Reducing Semantic Ambiguity In Domain Adaptive Semantic Segmentation Via Probabilistic Prototypical Pixel Contrast

Xiaoke Hao[a], Shiyu Liu[a], Chuanbo Feng[a], Ye Zhu[a]

[a]*School of Artificial Intelligence, Hebei University of Technology, No. 5340, Xiping Road, Tianjin, 300401, China*

Abstract

Domain adaptation aims to reduce the model degradation on the target domain caused by the domain shift between the source and target domains. Although encouraging performance has been achieved by combining contrastive learning with the self-training paradigm, they suffer from ambiguous scenarios caused by scale, illumination, or overlapping when deploying deterministic embedding. To address these issues, we propose probabilistic prototypical pixel contrast (PPPC), a universal adaptation framework that models each pixel embedding as a probability via multivariate Gaussian distribution to fully exploit the uncertainty within them, eventually improving the representation quality of the model. In addition, we derive prototypes from probability estimation posterior probability estimation which helps to push the decision boundary away from the ambiguity points. Moreover, we employ an efficient method to compute similarity between distributions, eliminating the need for sampling and reparameterization, thereby significantly reducing computational overhead. Further, we dynamically select the ambiguous crops at the image level to enlarge the number of boundary points involved in contrastive learning, which benefits the establishment of precise distributions for each category. Extensive experimentation demonstrates that PPPC not only helps to address ambiguity at the pixel level, yielding discriminative representations but also achieves significant improvements in both synthetic-to-real and day-to-night adaptation tasks. It surpasses the previous state-of-the-art (SOTA) by +5.2% mIoU in the most challenging daytime-to-nighttime adaptation scenario, exhibiting stronger generalization on other unseen datasets. The code and models are available at https://github.com/DarlingInTheSV/Probabilistic-Prototypical-Pixel-Contrast.

Keywords: Semantic segmentation, probabilistic embedding, domain adaptation, contrastive learning.

1. Introduction

Semantic segmentation is a dense prediction task at the pixel level, with the objective of assigning each pixel a corresponding label. It is a crucial step for enabling autonomous driving [1] and assisting medical analysis [2]. Though neural networks have achieved impressive performance on segmentation tasks, they require large-scale annotated datasets to stabilize the training process and prevent overfitting or inaccuracies in data statistics. It is particularly laborious to annotate each pixel for high-resolution images, for example, it takes 1.5 hours to annotate a single image of Cityscapes [3], and for adverse weather conditions, it is even 3.3 hours. Therefore, adopting synthetic data rendered from game engines is a promising option. However, the model trained on synthetic images often experiences performance degradation on real data due to its sensitivity to domain shift. Additionally, the architecture that achieves higher performance in the supervised way does not necessarily mitigate this influence. For instance, DeepLabV3+ [4] outperforms DeepLabV2 [5] in several supervised segmentation tasks, but it is more susceptible to the impact of domain gaps [6].

To address this issue, unsupervised domain adaptation (UDA) methods have been proposed, which facilitate the adaptation of models trained on labeled source domains to unlabeled target domains. A lot of works delve into diminishing the domain shift in input [7, 8], feature [9, 10], or output [11] space. A line of work employs adversarial learning, enforcing the encoder to generate domain-invariant features. However, alignment in a holistic way often leads to class-mismatching (e.g., style transfer and domain mix-up). Another line of work focuses on the self-training paradigm. They improve the self-training framework by refining the pseudo-label [12], using consistency regulation between different views of the same input [13], selecting reference pairs in the target domain [14, 15] or extracting fine-grain features [16]. Though these methods achieve remarkable improvement, rather than directly addressing the domain discrepancy, they implicitly mitigate the domain gap by separately optimizing performance in the source and target domain. Hence, several works combine adversarial training with self-training to help embedding space less vulnerable to domain discrepancy [17, 18]. Ad-

ditionally, auxiliary tasks like self-supervised depth estimation [19, 20] and pixel-level contrast [21] are also integrated into self-training.

Contrastive learning serves as a self-supervised strategy, enabling neural networks to learn to extract general features by optimizing label-agnostic tasks. The core of incorporating contrastive learning into UDA is to construct a coherent pixel embedding space across two domains. Therefore, most of the previous works deploy contrastive learning to explore the cross-domain pixel contrast, which attracts positive pairs and repels negative pairs.

However, there are two main challenges for employing contrastive learning in UDA. One is the difficulty in obtaining a certain decision boundary. Due to a lack of supervision in the target domain, we utilize pseudo-label as the class label for pixel embedding. However, as shown in Fig. 1(a), some pixel embeddings have been incorrectly assigned labels, causing the decision boundary to move in the wrong direction. This bias becomes more serious with the training proceeding and eventually leads to a decrease in the model's capability to discriminate similar classes in the target domain. Meanwhile, it is infeasible to perform pixel-wise contrast if we want to reduce the computational overhead. To address these problems, prototype contrast is proposed. A prototype renders the overall appearance of a category by averaging the embeddings, as shown in Fig. 1(b). Although the computation is significantly reduced, it ignores the variety of pixel embedding of the category, thus still leading to part of pixel embedding incorrectly divided by decision boundary. The other challenge is that, whether using pixel-wise or prototype-based methods, deterministic embeddings fail to address the classification of ambiguous pixel embeddings. They can only determine classes based on the maximum value of the Softmax prediction, and in the worst case, each class has a similar prediction probability, much like a random guess. Therefore, they can't provide an effective gradient for the loss function, thus leading the network into local optima.

To address the above issues, we propose PPPC to deal with ambiguous embeddings in a more effective way by explicitly modeling uncertainty for each pixel embedding. Our idea is straightforward: for pixel embeddings with low uncertainty, we include them in the contrastive process. However, for pixel embeddings with high uncertainty that surpass the current model's discriminative capability, we maintain their original positions, rather than randomly labeling them, leading to an incorrect optimization direction, as illustrated in Fig. 1(a) and Fig. 1(b). Therefore, in the worst-case scenario, the decision boundary will remain unchanged rather than moving in

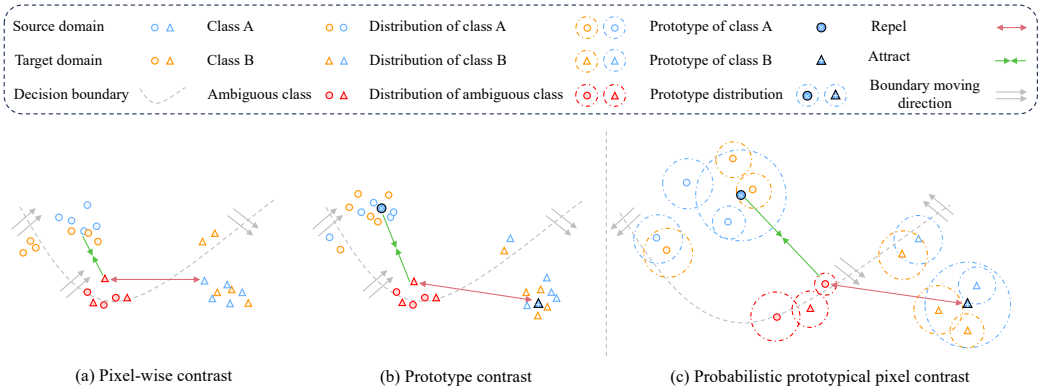

Figure 1: Illustration of the existing issues in self-training contrastive learning. (a) The decision boundary crosses the part of target pixel embeddings, leading to incorrect pseudo-label predictions. (b) The use of prototype contrast ignores the diversity of pixel embeddings, resulting in the decision boundary being unable to distinguish a few embeddings from both the source and target domains. (c) Our probabilistic prototypical pixel contrast not only better adjusts the decision boundaries but also addresses the issue of ambiguous classes that the previous two methods did not involve.

the wrong direction. Specifically, as shown in Fig. 1(c), in contrast to deterministic embeddings, we represent each pixel embedding as a multivariate Gaussian distribution, thus setting a dynamic uncertain range for them according to the model's current discriminative capability. The prototype for each category is calculated based on the posterior distribution of all observations rather than being a simple average. Its position tends to align with high-confidence pixel embedding and the uncertainty range is affected by embeddings with a higher covariance. Therefore, our composed prototype can reduce the loss of diversity through a reasonable uncertainty range while ensuring an accurate position. By employing the expected likelihood kernel (ELK) as the measure of similarity between pixel and prototype in contrastive learning, we successfully direct the decision boundary towards the correct direction. Moreover, boundary points contain a lot of unexplored fuzzy information, enlarging the number of these points would be advantageous for better handling ambiguous classes. Hence, we propose ambiguity-guided cropping (AGC), which dynamically selects boundary points at the image level in a global manner based on prototype covariance. These prototypes serve as an effective approximation of the model's current discriminative capability.

The main contributions of this paper are summarized as follows:

- We propose a universal probabilistic adaptation framework, called PPPC, which can fully exploit the uncertainty information of each pixel embedding from the feature level to the data level. Consequently, it can significantly enhance segmentation performance in ambiguous scenarios.

- To facilitate efficiency, we carefully design each module and loss function. Specifically, we demonstrate that prototypes calculated from posterior probabilities can better account for the uncertainty of pixel embeddings compared to directly averaging. Additionally, we find that kernel functions are more stable and efficient than other divergence-based methods as a similarity metric under UDA settings. Finally, we introduce more boundary points into contrastive learning by approximating image norm through prototype variance, avoiding the manual setting of thresholds. As a result, PPPC significantly improves performance while only slightly increasing computation overhead and GPU memory footprint.

- Extensive experiments on three typical UDA tasks show that PPPC achieves superior performance on three wisely adopted UDA benchmarks. Particularly, we obtain mIoUs of 63.7%, 58.7%, and 50.6% on benchmarks GTAV $\rightarrow$ Cityscapes, SYNTHIA $\rightarrow$ Cityscapes and Cityscapes $\rightarrow$ Dark Zurich. Especially on the most challenging day-to-night task, we improve the previous SOTA performance by +5.2% mIoU.

2. Related work

2.1. Unsupervised Domain Adaptation Segmentation

Numerous approaches have been developed to address domain shifts in semantic segmentation, with the majority falling into two main categories: adversarial training and self-training. In adversarial training methods, typically two networks are employed: one network generates images, features, or segmentation maps, which may belong to either the source or target domain, while another network functions as a discriminator. The discriminator takes the generated output from the generator network as input and attempts to

predict the domain of the input. Hoffman et al. [22] first apply the adversarial training to UDA semantic segmentation. Other works [23, 18] use style transfer inputs, transferring the image style of the target domain onto the source domain images, thus bridging the domain gap by increasing domain confusion. Saito et al. [24] propose to utilize task-specific classifiers to align distributions, avoiding generating target features near class boundaries. In [11], to handle the problem that predictions on target images are less certain, resulting in noisy, high entropy output, the entropy of the pixel-wise predictions is incorporated in the adversarial loss.

In self-training, pseudo-labels are generated for the target domain using confidence thresholds [25, 26] or pseudo-label prototypes [27, 12]. To avoid training instabilities, several methods have been proposed. For example, Zhou et al. [28] employ a novel knowledge distillation strategy to benefit the mutual learning of teacher and student networks. Tranheden et al. [29] propose cross-domain mixed sampling, using mixed images from two domains along with their labels. In [30], they employ a distribution alignment technique to enforce the consistency between the marginal distribution of clusters and pseudo labels to overcome the class imbalance problem that exists in pseudo labels. Hoyer et al. [31] utilize a multi-resolution input fusion strategy that combines the strengths of high-resolution and low-resolution crops which provide fine details and long-range dependencies. Hoyer et al. [32] use a masking strategy to enforce consistency between predictions of masked target images. Most self-training studies encode images to deterministic embeddings and perform alignment or pseudo-label refinement, which is often problematic as they can't deal with inputs near the decision boundary that are ambiguous in both image and feature levels. We overcome this problem by fully exploring the potential of probabilistic embedding in self-training. Specifically, we not only leverage the uncertainty provided by probabilistic embedding to aid in addressing the alignment of boundary points in contrastive learning but also apply it to the dynamic selection of input crops.

2.2. Probabilistic embedding

The purpose of probabilistic embedding (PE) is to represent levels of specificity or uncertainty that traditional embeddings struggle to do. Vilnis et al. [33] propose directly working in probability distribution rather than applying Bayes's rule to infer the latent distribution from observed data. Kingma et al. [34] first propose to use MLP to model the mean and variance

of the probability distribution. Oh et al. [35] propose to design the embedding function to be stochastic and map the input to a certain probability distribution (e.g., Gaussian) instead of a single point. Li et al. [36] further extend the Gaussian to r-radius von Mises Fisher (vMF), and Kirchhof et al. [37] use a non-isotropic vMF. Shi et al. [38] apply PE to face recognition, which not only improves the accuracy but shows a potential use in a risk-controlled system with estimated uncertainty. Recent literatures have delved deeper into PE, where Chun et al. [39] apply it to cross-modal retrieval using an attention-based architecture to encode mean and variance, Neculai et al. [40] apply it to multimodal image retrieval with a probabilistic composition rule, and Xie et al. [41] apply it in semi-supervised segmentation to deal with inaccurate pseudo-label. However, these methods are typically applied in sparse tasks like classification and retrieval, which encode the whole input into a single vector. Furthermore, they employ a complex Monte Carlo sampling strategy to compute the similarity of two distributions, which also requires additional reparametrization tricks to enable backpropagation during training. Instead, semantic segmentation is a dense prediction task, we treat each pixel embedding in the feature map as a probability distribution. We further demonstrate by experiment that measuring distribution similarity directly through mean and variance is more efficient than Monte Carlo sampling.

2.3. Contrastive learning

Contrastive learning, first proposed in [42], involves the network minimizing the contrastive loss in the latent space. This is accomplished by pulling positive pairs closer while pushing negative pairs away, ultimately enhancing the network's discriminative ability. The pairs used for contrastive learning can be images [43], pixels [44], and prototypes [12]. For instance, Zhang et al. [12] contrast the pixel representation with prototypes which are updated in a moving average way to learn a more compact embedding space. Jiang et al. [45] further explore the inter-class alignment by adopting the class-centered distribution alignment for adaptation. Li et al. [18] bridge the domain gap at both pixel and feature levels via contrastive learning. Huang et al. [44] increase the quantity of prototypes by establishing a domain-mixed memory bank that stores class-wise prototypes from both the source domain and target domain. Xie et al. [21] pushe the number of contrast instances to infinite by approximating the true distribution of each semantic category in the source domain. However, they model distribution implicitly by using statis-

tics from the source domain, we explicitly calculate the distribution for each instance involved in contrastive learning, including pixels and prototypes in both domains. Modeling the distribution of representations for each pixel in the feature map introduces only a little overhead, yet retains their original semantic information and preserves prototype diversity, which is hard to achieve even using the statistical distribution in deterministic embeddings. Xie et al. [41] also employ the same idea in semi-supervised segmentation. However, it requires complex sampling strategies for valid samples, anchors, and negative samples. In contrast, our approach contrasts pixel embeddings only with prototypes, significantly reducing the computation.

3. Methodology

3.1. Problem definition and overall framework

Given the labeled source domain data $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ and unlabeled target domain data $\mathcal{D}_t = \{(x_i^t)\}_{i=1}^{N_t}$, the objective of unsupervised domain adaptation of semantic segmentation is to train a model on $\mathcal{D}_s$ and $\mathcal{D}_t$ that can classify each pixel in a target domain image into one of K predefined classes. We adopt the self-training framework as our basic architecture, as shown in Fig. 2. The self-training segmentation model has two networks sharing the same structure but different weight update strategies. The student networks have an encoder $f_\theta(\cdot)$, a segmentation head $g_c(\cdot)$, and an auxiliary projection head $g_p(\cdot)$. Similarly, the teacher networks are denoted as $f'_\theta(\cdot)$, $g'_c(\cdot)$ and $g'_p(\cdot)$.

First, the student networks $g(f(\cdot))$ are trained on the labeled source domain in a supervised manner by minimizing the categorical cross-entropy between the model's prediction $P_{i,c}^s$ and the ground truth label $Y_{i,c}^s$. We formulate this problem as:

$$\mathcal{L}_s = -\frac{1}{HW} \sum_{n=1}^{N_s} \sum_{i=1}^{H \times W} \sum_{c=1}^{K} Y_{n,i,c}^s \log P_{n,i,c}^s, \tag{1}$$

where N_s is the total number of images in the source domain, H and W denote the height and the width of a source image, K is the number of classes, n is the index of image, i is the pixel index of image, c is the class index. $Y_{n,i,c}^s \in \{0, 1\}$ is the one-hot representation of ground truth label for pixel i in image n about class c. $P_{i,c}^s \in \mathbb{R}$ is the predicted probability for pixel i about class c, which is obtained by up-sampling the output of student

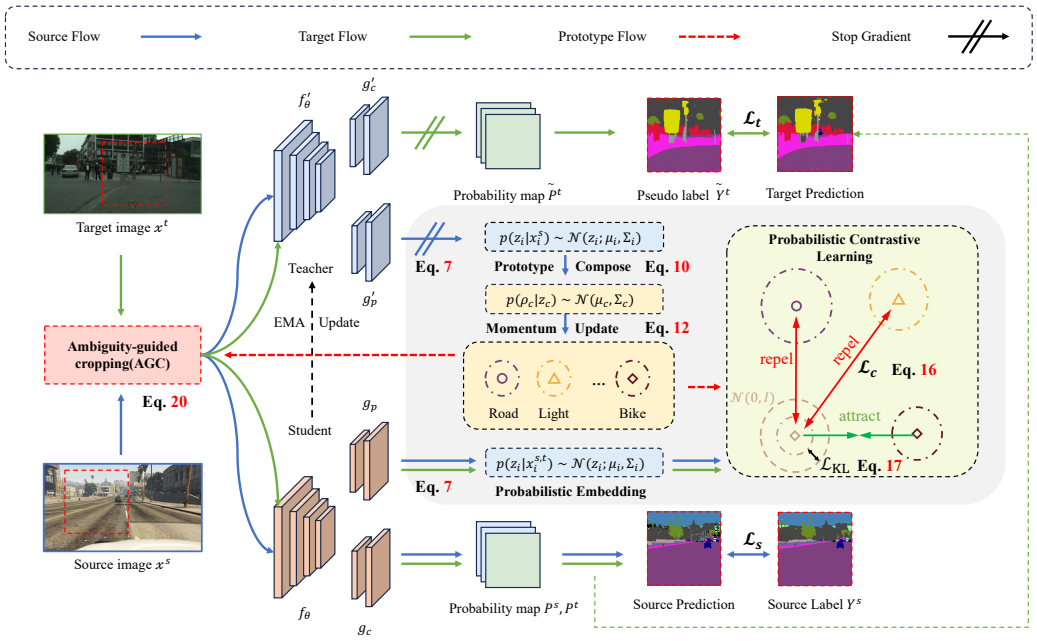

Figure 2: Overview of our framework. The model is trained with both the supervised segmentation loss $\mathcal{L}_s$ and the unsupervised adaptation loss $\mathcal{L}_t$. Specifically, the predictions for the source and target domain P_s, P_t are provided by the student model, and the pseudo-labels $\tilde{Y}^t$ are generated by the teacher model. Furthermore, the distribution for each pixel embedding $p(z_i|x_i^{s,t})$ is modeled by the projection head of the student model. For prototypes, we first obtain pixel-wise probabilistic embeddings $p(z_i|x_i^s)$from the source image x^s via the projection head g'_p within the teacher model. Then, we estimate each composed prototype $p(\rho_c|z_c)$ with these embeddings. Lastly, to stabilize the updating of prototypes we deploy the momentum update strategy. In the embedding space, apart from conducting distributional contrast, we also utilize an additional regularization term $\mathcal{L}_{KL}$ to prevent the covariance collapse to zero. Additionally, AGC uses the ambiguity provided by the online-updated prototypes to select ambiguous scenarios dynamically.

networks $g_c(f(x_n^s))$. To better utilize the information in the unlabeled target domain, we generate reliable pseudo label $\tilde{Y}_j^t$ by teacher networks for a given target image:

$$\tilde{Y}_j^t = \underset{c}{\text{argmax}} \tilde{P}_{j,c}^t. \tag{2}$$

Without supervised information, the generated pseudo label is prone to be noisy. Thus, we empirically set a threshold α to filter out low-confidence pixels:

$$M(x_j^t) = \mathbb{1}_{[max_c \tilde{P}_{j,c}^t > \alpha]}, j \in 1, 2, \cdots, H \times W. \tag{3}$$

$M(\cdot)$ is sample mask over the pixels of a target image, $\mathbb{1}_{[\cdot]}$ is the indicator function. Only the pixels whose predicted probability exceeds the threshold will participate in the re-training on the target domain:

$$\mathcal{L}_t = -\frac{1}{HW}\sum_{n=1}^{N_t}\sum_{i=1}^{H\times W}\sum_{c=1}^{K} M_{n,i}^t \tilde{Y}_{n,i,c}^t \log\, P_{n,i,c}^t. \tag{4}$$

The parameters of student networks are optimized via gradient descent, while the teacher network parameters are updated by the Exponential Moving Average (EMA) of the student network parameters. The contrastive loss is computed based on the output of the projection head $q = g_p(f(x^{s,t}))$:

$$\mathcal{L}_c = \mathbb{E}_{q,k^+,k^-}[-\log \frac{e^{s(q,k^+)/\tau}}{e^{s(q,k^+)/\tau} + \sum_{n=1}^{N} e^{s(q,k^-)/\tau}}], \tag{5}$$

where k^+ and k^- are positive and negative pairs with respect to query q, they can be either pixel representation or prototype of class. $s(\cdot,\cdot)$ denotes the similarity measurement between two representations, τ denotes the temperature. Intuitively, the goal of contrastive learning is to minimize the distance when pixels belong to the same category while maximizing the distance otherwise. By leveraging contrastive learning as an auxiliary task, the representations produced by the backbone encoder $f_\theta(\cdot)$, used for downstream semantic segmentation task, are refined to have clearer decision boundaries among each class. The overall loss for UDA semantic segmentation is formulated as:

$$\mathcal{L} = \mathcal{L}_s + \lambda_t \mathcal{L}_t + \lambda_c \mathcal{L}_c, \tag{6}$$

where λ_t and λ_c are hyper-parameters controlling the strength of corresponding loss.

As previously mentioned, conventional deterministic embedding is susceptible to the influence of unreliable pseudo labels and ambiguous classes, particularly during the initial stages of training. This can result in the model displaying a bias towards the majority and easily distinguishable classes, while allocating less attention to the minority and ambiguous classes, such as train, bus, bike, and motorbike. Once this bias is established, the model cannot rectify it solely through the supervised signals from the source domain, as there exists a gap between the source and target domain. If this trend persists, the decision boundary between each class will become indistinct, resulting in embedding points being situated closer to the boundary

rather than being maximally separated. Consequently, to overcome this issue, we promote to use of probabilistic embedding which has more statistical information compared to deterministic embedding. Moreover, this probabilistic mechanism could be viewed as utilizing distributional uncertainty to expand the embedding space for hard, confusing classes, which is hard to achieve by using deterministic embedding.

3.2. Probabilistic embedding

The prevailing contrastive learning method focuses on deterministic embeddings, whose similarities are easily measured by dot production. However, external factors like illumination, overlapping, and object size may confuse the model thus could lead to ambiguous embeddings. Although we can't eliminate these factors, we can explicitly model each pixel as a distribution in the embedding space, which allows the model to express uncertainty when the input is ambiguous. We parametrize the embedding of pixels as a multivariate normal distribution with mean vectors and covariance matrices in $\mathbb{R}^d$:

$$p(z_i|x_i) = \mathcal{N}(z_i; \mu_i, \Sigma_i), \tag{7}$$

where μ_i denotes the mean of the embedding pixel i, Σ_i denotes the covariance, we only consider a diagonal covariance matrix to reduce the complexity (i.e., we model the covariance matrix of pixel embeddings and prototypes as corresponding diagonal scalars). μ_i and Σ_i are predicted by two fully connected layers in projection head $g_p(f(x_i))$ respectively, the details are elaborated in the experiment section 4.1.

3.3. Composed probabilistic prototype

Let $Z_c = \{z_1, z_2, \cdots, z_k\}, k \in n_c$ be the set of k embeddings belonging to the same class c. The objective of aggregating all pixel embeddings that share the same class label is to determine a probability distribution $p(\rho_c|Z_c)$that maximally unifies all the individual distributions $p(\rho_c|z_c) \sim \mathcal{N}(\mu_c, \Sigma_c)$. To be noticed, we do not directly estimate prototypes from pixel embeddings as done in [21], but instead, we utilize the posterior probabilities constructed from the likelihood function, similar to [40]. Since each pixel is modeled as a distribution, to maintain inter-class diversity as much as possible, we cannot simply average those observations. Additionally, due to the insufficient discriminative capability of the model, we must also account for their uncertainty. Therefore, we compute the posterior probabilities of n observations,

and the mean of the prototypes will be constrained by the covariances of individual observations. A simple way to compose this prototype after n_c observed embeddings is making a product of n_c Gaussian probability density functions (PDFs). Formally, the multivariate Gaussian PDF of pixel embedding Eq. (7) can be written as:

$$p(z) = \exp[\zeta + \eta^T z - \frac{1}{2} z^T \Lambda z], \tag{8}$$

where

$$\Lambda = \Sigma^{-1}, \eta = \Sigma^{-1}\mu \text{ and } \zeta = -\frac{1}{2}(d\log 2\pi - log|\Lambda| + \eta^T \Lambda^{-1} \eta),$$

where d is the dimensionality of z, μ is the d-dimensional mean vector, and Σ is the d-by-d dimensional covariance matrix. The posterior distribution of prototype can be written as:

$$\begin{aligned} p(\rho_c) &= \prod_{i=1}^{n_c} p(z_i) \\ &= \exp[\zeta_{i=1\ldots n_c} + (\sum_{i=1}^{n_c} \eta_i)^T z - \frac{1}{2} z^T (\sum_{i=1}^{n_c} \Lambda_i) z] \\ &= \exp(\zeta_{i=1\ldots n_c} - \zeta_{n_c}) \exp[\zeta_{n_c} + \eta_{n_c}^T x - \frac{1}{2} z^T \Lambda_{n_c} z], \end{aligned} \tag{9}$$

where

$$\Lambda_{n_c} = \sum_{i=1}^{n_c} \Lambda_i \text{ and } \eta_{n_c} = \sum_{i=1}^{n_c} \eta_i.$$

Comparing Eq. (8) to Eq. (9), it can be seen that the new composite prototype distribution is a constant scaled Gaussian with a mean vector and covariance matrix given by:

$$\mu_\rho = \frac{\sum_{i=1}^{n_c} \Sigma_i^{-1} \mu_i}{\Sigma_\rho^{-1}} \text{ and } \Sigma_\rho^{-1} = \sum_{i=1}^{n_c} \Sigma_i^{-1}. \tag{10}$$

To make sure our composite prototypes are representative and reliable, we use the teacher network to extract the mean and covariance for each pixel embedding involved in the computation of Eq. (10):

$$\begin{aligned} \mu_i &= g'_p(f'(x_i^s)) \\ \Sigma_i &= g'_p(f'(x_i^s)). \end{aligned} \tag{11}$$

To be noticed, we only calculate prototypes on the source domain which has label information rather than on the target domain which causes disturbance during self-training. Unlike some existing methods using a memory bank to store the prototypes at each training stage, we conjecture that probabilistic embedding is capable enough to deal with unstable changes of prototypes in embedding space during the initial training stage with a high learning rate. We only adopt an EMA update strategy to limit the prototypes to be similar in a few iterations in case of random error.

$$\begin{aligned} \mu' &\leftarrow \beta\mu' + (1-\beta)\mu \\ \Sigma' &\leftarrow \beta\Sigma' + (1-\beta)\Sigma, \end{aligned} \tag{12}$$

where μ' and Σ' are mean and covariance from the current training loop, μ and Σ are from the previous iteration.

3.4. Distribution to distribution distance

Since pixel embeddings and prototypes are not modeled as points but as distributions, conventional Euclidean distance cannot measure the similarity between two distributions. To address this issue, we leverage a kernel function to measure the similarity between two distributions, which combines a weighted l_2 distance with a logarithmic regularization term, taking into account the corresponding reliability. Probability Product Kernels (PPK) are a family of metrics to compare two distributions by the product of their PDFs. Given two d-dimensional Gaussian distribution $p \sim \mathcal{N}(\mu_1, \Sigma)$ and $p \sim \mathcal{N}(\mu_2, \Sigma')$, the PPK can be denoted as:

$$PPK(p,q) = \int p(z)^\rho q(z)^\rho dz. \tag{13}$$

Combining with Eq. (8), PPK for two distributions can be written as:

$$\begin{aligned} PPK(p,q) = (2\pi)^{(1-2\rho)D/2}\rho^{-D/2}\left|\Sigma^\dagger\right|^{1/2}|\Sigma|^{-\rho/2}|\Sigma'|^{-\rho/2} \\ \exp\left(-\frac{\rho}{2}\left(\mu^T\Sigma^{-1}\mu + \mu'^{-1} - \mu^{\dagger T}\Sigma^\dagger\mu^\dagger\right)\right), \end{aligned} \tag{14}$$

where

$$\Sigma^\dagger = \left(\Sigma^{-1} + \Sigma'^{-1}\right)^{-1} \text{ and } \mu^\dagger = \Sigma^{-1}\mu + \Sigma'^{-1}\mu'.$$

Here, we consider the PPK with $\rho = 1$, it is called the ELK. In practice, we compute them with logarithm to ensure numerical stability as follows:

$$ELK(p,q) = -\frac{D}{2}\log 2\pi - \frac{1}{2}\log(\Sigma_1 + \Sigma_2) + \frac{(\mu_1 - \mu_2)^2}{\Sigma_1 + \Sigma_2}. \tag{15}$$

In essence, ELK provides a tractable probabilistic formulation to measure the similarity between two distributions. Other metrics like divergence and cosine similarity are less effective and efficient than ours, as we will illustrate in the experiment section 5.

3.5. Probabilistic contrastive learning

Similar to conventional deterministic contrastive learning, our objective is imposed to pull the distribution of the source-composed prototype embeddings and both the source and target image pixel embeddings closer, while pushing away the distributions of negative pairs. Compared to the most used cosine similarity $s(z_1, \rho) = \frac{z_1 \cdot \rho}{\|z_1\|\|\rho\|}$, which only leverages angles between them, leaving class-specific distribution covariance ignored, we propose to use distribution similarity metrics mentioned in Eq. (15), accounting for either mean and covariance.

$$\mathcal{L}_c = \mathbb{E}_{q,\mu^+,\mu^-}[-\log \frac{e^{\kappa(q,\mu^+)/\tau}}{e^{\kappa(q,\mu^+)/\tau} + \sum_{n=1}^{N} e^{\kappa(q,\mu^-)/\tau}}], \tag{16}$$

where q denotes query pixel embedding, μ^+ and μ^- denote positive and negative prototype embedding with respect to query q. In practice, we extract pixel embeddings from both the source and target domains as q, and based on the corresponding labels or pseudo-labels, we consider prototypes belonging to the same class as q as μ^+, while those belonging to different classes are considered as μ^-. $\kappa(\cdot, \cdot)$ denotes kernel function which measures the similarity between two distributions. It is obvious that the $\kappa(\cdot, \cdot)$ is left unbounded if the covariance Σ converges to zero. To make covariance a valid representation, we employ the additional KL regularization term between the pixel embedding distribution and the unit Gaussian prior $\mathcal{N}(0, I)$ to prevent the covariance collapse to zero:

$$\mathcal{L}_{\text{KL}} = \text{KL}(p(z_i|x^{x,t})||\mathcal{N}(0, I)). \tag{17}$$

Therefore, the overall training objective is formulated as:

$$\mathcal{L} = \mathcal{L}_s + \lambda_t \mathcal{L}_t + \lambda_c \mathcal{L}_c + \lambda_{KL} \mathcal{L}_{KL}, \tag{18}$$

where λ_t, λ_c, λ_{KL} control the trade-off of the corresponding loss.

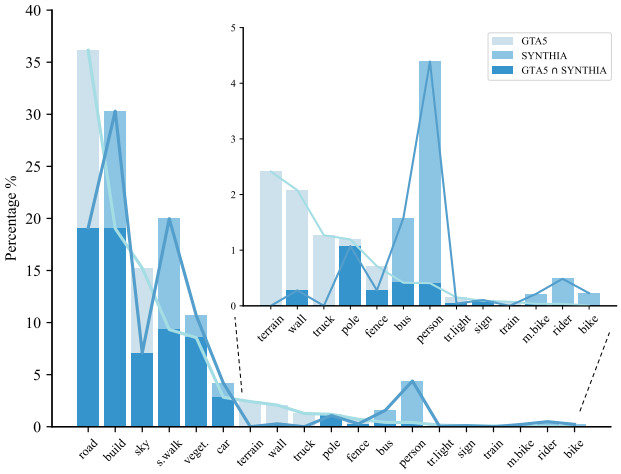

Figure 3: The Class distribution of the Cityscapes and SYNTHIA datasets.

3.6. Ambiguity-guided cropping

Class balance is hard to achieve for almost all of the dataset except with careful splitting. An imbalanced dataset often leads to overfitting to majority classes in the source domain. Rare class sampling (RCS) [6] is one of the ways to solve the probslem, however, there are no labels available in the target domain. Class-balanced cropping (CBC) [21] proposes to utilize generated target pseudo labels to crop image regions that jointly promote class balance in pixel number and diversity of internal categories. Though this strategy reduces the influence of imbalance, the discriminative capability improves less significantly because the model gains little on a balanced crop which only considers the numerical class balance. We suggest that in the latter stages of training, the difficulty of classes has a greater impact on the model's performance than class balance. To handle this issue, we propose Ambiguity-guided cropping (AGC) which utilizes the class-wise ambiguity provided by the prototypes rather than pixel embeddings, as the sample size for each class varies greatly, leading to inaccurate estimation. More specifically, we first calculate the ambiguity of each class by:

$$a_i = \text{softmax}(\Sigma_i/\tau), \tag{19}$$

where Σ_i denotes prototype covariance of class i, τ denotes the temperature. As shown in Fig. 3, there are significant differences in the occurrence

frequencies of different classes. Since the labels in the target domain are not available, to mitigate the impact of this imbalance on the estimation of crop uncertainty, we assume that the class distribution in the target domain is similar to that in the source domain. It can be observed that although GTAV and SYNTHIA have inconsistent percentages in a very few classes, they both exhibit a long-tailed distribution overall. Specifically, we only select the top-k classes with similar frequency levels as candidate classes, reducing the impact of class imbalance on crop ambiguity estimation. We calculate the overall score for each random select crop by:

$$score_j = \sum_{i=0}^{n-1} \mathbb{1}_{i \in S_{\text{top\_k}}} a_i n_{ij}, \tag{20}$$

where j is the crop index, i denotes the class, $S_{\text{top\_k}}$ denotes the set only contains top-k ambiguity classes, $\mathbb{1}_{[\cdot]}$ is an indicator function, n_{ij} denotes pixel number for class i in crop j. In practice, given a target image with the original resolution, we random crop it for N times, and calculate the score for each crop, the crop with the highest score is selected as the input image for the network. From Eq. (19) and Eq. (20), we can see that RCS is a special case of AGC when at the initial stage of training, all Σ_i have the same value, however, with the update of the prototypes, the crops are prone to involve more ambiguous scenarios, which better helps our PPPC in adjusting the decision boundary. Moreover, our AGC can be regarded as introducing image norm through prototypes as a selection criterion to boost the model's discriminative capability without a forward propagation. This is in line with the finding that CNNs encode the amount of visible class discriminative features in the norm of the embedding [46].

4. EXPERIMENT

4.1. Datasets and Evaluation Metrics

We evaluate our method on three cross-domain semantic segmentation tasks: GTAV → Cityscapes, SYNTHIA → Cityscapes and Cityscapes → Dark Zurich. Two synthetic datasets GTAV and SYNTHIA are served as labeled source domains, and Cityscapes is served as unlabeled target domain. For day-to-night adaptation, Cityscapes is utilized as source domain, Dark Zurich is served as the target domain. The overview of all tasks is shown in Fig. 4.

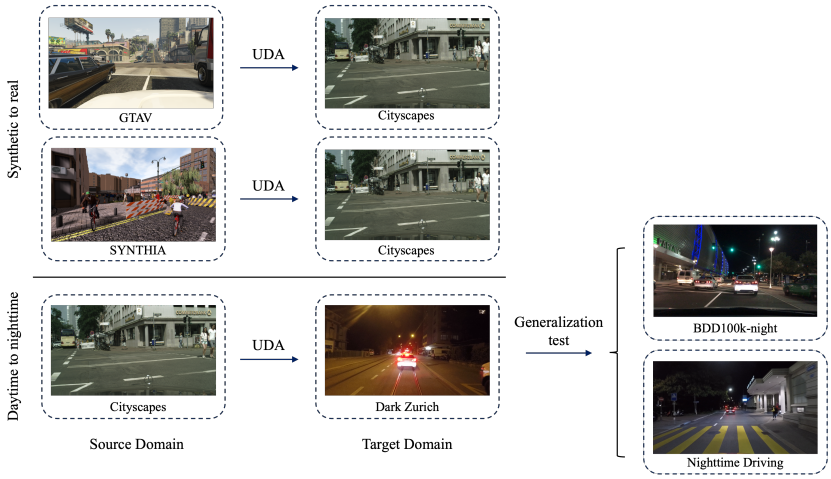

Figure 4: Overview of all our tasks. Our tasks include two synthetic-to-real tasks and one daytime-to-nighttime task. Additionally, we conduct extra generalization tests on the daytime-to-nighttime task.

1. GTAV [47] is a composite dataset sharing 19 classes with Cityscapes, which contains 24,966 images with a resolution of 1914×1052.

2. SYNTHIA [48] is a synthetic urban scene dataset, we select its subset SYNTHIA-RAND-CITYSCAPES which has 16 common semantic annotations with Cityscapes and contains 9,400 training images with a resolution of 1280×760.

3. Cityscapes [3] is a real urban scenes dataset taken from cities in Europe, we use 2,975 annotated images at the training stage, and 500 validation images during evaluation, all the images have a resolution of 2048×1024.

4. Dark Zurich [15] is a large dataset with urban driving scenes. It consists of 2,416 nighttime images, 2,920 twilight images, and 3,041 day-time images, with a resolution of 1920×1080. Following [21], we use 2,416 day-night image pairs as target training data and another 151 test images as target test data. We obtain the mIoU result by submitting the segmentation predictions to the online evaluation website[1].

5. BDD100k-night [49]. To verify the generalization of our model, we

[1]https://competitions.codalab.org/competitions/23553

Table 1: Hyperparameter values for the GTAV → Cityscapes (G → C), SYNTHIA → Cityscapes (S→ C) and Cityscapes → Dark Zurich (C → D) tasks.

Settings	G→C	S→C	C→D
Top-k ambiguity class k	10	10	10
Cropping times N	10	10	10
Total training iterations	40k	40k	60k
Learning rate	6e−5	6e−5	6e−5
λ_t	1	1	1
λ_c	1	1	1
λ_{KL}	1e−6	1e−6	1e−6

directly use our model trained from Cityscapes → Dark Zurich to predict the test set of BDD100k-night. BDD100k-night contains 87 images with a resolution of 1280×720.

6. Nighttime Driving [50]. The Nighttime Driving includes 50 nighttime driving-scene images with a resolution of 1920×1080. We also adopt it to verify the generalization of our model trained from Cityscapes → Dark Zurich.

We employ per-class intersection-over-union(IoU) and mean IoU over all classes as the evaluation metric. We perform the evaluation on 19 categories for GTAV → Cityscapes, Cityscapes → Dark Zurich tasks, and both 16 and 13 categories for SYNTHIA → Cityscapes task.

4.2. Implementation Details

4.2.1. Network architecture

We adopt DeepLab-V3+ with ResNet101 as our base segmentation architecture. Meanwhile, both the mean and covariance of the embeddings are generated by the projection head. Specifically, they are produced separately by two fully-connected layers with BN and ReLU, mapping the high-dimensional pixel embedding from the backbone into a 512-d l_2-normalized vector. However, we replace the last ReLU function with the exponential (Exp) function when generating covariance to ensure the covariance is non-negative. Similar to [21], the backbone is initialized using the weights pre-trained on ImageNet, and the other layers are initialized randomly.

4.2.2. Training

We train the network with a batch of two 640 × 640 random crops. Threshold α is set to 0.968. Cropping times N is set to 10, the default value of k for the top-k ambiguity class selection is set to 10. We adopt AdamW as our optimizer with betas (0.9, 0.999) and weight decay 0.01. The initial learning rate for the backbone and projection head is set to 6×10^{-5}, 6×10^{-4}. Moreover, linear learning rate warmup and polynomial learning rate decay are used. For all experiments, we set λ_t, λ_c to 1.0 , λ_{KL} to 10^{-6}, and momentum for prototype update to 0.999 respectively. We train the model for a total of 40k iterations for synthetic-to-real tasks and 60k iterations for day-to-night adaptation. The distribution contrast starts from 3k iterations. The hyperparameters for all tasks are shown in Table 1. We keep all hyperparameters the same as our baseline [21], except for k, which we select through experimentation. All experiments are implemented on a single NVIDIA A30 GPU with PyTorch.

4.2.3. Testing

During test time, all projection heads, the student model, and prototypes are discarded. We resize the test images to 1280×640 and obtain the segmentation results directly through the adapted teacher model.

4.3. Comparisons with the state-of-the-arts

We compare our PPPC with the recently leading approaches in two challenging synthetic-to-real cross-domain semantic segmentation tasks including GTAV → Cityscapes and SYNTHIA → Cityscapes. Additionally, a day-time to night-time adaptation task Cityscapes → Dark Zurich is also included.

Table 2 shows the adaptation results on the task of GTAV → Cityscapes with comparisons to the state-of-the-art methods [12, 45, 30, 51, 21, 52, 53, 18, 54, 55, 31, 56]. It can be seen that our PPPC achieves superior performance. Specifically, our methods achieve the best mIoU of 63.7%, significantly outperforming the deterministic feature alignment approach SePiCo [21] and also surpassing OTCLDA [56], which aligns distributions through optimal transport. Due to the proposed probabilistic pixel embedding inherent with the capability to gather information about the style gap between two domains, thus outperforming the style transfer based method CONFETI [18] by +1.5% and DiGA [55] which additionally combines a distillation procedure by +1.0%. Moreover, our PPPC achieves mIoU even higher than HRDA [31] which uses high-resolution images as input. This shows that the

Table 2: Comparison results of **GTAV → Cityscapes**. All methods are based on ResNet-101 for a fair comparison. The best result is highlighted in **bold**,with the second best results underlined.† indicate method trained at higher resolution.

Method	Road	S.walk	Build	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
ProCA [45]	91.9	48.4	87.3	41.5	31.8	41.9	47.9	36.7	86.5	42.3	84.7	68.4	43.1	88.1	39.6	48.8	40.6	43.6	56.9	56.3
ProDA [12]	87.8	56.0	79.7	46.3	44.8	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1.0	48.9	56.4	57.5
ADPL [51]	93.4	60.6	87.5	45.3	32.6	37.3	43.3	55.5	87.2	44.8	88.0	64.5	34.2	88.3	52.6	61.8	49.8	41.8	59.4	59.4
PBAL [52]	93.4	59.5	87.7	49.7	41.3	45.2	53.8	43.7	88.6	41.7	**91.2**	70.5	36.0	89.4	49.9	52.1	**57.8**	49.0	46.7	60.4
CPSL [30]	92.3	59.9	84.9	45.7	29.7	**52.8**	**61.5**	59.5	87.9	41.5	85.0	73.0	35.5	90.4	48.7	**73.9**	26.3	**53.8**	53.9	60.8
SePiCo [21]	95.2	67.8	88.7	41.4	38.4	43.4	55.5	63.2	88.6	46.4	88.3	73.1	49.0	91.4	63.2	60.4	0.0	45.2	60.0	61.0
FREDOM [54]	90.9	54.1	87.8	44.1	32.6	45.2	51.4	57.1	88.6	42.6	89.5	68.8	40.0	89.7	58.4	62.6	55.3	47.7	58.1	61.3
RTea [53]	95.4	67.1	87.9	46.1	**44.0**	46.0	53.8	59.5	**89.7**	**49.8**	89.8	71.5	40.5	90.8	55.0	57.9	22.1	47.7	62.5	61.9
CONFETI [18]	95.7	69.9	89.5	34.6	42.6	40.9	57.5	59.4	88.6	49.0	88.2	72.8	**53.4**	90.1	61.8	54.9	13.9	50.2	63.4	62.2
DiGA [55]	95.6	67.4	**89.8**	**51.6**	38.1	52.0	59.0	51.5	86.4	34.5	87.7	**75.6**	48.8	**92.5**	**66.5**	63.8	19.7	49.6	61.6	62.7
OTCLDA [56]	**95.8**	**70.5**	89.1	40.4	40.6	45.7	55.7	**67.1**	89.4	48.1	90.6	73.8	49.3	91.4	66.4	62.1	0.1	53.4	63.5	62.8
PPPC(Ours)	94.9	65.2	88.4	45.5	41.1	50.1	56.3	65.5	88.7	42.0	89.0	73.4	43.3	90.8	60.2	62.8	39.9	47.6	**64.9**	**63.7**
HRDA† [31]	96.2	73.1	89.7	43.2	39.9	47.5	60.0	60.0	89.9	47.1	90.2	75.9	49.0	91.8	61.9	59.3	10.2	47.0	65.3	63.0

Table 3: Comparison results of **SYNTHIA → Cityscapes**. mIoU* denotes the mean IoU of 13 classes excluding the classes with *.All methods are based on ResNet-101 for a fair comparison. The best result is highlighted in **bold**,with the second best results underlined.

Method	Road	S.walk	Build	Wall*	Fence*	Pole*	Tr.Light	Sign	Veget.	Sky	Person	Rider	Car	Bus	M.bike	Bike	mIoU	mIoU*
DACS [29]	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24.0	83.7	90.8	67.6	38.3	83.0	38.9	28.5	47.6	48.4	54.8
ProCA [45]	90.5	52.1	84.6	29.2	3.3	40.3	37.4	27.3	86.4	85.9	69.8	28.7	88.7	53.7	14.8	54.8	53.0	59.6
ProDA [12]	87.8	45.7	84.6	**37.1**	0.6	44.0	54.6	37.0	88.1	84.4	74.2	24.3	88.2	51.1	40.5	45.6	55.5	62.0
ADPL [51]	86.1	38.6	85.9	29.7	1.3	36.6	41.3	47.2	85.0	90.4	67.5	44.3	87.4	57.1	43.9	51.4	55.9	63.6
DecoupleNet [57]	77.8	48.6	75.6	32.0	1.9	44.4	52.9	38.5	87.8	88.1	71.1	34.3	88.7	58.8	50.2	61.4	57.0	64.1
PBAL [52]	86.5	46.5	85.1	23.6	0.4	44.1	49.7	46.5	**88.4**	**91.6**	72.6	41.5	89.5	63.1	43.1	48.1	57.5	65.6
SePiCo [21]	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	**69.7**	50.9	**66.5**	58.1	66.5
CPSL [30]	87.2	43.9	85.5	33.6	0.3	47.7	**57.4**	37.2	87.8	88.5	**79.0**	32.0	**90.6**	49.4	50.8	59.8	58.2	65.3
OTCLDA [56]	87.3	45.9	**87.0**	17.0	**5.7**	**49.0**	55.9	56.7	81.6	75.7	74.6	41.4	89.0	53.7	**51.1**	63.8	58.5	66.4
Ours	**90.9**	**52.9**	84.7	21.9	2.1	45.4	49.4	**57.0**	85.2	87.5	72.9	**46.3**	86.8	47.2	50.4	61.0	**58.8**	**67.1**

proposed probabilistic pixel embedding and prototypical contrast approach can effectively deal with ambiguous regions in images by considering the uncertainty of each pixel and prototype without the need for high-resolution images.

Table 3 shows the adaptation results on the task SYNTHIA → Cityscapes. This adaptation task is more challenging than the previous one due to the large domain gap, but our PPPC still achieves significant improvements over competing methods. PPPC attains 58.8% mIoU and 67.1% mIoU* achieving a significant improvement with ProDA [12] and PBAL [52], outperforming them by +3.2% mIoU and +1.2% mIoU respectively. Our method can outperform these approaches without the need for any complex strategies

Table 4: Comparison results of **Cityscapes** → **Dark Zurich**. All methods are based on ResNet-101 for a fair comparison. The best result is highlighted in **bold**,with the second best results underlined.

Method	Road	S.walk	Build	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
DANNet [14]	90.4	60.1	71.0	33.6	22.9	30.6	34.3	33.7	70.5	31.8	80.2	45.7	41.6	67.4	16.8	0.0	73.0	31.6	22.9	45.2
DIAL-Filters [58]	90.6	60.8	70.9	40.2	21.1	39.6	34.4	38.3	**73.2**	30.2	72.9	**48.6**	41.6	72.8	8.8	0.0	**74.6**	33.0	22.8	46.0
LoopDA [59]	86.3	46.3	**76.1**	30.3	22.5	32.5	34.1	34.8	62.6	19.5	84.3	46.6	51.5	**73.2**	60.7	3.1	73.4	26.2	24.8	46.8
DANIA [60]	90.8	59.7	73.7	39.9	26.3	36.7	33.8	32.4	70.5	32.1	**85.1**	43.0	42.2	72.8	13.4	0.0	71.6	**48.9**	23.9	47.2
CCDistill [61]	89.6	58.1	70.6	36.6	22.5	33.0	27.0	30.5	68.3	**33.0**	80.9	42.3	40.1	69.4	58.1	0.1	72.6	47.7	21.3	47.5
SePiCo [21]	**91.2**	61.3	67.0	28.5	15.5	44.7	**44.3**	**41.3**	65.4	22.5	80.4	41.3	**52.4**	71.2	39.3	0.0	39.6	27.5	28.8	45.4
PPPC(Ours)	90.9	**63.0**	75.6	**43.9**	**27.6**	**53.3**	40.9	33.1	67.1	29.2	79.8	46.3	51.7	53.6	**61.6**	**14.2**	65.0	33.6	**30.8**	**50.6**

like muti-stage self-training in ProDA or knowledge distillation in PBAL, which further verifies the benefits of distributional pixel contrast with prototypes. Notably, our methods achieve the best mIoU* at 67.1%, surpassing the approach CPSL [30], which addresses class imbalance and refines pseudo-labels through clustering pixels, further highlighting the superiority of distributional embedding under class imbalance scenarios. Additionally, the higher performance compared to OTCLDA [56] demonstrates that our kernel function is more advantageous than optimal transport (i.e., Wasserstein Distance). A detailed comparison can be found in Section 4.4.5.

Table 4 shows the adaptation results on the task Cityscapes → Dark Zurich. Our PPPC achieves significant improvement over the methods explicitly tailored for addressing day-time to night-time adaptation, outperforming CCDstill [61] by +3.1% mIoU and LoopDA [59] by +3.8%. Like SePiCo [21], our PPPC is a universal framework for domain adaptation, not limited to addressing day-time to night-time tasks. Compared to SePiCo, we not only exhibit noticeable improvement in majority groups such as buildings, vegetation, and person but also achieve significant enhancement in minority groups such as fences, poles, and trucks, eventually leading gain of +5.2%.

Fig. 5 shows the qualitative segmentation results on Cityscapes. We compare our results predicted by PPPC to the others obtained by the Source Only, SePiCo [21], and CONFETI [18]. Our probabilistic methods PPPC predict a clearer outline in ambiguous objects like backlit walls, dense pedestrian crowds, overlapping bicycles, and distant traffic signs, showing a significant improvement over the deterministic methods.

4.4. Ablation study

We conduct comprehensive ablation studies on each component present in the proposed method. For a fair comparison, all experiments are imple-

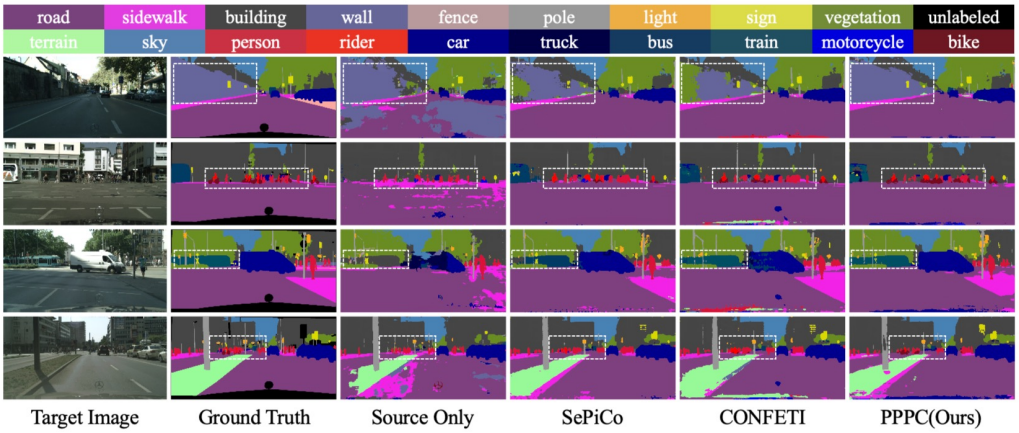

Figure 5: Qualitative results of domain adaptive semantic segmentation task GTAV → Cityscapes. Better segmentation results are highlighted in dash boxes. PPPC improves the segmentation of ambiguous scenarios such as inadequately illuminated walls, crowded persons, and challenging classes like bus.

mented under the same training settings.

1. ***Effect of probabilistic prototypical pixel contrast***. As mentioned before, the probabilistic contrast not only reduces the intra-class distance and increases inter-class distance, which is easy to realize by deterministic contrast, but makes the decision boundary clearer near the ambiguous embedding point. We investigate the effect of probabilistic contrast with and without the self-training framework separately. As shown in Table 5, probabilistic contrast brings mIoU gains of +6.4% on GTAV → Cityscapes and +2.7% on SYNTHIA → Cityscapes without self-training which is less than using self-training techniques alone. Interestingly, when we combine the probabilistic contrast with self-training, the performance is slightly degraded compared with the result using only self-training. We conjecture that the model trained without self-training lacks the ability to correlate the two domains, hence resulting in a substantial mean difference ,i.e., $(\mu_1-\mu_2)^2$, which can be easily addressed by probabilistic contrast. However, the self-training strategy boosts the transfer capability of the model between two domains, leading to relatively lower values on the numerator in Eq. (15). which could cause erroneous optimization direction if the covariance on the denominator and logarithm is not properly regulated.

2. ***Effect of*** $\mathcal{L}_{KL}$. Because our loss function is left-unbounded, an in-

Table 5: Ablation study on the GTAV → Cityscapes (G → C) and SYNTHIA → Cityscapes tasks(S→ C). AGC denotes Ambiguity-guided Cropping.

Method	$\mathcal{L}_{ssl}$	$\mathcal{L}_{cl}$	$\mathcal{L}_{KL}$	AGC	G → C		S→ C	
					mIoU	Δ	mIoU	Δ
w/o self-training					37.1	-	26.5	-
		✓			43.5	6.4	29.2	2.7
		✓	✓		45.2	8.1	30.3	3.8
		✓	✓	✓	47.2	10.1	32.5	6.0
PPPC(Ours)	✓				59.4	-	55.1	-
	✓	✓			58.7	-0.7	53.3	-1.8
	✓	✓	✓		62.0	2.6	56.1	1.0
	✓	✓	✓	✓	**63.7**	4.3	**58.9**	3.8

Table 6: Comparison results of mIoU for different cropping methods on the target domain.

Method	Random	CBC	AGC (Constant)	AGC
mIoU	61.4	60.5	61.6	**63.7**

correct optimization direction results in the covariance converging to ∞. We study the advantages of using KL divergence as a regularization term on the similarity measurement of two distributions in Table 5. It is obviously observed that by adding the KL divergence, the covariance is limited to a meaningful range, and the full potential of probabilistic contrast is released, achieving a significant gain of +1.7% and +3.3% on GTAV → Cityscapes, +1.1% and +2.8% on SYNTHIA → Cityscapes respectively.

3. ***Effect of ambiguity-guided cropping(AGC)*** . As shown in Table 5, our cropping method achieves noticeable gains of +2.0% and +1.7% on GTAV → Cityscapes, +2.2% and +2.8% on SYNTHIA → Cityscapes. The results imply that the covariance predicted by the model not only benefits the alignment at the feature level but also helps to select reasonable inputs according to the transfer capability of the model dynamically at the data level. Next, we analyze the parameter sensitivity of AGC, and compare it with other strategies on target domains. From Fig. 6, we can see that increasing the value of k within the range of not greater than 10 slightly improves the performance, but the performance drops when k is greater than 10. We conjecture that although the precise trend is not aligned due

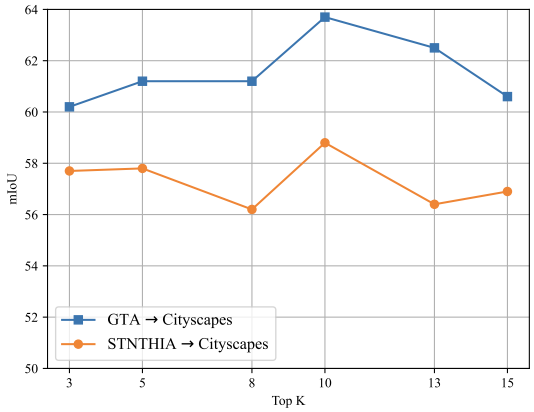

Figure 6: Parameter sensitivity analysis for AGC.

to different source domains (e.g. the STNTHIA has 16 classes which is fewer than 19 classes in GTAV), the head classes with the highest covariance are generally the same. Thus, we choose 10 as the default value of k for all experiments.

Furthermore, Table 6 shows the performance of different cropping methods implemented on target domain inputs. Random stands for cropping a image to given size randomly, CBC stands for class-balanced cropping which gives crops by evaluating the max ratio of each class and the score of classes meet the condition, AGC (Constant) stands for the same configuration with AGC except we fixed the covariances of prototypes to the reciprocal of the class frequency in the source dataset. As seen, CBC achieves the worst result even lower than random selection, showing that CBC is not suitable for probabilistic-based methods. Moreover, AGC (Constant) exhibits a marginal performance gap + 0.2% with random selection, indicating that our probabilistic embedding has, to some extent, addressed the class imbalance issue, the model will no longer benefit from continuing to use class-balance strategies. Thus, the main issue with these methods lies in their inability to effectively introduce ambiguous points near the decision boundary at the data level. Conversely, our AGC is designed to tackle this problem by scoring each crop with class-wise uncertainty provided by prototypes. With dynamically adjusted crop output by our AGC strategy, we achieve +2.3% mIoU compared with random selection.

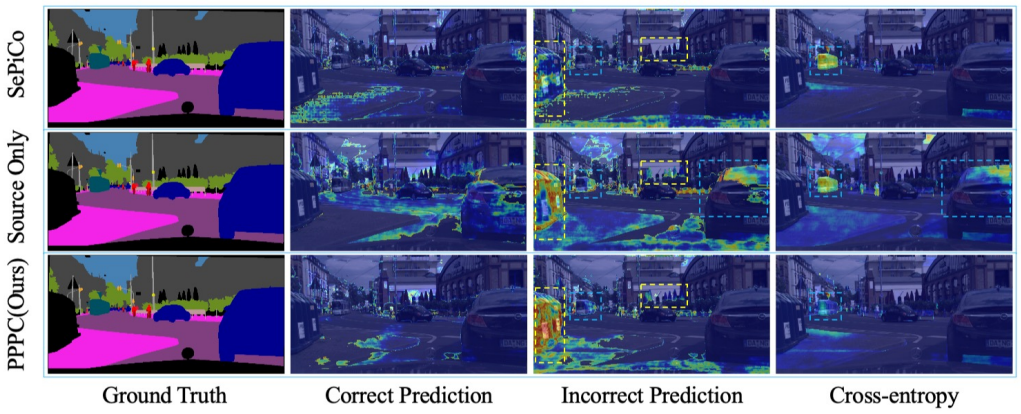

Figure 7: Visualization of the entropy map for correct predictions, incorrect predictions, and the cross-entropy map with ground truth. PPPC shows low uncertainty for correct predictions, and high uncertainty for both unlabeled and incorrect classes. For incorrect predictions, apart from the unlabeled classes, the more consistent it is with corss-entropy map, the model is less likely to assign high confidence to the incorrect class.

Table 7: The degree of freedom (DoF) for Σ and mIoU on GTAV $\rightarrow$ Cityscapes (G $\rightarrow$ C) and SYNTHIA $\rightarrow$ Cityscapes tasks(S$\rightarrow$ C).

Method	DoF(σ)	G $\rightarrow$ C	S $\rightarrow$ C
PPPC μ only	0	59.4	56.0
PPPC iostropic	1	59.1	56.6
PPPC	512	**63.7**	**58.8**

4. ***Effect of probabilistic embedding***. We model the probabilistic embedding with both μ and Σ. We remove the covariance Σ to illustrate its necessity. As shown in Table 7, DoF denotes the degree of freedom for covariance. Removing the covariance of embedding leads our method to degrade to a deterministic approach, resulting in a decrease of -4.3% and -2.8% on GTAV$\rightarrow$ Cityscapes and SYNTHIA $\rightarrow$ Cityscapes, demonstrating the significance of modeling covariance in probabilistic methods. Furthermore, although we parameterize the covariance matrix $\Sigma \in \mathbb{R}^{D\times D}$ with its diagonal element vector $\sigma \in \mathbb{R}^{D}$, one may consider modeling the covariance Σ with the same value on each position of diagonal elements, leading to isotropic gaussian distributions. From Table 7, we can observe that increasing the freedom of covariance Σ brings continuous improvement, however, to increase the

training efficiency, we just choose 512-d for all experiments.

To illustrate how our model leverages uncertainty, we visualize the entropy map for both correct and incorrect predictions, as well as the cross-entropy map in Fig. 7. Entropy map for predictions is derived from the calculation of max probability predicted from the unlabeled image for each pixel, describing the model's uncertainty for each part of the image. On the other hand, the cross-entropy map is generated based on the cross-entropy between the model's prediction and the ground truth label, depicting the extent of deviation from the correct results. For a discriminative model, lower uncertainty should be assigned to correct predictions, while higher uncertainty for incorrect predictions. From these visualizations, we can observe that (i) for correct predictions, our PPPC shows less uncertainty about each pixel; (ii) for incorrect predictions, by comparing with the cross-entropy map, the deterministic approach SePiCo is prone to assign high confidence to the wrong class (marked as the blue box), which would lead to incorrect optimization direction as shown in Fig. 1(b). However, our probabilistic method, even the source-only version, is capable of giving high uncertainty to ambiguous pixels, ensuring most of the incorrect predictions are among the high-uncertainty groups; (iii) Although the unlabeled class is not taken into account in the cross-entropy calculation, our method still can discriminative it from the other class, and assign high uncertainty to it (marked as the yellow box), which is struggled for deterministic method. This further indicates the generalization of our PPPC and its potential application in risk control for autonomous driving.

5. ***Effect of distribution measurement***. To demonstrate the effectiveness and efficiency of our method, we compare it with other probabilistic distance variants to measure the distance between two Gaussian distributions. One may consider using the Monte-Carlo sampling strategy to approximate the similarity between two distributions as:

$$sim(p(z_i|x_i), p(z_i|\rho_i)) = \frac{1}{J^2}\sum_{i=1}^{J}\sum_{j=1}^{J} d(z_i^{x_i}, z_j^{\rho_j}), \tag{21}$$

where $p(z_i|x_i)$ denotes the probability of pixel embedding, $p(z_i|\rho_i))$ denotes the embedding of prototype, J is the sampling number for each distribution and $d(\cdot,\cdot)$ refers to cosine similarity. We also compare with divergence-based methods, including KL divergence, JS divergence, and Wasserstein Distance.

Table 8: Comparison results of mIoU for different distribution measurements trained on GTAV → Cityscapes.The best result is highlighted in **bold**,with the second best results underlined, the third best results double underlined.

Method	Sampling	mIoU
Cosine Similarity	10	62.0
	20	62.3
	30	<u>63.4</u>
	40	61.7
	50	60.8
KL Divergence	✗	60.1
JS Divergence	✗	60.0
Wasserstein Distance	✗	60.4
Bhattacharyya Kernel	✗	<u>62.7</u>
Expected Likelihood Kernel	✗	**63.7**

Furthermore, we introduce another special case of PPK when $\rho = \frac{1}{2}$, it is called Bhattacharyya Kernel (BK):

$$BK(p,q) = -\frac{D}{2}\log\frac{1}{2} + \frac{1}{2}\log\left(\frac{\Sigma_1\Sigma_2}{\Sigma_1^{\frac{1}{2}} + \Sigma_2^{\frac{1}{2}}}\right) - \frac{1}{4}\frac{(\mu_1-\mu_2)^2}{\Sigma_1+\Sigma_2}. \tag{22}$$

As shown in Table 8, it can be observed that the divergence-based method performs the worst and is numerically unstable during the experiments. This instability arises because it involves division terms by variances Σ_1 and Σ_2 (i.e., $\frac{\Sigma_1}{\Sigma_2}$), which can lead to numerical instability when the variances are very small. That is to say, when we have a highly certain embedding with nearly zero variance, the KL divergence between p and q will explode. Increasing the number of samples helps to improve the performance of measuring the distribution similarity. Notably, when the number of samples is set to 30, it achieves a comparable performance to kernel methods. However, it significantly increase computation overhead and the space complexity is $O(NJ^2)$, where N refers the number of pixel embeddings. Instead, our kernel methods achieve the best results with space complexity of only $O(2N)$ and does not require complex operations such as sampling and reparameterization.

6. ***Effect of combination of all components***. When we combine all ingredients properly in a unified pipeline, the full potential of our model

	Wall	Fence	Tr.Light	Sign	Rider	Truck	Bus	Train	M.bike	Bike	mIoU†
Source Only	10.3	23.8	39.3	24.8	34.3	29.8	24.0	11.3	23.0	15.0	23.6
ProCA	41.5	31.8	47.9	36.7	43.1	39.6	48.8	40.6	43.6	56.9	43.1
ProDA	46.3	44.8	53.5	53.5	39.2	45.5	59.4	1.0	48.9	56.4	44.9
SePiCo	41.4	38.4	55.5	63.2	49.0	63.2	60.4	0.0	45.2	60.0	47.6
PPPC(Ours)	45.5	41.1	56.3	65.5	43.3	60.2	62.8	39.9	47.6	64.9	52.7

Figure 8: The results of different methods on long-tail classes. mIoU† indicates mIoU for tail classes.

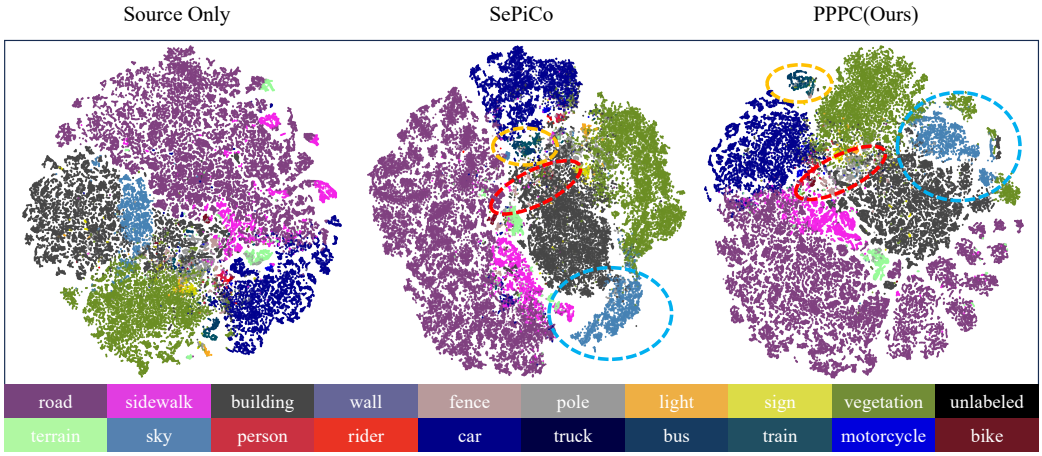

Figure 9: T-SNE analysis for the bottleneck features for ResNet-101 on the Cityscapes val. set.

is released, resulting a 63.7% mIoU surpassing even the generative-based method CONFETI [18]. Our method has not only achieved considerable improvement in the head class but also notable enhancements in the tail class. As shown in Fig. 8, the majority of improvement lies in easily confused pairs like train and bus, motorbike and bike. Additionally, smaller objects such as fences, traffic signs, and traffic lights also show a slight increase, achieving a 52.7% tail-class mIoU†.

4.5. t-SNE visualization

To gain insight on the improved discriminative capability, Fig. 9 visualizes the features of the deterministic method SePiCo, our probabilistic method, and source only method for reference. We select a few images from the target domain to ensure that all 19 classes are represented. Then we

Table 9: Comparison results of Cityscape → Dark Zurich trained models for generalization on two unseen target domains: Nighttime Driving and BDD100k-night test sets. All methods are based on ResNet-101 for a fair comparison.

Method	Dark Zurich	Nighttime Driving	BDD100k-night	Cityscapes
DANNet [14]	45.2	47.7	-	-
DANIA [60]	-	<u>48.4</u>	-	-
MGCDA [62]	42.5	**49.4**	**34.9**	-
CCDistill [61]	<u>47.5</u>	-	<u>33.0</u>	-
SePiCo [21]	45.4	47.0	22.4	<u>74.4</u>
PPPC(Ours)	**50.6**	47.6	24.5	**77.3**

map the feature representations, derived from the final stage of ResNet-101 backbone, also served as input to the decoder, into 2D space to create t-SNE visualization. We can observe that SePiCo tends to treat a class as a single cluster, employing contrastive learning to make the intra-class compact and inter-class distant. Conversely, our PPPC employs a probabilistic approach, resulting in multiple clusters within the same class, preserving the diversity of features in each class. Additionally, it is noteworthy that we successfully distinguish between train and bus, which are easily confused by other methods (marked as the orange circle). Interestingly, it seems that SePiCo has a more distinct boundary than our method (e.g., sky class exhibits a far distance from vegetation and building, marked as the blue circle), however, its mIoU performance is lower than our method. Hence, we conjecture that intra-class compactness and inter-class separability are not necessary for enhancing model discriminative capability, while handling boundary points properly might be the most crucial, which aligns with the core idea of our proposed method as shown in Fig. 1(c). Moreover, this assumption is further validated by our PPPC, which significantly reduces the overlapping area (marked as the red circle).

4.6. Generalization to unseen domains

We also validate the generalization of PPPC, the model trained on Cityscapes → Dark Zurich is directly tested on two unseen domains, Nighttime Driving and BDD100k-night. In Table 9, we can see that our PPPC achieves a comparable performance, although slightly lower than the method carefully designed for night-time adaptation tasks. Moreover, our PPPC outperforms SePiCo on the source domain, indicating that our probabilistic approach is

Figure 10: Cityscapes val. image crops with lowest(right) to highest(left) embedding norms and crops with lowest(left) to highest(right) variances. It can be observed that crops with high variance or low norm show the same trend, both involving ambiguous scenarios, vice versa.

not only beneficial to the target domain and target-related domains but also contributes to the model's discriminative capability on the source domain.

5. Efficiency Analysis

5.1. Relationship between embedding norms and variance

We further analyze the relationship between norms and variance to highlight the superiority of our method. We calculate image norms for each crop by averaging the l_2-norms of all the pixel embeddings. Fig. 10 shows the images with the highest and lowest embedding norm in the validation set. In high-norm images, the characteristic parts of objects are particularly prominent, facilitating the detection of class-discriminative features. However, for samples with low norm, illumination appears dim and objects are overlaid, hindering the learning, in line with recent findings [46]. Similarly, we computed image variance and prototype variance respectively, observing a consistent trend between low variance and high norms images, and vice versa. Therefore, we believe that variance serves as another means to describe image ambiguity, much like the role of norms. More importantly, the computation of prototypes is independent of the network's forward propagation (i.e., Eq. (12)), unlike image norms and variance, which require obtaining the values through the entire backbone. Thus, our AGC poses as an efficient approximation for selection via image norms.

Table 10: Efficiency comparison with different methods on GTAV → Cityscapes. 'Memory' represents GPU memory usage, 'Time' represents the average training time per iteration, 'Params' denotes trainable parameters, and 'FLOPs' denotes floating point operations. Results are obtained using NVIDIA A30 with a batch size of 2.

Method	Memory (GB)	Time (s)	Params (M)	FLOPs (G)	mIoU
Source Only	10.9	1.3	353.5	41.2	37.1
SePiCo [21]	12.2	3.1	645.2	49.1	61.0
HRDA [31]	22.1	3.2	734.0	43.9	63.0
PPPC(Ours)	15.2	3.3	713.1	50.2	**63.7**

5.2. Throughout

The efficiency comparison of representative methods are shown in Table 10. We can observe that PPPC, unlike HRDA, doesn't require the use of high-resolution images, which nearly doubles the GPU memory footprint. It only introduces a slight increase in computation and memory usage compared to SePiCo but achieves a significant enhancement. Moreover, compared to deterministic methods like SePiCo, our probabilistic approach adds only an additional probability head to encode means and variances, leading to a relative increase in the number of parameters. However, because we use posterior probability estimates for prototypes and PPK for contrastive learning, the FLOPs increases only slightly despite the increase in parameters.

6. Limitations and future work

Our model has demonstrated its efficacy; however, like most existing approaches, it also has some limitations. For example, we assume all observations are conditionally independent given the prototypes in Eq. (9). This assumption may not be appropriate in practical scenarios. Additionally, our method only addresses inter-class confusion issues and does not improve the decoder structure, resulting in somewhat rough segmentation results for object boundaries. Finally, our method cannot yet be extended to Transformer models because the KL divergence cannot constrain the variance to a reasonable range, leading to training instability and early collapse.

In future work, we will explore how to use multi-task learning (MTL) to stabilize pixel embedding modeling and extend it to a wider range of backbone architectures. Additionally, we will investigate decoder structures that facilitate fine-grained segmentation in the context of UDA.

7. Conclusion

In this paper, we propose PPPC, a universal probabilistic adaptation framework tailored for semantic segmentation, which successfully addresses the issue of ambiguous classes commonly existing in the self-training paradigm for UDA. Specifically, we model each pixel embedding as a multivariate Gaussian distribution, and prototypes are computed based on the observation of these embeddings, preserving both the semantic diversity for each class and the uncertainty for each embedding. Moreover, an effective and efficient metric for measuring the similarity between two distributions is adopted, which helps the model better deal with ambiguous embeddings. Additionally, a sampling strategy, AGC, which obtains class-wise uncertainty from online-updated prototypes, is employed to increase the number of ambiguous embeddings at the input level. Extensive experiments demonstrate the superiority of PPPC in both synthetic-to-real and day-to-night adaptation tasks.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 62276088 and 62102129.

References

[1] J. Janai, F. Güney, A. Behl, A. Geiger, et al., Computer vision for autonomous vehicles: Problems, datasets and state of the art, Foundations and Trends® in Computer Graphics and Vision 12 (1–3) (2020) 1–308.

[2] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.

[4] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 801–818.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2017) 834–848.

[6] L. Hoyer, D. Dai, L. Van Gool, Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9924–9935.

[7] A. Dundar, M.-Y. Liu, Z. Yu, T.-C. Wang, J. Zedlewski, J. Kautz, Domain stylization: A fast covariance matching framework towards domain adaptation, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (7) (2020) 2360–2372.

[8] Y. Yang, S. Soatto, Fda: Fourier domain adaptation for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4085–4095.

[9] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, T. Darrell, Cycada: Cycle-consistent adversarial domain adaptation, in: International Conference on Machine Learning, Pmlr, 2018, pp. 1989–1998.

[10] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, M. Chandraker, Learning to adapt structured output space for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7472–7481.

[11] T.-H. Vu, H. Jain, M. Bucher, M. Cord, P. Pérez, Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2517–2526.

[12] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, F. Wen, Prototypical pseudo label denoising and target structure learning for domain adaptive

semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12414–12424.

[13] N. Araslanov, S. Roth, Self-supervised augmentation consistency for adapting semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15384–15394.

[14] X. Wu, Z. Wu, H. Guo, L. Ju, S. Wang, Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15769–15778.

[15] C. Sakaridis, D. Dai, L. V. Gool, Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7374–7383.

[16] H. Wang, T. Shen, W. Zhang, L.-Y. Duan, T. Mei, Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 642–659.

[17] Y. Li, L. Yuan, N. Vasconcelos, Bidirectional learning for domain adaptation of semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6936–6945.

[18] T. Li, S. Roy, H. Zhou, H. Lu, S. Lathuilière, Contrast, stylize and adapt: Unsupervised contrastive learning framework for domain adaptive semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 4868–4878.

[19] T.-H. Vu, H. Jain, M. Bucher, M. Cord, P. Pérez, Dada: Depth-aware domain adaptation in semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7364–7373.

[20] Q. Wang, D. Dai, L. Hoyer, L. Van Gool, O. Fink, Domain adaptive semantic segmentation with self-supervised depth estimation, in: Proceed-

ings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8515–8525.

[21] B. Xie, S. Li, M. Li, C. H. Liu, G. Huang, G. Wang, Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).

[22] J. Hoffman, D. Wang, F. Yu, T. Darrell, Fcns in the wild: Pixel-level adversarial and constraint-based adaptation, arXiv preprint arXiv:1612.02649 (2016).

[23] D. Peng, P. Hu, Q. Ke, J. Liu, Diffusion-based image translation with label guidance for domain adaptive semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 808–820.

[24] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, Maximum classifier discrepancy for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3723–3732.

[25] K. Mei, C. Zhu, J. Zou, S. Zhang, Instance adaptive self-training for unsupervised domain adaptation, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 415–430.

[26] W. Zhang, W. Ouyang, W. Li, D. Xu, Collaborative and adversarial network for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3801–3809.

[27] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, T. Mei, Transferrable prototypical networks for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2239–2247.

[28] L. Zhou, S. Xiao, M. Ye, X. Zhu, S. Li, Adaptive mutual learning for unsupervised domain adaptation, IEEE Transactions on Circuits and Systems for Video Technology (2023).

[29] W. Tranheden, V. Olsson, J. Pinto, L. Svensson, Dacs: Domain adaptation via cross-domain mixed sampling, in: Proceedings of the

IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1379–1389.

[30] R. Li, S. Li, C. He, Y. Zhang, X. Jia, L. Zhang, Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11593–11603.

[31] L. Hoyer, D. Dai, L. Van Gool, Hrda: Context-aware high-resolution domain-adaptive semantic segmentation, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 372–391.

[32] L. Hoyer, D. Dai, H. Wang, L. Van Gool, Mic: Masked image consistency for context-enhanced domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 11721–11732.

[33] L. Vilnis, A. McCallum, Word representations via gaussian embedding, arXiv preprint arXiv:1412.6623 (2014).

[34] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[35] S. J. Oh, K. Murphy, J. Pan, J. Roth, F. Schroff, A. Gallagher, Modeling uncertainty with hedged instance embedding, arXiv preprint arXiv:1810.00319 (2018).

[36] S. Li, J. Xu, X. Xu, P. Shen, S. Li, B. Hooi, Spherical confidence learning for face recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15629–15637.

[37] M. Kirchhof, K. Roth, Z. Akata, E. Kasneci, A non-isotropic probabilistic take on proxy-based deep metric learning, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 435–454.

[38] Y. Shi, A. K. Jain, Probabilistic face embeddings, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6902–6911.

[39] S. Chun, S. J. Oh, R. S. De Rezende, Y. Kalantidis, D. Larlus, Probabilistic embeddings for cross-modal retrieval, in: Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8415–8424.

[40] A. Neculai, Y. Chen, Z. Akata, Probabilistic compositional embeddings for multimodal image retrieval, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4547–4557.

[41] H. Xie, C. Wang, M. Zheng, M. Dong, S. You, C. Fu, C. Xu, Boosting semi-supervised semantic segmentation with probabilistic representations, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 2938–2946.

[42] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 1735–1742.

[43] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.

[44] J. Huang, D. Guan, A. Xiao, S. Lu, L. Shao, Category contrast for unsupervised domain adaptation in visual tasks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1203–1214.

[45] Z. Jiang, Y. Li, C. Yang, P. Gao, Y. Wang, Y. Tai, C. Wang, Prototypical contrast adaptation for domain adaptive semantic segmentation, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 36–54.

[46] T. R. Scott, A. C. Gallagher, M. C. Mozer, von mises-fisher loss: An exploration of embedding geometries for supervised learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10612–10622.

[47] S. R. Richter, V. Vineet, S. Roth, V. Koltun, Playing for data: Ground truth from computer games, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 102–118.

[48] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A. M. Lopez, The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 3234–3243.

[49] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, T. Darrell, Bdd100k: A diverse driving dataset for heterogeneous multitask learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2636–2645.

[50] D. Dai, L. Van Gool, Dark model adaptation: Semantic image segmentation from daytime to nighttime, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018, pp. 3819–3824.

[51] Y. Cheng, F. Wei, J. Bao, D. Chen, W. Zhang, Adpl: Adaptive dual path learning for domain adaptation of semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).

[52] Q. Ren, Q. Mao, S. Lu, Prototypical bidirectional adaptation and learning for cross-domain semantic segmentation, IEEE Transactions on Multimedia (2023).

[53] D. Zhao, S. Wang, Q. Zang, D. Quan, X. Ye, R. Yang, L. Jiao, Learning pseudo-relations for cross-domain semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 19191–19203.

[54] T.-D. Truong, N. Le, B. Raj, J. Cothren, K. Luu, Fredom: Fairness domain adaptation approach to semantic scene understanding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19988–19997.

[55] F. Shen, A. Gurram, Z. Liu, H. Wang, A. Knoll, Diga: Distil to generalize and then adapt for domain adaptive semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15866–15877.

[56] Q. Fan, X. Shen, S. Ying, S. Du, Otclda: Optimal transport and contrastive learning for domain adaptive semantic segmentation, IEEE Transactions on Intelligent Transportation Systems (2024).

[57] X. Lai, Z. Tian, X. Xu, Y. Chen, S. Liu, H. Zhao, L. Wang, J. Jia, Decouplenet: Decoupled network for domain adaptive semantic segmentation, in: Proceedings of the European Conference on Computer Visionn, Springer, 2022, pp. 369–387.

[58] W. Liu, W. Li, J. Zhu, M. Cui, X. Xie, L. Zhang, Improving nighttime driving-scene segmentation via dual image-adaptive learnable filters, IEEE Transactions on Circuits and Systems for Video Technology (2023).

[59] F. Shen, Z. Pataki, A. Gurram, Z. Liu, H. Wang, A. Knoll, Loopda: Constructing self-loops to adapt nighttime semantic segmentation, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 3256–3266.

[60] X. Wu, Z. Wu, L. Ju, S. Wang, A one-stage domain adaptation network with image alignment for unsupervised nighttime semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (1) (2021) 58–72.

[61] H. Gao, J. Guo, G. Wang, Q. Zhang, Cross-domain correlation distillation for unsupervised domain adaptation in nighttime semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9913–9923.

[62] C. Sakaridis, D. Dai, L. Van Gool, Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (6) (2020) 3139–3153.