that serves as an online benchmark evaluating via online site<sup>2</sup>.

#### 4.1.2 Implementation Details

**Network architecture.** Our implementation is based on the mmsegmentation toolbox<sup>3</sup>. For CNN-based architectures, we utilize the DeepLab-V2 [8] with ResNet101 [76] as the backbone. For recent Transformer-based ones, we adopt the same framework used in DAFormer [28] as a strong backbone. As for the segmentation head, We follow the mainstream pipelines [20], [28], [30], [34]. Subsequently, a projection head is integrated into the network that maps high-dimensional pixel embedding into a 512-d  $\ell_2$ -normalized feature vector [67]. It consists of two 1×1 convolutional layers with ReLU. For fairness, all backbones are initialized using the weights pre-trained on ImageNet [77], with the remaining layers being initialized randomly.

**Training.** Our model is implemented in PyTorch [78] and trained on a single NVIDIA Tesla V100 GPU. We use the AdamW [79] as our optimizer with betas (0.9, 0.999) and weight decay 0.01. The learning rate is initially set to  $6 \times 10^{-5}$  for the encoder and  $6 \times 10^{-4}$  for decoders. Similar to [28], learning rate warmup policy and rare class sampling are also applied. In all experiments, we set trade-offs  $\lambda_{cl}, \lambda_{reg}$  to 1.0, 1.0 and threshold  $\alpha$ , momentum  $\beta$ , and bank size  $B$  to 0.968, 0.999, 200 respectively. We train the network with a batch of two 640×640 random crops for a total of 40k iterations. The statistics in Section 3.2 are estimated right from the beginning, but pixel contrast starts from  $L_w$  (default 3k) iteration to stabilize training.

**Testing.** At the test stage, we only resize the validation images to 1280×640 as the input image. Note that there is no extra inference step inserted into the basic segmentation model, that is, the teacher network, projection head  $\Theta_p$ , and memory bank  $\mathcal{B}$ , are directly discarded. We employ per-class intersection-over-union (IoU) and mean IoU over all classes as the evaluation metric which is broadly adopted in semantic segmentation [20], [28], [36], [51].

<sup>2</sup><https://competitions.codalab.org/competitions/23553>

<sup>3</sup><https://github.com/open-mmlab/msegmentation>



TABLE 1: Comparison results of GTAV  $\rightarrow$  Cityscapes. All methods are based on DeepLab-V2 with ResNet-101 for a fair comparison. The best result is highlighted in bold.

Method	road	side.	buil.	wall	fence	pole	light	sign	veg.	terr.	sky	pers.	rider	car	truck	bus	train	mbike	bike	mIoU
Source Only	70.2	14.6	71.3	24.1	15.3	25.5	32.1	13.5	82.9	25.1	78.0	56.2	33.3	76.3	26.6	29.8	12.3	28.5	18.0	38.6
AdaptSeg [20]	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
CLAN [23]	88.7	35.5	80.3	27.5	25.0	29.3	36.4	28.1	84.5	37.0	76.6	58.4	29.7	81.2	38.8	40.9	5.6	32.9	28.8	45.5
CBST [63]	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
MRKLD [33]	91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	26.9	26.0	42.3	47.1
PLCA [62]	84.0	30.4	82.4	35.3	24.8	32.2	36.8	24.5	85.5	37.2	78.6	66.9	32.8	85.5	40.4	48.0	8.8	29.8	41.8	47.7
BDL [38]	91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	43.6	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5
SIM [25]	90.6	44.7	84.8	34.3	28.7	31.6	35.0	37.6	84.7	43.3	85.3	57.0	31.5	83.8	42.6	48.5	1.9	30.4	39.0	49.2
CaCo [60]	91.9	54.3	82.7	31.7	25.0	38.1	46.7	39.2	82.6	39.7	76.2	63.5	23.6	85.1	38.6	47.8	10.3	23.4	35.1	49.2
ConDA [32]	93.5	56.9	85.3	38.6	26.1	34.3	36.9	29.9	85.3	40.6	88.3	58.1	30.3	85.8	39.8	51.0	0.0	28.9	37.8	49.9
FADA [26]	91.0	50.6	86.0	43.4	29.8	36.8	43.4	25.0	86.8	38.3	87.4	64.0	38.0	85.2	31.6	46.1	6.5	25.4	37.1	50.1
LTIR [59]	92.9	55.0	85.3	34.2	31.1	34.9	40.7	34.0	85.2	40.1	87.1	61.0	31.1	82.5	32.3	42.9	0.3	36.4	46.1	50.2
CAG-UDA [27]	90.4	51.6	83.8	34.2	27.8	38.4	25.3	48.4	85.4	38.2	78.1	58.6	34.6	84.7	21.9	42.7	41.1	29.3	37.2	50.2
PixMatch [58]	91.6	51.2	84.7	37.3	29.1	24.6	31.3	37.2	86.5	44.3	85.3	62.8	22.6	87.6	38.9	52.3	0.7	37.2	50.0	50.3
Seg-Uncert. [31]	90.4	31.2	85.1	36.9	25.6	37.5	48.8	48.5	85.3	34.8	81.1	64.4	36.8	86.3	34.9	52.2	1.7	29.0	44.6	50.3
FDA-MBT [18]	92.5	53.3	82.4	26.5	27.6	36.4	40.6	38.9	82.3	39.8	78.0	62.6	34.4	84.9	34.1	53.1	16.9	27.7	46.4	50.5
KATPAN [29]	90.8	49.8	85.1	39.5	28.4	30.5	43.1	34.7	84.9	38.9	84.7	62.6	31.6	85.1	38.7	51.8	26.2	35.4	42.6	51.8
DACS [30]	89.9	39.7	87.9	30.7	39.5	38.5	46.4	52.8	88.0	44.0	88.8	67.2	35.8	84.5	45.7	50.2	0.0	27.3	34.0	52.1
MetaCorrection [73]	92.8	58.1	86.2	39.7	33.1	36.3	42.0	38.6	85.5	37.8	87.6	62.8	31.7	84.8	35.7	50.3	2.0	36.8	48.0	52.1
IAST [61]	94.1	58.8	85.4	39.7	29.2	25.1	43.1	34.2	84.8	34.6	88.7	62.7	30.3	87.6	42.3	50.3	24.7	35.2	40.2	52.2
UPLR [37]	90.5	38.7	86.5	41.1	32.9	40.5	48.2	42.1	86.5	36.8	84.2	64.5	38.1	87.2	34.8	50.4	0.2	41.8	54.6	52.6
DPL-dual [57]	92.8	54.4	86.2	41.6	32.7	36.4	49.0	34.0	85.8	41.3	86.0	63.2	34.2	87.2	39.3	44.5	18.7	42.6	43.1	53.3
SAC [34]	90.4	53.9	86.6	42.4	27.3	45.1	48.5	42.7	87.4	40.1	86.1	67.5	29.7	88.5	49.1	54.6	9.8	26.6	45.3	53.8
CTF [35]	92.5	58.3	86.5	27.4	28.8	38.1	46.7	42.5	85.4	38.4	<b>91.8</b>	66.4	37.0	87.8	40.7	52.4	<b>44.6</b>	41.7	59.0	56.1
CorDA [40]	94.7	63.1	87.6	30.7	40.6	40.2	47.8	51.6	87.6	<b>47.0</b>	89.7	66.7	35.9	90.2	48.9	57.5	0.0	39.8	56.0	56.6
ProDA [36]	87.8	56.0	79.7	<b>46.3</b>	<b>44.8</b>	<b>45.6</b>	53.5	53.5	88.6	45.2	82.1	70.7	39.2	<b>88.8</b>	45.5	59.4	1.0	<b>48.9</b>	56.4	57.5
SePiCo (ProtoCL)	95.6	69.2	<b>89.0</b>	40.8	38.6	44.3	<b>56.3</b>	<b>64.4</b>	88.3	46.5	88.6	<b>73.1</b>	47.6	90.7	58.9	53.8	5.4	22.4	43.8	58.8
SePiCo (BankCL)	<b>96.1</b>	<b>72.1</b>	88.6	43.1	42.4	43.7	56.0	63.5	<b>88.9</b>	44.5	89.0	72.7	45.7	91.1	61.7	59.6	0.0	24.7	53.6	59.8
SePiCo (DistCL)	95.2	67.8	88.7	41.4	38.4	43.4	55.5	63.2	88.6	46.4	88.3	<b>73.1</b>	<b>49.0</b>	<b>91.4</b>	<b>63.2</b>	<b>60.4</b>	0.0	45.2	<b>60.0</b>	<b>61.0</b>

## 4.2 Experimental Results

We comprehensively compare our SePiCo with the recently leading approaches in two representative synthetic-to-real adaptation scenarios: GTAV  $\rightarrow$  Cityscapes in TABLE 1, and SYNTHIA  $\rightarrow$  Cityscapes in TABLE 2 and a challenging daytime-to-nighttime scenario: Cityscapes  $\rightarrow$  Dark Zurich in TABLE 3. Additionally, we provide some qualitative results in Fig. 4 and Fig. 5. Next, due to the great potential of Vision Transformer, we evaluate our framework on the above three benchmarks and list results in TABLE 4. Last but not least, SePiCo can be applied to domain generalization setting in TABLE 5 and detection task in TABLE 6.

### 4.2.1 Comparisons with the state-of-the-arts

**GTAV  $\rightarrow$  Cityscapes.** We first present the adaptation results on the task of GTAV  $\rightarrow$  Cityscapes in TABLE 1, with comparisons to the state-of-the-art DA approaches [29], [32], [35], [36], [40], [60], and the best results are highlighted in bold. Overall, our SePiCo (ProtoCL) sets the new state of the art. Particularly, we observe: (i) SePiCo (DistCL) achieves 61.0% mIoU, outperforming the baseline model trained merely on source data by a large margin of +22.4% mIoU; (ii) Due to the rare presence of “train” class in an image and its significant appearance difference across domains, our SePiCo fails to predict them well. (iii) Adversarial training methods, e.g., AdaptSeg [20], CLAN [23], FADA [26], can improve the transferability, but the effect is not as obvious as using self-training methods, e.g., Seg-Uncert. [31], DACS [30], IAST [61], SAC [34]; (iv) On top of that, our SePiCo (DistCL) beats the best-performing model, ProDA [36], by a considerable margin of +3.5% mIoU, while ProDA has three complex training stages including warm up, self-training, and knowledge distillation.

Comparing the three variants of our framework, SePiCo (ProtoCL) and SePiCo (BankCL) also achieve remarkable

mIoUs of 59.5% and 60.4% respectively. It is clear that BankCL and DistCL perform much better than ProtoCL, indicating features of higher quality are generated thanks to semantic concepts with greater diversity. It is worth reminding that methods built on memory banks are generally slower and demands more memory in training, while SePiCo (DistCL) eases such burden and still manages to surpass SePiCo (BankCL) at the same time.

**SYNTHIA  $\rightarrow$  Cityscapes.** As revealed in TABLE 2, our SePiCo remains competitive on SYNTHIA  $\rightarrow$  Cityscapes. SePiCo (DistCL) attains 58.1% mIoU and 66.5% mIoU\*, achieving a significant gain of +24.6% mIoU and +27.9% mIoU\* in comparison with “Source Only” model. It is noticeable that our SePiCo (DistCL) ranks among the best in both mIoU and mIoU\*, outperforming ProDA [36] by +2.6% mIoU and CorDA [40] by +3.7% mIoU\*. The former is a multi-stage self-training framework and the latter combines auxiliary tasks, i.e., depth estimation, to facilitate knowledge transfer to the target domain. SePiCo (ProtoCL/BankCL) also obtain a comparable performance in terms of mIoU\* compared with SePiCo (DistCL), but under-perform or tie with it in mIoU, indicating that a more class-balanced performance is done by SePiCo (DistCL).

**Cityscapes  $\rightarrow$  Dark Zurich.** TABLE 3 highlights the capability of our SePiCo on the challenging daytime-to-nighttime task Cityscapes  $\rightarrow$  Dark Zurich. To show that the current daytime-trained semantic segmentation models face significant performance degradation at night, we compare with AdaptSeg [20], AdvEnt [21], and BDL [38], adopting DeepLab-V2 as backbone network. Our framework, especially SePiCo (BankCL) and SePiCo (DistCL), outperforms the comparison counterparts by a large margin. The less powerful variant, SePiCo (ProtoCL), is still able to win by a narrow margin when compared to the previous SOTA DANNet [51]. Due to the huge domain divergence between daytime and nighttime scenarios, there are always two steps

TABLE 2: Comparison results of **SYNTHIA**  $\rightarrow$  **Cityscapes**. mIoU\* denotes the mean IoU of 13 classes excluding the classes with \*. All methods are based on DeepLab-V2 with ResNet-101 for a fair comparison. The best result is highlighted in **bold**.

Method	road	side.	buil.	wall*	fence*	pole*	light	sign	veg.	sky	pers.	rider	car	bus	mbike	bike	mIoU	mIoU*
Source Only	55.6	23.8	74.6	9.2	0.2	24.4	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	33.5	38.6
AdaptSeg [20]	79.2	37.2	78.8	10.5	0.3	25.1	9.9	10.5	78.2	80.5	53.5	19.6	67.0	29.5	21.6	31.3	39.5	45.9
CLAN [23]	82.7	37.2	81.5	-	-	-	17.7	13.1	81.2	83.3	55.5	22.1	76.6	30.1	23.5	30.7	-	48.8
CBST [63]	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	78.3	60.6	28.3	81.6	23.5	18.8	39.8	42.6	48.9
LTIR [59]	92.6	53.2	79.2	-	-	-	1.6	7.5	78.6	84.4	52.6	20.0	82.1	34.8	14.6	39.4	-	49.3
MRKLD [33]	67.7	32.2	73.9	10.7	1.6	37.4	22.2	31.2	80.8	80.5	60.8	29.1	82.8	25.0	19.4	45.3	43.8	50.1
BDL [38]	86.0	46.7	80.3	-	-	-	14.1	11.6	79.2	81.3	54.1	27.9	73.7	42.2	25.7	45.3	-	51.4
SIM [25]	83.0	44.0	80.3	-	-	-	17.1	15.8	80.5	81.8	59.9	33.1	70.2	37.3	28.5	45.8	-	52.1
FDA-MBT [18]	79.3	35.0	73.2	-	-	-	19.9	24.0	61.7	82.6	61.4	31.1	83.9	40.8	38.4	51.1	-	52.5
CAG-UDA [27]	84.7	40.8	81.7	7.8	0.0	35.1	13.3	22.7	84.5	77.6	64.2	27.8	80.9	19.7	22.7	48.3	44.5	-
MetaCorrection [73]	92.6	52.7	81.3	8.9	2.4	28.1	13.0	7.3	83.5	85.0	60.1	19.7	84.8	37.2	21.5	43.9	45.1	52.5
FADA [26]	84.5	40.1	83.1	4.8	0.0	34.3	20.1	27.2	84.8	84.0	53.5	22.6	85.4	43.7	26.8	27.8	45.2	52.5
ConDA [32]	88.1	46.7	81.1	10.6	1.1	31.3	22.6	19.6	81.3	84.3	53.9	21.7	79.8	42.9	24.2	46.8	46.0	53.3
CaCo [60]	87.4	48.9	79.6	8.8	0.2	30.1	17.4	28.3	79.9	81.2	56.3	24.2	78.6	39.2	28.1	48.3	46.0	53.6
PixMatch [58]	92.5	54.6	79.8	4.78	0.08	24.1	22.8	17.8	79.4	76.5	60.8	24.7	85.7	33.5	26.4	54.4	46.1	54.5
PLCA [62]	82.6	29.0	81.0	11.2	0.2	33.6	24.9	18.3	82.8	82.3	62.1	26.5	85.6	48.9	26.8	52.2	46.8	54.0
DPL-Dual [57]	87.5	45.7	82.8	13.3	0.6	33.2	22.0	20.1	83.1	86.0	56.6	21.9	83.1	40.3	29.8	45.7	47.0	54.2
Seg-Uncert. [31]	87.6	41.9	83.1	14.7	1.7	36.2	31.3	19.9	81.6	80.6	63.0	21.8	86.2	40.7	23.6	53.1	47.9	54.9
UPLR [37]	79.4	34.6	83.5	19.3	2.8	35.3	32.1	26.9	78.8	79.6	66.6	30.3	86.1	36.6	19.5	56.9	48.0	54.6
CTF [35]	75.7	30.0	81.9	11.5	2.5	35.3	18.0	32.7	86.2	90.1	65.1	33.2	83.3	36.5	35.3	54.3	48.2	55.5
DACS [30]	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24.0	83.7	<b>90.8</b>	67.6	38.3	82.9	38.9	28.5	47.6	48.3	54.8
IAST [61]	81.9	41.5	83.3	17.7	4.6	32.3	30.9	28.8	83.4	85.0	65.5	30.8	86.5	38.2	33.1	52.7	49.8	57.0
KATPAN [29]	82.3	40.8	83.7	19.2	1.8	34.6	29.5	32.7	82.9	83.4	67.3	32.8	86.1	41.2	33.5	52.1	50.2	57.6
SAC [34]	89.3	47.2	85.5	26.5	1.3	43.0	45.5	32.0	87.1	89.3	63.6	25.4	86.9	35.6	30.4	53.0	52.6	59.3
CorDA [40]	<b>93.3</b>	<b>61.6</b>	85.3	19.6	<b>5.1</b>	<b>37.8</b>	36.6	42.8	84.9	90.4	69.7	41.8	85.6	38.4	32.6	53.9	55.0	62.8
ProDA [36]	87.8	45.7	84.6	<b>37.1</b>	0.6	<b>44.0</b>	<b>54.6</b>	<b>37.0</b>	<b>88.1</b>	<b>84.4</b>	<b>74.2</b>	<b>41.8</b>	<b>88.2</b>	<b>51.1</b>	<b>40.5</b>	<b>45.6</b>	55.5	62.0
<b>SePiCo (ProtoCL)</b>	79.2	42.9	<b>85.6</b>	9.9	4.2	38.0	52.5	53.3	80.6	81.2	73.7	<b>47.4</b>	86.2	63.1	48.0	63.2	56.8	65.9
<b>SePiCo (BankCL)</b>	76.7	34.3	84.9	18.7	2.9	38.4	51.8	<b>55.6</b>	85.0	84.6	73.2	45.0	<b>89.7</b>	63.7	50.5	63.8	57.4	66.1
<b>SePiCo (DistCL)</b>	77.0	35.3	85.1	23.9	3.4	38.0	51.0	55.1	85.6	80.5	73.5	46.3	87.6	<b>69.7</b>	<b>50.9</b>	<b>66.5</b>	<b>58.1</b>	<b>66.5</b>

TABLE 3: Comparison results of **Cityscapes**  $\rightarrow$  **Dark Zurich**. The DeepLab-V2 (D) [8] and RefineNet (R) [80] architecture with ResNet-101 trained on Cityscapes are used as Source Only baselines. The best result is highlighted in **bold**.

Method	road	side.	buil.	wall	fence	pole	light	sign	veg.	terr.	sky	pers.	rider	car	truck	bus	train	mbike	bike	mIoU	
Source Only	R	68.8	23.2	46.8	20.8	12.6	29.8	30.4	26.9	43.1	14.3	0.3	36.9	49.7	63.6	6.8	0.2	24.0	33.6	9.3	28.5
DMAda [49]	R	75.5	29.1	48.6	21.3	14.3	34.3	36.8	29.9	49.4	13.8	0.4	43.3	50.2	69.4	18.4	0.0	27.6	34.9	11.9	32.1
GCMa [75]	R	81.7	46.9	58.8	22.0	20.0	41.2	40.5	<b>41.6</b>	64.8	31.0	32.1	<b>53.5</b>	47.5	<b>75.5</b>	39.2	0.0	49.6	30.7	21.0	42.0
MGCA [50]	R	80.3	49.3	66.2	7.8	11.0	41.4	38.9	39.0	64.1	18.0	55.8	52.1	53.5	74.7	<b>66.0</b>	0.0	37.5	29.1	22.7	42.5
DANNet [51]	R	90.0	54.0	<b>74.8</b>	<b>41.0</b>	<b>21.1</b>	25.0	26.8	<b>30.2</b>	<b>72.0</b>	26.2	<b>84.0</b>	47.0	33.9	68.2	19.0	<b>0.3</b>	<b>66.4</b>	38.3	23.6	44.3
CDAda [81]	R	90.5	60.6	67.9	<b>37.0</b>	19.3	42.9	36.4	35.3	66.9	24.4	79.8	45.4	42.9	70.8	51.7	0.0	29.7	27.7	26.2	45.0
Source Only	D	79.0	21.8	53.0	13.3	11.2	22.5	20.2	22.1	43.5	10.4	18.0	37.4	33.8	64.1	6.4	0.0	52.3	30.4	7.4	28.8
AdaptSeg [20]	D	86.1	44.2	55.1	22.2	4.8	21.1	5.6	16.7	37.2	8.4	1.2	35.9	26.7	68.2	45.1	0.0	50.1	33.9	15.6	30.4
AdvEnt [21]	D	85.8	37.9	55.5	27.7	14.5	23.1	14.0	21.1	32.1	8.7	2.0	39.9	16.6	64.0	13.8	0.0	58.8	28.5	20.7	29.7
BDL [38]	D	85.3	41.1	61.9	32.7	17.4	20.6	11.4	21.3	29.4	8.9	1.1	37.4	22.1	63.2	28.2	0.0	47.7	<b>39.4</b>	15.7	30.8
DANNet [51]	D	88.6	53.4	69.8	34.0	20.0	25.0	31.5	35.9	69.5	<b>32.2</b>	82.3	44.2	43.7	54.1	22.0	0.1	40.9	36.0	24.1	42.5
<b>SePiCo (ProtoCL)</b>	D	87.3	50.9	64.5	25.6	12.1	38.3	40.8	37.5	61.0	21.9	77.6	37.4	47.0	67.8	54.5	0.0	33.7	27.0	23.7	42.6
<b>SePiCo (BankCL)</b>	D	88.5	54.8	66.5	25.1	13.5	40.0	39.6	40.8	62.5	25.1	79.0	37.8	<b>54.8</b>	70.4	63.7	0.0	36.8	15.6	23.4	44.1
<b>SePiCo (DistCL)</b>	D	<b>91.2</b>	<b>61.3</b>	67.0	28.5	15.5	<b>44.7</b>	<b>44.3</b>	41.3	65.4	22.5	80.4	41.3	52.4	71.2	39.3	0.0	39.6	27.5	<b>28.8</b>	<b>45.4</b>

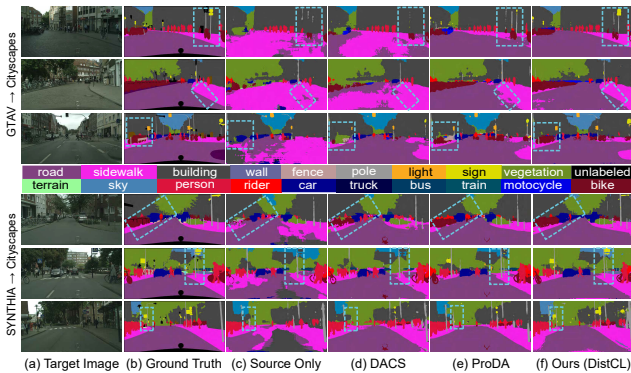


Fig. 4: Qualitative results on Cityscapes (val). From left to right: target image, ground truth, the maps predicted by Source Only, DACS, ProDA and Ours (DistCL) are shown one by one. Our method shows a clear visual improvement.

in prior methods. Take CDAda [81] as an example, it consists of inter-domain style transfer and intra-domain gradual self-training. While our SePiCo aims at ensuring pixel-wise

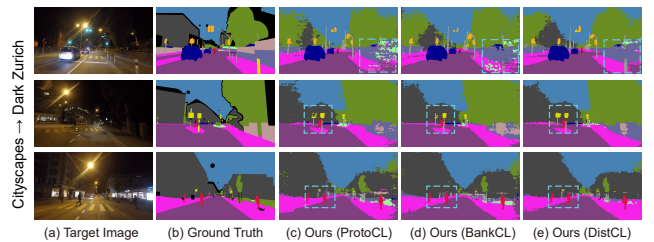


Fig. 5: Qualitative results on Dark Zurich (val). From left to right: target image, Ground Truth, the maps predicted by Ours (ProtoCL), Ours (BankCL) and Ours (DistCL).

representation consistency between daytime and nighttime images, it is complementary to models designed for the nighttime and can still be trained in one stage. It is worth noting that our methods based on DeepLab-V2 are even superior or comparable to CDAda based on RefineNet [80], which further demonstrates the efficacy of our method.

**Qualitative results.** In Fig. 4, we first visualize the segmentation results on two synthetic-to-real scenarios, pre-

TABLE 4: Comparison results using Swin-B ViT [82] and SegF. MiT-B5 [48]. The best result is highlighted in **bold**.

(a) GTAV → Cityscapes																				
Method	road	side.	buil.	wall	fence	pole	light	sign	veg.	sky	pers.	rider	car	truck	bus	train	mbike	bike	mIoU	
Swin-B ViT (88M)	63.3	28.6	68.3	16.8	23.4	37.8	51.0	34.3	83.8	42.1	85.7	68.5	25.4	83.5	36.3	17.7	2.9	36.1	42.3	44.6
TransDA-B [83]	94.7	64.2	89.2	48.1	45.8	50.1	60.2	40.8	<b>90.4</b>	50.2	<b>93.7</b>	<b>76.7</b>	<b>47.6</b>	92.5	56.8	60.1	47.6	49.6	55.4	63.9
SegF. MiT-B5 (84.7M)	77.1	15.2	83.8	30.8	32.0	27.9	41.5	18.5	86.5	42.5	86.8	62.6	22.2	87.0	42.7	36.8	6.1	33.5	12.5	44.5
DAFormer [28]	95.7	70.2	89.4	53.5	48.1	49.6	55.8	59.4	89.9	47.9	92.5	72.2	44.7	92.3	74.5	78.2	<b>65.1</b>	55.9	61.8	68.3
<b>SePiCo (ProtoCL)</b>	96.1	72.9	<b>89.7</b>	54.4	48.8	53.5	60.4	<b>65.3</b>	90.0	48.4	91.6	75.2	47.1	<b>93.3</b>	74.4	74.6	41.2	58.8	<b>65.9</b>	68.5
<b>SePiCo (BankCL)</b>	96.3	73.6	89.6	53.7	47.8	<b>53.8</b>	<b>60.8</b>	60.0	89.9	48.8	91.5	74.6	45.1	93.1	74.8	73.8	51.5	60.3	65.3	68.7
<b>SePiCo (DistCL)</b>	<b>96.9</b>	<b>76.7</b>	<b>89.7</b>	<b>55.5</b>	<b>49.5</b>	53.2	60.0	64.5	<b>90.2</b>	<b>50.3</b>	90.8	74.5	44.2	<b>93.3</b>	<b>77.0</b>	<b>79.5</b>	63.6	<b>61.0</b>	65.3	<b>70.3</b>
(b) SYNTHIA → Cityscapes																				
Method	road	side.	buil.	wall*	fence*	pole*	light	sign	veg.	sky	pers.	rider	car	bus	mbike	bike	mIoU	mIoU*		
Swin-B ViT (88M)	57.3	33.8	56.0	6.3	0.2	33.8	35.5	18.9	79.9	74.8	63.1	10.9	78.3	39.0	20.8	19.4	39.2	45.2		
TransDA-B [83]	<b>90.4</b>	<b>54.8</b>	86.4	31.1	1.7	<b>53.8</b>	<b>61.1</b>	37.1	<b>90.3</b>	<b>93.0</b>	71.2	25.3	92.3	66.0	44.4	49.8	59.3	66.3		
SegF. MiT-B5 (84.7M)	69.9	27.8	82.9	21.6	2.3	39.2	36.3	29.9	84.2	84.9	61.6	22.6	83.8	48.0	14.9	19.7	45.6	51.3		
DAFormer [28]	84.5	40.7	88.4	<b>41.5</b>	6.5	50.0	55.0	54.6	86.0	89.8	73.2	48.2	87.2	53.2	53.9	61.7	60.9	67.4		
<b>SePiCo (ProtoCL)</b>	85.9	45.5	<b>88.9</b>	38.2	2.5	52.3	57.7	58.2	89.3	88.4	74.0	50.5	92.3	70.6	<b>56.2</b>	56.7	62.9	70.3		
<b>SePiCo (BankCL)</b>	88.2	49.3	88.6	36.1	4.7	53.1	58.9	<b>58.4</b>	88.5	84.8	72.4	49.3	<b>92.8</b>	76.3	55.5	55.2	63.3	70.6		
<b>SePiCo (DistCL)</b>	87.0	52.6	88.5	40.6	<b>10.6</b>	49.8	57.0	55.4	86.8	86.2	<b>75.4</b>	<b>52.7</b>	92.4	<b>78.9</b>	53.0	<b>62.6</b>	<b>64.3</b>	<b>71.4</b>		
(c) Cityscapes → Dark Zurich																				
Method	road	side.	buil.	wall	fence	pole	light	sign	veg.	terr.	sky	pers.	rider	car	truck	bus	train	mbike	bike	mIoU
SegF. MiT-B5 <sup>‡</sup> (84.7M)	80.3	37.1	57.5	28.1	7.9	35.5	33.2	29.3	41.7	14.8	4.7	48.9	48.0	66.6	5.7	<b>7.9</b>	63.3	31.4	23.3	35.0
DAFormer <sup>‡</sup> [28]	92.0	63.0	67.2	28.9	13.1	44.0	42.0	42.3	70.7	28.2	83.6	51.1	39.1	76.4	31.7	0.0	78.3	43.9	26.5	48.5
<b>SePiCo (ProtoCL)</b>	90.1	57.7	<b>75.0</b>	<b>34.9</b>	16.4	53.5	47.0	47.8	70.1	31.7	84.1	57.3	<b>53.3</b>	80.5	42.4	2.3	83.6	42.6	<b>30.1</b>	52.7
<b>SePiCo (BankCL)</b>	91.1	61.2	73.4	31.9	<b>18.0</b>	51.6	48.6	47.7	72.8	<b>33.0</b>	85.5	57.0	51.1	80.6	48.4	3.1	<b>84.6</b>	<b>45.3</b>	28.2	53.3
<b>SePiCo (DistCL)</b>	<b>93.2</b>	<b>68.1</b>	73.7	32.8	16.3	<b>54.6</b>	<b>49.5</b>	<b>48.1</b>	<b>74.2</b>	31.0	<b>86.3</b>	<b>57.9</b>	50.9	<b>82.4</b>	<b>52.2</b>	1.3	83.8	43.9	29.8	<b>54.2</b>

<sup>‡</sup> Implement according to source code.

dicted by our SePiCo (DistCL), and compare our results to those predicted by the Source Only, DACS and ProDA models. The results predicted by SePiCo (DistCL) are smoother and contain fewer spurious areas than those predicted by other models, showing that the performance has been largely improved. Next, as the daytime-to-nighttime task is far more challenging than the previous two, we further show several qualitative segmentation results in Fig. 5 to illustrate the advantage of SePiCo (DistCL) over the other two variants SePiCo (ProtoCL) and SePiCo (BankCL).

#### 4.2.2 More Experimental Results

**Advanced network architecture.** Vision Transformer-based DA methods have been actively studied not long ago [28], [83]. Hoyer et. al [28] analyze different architectures for adaptation and propose a new architecture, DAFormer, based on a Transformer encoder [48] and a context-aware fusion decoder. Lately, Chen et al. [83] introduce a momentum network and dynamic of discrepancy measurement to smooth the learning dynamics for target data. Therefore, we further adopt one of architectures such as DAFormer [28], to support our claims. Inspired by a multi-level context-aware feature fusion decoder, we also fuse all stacked multi-level features from the decoder to provide valuable concepts for contrastive learning. From TABLE 4, we have the following observations: (i) Approaches based on Transformer perform generally better than those based on DeepLab-V2, confirming the strength of these advanced architectures; (ii) Our SePiCo is still competitive on the new architecture. All variants of SePiCo achieve an extraordinary improvement of around +20% mIoU on each task when compared with the models trained merely on source data, i.e., Swin-B ViT [82] and SegF. MiT-B5 [48]; (iii) SePiCo (DistCL) improves the state-of-the-art DAFormer by +2.0% mIoU for GTAV → Cityscapes, +3.4% mIoU for SYNTHIA → Cityscapes, and +5.7% mIoU for Cityscapes → Dark Zurich.

**Generalization to unseen domains.** In TABLE 3 and TABLE 4(c), we have benchmarked our method on the Dark Zurich test. To showcase the better generalization of

TABLE 5: Comparison results of Cityscapes → Dark Zurich trained models for generalization on two unseen target domains: Nighttime Driving and BDD100k-night test sets.

Method	Dark Zurich	Nighttime Driving	BDD100k-night	Cityscapes
DMAda (RefineNet) [49]	32.1	36.1	28.3	-
GCMA (RefineNet) [75]	42.0	45.6	33.2	-
MGCDA (RefineNet) [50]	42.5	49.4	34.9	-
CDAda (RefineNet) [81]	45.0	50.9	33.8	-
SegF. MiT-B5 [48]	35.0	46.9	34.0	76.8
DAFormer [28]	48.5	51.8	33.9	76.4
<b>SePiCo (ProtoCL)</b>	52.7	55.5	37.5	<b>79.5</b>
<b>SePiCo (BankCL)</b>	53.3	54.9	39.1	78.9
<b>SePiCo (DistCL)</b>	<b>54.2</b>	<b>56.9</b>	<b>40.6</b>	78.9

TABLE 6: Experiments over weather DA object detection: Cityscapes → Foggy Cityscapes based on Faster R-CNN.

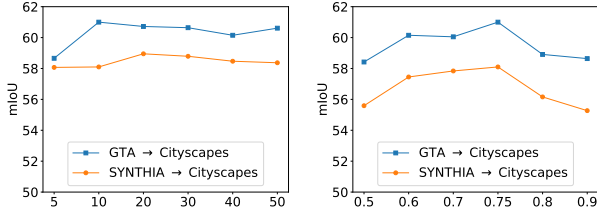
Method	person	rider	car	truck	bus	train	mcycle	bicycle	mAP <sub>0.5</sub> <sup>r</sup>
Faster R-CNN [84]	17.8	23.6	27.1	11.9	23.8	9.1	14.4	22.8	18.8
DA-Faster [85]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
SW-Faster [86]	36.2	35.3	43.5	30.0	29.9	42.3	32.6	24.5	34.3
PDA [87]	36.0	45.5	54.4	24.3	44.1	25.8	29.1	35.9	36.9
EveryPixelMatters [88]	41.9	38.7	56.7	22.6	41.5	26.8	24.6	35.5	36.0
<b>SePiCo (ProtoCL)</b>	41.9	40.4	58.2	24.9	40.3	32.6	25.5	35.0	37.4
<b>SePiCo (BankCL)</b>	43.8	42.8	57.9	22.5	43.0	26.4	27.0	38.9	37.8
<b>SePiCo (DistCL)</b>	43.9	41.2	57.5	25.1	42.8	26.1	29.1	39.1	<b>38.1</b>
Faster RCNN (oracle)	47.4	40.8	66.8	27.2	48.2	32.4	31.2	38.3	41.5

SePiCo, the trained Dark Zurich models are also tested on two unseen target domains, i.e., Nighttime Driving [49] and BDD100k-night [89]. From TABLE 5, we can find that the generalization ability of previous self-training methods is limited, and they often fail to transfer well to unseen domains or concepts. On the contrary, our SePiCo markedly improves over the sophisticated baselines. Notably, SePiCo (DistCL) achieves mIoUs of 56.9% and 40.6%, respectively, releasing the newest records on both. Another interesting finding is that our SePiCo indeed boosts the segmentation performance on the source domain even compared with SegF. MiT-B5 (Source Only) model that is only trained on source images. There is some evidence that a well-structured pixel embedding space provides the best of both worlds: reducing distribution shift, plus promoting the source task.

**Adaptation for object detection.** We further extend our SePiCo to a weather adaptive object detection task, i.e., Cityscapes [74] → Foggy Cityscapes [90]. More specifically, Foggy Cityscapes is a synthetic foggy dataset that applies

TABLE 7: Ablation study on GTAV  $\rightarrow$  Cityscapes. All models are trained end-to-end in a total of 40k iterations.

Method	$\mathcal{L}_{ssl}$	$\mathcal{L}_{cl}$	$\mathcal{L}_{reg}$	CBC	mIoU	$\Delta$
w/o self-training		✓			38.6 $\pm$ 0.5	-
		✓			48.5 $\pm$ 0.8	9.9
		✓	✓		49.2 $\pm$ 0.7	10.6
		✓	✓	✓	49.8 $\pm$ 0.4	11.2
SePiCo (DistCL)	✓				52.1 $\pm$ 2.0	-
	✓	✓			59.3 $\pm$ 1.7	7.2
	✓	✓	✓		60.4 $\pm$ 1.3	8.3
	✓	✓	✓	✓	61.0 $\pm$ 0.7	8.9



(a) Study on  $N_{crop}$ . (b) Study on  $cat\_max\_ratio$ .  
Fig. 6: Parameter sensitivity analysis for CBC.

simulated fog to scenes of Cityscapes. Build upon [88], we employ Faster R-CNN [84] with VGG-16 [91] as the backbone. The model is trained with learning rate of  $5 \times 10^{-3}$ , momentum of 0.9 and weight decay of  $5 \times 10^{-4}$ . The image’s shorter side is set to 800 and RoIAlign is employed for feature extraction. From TABLE 6, we observe that SePiCo achieves comparable results with other task-specific and well-optimized detection algorithms [86], [87], [88].

### 4.3 Ablation Studies

We evaluate the contribution of each component present in our one-stage framework. Specifically, we testify SePiCo on the task of GTAV  $\rightarrow$  Cityscapes, and the results are reported in TABLE 7, TABLE 8, TABLE 9 and Fig. 6. As can be seen, each of these components contributes to the ultimate success. Eventually, we achieve 49.8% and 61.0% mIoU under “w/o self-training” and “SePiCo (DistCL)” respectively, outperforming the corresponding baselines by +11.2% and +8.9%.

**Effect of semantic-guided pixel contrast.** As discussed in Section 3.3, centroid-aware and distribution-aware pixel contrast can build up stronger intra-/inter-category connections and minimize the domain divergence efficiently. We validate the performance increments by separately training models with and without self-training. As shown in Table 7, contrastive learning alone can improve the segmentation performance, but the effect is not as noticeable as using self-training (48.5% mIoU vs. 52.1% mIoU). When they are adopted properly in a unified pipeline, the full potential of the model is released, further promoting gains of +7.2% mIoU. The results imply the effect and necessity of representation learning for the classical self-training paradigm.

**Effect of  $\mathcal{L}_{reg}$ .** We study the advantages of diversity-promoting regularization term  $\mathcal{L}_{reg}$  in TABLE 7. It is clearly shown that using  $\mathcal{L}_{reg}$  also brings an extra increase (+0.7% mIoU and +1.1% mIoU, respectively), verifying the effectiveness of smoothing the learned representations.

	road	side	buil.	wall	fence	pole	light	sign	veg.	terr.	sky	pers.	rider	car	truck	bus	train	bike	mIoU	
SePiCo (DistCL)	98.2	97.8	98.7	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4
SePiCo (DistCL) w/o RCS	98.4	98.7	98.3	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4
SePiCo (DistCL) w/o CBC	98.2	98.3	98.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4	91.4

Fig. 7: Comparison results of per-class IoU for CBC and RCS.

TABLE 8: Effect of the teacher network and the bank size on GTAV  $\rightarrow$  Cityscapes. The default choice is colored in gray.

(a) Effect of the teacher network.

	w/ teacher				w/o teacher
$\beta$	0.99	0.999	0.9995	0.9999	0.0
mIoU	60.8	61.0	60.6	60.7	56.1

(b) Effect of the bank size  $B$ .

	SePiCo (BankCL)				SePiCo (DistCL)
$B$	50	100	200	500	$\infty$
mIoU	59.4	59.5	59.8	59.7	61.0

**Effect of class-balanced cropping (CBC).** We first remove the class-balanced cropping discussed in Section 3.4.1 to verify its necessity. As shown in TABLE 7, as expected, the mIoU of the adapted model decreases moderately without CBC, supporting the importance of class imbalance cropping. Furthermore, we test CBC from parameter sensitivity and compare it with other strategies. From Fig. 6, we can observe that increasing cropping times ( $N_{crop}$ ) helps us to select more class-balanced regions and bring slight gains. However, to increase the training efficiency, we just make 10 croppings for all experiments. As for  $cat\_max\_ratio$ , a high threshold (e.g.,  $> 0.9$ ) will result in too many futile candidates and, conversely, a low threshold (e.g.,  $< 0.5$ ) will result in no candidate (In this case, the first crop will be selected). Both cases invalidate CBC, which leads to random cropping. Thus,  $cat\_max\_ratio$  is default set to 0.75.

Furthermore, Fig. 7 shows the results of class-wise IoU of ablating two class-balanced strategies (RCS and CBC), respectively. As seen, RCS mainly improves the performance of minority classes (e.g., large gains on “rider”, “mbike” and “bike”). While our CBC mainly aims to balance the IoUs among all categories, in which some categories such as “side”, “wall”, and “fence” even show slight decreases in IoU while others show tremendous increases in IoU. In summary, the RCS is a direct yet effective class-balanced strategy since supervised annotations are utilized. In contrast, the proposed CBC serves as an alternative choice in the absence of annotations but is also able to balance the IoUs among all categories and improve the overall performance.

**Effect of the teacher network.** The teacher-student architecture is frequently adopted to introduce a strong regularization during training [69]. Essentially, a larger momentum value  $\beta$  indicates a stronger effect from the teacher net. We adjust  $\beta$  to change the amount of regularization and report the results in TABLE 8a. A performance gain of more than +3.0% mIoU is brought about by the teacher net, confirming its efficacy. Thus  $\beta$  is fixed to 0.999 for proper regularization.

**Effect of the bank size  $B$ .** TABLE 8b lists the effect of bank size for SePiCo (BankCL). As we enlarge the memory bank from 50 to 500, a gradual gain can be witnessed in performance, with a slight drop when  $B=500$ . Generally, a

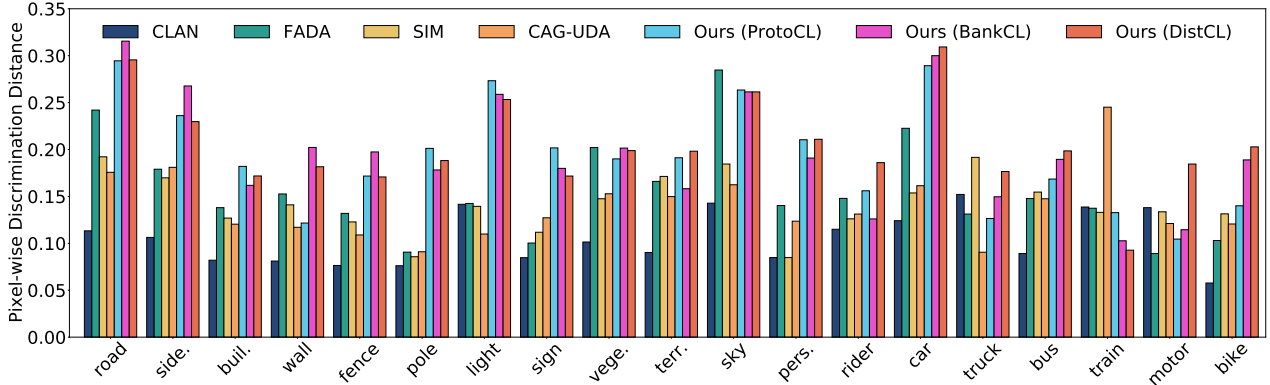


Fig. 8: Quantitative analysis of the discrimination of features. For each class, we show the values of pixel-wise discrimination distance (PDD) as defined in Eq. (18) on Cityscapes validation set. These comparison results are from 1) category adversarial learning methods, i.e, CLAN and FADA; 2) category centroid-based alignment methods i.e., SIM and CAG-UDA; 3) pixel contrast methods, i.e., Ours (ProtoCL/BankCL/DistCL), respectively. A high PDD suggests the pixel-wise representations of same category are clustered densely while the distance between different categories is relatively large.

TABLE 9: Ablation of feature selection in SePiCo variants for GTAV  $\rightarrow$  Cityscapes based on DeepLab-V2.

Method	layer 1	layer 2	layer 3	layer 4	mIoU
SePiCo (ProtoCL)	✓				58.7
		✓			58.6
			✓		58.6
				✓	58.8
		{1,2,3,4}-fusion			58.4
SePiCo (BankCL)	✓				58.5
		✓			58.4
			✓		58.0
				✓	59.8
		{1,2,3,4}-fusion			58.6
SePiCo (DistCL)	✓				60.7
		✓			59.3
			✓		58.8
				✓	61.0
		{1,2,3,4}-fusion			60.2

larger bank size means more diversity in semantic concepts, leading to better performance. However, a huge bank will result in prolonged retention of outdated representations, which may exert a negative effect on pixel-level guidance. In comparison, SePiCo (DistCL) overcomes this issue by using the distribution to simulate infinite bank size on-the-fly, thus exceeding SePiCo (BankCL) by a considerable margin.

**Effect of multi-level features.** We provide the performance of applying SePiCo to intermediate layers. In particular, there are four residual blocks in the original ResNet-101 backbone [76]. The four layers (denoted by layer 1 - layer 4) are taken from the output of each residual block and {1,2,3,4}-fusion means that features from all four layers are concatenated together. Interestingly, features from layer 1 also exhibit distinctive information. This is expected because earlier features provide valuable low-level concepts for semantic segmentation at a high resolution. However, if we fuse all the features for adaptation, the results are slightly degraded, which is different from the Transformer-based architecture. We conjecture that the ViTs have more similarity between the representations obtained in shallow and deep layers compared to CNNs. Overall, the features from layer 4 prove to be the best choice for all three variants of SePiCo. It can be seen that SePiCo (DistCL) performs nearly equally well while adopting features from the last

layer or the fusion of multiple layers, indicating that the distribution indeed increases the diversity of features and is more robust.

## 4.4 Further Analysis

### 4.4.1 Pixel-wise Discrimination Distance

To verify whether our adaptation framework can yield a discriminative embedding space, we design a metric to take a closer look at what degree the pixel-wise representations are aligned. In the literature, CLAN [23] defines a Cluster Center Distance as the ratio of the intra-category distance between the initial model and the aligned model and FADA [26] proposes a new Class Center Distance to consider inter-category distance. To better evaluate the effectiveness of pixel-wise representation alignment, we introduce a new Pixel-wise Discrimination Distance (PDD) by taking intra- and inter-category affinities of pixel representations into account. Formally, a PDD value for category  $k$  is given by:

$$PDD(k) = \frac{1}{|\Lambda^k|} \sum_{x \in \Lambda^k} \frac{sim(x, \mu^k)}{\sum_{i=1, i \neq k}^K sim(x, \mu^i)}, \quad (18)$$

where  $sim(\cdot, \cdot)$  is the similarity metric, and we adopt cosine similarity.  $\Lambda^k$  denotes the pixel set that contains all the pixel representations belonging to the  $k^{th}$  semantic class.

With PDD, we could investigate the relative magnitude of inter-category and intra-category pixel feature distances. Specifically, we calculate the PDD on the whole Cityscapes validate set and compare PDD values with other state-of-the-art category alignment methods: CLAN [23] and FADA [26] for category-level adversarial training and SIM [25] and CAG-UDA [27] for category centroid based counterparts that do not tackle the distance between different category features. As shown in Fig. 8, we observe that: (i) CLAN and FADA could not cope well with the distance between different category features, thus obtaining lower PDD values. (ii) Both SIM and CAG-UDA adopt category anchors computed on the source domain to guide the alignment but they do not regularize the distance among different category features. Thus, the PDD values of some categories such as “road”, “wall”, “light” and “car” are even lower than those of adversarial training methods while the

$\lambda_{cl}$	0.01	0.1	0.5	1.0	2.0	$\lambda_{reg}$	0.01	0.1	0.5	1.0	2.0	$L_w$	0	1500	3000	5000	10000
G $\rightarrow$ C	58.9	59.9	60.9	<b>61.0</b>	59.3	G $\rightarrow$ C	59.5	59.3	59.3	<b>61.0</b>	58.7	G $\rightarrow$ C	59.2	59.5	<b>61.0</b>	59.7	58.4
S $\rightarrow$ C	57.3	57.4	57.8	<b>58.1</b>	57.9	S $\rightarrow$ C	57.8	57.9	<b>58.2</b>	58.1	57.6	S $\rightarrow$ C	57.2	58.0	<b>58.1</b>	57.8	57.3

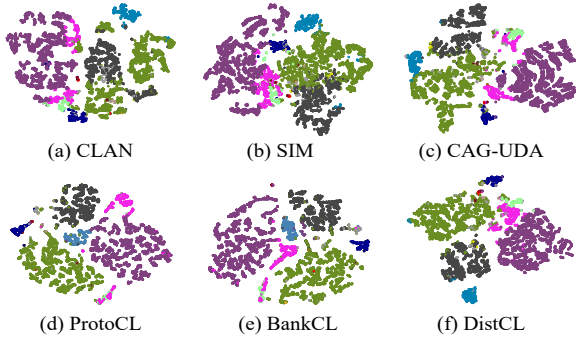
(a) Study on  $\lambda_{cl}$ .(b) Study on  $\lambda_{reg}$ .(c) Study on  $L_w$ .TABLE 10: Parameter sensitivity on GTAV  $\rightarrow$  Cityscapes (G  $\rightarrow$  C) and SYNTHIA  $\rightarrow$  Cityscapes (S  $\rightarrow$  C) tasks.

Fig. 9: t-SNE analysis of existing comparable alignment methods and our SePiCo. As seen, the proposed pixel contrast objectives (ProtoCL/BankCL/DistCL) beget well-structured embedding spaces. Please zoom in for details.

PDD values of some categories are sometimes higher. (iii) Considering cross-domain pixel contrast, our SePiCo (ProtoCL/BankCL/DistCL) can achieve much higher PDD values in most categories. Based on these quantitative results, together with the t-SNE analysis in Fig. 9, it is clear that our SePiCo can achieve better pixel-wise category alignment and largely improve the pixel-wise accuracy of predictions.

#### 4.4.2 t-SNE Visualization

To better develop intuition, we draw t-SNE visualizations [92] of learned representations for three competitive category alignment methods (CLAN [23], SIM [25], CAG-UDA [27]) and compare them with all variants of our SePiCo (ProtoCL, BankCL, DistCL) in Fig. 9. With this in mind, we first randomly select an image from target domain and then map its high-dimensional latent feature representations to a 2D space. From the t-SNE visualizations, we can observe that (i) Existing category alignment methods could produce separated features, but it may be hard for dense prediction since the margins between different category features are not obvious and the distribution is still dispersed; (ii) When we apply pixel contrast, features among different categories are better separated, demonstrating that the semantic distributions can provide correct supervision signal for target data; (iii) More importantly, the representations of SePiCo (DistCL) exhibit clear clusters, revealing the discriminative capability of the distribution-aware contrastive adaptation.

#### 4.4.3 Parameter Sensitivity

We conduct parameter sensitivity analysis to evaluate the sensitivity of SePiCo (DistCL) on two synthetic-to-real benchmarks. As shown in TABLE 10a, 10b and 10c, we select loss weights  $\lambda_{cl}$  and  $\lambda_{reg} \in \{0.01, 0.1, 0.5, 1.0, 2.0\}$ , the iteration at which to start contrastive learning  $L_w \in \{0, 1500, 3000, 5000, 10000\}$ , respectively. While altering  $\lambda_{cl}$  and  $\lambda_{reg}$  in a large range, we find that both losses are slightly sensitive to their assigned weight on GTAV

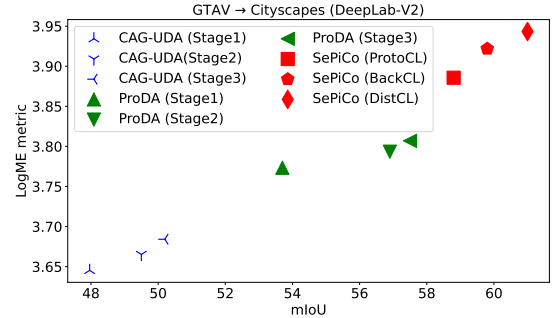


Fig. 10: Comparison results of model transferability for different methods trained on GTAV  $\rightarrow$  Cityscapes.

$\rightarrow$  Cityscapes, and probably their relative weight due to their resemblance. Nevertheless, our method keeps outperforming the previous SOTA in different compositions of loss weights. We also explore the sensitivity of our method on the iteration to start contrastive learning and observe that SePiCo (DistCL) is relatively robust to  $L_w$ , peaking at  $L_w = 3000$ . The result could be attributed to better category information learned through warm-up iterations.

#### 4.4.4 Quality of Model Generalization

To quantify the generalization of our SePiCo, we adopt a transferability metric (i.e., LogME [93]) to accurately assess the transferability of the model trained on GTAV  $\rightarrow$  Cityscapes to the target dataset. Specifically, LogME calculates the maximum value of label evidence given extracted features by the adapted models and can measure the quality of models. A model with a higher LogME value is likely to have good transfer performance. In the trial, we consider each pixel and its ground-truth label as a separate observation. Since using all observations of target data to calculate is too computationally expensive, we instead calculate the LogME metric within a single target image and then average them. Fig. 10 shows the comparison results at different stages of CAG-UDA [27] (Stage1, Stage2, and Stage3) and ProDA [36] (Stage1, Stage2, and Stage3) and our one-stage pipeline SePiCo (ProtoCL, BankCL, and DistCL). This case study confirms the strong generalization of our pixel contrast paradigm, which essentially learns a well-structured pixel embedding space by making full use of the prototype, bank, or distribution-aware semantic similarities from the source domain.

#### 4.4.5 Throughput

We first compare inputs of different baseline methods. Source Only [8], [48], for instance, contains a student model and takes source ( $I_s$ ) images as input. DACS [30], an advanced self-training method, introduces the teacher-student model. The student model takes both source ( $I_s$ ) and target ( $I_t$ ) images as input, and the target model takes target

TABLE 11: Throughput measured for different networks (DeepLab-V2 [8] and SegF. Mit-B5 [48]) on GTAV  $\rightarrow$  Cityscapes. Results are obtained using V100-32G and throughput is measured using a batch size of 1.

Method	mIoU	Student model	Teacher model	ImageNet pretrained model	Training time per iteration (s)
Source Only [8]	38.6	$I_s$	-	-	0.32 (1.00 $\times$ )
DACS [30]	52.1	$I_s + I_t$	$I_t$	-	1.16 (3.63 $\times$ )
<b>SePiCo (DistCL)</b>	<b>61.0</b>	$I_s + I_t$	$I_s + I_t$	-	<b>1.34 (4.18<math>\times</math>)</b>
Source Only [48]	44.5	$I_s$	-	-	0.35 (1.00 $\times$ )
DAFormer [28]	68.3	$I_s + I_t$	$I_t$	$I_s$	1.33 (3.80 $\times$ )
<b>SePiCo (DistCL)</b>	<b>70.3</b>	$I_s + I_t$	$I_s + I_t$	-	<b>1.45 (4.14<math>\times</math>)</b>

images as input to generate target pseudo labels. As for DAFormer [28], the state-of-the-art method based on a Transformer backbone, it also contains a teacher model and a student model. The training process is similar to DACS, except that DAFormer introduces an auxiliary ImageNet pre-trained model and takes source ( $I_s$ ) images as input to distill knowledge from expressive thing features of ImageNet. In this work, we also contain a teacher model and a student model and introduce a very lightweight projection head into the network that generates a new pixel embedding space. And both teacher and student models take source ( $I_s$ ) and target ( $I_t$ ) images as input, as shown in Fig. 3.

Then, we compute the throughput of the mentioned methods on the GTAV  $\rightarrow$  Cityscapes task using different networks. Since the throughput of the test phase is the same with the same network, we compare the training time of one iteration in TABLE 11. Compared to existing methods, we can observe that SePiCo only introduces slight extra computation on either CNN-based or Transformer-based networks but significantly surpasses comparison methods.

## 5 CONCLUSION

In this paper, we present SePiCo, a novel end-to-end adaptation framework tailored for semantic segmentation, which successfully enhances the potential of the self-training paradigm in conjunction with representation learning. Our main contribution is the discovery of pixel contrast guided by different semantic concepts. Eventually, we propose a particular form of contrastive loss at the pixel level, which implicitly involves the joint learning of an infinite number of similar/dissimilar pixel pairs for each pixel representation of both domains. Additionally, we derive an upper bound on this formulation and transfer the originally intractable loss function into practical implementation. Though simple yet effective, it works surprisingly well. Extensive experiments demonstrate the superiority of SePiCo on both daytime and nighttime segmentation benchmarks.

## ACKNOWLEDGMENTS

This paper was supported by National Key R&D Program of China (No. 2021YFB3301503), and also supported by the National Natural Science Foundation of China under Grant No. U21A20519.

## REFERENCES

- [1] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Found. Trends Comput. Graph. Vis.*, vol. 12, no. 1-3, pp. 1-308, 2020.
- [2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *CVPR*, 2012, pp. 3354-3361.
- [3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234-241.
- [4] M. H. Hesamian, W. Jia, X. He, and P. J. Kennedy, "Deep learning techniques for medical image segmentation: Achievements and challenges," *J. Digit. Imaging*, vol. 32, no. 4, pp. 582-596, 2019.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012, pp. 1097-1105.
- [6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431-3440.
- [7] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.
- [8] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834-848, 2018.
- [9] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *ECCV*, 2016, pp. 102-118.
- [10] G. Ros, L. Sellart, J. Materzynska, D. Vázquez, and A. M. López, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *CVPR*, 2016, pp. 3234-3243.
- [11] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.
- [12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345-1359, 2010.
- [13] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *CVPR*, 2017, pp. 7167-7176.
- [14] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3071-3085, 2019.
- [15] S. Li, B. Xie, Q. Lin, C. H. Liu, G. Huang, and G. Wang, "Generalized domain conditioned adaptation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4093-4109, 2022.
- [16] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096-2030, 2016.
- [17] A. Dundar, M. Liu, Z. Yu, T. Wang, J. Zedlewski, and J. Kautz, "Domain stylization: A fast covariance matching framework towards domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2360-2372, 2021.
- [18] Y. Yang and S. Soatto, "FDA: fourier domain adaptation for semantic segmentation," in *CVPR*, 2020, pp. 4084-4094.
- [19] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *ICML*, 2018, pp. 1989-1998.
- [20] Y.-H. Tsai, W.-C. Hung, S. Schuster, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *CVPR*, 2018, pp. 7472-7481.
- [21] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *CVPR*, 2019, pp. 2517-2526.
- [22] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon, "Unsupervised intra-domain adaptation for semantic segmentation through self-supervision," in *CVPR*, 2020, pp. 3764-3773.
- [23] Y. Luo, P. Liu, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Category-level adversarial adaptation for semantic segmentation using purified features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 3940-3956, 2022.
- [24] S. Li, M. Xie, K. Gong, C. H. Liu, Y. Wang, and W. Li, "Transferable semantic augmentation for domain adaptation," in *CVPR*, 2021, pp. 11516-11525.
- [25] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W.-m. Hwu, T. S. Huang, and H. Shi, "Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation," in *CVPR*, 2020, pp. 12635-12644.
- [26] H. Wang, T. Shen, W. Zhang, L. Duan, and T. Mei, "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation," in *ECCV*, 2020, pp. 642-659.

- [27] Q. Zhang, J. Zhang, W. Liu, and D. Tao, "Category anchor-guided unsupervised domain adaptation for semantic segmentation," in *NeurIPS*, 2019, pp. 433–443.
- [28] L. Hoyer, D. Dai, and L. Van Gool, "Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation," in *CVPR*, 2022, pp. 9924–9935.
- [29] J. Dong, Y. Cong, G. Sun, Z. Fang, and Z. Ding, "Where and how to transfer: Knowledge aggregation-induced transferability perception for unsupervised domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021.
- [30] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson, "DACs: domain adaptation via cross-domain mixed sampling," in *WACV*, 2021, pp. 1378–1388.
- [31] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 1106–1120, 2021.
- [32] C. Corbiere, N. Thome, A. Saporta, T.-H. Vu, M. Cord, and P. Perez, "Confidence estimation via auxiliary models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6043–6055, 2022.
- [33] Y. Zou, Z. Yu, X. Liu, B. V. K. V. Kumar, and J. Wang, "Confidence regularized self-training," in *ICCV*, 2019, pp. 5982–5991.
- [34] N. Araslanov and S. Roth, "Self-supervised augmentation consistency for adapting semantic segmentation," in *CVPR*, 2021, pp. 15384–15394.
- [35] H. Ma, X. Lin, Z. Wu, and Y. Yu, "Coarse-to-fine domain adaptive semantic segmentation with photometric alignment and category-center regularization," in *CVPR*, 2021, pp. 4051–4060.
- [36] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *CVPR*, 2021, pp. 12414–12424.
- [37] Y. Wang, J. Peng, and Z. Zhang, "Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation," in *ICCV*, 2021, pp. 9092–9101.
- [38] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *CVPR*, 2019, pp. 6936–6945.
- [39] T. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "DADA: depth-aware domain adaptation in semantic segmentation," in *ICCV*, 2019, pp. 7363–7372.
- [40] Q. Wang, D. Dai, L. Hoyer, L. Van Gool, and O. Fink, "Domain adaptive semantic segmentation with self-supervised depth estimation," in *ICCV*, 2021, pp. 8515–8525.
- [41] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9726–9735.
- [42] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020, pp. 1597–1607.
- [43] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, "Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning," in *CVPR*, 2021, pp. 16684–16693.
- [44] Q. Cai, Y. Wang, Y. Pan, T. Yao, and T. Mei, "Joint contrastive learning with infinite possibilities," in *NeurIPS*, 2020.
- [45] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 6230–6239.
- [46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.
- [47] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.
- [48] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *NeurIPS*, 2021.
- [49] D. Dai and L. Van Gool, "Dark model adaptation: Semantic image segmentation from daytime to nighttime," in *ITSC*, 2018, pp. 3819–3824.
- [50] C. Sakaridis, D. Dai, and L. V. Gool, "Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3139–3153, 2022.
- [51] X. Wu, Z. Wu, H. Guo, L. Ju, and S. Wang, "Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation," in *CVPR*, 2021, pp. 15769–15778.
- [52] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *NeurIPS*, 2006, pp. 137–144.
- [53] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *ICML*, 2019, pp. 7404–7413.
- [54] B. Xie, L. Yuan, S. Li, C. H. Liu, X. Cheng, and G. Wang, "Active learning for domain adaptation: An energy-based approach," in *AAAI*, 2022, pp. 8708–8716.
- [55] C. Chu, R. Dabre, and S. Kurohashi, "An empirical comparison of domain adaptation methods for neural machine translation," in *ACL*, 2017, pp. 385–391.
- [56] F. Lv, J. Liang, K. Gong, S. Li, C. H. Liu, H. Li, D. Liu, and G. Wang, "Pareto domain adaptation," in *NeurIPS*, 2021.
- [57] Y. Cheng, F. Wei, J. Bao, D. Chen, F. Wen, and W. Zhang, "Dual path learning for domain adaptation of semantic segmentation," in *ICCV*, 2021, pp. 9082–9091.
- [58] L. Melas-Kyriazi and A. K. Manrai, "Pixmatch: Unsupervised domain adaptation via pixelwise consistency training," in *CVPR*, 2021, pp. 12435–12445.
- [59] M. Kim and H. Byun, "Learning texture invariant representation for domain adaptation of semantic segmentation," in *CVPR*, 2020, pp. 12975–12984.
- [60] J. Huang, D. Guan, A. Xiao, S. Lu, and L. Shao, "Category contrast for unsupervised domain adaptation in visual tasks," in *CVPR*, 2022, pp. 1203–1214.
- [61] K. Mei, C. Zhu, J. Zou, and S. Zhang, "Instance adaptive self-training for unsupervised domain adaptation," in *ECCV*, 2020, pp. 415–430.
- [62] G. Kang, Y. Wei, Y. Yang, Y. Zhuang, and A. G. Hauptmann, "Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation," in *NeurIPS*, 2020.
- [63] Y. Zou, Z. Yu, B. V. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *ECCV*, 2018, pp. 289–305.
- [64] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, 2006, pp. 1735–1742.
- [65] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv:1807.03748*, 2018.
- [66] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense contrastive learning for self-supervised visual pre-training," in *CVPR*, 2021, pp. 3024–3033.
- [67] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *ICCV*, 2021, pp. 7303–7313.
- [68] I. Alonso, A. Sabater, D. Ferstl, L. Montesano, and A. C. Murillo, "Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank," in *ICCV*, 2021, pp. 8219–8228.
- [69] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017, pp. 1195–1204.
- [70] V. Olsson, W. Tranheden, J. Pinto, and L. Svensson, "Classmix: Segmentation-based data augmentation for semi-supervised learning," in *WACV*, 2021, pp. 1368–1377.
- [71] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *ICML*, 2020, pp. 6028–6039.
- [72] Y. Wang, G. Huang, S. Song, X. Pan, Y. Xia, and C. Wu, "Regularizing deep networks with semantic data augmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3733–3748, 2022.
- [73] X. Guo, C. Yang, B. Li, and Y. Yuan, "Metaarxiv:action: Domain-aware meta loss arxiv:action for unsupervised domain adaptation in semantic segmentation," in *CVPR*, 2021, pp. 3927–3936.
- [74] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016, pp. 3213–3223.
- [75] C. Sakaridis, D. Dai, and L. V. Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *ICCV*, 2019, pp. 7373–7382.
- [76] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
- [77] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.
- [78] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch:



An imperative style, high-performance deep learning library," in *NeurIPS*, 2019, pp. 8024–8035.

- [79] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.
- [80] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *CVPR*, 2017, pp. 1925–1934.
- [81] Q. Xu, Y. Ma, J. Wu, C. Long, and X. Huang, "Cdada: A curriculum domain adaptation for nighttime semantic segmentation," in *ICCVW*, 2021, pp. 2962–2971.
- [82] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021, pp. 9992–10 002.
- [83] R. Chen, Y. Rong, S. Guo, J. Han, F. Sun, T. Xu, and W. Huang, "Smoothing matters: Momentum transformer for domain adaptive semantic segmentation," *arXiv:2203.07988*, 2022.
- [84] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [85] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *CVPR*, 2018, pp. 3339–3348.
- [86] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Strong-weak distribution alignment for adaptive object detection," in *CVPR*, 2019, pp. 6956–6965.
- [87] H. Hsu, C. Yao, Y. Tsai, W. Hung, H. Tseng, M. K. Singh, and M. Yang, "Progressive domain adaptation for object detection," in *WACV*, 2020, pp. 738–746.
- [88] C. Hsu, Y. Tsai, Y. Lin, and M. Yang, "Every pixel matters: Center-aware feature alignment for domain adaptive object detector," in *ECCV*, 2020, pp. 733–748.
- [89] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *CVPR*, 2020, pp. 2633–2642.
- [90] C. Sakaridis, D. Dai, S. Hecker, and L. V. Gool, "Model adaptation with synthetic and real data for semantic dense foggy scene understanding," in *ECCV*, 2018, pp. 707–724.
- [91] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
- [92] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.
- [93] K. You, Y. Liu, J. Wang, and M. Long, "LogME: Practical assessment of pre-trained models for transfer learning," in *ICML*, 2021, pp. 12 133–12 143.



**Binhui Xie** is a Ph.D. student at the School of Computer Science and Technology, Beijing Institute of Technology. His research interests focus on computer vision and transfer learning.



**Shuang Li** received the Ph.D. degree in control science and engineering from the Department of Automation, Tsinghua University, Beijing, China, in 2018.

He was a Visiting Research Scholar with the Department of Computer Science, Cornell University, Ithaca, NY, USA, from November 2015 to June 2016. He is currently an Associate Professor with the school of Computer Science and Technology, Beijing Institute of Technology, Beijing. His main research interests include machine learning and deep learning, especially in transfer learning and domain adaptation.

chine learning and deep learning, especially in transfer learning and domain adaptation.



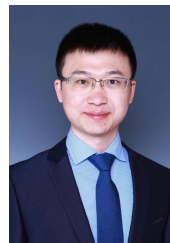
**Mingjia Li** is an undergraduate student at the School of Computer Science and Technology, Beijing Institute of Technology. His research interests focus on computer vision and transfer learning.



**Chi Harold Liu** (SM'15) receives a Ph.D. degree in Electronic Engineering from Imperial College, UK in 2010, and a B.Eng. degree in Electronic and Information Engineering from Tsinghua University, China in 2006.

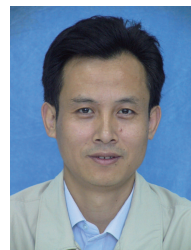
He is currently a Full Professor and Vice Dean at the School of Computer Science and Technology, Beijing Institute of Technology, China. Before moving to academia, he worked for IBM Research - China as a staff researcher and project manager from 2010 to 2013, worked as

a postdoctoral researcher at Deutsche Telekom Laboratories, Germany in 2010, and as a Research Staff Member at IBM T. J. Watson Research Center, USA in 2009. His current research interests include the big data analytics, mobile computing, and machine learning. He received the IBM First Plateau Invention Achievement Award in 2012, ACM SigKDD'21 Best Paper Runner-up Award, and IEEE DataCom'16 Best Paper Award. He has published more than 100 prestigious conference and journal papers and owned 26 EU/UK/US/Germany/Spain/China patents. He serves as the Associate Editor for IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, Area Editor for KSII Trans. on Internet and Information Systems, the Symposium Chair for IEEE ICC 2020 on Next Generation Networking, and served as the (Lead) Guest Editor for IEEE Transactions on Emerging Topics in Computing and IEEE Sensors Journal. He was the book editor for 11 books published by Taylor & Francis Group, USA and China Machine Press, China. He also has served as the general chair of IEEE SECON'13 workshop on IoT Networking and Control, IEEE WCNC'12 workshop on IoT Enabling Technologies, and ACM UbiComp'11 Workshop on Networking and Object Memories for IoT. He was a consultant to Asian Development Bank, Bain & Company, and KPMG, USA, and the peer reviewer for Qatar National Research Foundation, National Science Foundation, China, Ministry of Education and Ministry of Science and Technology, China. He is a senior member of IEEE and a Fellow of IET, British Computer Society, and Royal Society of Arts.



**Gao Huang** is an Associate Professor in the Department of Automation, Tsinghua University. He was a Postdoctoral Researcher in the Department of Computer Science at Cornell University. He received the PhD degree in Control Science and Engineering from Tsinghua University in 2015, and B.Eng degree in Automation from Beihang University in 2009. He was a visiting student at Washington University at St. Louis and Nanyang Technological University in 2013 and 2014, respectively. His research interests

include machine learning and computer vision.



**Guoren Wang** received the BSc, MSc, and PhD degrees from the Department of Computer Science, Northeastern University, China, in 1988, 1991 and 1996, respectively. Currently, he is a Professor and the Dean with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. His research interests include XML data management, query processing and optimization, bioinformatics, high dimensional indexing, parallel database systems, and cloud data management. He has published

more than 100 research papers.