Contrastive Learning With Multiple Prototypes for Unsupervised Domain Adaptive Semantic Segmentation

Jun Yu, Guochen Xie, Quansheng Liu, Zhen Kan, *Senior Member, IEEE*, Lei Wang, Tianyu Liu, Qiang Ling, *Senior Member, IEEE*, Wei Xu, and Fang Gao, *Member, IEEE*

***Abstract*—Unsupervised domain adaptive semantic segmentation aims to transfer knowledge from the annotated source domain to the unlabeled target domain. Recently, self-training methods have gained substantial attention, which leverage high-confidence predictions in the target domain as pseudo labels for supervision. However, limited exploration of intra-class variations across domains, including significant visual differences within each category, has led to misalignment between feature distribution across domains. In this article, we present a unified non-parametric distance-based online clustering method to efficiently maintain multiple centroid-based prototypes within each category subspace instead of one prototype for each category subspace, which enables prototypes to possess the capacity for richer feature representation. Then, considering the variance across different dimensions of a feature representation, we then extend the prototypes from centroid-based ones to distribution-based ones. Specifically, each subspace is modeled using a Gaussian mixture model which includes several anisotropic Gaussian distributions, aimed at prioritizing discriminative dimensions and obtaining a finer measurement of the pixel-to-prototype similarity. Meanwhile, a category-aware feature space is achieved through pixel-to-prototype contrastive learning to ensure the compactness of pixel features in the same subcategory and drive the separation between pixel features of different subcategories. What's more, multi-resolution features are utilized to promote diversity and robustness among intra-class prototypes. Experiments validate the competitiveness of our two prototype-based methods against existing state-of-the-art methods, with a mIoU of 76.8% on GTA → Cityscapes, 68.4% on Synthia → Cityscapes, 54.5% on Cityscapes → DarkZurich and 56.4% on Cityscapes → ACDC. Notably, our method is able to seamlessly integrate with existing UDA methods.**

***Index Terms*—Unsupervised domain adaptation, semantic segmentation, prototype learning, contrastive learning.**

I. Introduction

SEMANTIC segmentation holds a pivotal position in computer vision, serving as a cornerstone for visual scene understanding. Its significance reverberates through a multitude of real-world applications, prominently manifesting its value in autonomous driving. With the advent of various Convolutional Neural Network (CNN) architectures [1], [2], and Vision Transformer-based models [3], remarkable advances in performance have been made in this domain.

The training of semantic segmentation models relies on the availability of adequate annotated images. Nonetheless, the manual annotation at the pixel level is a labor-intensive process, requiring up to 3 hours per image [4]. Consequently, creating a fully labeled dataset is a tedious process. Thus, researchers have turned to synthetic data [5], [6] to assist in training segmentation models. Pixel-level annotations can be automatically extracted from images generated by the video engine, greatly reducing the need for human interaction and thus increasing the scalability of annotated datasets. However, the distribution shift between synthetic and real data, which typically includes variations in image styles, weather, and lighting, makes the generalization of models from the synthetic domain to the real domain a challenging task. To close the domain gap and improve model performance on unlabeled target domains, Unsupervised Domain Adaptation (UDA) methods have attracted significant interest.

Previous research in UDA has concentrated on distributional alignment with adversarial training at the input [7], [8], feature [9], [10], and output level [11], [12]. Currently, self-training methods [13], [14], [15], [16] have received growing interest due to their promising performance. These methods utilize high-confidence target domain predictions as pseudo-labels, offering supervision for target domain adaptation. The generalization abilities of the model are closely linked to the accuracy of the pseudo-labels. Insufficient supervision and regularization in the target domain will lead to unreliable pseudo-labels, introducing additional noise during the training process and ultimately

Received 12 April 2024; revised 8 September 2024 and 2 November 2024; accepted 10 December 2024. Date of publication 17 February 2025; date of current version 27 August 2025. This work was supported in part by the Natural Science Foundation of China under Grant 62276242, in part by National Aviation Science Foundation under Grant 2022Z071078001, in part by the Dreams Foundation of Jianghuai Advance Technology Center under Grant 2023-ZM01Z001. The work of Fang Gao was supported by Guangxi Science and Technology Base and Talent Project under Grant 2020AC19253. The associate editor coordinating the review of this article and approving it for publication was Dr. Guosheng Lin. *(Jun Yu and Guochen Xie are co-first authors.) (Corresponding authors: Wei Xu; Fang Gao.)*

Jun Yu, Guochen Xie, Quansheng Liu, Zhen Kan, Lei Wang, and Qiang Ling are with the School of Information Science and Technology University of Science and Technology of China, Hefei 230026, China (e-mail: harryjun@ustc.edu.cn; xiegc@mail.ustc.edu.cn; liuqs29@mail.ustc.edu.cn; zkan@ustc.edu.cn; wangl@ustc.edu.cn; qling@ustc.edu.cn).

Tianyu Liu is with Jianghuai Advance Technology Center, Hefei 230088, China (e-mail: liutianyu18@mails.ucas.ac.cn).

Wei Xu is with The First Affiliated Hospital of USTC, Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei 230001, China (e-mail: xw199807@163.com).

Fang Gao is with the College of Electrical Engineering, Guangxi University, Nanning 530000, China (e-mail: fgao@gxu.edu.cn).

Digital Object Identifier 10.1109/TMM.2025.3543115

1520-9210 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

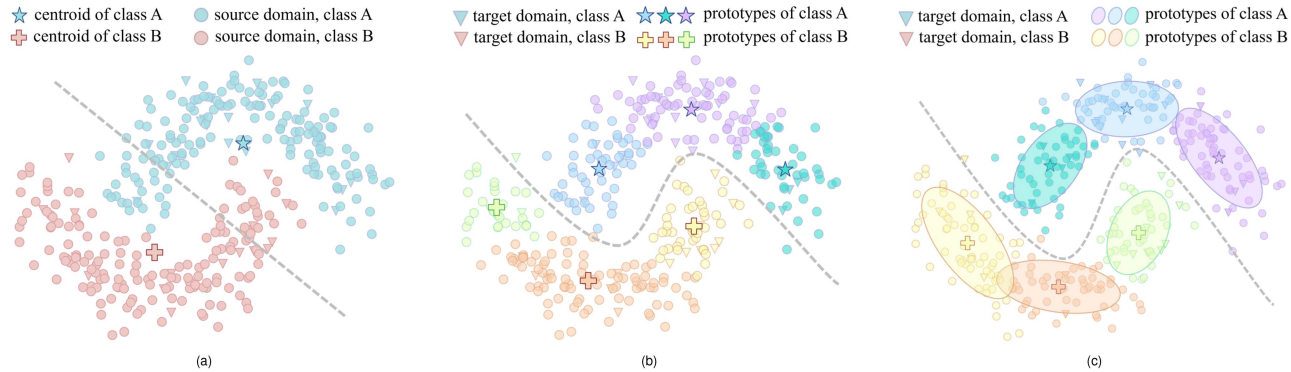


Fig. 1. Illustration of the main idea. (a) Single-centroid prototypes struggle to represent skewed distribution, resulting in the misclassification of outliers. (b) Centroid-based prototypes capture intra-class patterns within each subspace. (c) Modeling multiple Gaussian-based prototypes effectively captures variance on different dimensions for each category, leading to a more discriminative decision boundary.

leading to a degradation in overall performance. To tackle these challenges, researchers have adopted strategies such as dynamic confidence thresholds [17] and uncertainty estimation [18] for filtering out unreliable pseudo-labels. Some methods [19], [20], [21] compute the centroid of feature clusters as category prototypes and rectify pseudo-labels based on the feature-to-prototype distances.

From a prototype perspective, feature classification can be understood as the assignment of features to their most similar prototypes [22], with each prototype associated with a specific category. This clustering process can be achieved in an unsupervised fashion, which is particularly suitable for us to tackle the UDA problem, where labeled data are not always available. The definition of similarity between prototypes and features can vary depending on the particular type of prototype being used. As depicted in Fig. 1, samples with identical class labels but originating from different domains often display a scattered distribution within the feature space. This dispersion arises from variations within the class, which can lead to biased centroids. Consequently, outliers situated near the decision boundary present challenges for accurate classification. Hence, it becomes evident that a single prototype may not adequately represent the diverse features within a category. To address the category feature misalignment across domains, we advocate implementing contrastive learning in a multi-prototype way. Specifically, we first adopt nonparametric online clustering to mine multiple **centroid-based prototypes** within the domain-mixed feature space for each category, where features from both domains are associated with the same set of category prototypes. Online clustering provides scalability even when dealing with large volumes of data.

Meanwhile, probabilistic distribution-based clustering methods, such as Gaussian mixture model(GMM) [23] shows promising performance in the field of unsupervised learning. This allows us to extend the centroid-based prototypes to **distribution-based prototypes**, which enables a more comprehensive representation of feature distributions across domains. The distribution-based prototypes can also be clustered within a unified framework, ensuring methodological coherence. As shown in Fig. 1(c), distribution-based prototypes in the same category correspond to a distinct subspace, effectively capturing intra-class distribution patterns at a finer granularity. Subsequently, pixel-to-prototype contrastive learning is conducted in both domains to accurately represent category features and reduce confusion among the features of different categories, selecting positive and negative pairs according to the optimal prototype assignments. In this way, a regularized category-discriminative feature space is learned by promoting compactness among features from the same subcategory while driving a clear separation between those from other subcategories. Furthermore, to enhance supervision, we maintain the consistency between the predictions of the linear classifier and the prototype clustering.

To further enhance segmentation performance, we introduce multi-scale training to improve the model's ability to handle objects of varying scales. While the HR detail crop is effective for segmenting small objects like poles or distant pedestrians, it struggles to capture long-range dependencies, which is a drawback when segmenting larger areas, such as expansive sections of sidewalk. Conversely, the LR context crop excels in this regard. To address these limitations, we fuse the predictions from both crops to enhance the effectiveness of prototypes.

In this article, we offer a more comprehensive perspective on the use of prototypes in UDA. Specifically, this work introduces novel contributions in the following areas:

- We present a multiple-prototype contrastive learning module for UDA semantic segmentation, which aligns subcategory features by leveraging pixel-to-prototype assignments and leads to more reliable pseudo labels. Our plug-and-play module is compatible with most modern self-training UDA methods.
- To enhance pattern discrimination across diverse feature spaces and improve the clustering robustness, we start with centroid-based prototypes and then extend to distribution-based Gaussian prototypes. Moreover, clustering for both centroid-based and distribution-based prototypes can be achieved through our unified non-parametric cross-domain online clustering pipeline. The non-learnable prototypes make the clustering of prototypes computationally efficient.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

- Furthermore, we adopt a multi-resolution framework for category prototype clustering, allowing the category prototypes to acquire detailed information from high-resolution inputs as well as contextual information from low-resolution inputs, thereby enhancing the diversity and robustness of the prototypes.
- Extensive experiments conducted on popular UDA benchmarks show that our methods consistently attain superior performance and compatibility. Specifically, we achieve 76.8% and 68.4% mIoU on GTA → Cityscapes and Synthia → Cityscapes, respectively. Furthermore, we conduct experiments on two real-world benchmarks, i.e. Cityscapes → ACDC and Cityscapes → DarkZurich. Our method also obtains the results of 56.4% and 54.5%, respectively, which are favorable improvements on the baseline. A comprehensive ablation study confirms the effectiveness of each proposed component.

II. Related Work

A. Unsupervised Domain Adaptive Semantic Segmentation

Unsupervised domain adaptation (UDA) encourages a model to transfer valuable semantics from the labeled source domain to the unlabeled target domain, resulting in improved adaptation performance. Generally, the majority of UDA segmentation methods can be categorized into two groups: adversarial training [7], [24], [25], [26] and self-training [14], [16], [26], [27].

Adversarial training method bridges the gap between distinct domain distributions through different adversarial strategies. This is typically done by confusing domain discriminator at feature level [2], [10] or output level [10], [25], [28]. Alternatively, some methods use style transfer [8], [29], [30] to translate source domain images to the target domain while preserving the dense labels. However, the unstable nature of adversarial training results in suboptimal adaptation performance.

Self-training instead generates pseudo-labels from target domain predictions and integrates them into the training process. Pseudo labels is created either before adaptation [13], [19] or in real-time [14], [31]. Due to notable variations in data distributions between the two domains, pseudo labels unavoidably contain noise. To prevent pseudo-label drift, strategies including curriculum learning [32], [33], consistency regularization with augmented data [34], [35], entropy minimization [36], and depth estimation [37] are adopted. Furthermore, some approaches [19], [27], [38], [39] rectify pseudo-labels by utilizing the centroids of features within each category. However, these centroid-based methods fail to prioritize different feature dimensions.

In addition, most of the methods mentioned above overlook the variations in features within each category and consider each class as a uniform entity. This over-simplified assumption leads to a less discriminative feature space.

B. Prototype Learning

Prototype learning enables the unsupervised categorization of samples based on their similarities. Prominent methods include distance-based hard-clustering approaches such as nearest neighbors [22], K-Means [40], and probabilistic-based soft-clustering methods like GMM [41]. The prototype, whether represented as a centroid or distribution, captures the shared representative patterns of features within the same cluster. Thus, prototype-based methods are naturally introduced to tackle the unsupervised domain adaption problem because of its representativeness. [42] formulates the unsupervised domain adaption task as the alignment of source and target prototypes. [43] presents Transferrable Prototypical Networks (TPN) to ensure that the prototypes for each class in source and target domains are close in the embedding space. Also, the score distributions predicted by prototypes separately on source and target data need to be similar. [44] proposes a bidirectional memorization mechanism that learns to remember useful and representative information to purify noisy pseudo labels on the fly for robust black-box unsupervised domain adaption. [45] proposes the Prototype-Guided Feature Learning (PGFL) method to learn domain-invariant features as well as reduce the negative effect of mislabeled samples.

Prior research [1], [2], [3] has demonstrated the effectiveness of deep neural networks in learning dense image representations. By utilizing the deep dense representation extraction by deep neural models, prototype networks excel in a variety of tasks, delivering impressive performance in few-shot [46] and zero-shot [47] learning. Recent progress in semantic segmentation, including fully-supervised [48], semi-supervised [49], and unsupervised [27], [50] has underscored the prospect of prototype networks in dense classification tasks.

However, many existing works either concentrate on intra-domain clustering [51] or rely on a single prototype to characterize an entire category space [27], thereby overlooking intra-class variance. In addition, most prototypes [19] are distance-based and isotropic, treating distances uniformly in all directions. When sample distributions are skewed, such prototypes are less effective, as they fail to adapt to the imbalanced feature distribution. In this article, we introduce an efficient online clustering method to maintain multiple prototypes for features in each category, with both centroid-based and distribution-based prototypes considered, thereby improving the characterization of the category subspace.

C. Contrastive Learning

Contrastive learning [52], [53], [54] plays a crucial role in self-supervised or unsupervised learning. It prioritizes the acquisition of distinctive feature representations by bringing positive samples closer together in the feature space while pushing negative samples farther apart.

In recent times, pixel-wise contrastive learning has emerged as a promising technique for improving performance in semantic segmentation tasks. For instance, ReCo [55] utilizes hard negative pixels to facilitate contrastive learning. Meanwhile, ProCA [17] define feature centroids as prototypes for computing the intra-domain contrast loss function. Multiple prototypes have been shown to boost the robustness of class representations against intra-class variance in prior research [51]. Yet, despite their effectiveness, most existing methods demand a

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

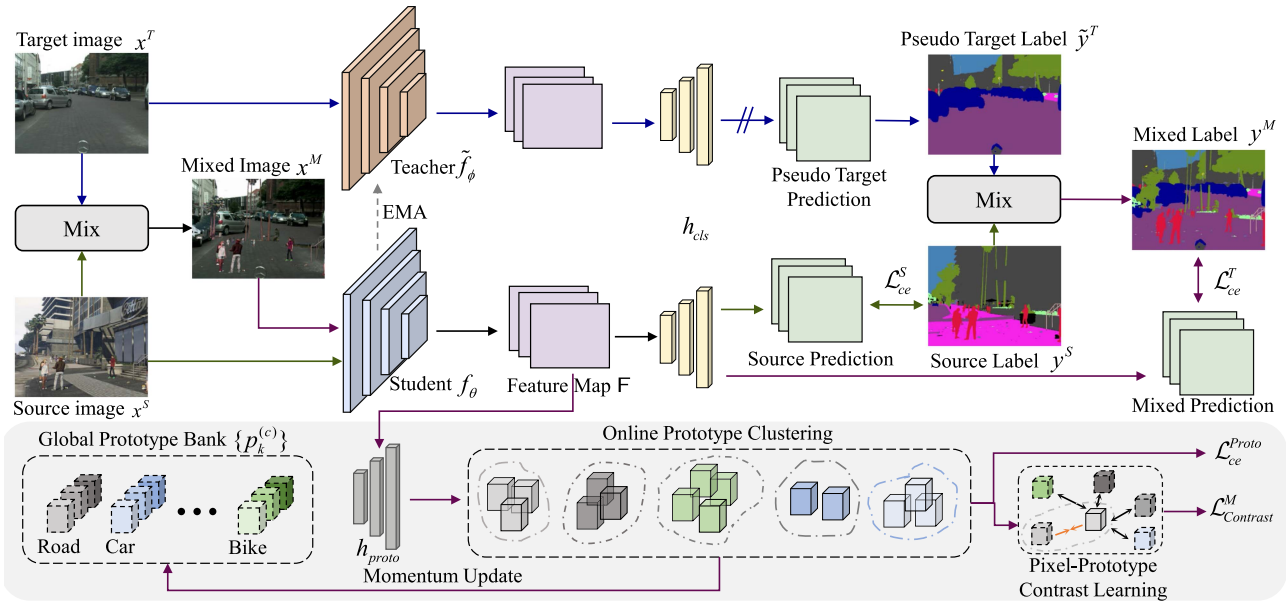


Fig. 2. Framework Overview. The network is trained with a supervised segmentation loss $\mathcal{L}_{ce}^S$ on the source domain (green) and an unsupervised adaptation loss $\mathcal{L}_{ce}^T$ on the mixed target domain (purple). Then, a multi-prototype contrastive learning module(MPCL, grey) is introduced. Both centroid-based and distribution-based prototypes are considered. Non-parametric intra-class clustering is conducted online, where pixel-to-prototype similarity is utilized to determine prototype assignments. The pixel-to-prototype contrastive loss $\mathcal{L}_{Contrast}^M$ and consistency loss $\mathcal{L}_{ce}^{Proto}$ are introduced to refine the shared feature space.

considerable number of pixel-to-pixel comparisons during the sampling stage. In contrast to prior works, our approach utilizes pixel-to-prototype contrastive learning at a more granular level. Through online clustering, each prototype can be considered a representative of feature samples within the corresponding sub-category space, and constructing positive-negative sample pairs between pixels and samples reduces the burden of inter-pixel sampling.

III. Methodology

A. Background

During UDA training, as shown in Fig. 2, the model has access to labeled source domain $\mathcal{S} = \{(x_i^S, y_i^S)\}_{i=1}^{N^S}$ and unlabeled target domain $\mathcal{T} = \{x_i^T\}_{i=1}^{N^T}$. Both domains share the same set of labels $\{y^T\} = \{y^S\} = \{1, 2, \cdots, C\}$.

The goal is to train a model that maps pixels in target image X^T to its real label Y^T, with the supervision obtained from both the target and source domains. Given the availability of source labels Y^S, the model can be effectively trained with the fully supervised categorical cross-entropy loss on the source domain:

$$\mathcal{L}_{ce}^S = \frac{-1}{H \times W} \sum_{i=1}^{H \times W} \sum_{j}^{C} y_{(i,j)}^S log(h_{cls}(f_\theta(x^S))_{(i,j)}) \quad (1)$$

where f_θ is the feature extractor of student model and h_{cls} is a MLP classifier that generate category predictions $P^{H \times W \times C}$ based on feature embeddings $\mathcal{F}^{H \times W \times D}$. D represents the channel of the feature.

Relying solely on supervision from the source domain proves inadequate for effective adaptation to the target domain. Following [16], self-training method with online-updating pseudo-labels $\tilde{Y}^T = \{\tilde{y}_i^T\}_i^{N^T}$ is employed. To maintain the stability of pseudo-label generation, pseudo-labels prediction are screened by a dynamic confidence threshold.

During each iteration t, the non-learnable teacher model is updated with the exponential moving average (EMA) of the student model's weights:

$$\phi^t \leftarrow \alpha \phi^{t-1} + (1 - \alpha)\theta^t \quad (2)$$

To bridge the domain gap in UDA, data mixing [14] is commonly employed during training, where half of the categories' pixels on the source image are masked, and the rest of the pixels are pasted onto the target domain image based on a random pixel mask M. The mixed-images is defined as $x^M = (\mathbf{1} - M) \odot x^T + M \odot x^S$. A similar mixing strategy is applied at the output level.

Therefore, the training objective for the target domain is given as:

$$\mathcal{L}_{ce}^T = \frac{-1}{H \times W} \sum_{i=1}^{H \times W} \sum_{j}^{C} y_{(i,j)}^M log(h_{cls}(f_\theta(x^M))_{(i,j)}) \quad (3)$$

The overall training objective of self-training is given as,

$$\mathcal{L}_{seg} = \mathcal{L}_{ce}^S + \mathcal{L}_{ce}^T \quad (4)$$

B. Prototype-Based Contrastive Learning

As seen in Fig. 3(a), without any regularization, features from the same class but originating from distinct domains are likely to be scattered in the shared feature space. This is mainly a result of neglecting intra-class diversity. To deal with this challenge, we introduce a multi-prototype contrastive learning (MPCL) method that refines the feature distribution by utilizing unsupervised clustering to explore distribution structure. In contrast to previous approaches where a single prototype is utilized to

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

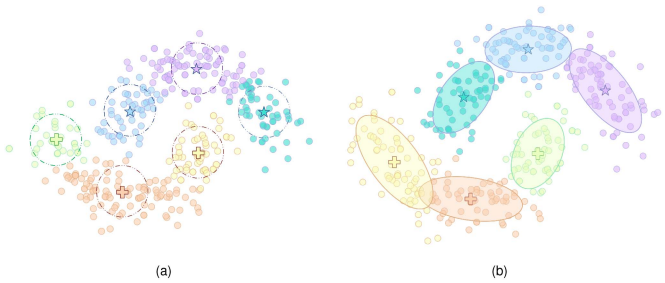


Fig. 3. Illustration of different choice of prototypes (a) Centroid-based prototypes treat features in each dimension equally and utilize the Euclidean distance metric; (b) Gaussian-distribution based prototypes where feature variance in each dimension is considered.

characterize each category feature space [19], [38], our perspective differs as we propose the adoption of multiple prototypes to represent each category.

We employ online clustering within the feature space of each category to obtain unbiased category prototypes. Essentially, this process can be viewed as assigning pixels of the same class to a distinct set of K prototypes created for that category through clustering. This clustering approach enables the model to explore feature distributions across domains, facilitating the identification of unique intra-class patterns and learning a robust representation.

Definition of Prototypes: Different types of prototypes correspond to different clustering methods. Distance-based clustering algorithms, such as k-means, are straightforward and use the spatial distance between feature embeddings and cluster centroids as the clustering criterion in an intuitive manner. Meanwhile, distribution-based clustering, represented by GMM, approaches the task with a probabilistic perspective. Feature points are generated independently from separate distributions. Consequently, each cluster's data points can be represented by this distribution.

As shown in Fig. 3, we establish different forms of prototypes for feature clustering within a unified framework. First, we will present the definitions of the two types of prototypes.

Centroid-Based Prototypes: For centroid-based clustering, each prototype, in theory, is identical to the centroid of its respective cluster, representing the features within the corresponding subspace. Specifically, for class $c \in \{1, 2, \cdots, C\}$, the category feature space is being clustered into K clusters, with the k-th cluster characterized by a centroid $p_k^{(c)}$

$$p_k^{(c)} = \mu_k^{(c)} \tag{5}$$

Here, $\mu_k^{(c)}$ denotes the mean vector of features within the subspace. As a result, a total of $C \times K$ prototypes $\mathcal{P} = \{p_k^{(c)}\}_{c,k=1}^{C,K}$ are generated for all categories.

Distribution-Based Prototypes: In the context of domain adaptation, centroid-based prototypes determine similarity based on pixel-to-prototype distances. However, with high-dimensional, sparse features, distance metrics struggle to prioritize different dimensions effectively, leading to a degradation in clustering performance. In contrast, distribution-based clustering methods, like the Gaussian mixture model (GMM) [23], introduce variance across feature dimensions that can prioritize the discriminative dimensions while suppressing the unrelated ones, making it better suited for high-dimensional, sparse feature scenarios.

The GMM method can be viewed as the weighted sum of K multivariate Gaussian distribution:

$$p^{(c)}(x) = \sum_{k=1}^{K} \pi_{c,k} \mathcal{N}(x|\mu_k^{(c)}, \Sigma_k^{(c)}) \tag{6}$$

where x is the data points with d dimensions and $\pi_{c,k}$ means the weight. Inspired by the fact that the Gaussian mixture model can fit arbitrary distributions, we represent each category feature space with a unique Gaussian mixture model, each with K components. Similar to the centroid-based prototype, we can view each component $p_k^{(c)}$ as a distribution-based prototype.

$$\begin{aligned} p_k^{(c)}(x) &= \mathcal{N}(x|\mu_k^{(c)}, \Sigma_k^{(c)}) \\ &= \frac{1}{\sqrt{(2\pi^d)|\Sigma_k^{(c)}|}} e^{-\frac{1}{2}(x-\mu_k^{(c)})^T {\Sigma_k^{(c)}}^{-1}(x-\mu_k^{(c)})} \end{aligned} \tag{7}$$

Here, each prototype is characterized by the mean vector $\mu_k^{(c)}$ and covariance matrix $\Sigma_k^{(c)}$.

To further express our motivation in real senarios, we extarct features from the baseline model [16] and then conduct the centroid-based and distribution-based clsutering on them. The final results are shown in Fig. 4. In Fig. 4, the geometric center of the feature distribution is adopted by the single-centroid method (subfigure (a)). But the geometric center fails to represent complex distributions, which can easily lead to erroneous classification boundaries. The centroid-based prototype method (subfigure (b)) assume equal weights in all directions, resulting in a rough depiction of boundaries in complex distributions. This might lead to situations where the distance to the central points of subclasses from other classes is very small, as highlighted by the dashed box. In contrast, the distribution-based prototype method (subfigure (c)), by flexibly considering different directions, has significant advantages in modeling distributions. This allows subclass centers of different classes to maintain a reasonable distance at the boundaries, making the boundaries clear and discriminative.

Prototype-Category Prediction: For centroid-based clustering, the feature-to-prototype similarity is measured by the Euclidean distance between the pixel embedding vector and the centroid prototypes. The similarity function sim is defined as:

$$sim(p_k^{(c)}, \mathcal{F}_i^{(c)}) = -||p^{(c)} - \mathcal{F}_i^{(c)}||_2 \tag{8}$$

For Gaussian-based clustering, sim is defined as the log-likelihood of the distribution prototype $p_k^{(c)}$:

$$sim(p_k^{(c)}, \mathcal{F}_i^{(c)}) = log\mathcal{N}(\mathcal{F}_i^{(c)}|\mu_k^{(c)}, \Sigma_k^{(c)}) \tag{9}$$

where $\mu_k^{(c)}$ and $\Sigma_k^{(c)}$ are the mean vector and covariance matrix that get updated during the training iteration. In practice, target domain features are grouped based on pseudo labels. The

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

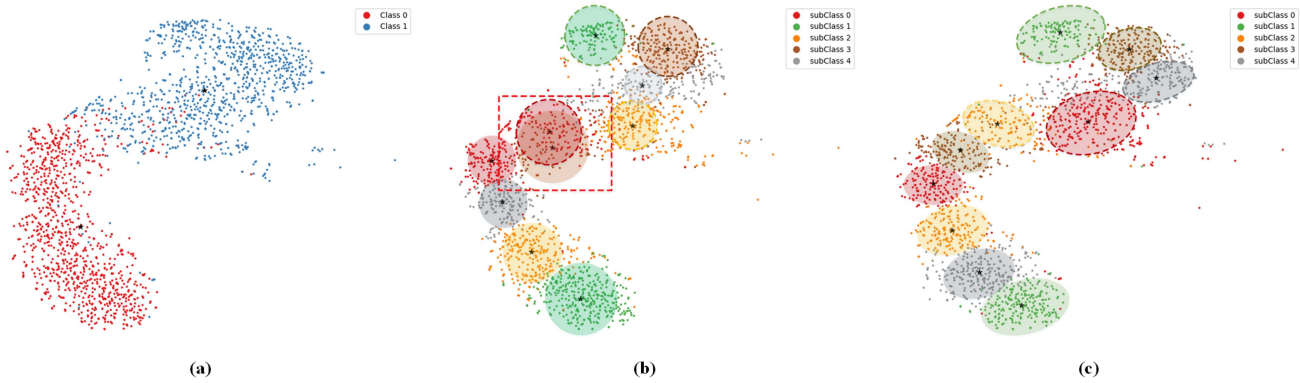


Fig. 4. The representational capacity of three methods for feature distribution in real scenarios. Each point in the figure represents a feature. Subfigure (a), (b) and (c) show the single-centroid method [16], centroid-based prototype method and distribution-based prototype method respectively. The subfigure (a) illustrates two different classes, while the subfigure (b) and (c) show 5 distinct subclasses of each class. Black stars illustrate the center of classes in subfigure (a) and subclasses in subfigure (b) and (c).

category prediction y^{proto} of the features in prototype space is determined by identifying the prototype among $C \times K$ prototypes that is most similar to the feature $\mathcal{F}_i$:

$$y^{proto} = c', \text{ where } (c', k') = \arg\max_{c,k} \{sim(p_k^{(c)}, \mathcal{F}_i)\}_{c,k=1}^{C,K} \tag{10}$$

where sim denotes the prototype similarity metric and $\mathcal{F}_i \in \mathrm{R}^D$ is the pixel embeddings in prototype feature space. For simplicity in notation, we will no longer distinguish between features extracted from source images and mixed images.

In formal terms, we can define the posterior probability that a pixel $\mathcal{F}_i$ being assigned to category c as follows:

$$\mathbb{P}_{c,i}^{proto} = \mathbb{P}(y^{proto} = c|\mathcal{F}_i) = \frac{\exp(g(c, \mathcal{F}_i)/\tau)}{\sum_{j=1}^{C} \exp(g(j, \mathcal{F}_i)/\tau)} \tag{11}$$

where $g(c, \mathcal{F}_i)$ represents the similarity between a pixel $\mathcal{F}_i$ and a class c, which is calculated as the maximal similarity between the K prototypes of class C and the embedding $\mathcal{F}_i : g(c, \mathcal{F}_i) = \max_k \{sim(p_k^{(c)}, \mathcal{F}_i)\}$.

To enforce consistency between the linear classifier and the prototype branch at feature level, the cross-entropy loss based on prototypical class predictions is introduced:

$$\mathcal{L}_{ce}^{Proto} = \frac{-1}{H \times W} \sum_{i=1}^{H \times W} \sum_{j}^{C} \bar{y}_{(i,j)}^{M} log(\mathbb{P}_{j,i}^{proto}) \tag{12}$$

where $\bar{y}_{(i,j)}^{M}$ represents the labels of the mixed images downsampled to the resolution of prototype feature embeddings. Here, we use the mixed images as they contain features from both domains and inherently facilitate the alignment of shared prototype features across domains.

Cross Domain Prototype Online Clustering: The GMM method, as a parameterized probabilistic model, is typically solved using the EM algorithm. This iterative approach initially estimates latent variables with observed value and then employs these estimates to update model parameters. However, for segmentation tasks, the immense volume of pixel-level data significantly increases the computational cost of iterative calculations. To ensure the scalability of our approach across different datasets, we introduce an online clustering method that dynamically assigns feature pixels from the current batch to their most similar prototypes. Drawing inspiration from [56], we adopt a unified clustering strategy for the two types of proposed prototypes.

Taking category c as an example, we denote the pixel features belonging to class c in the current batch as $\mathcal{F}^{(c)}$, and L is total number of pixels in the current batch, and the prototypes established within category c as $\mathcal{P}^{(c)} = \{p_k^{(c)}\}_{k=1}^{K}$. We can view the clustering process as an optimal matching problem, with the goal of finding the optimal assignment matrix $Q_c \in \{0, 1\}^{L \times K}$, which maximizes the overall similarity between pixels and prototypes. Here, $S_c \in \mathbb{R}^{L \times K}$ is the similarity

Algorithm 1: Overview of Proposed Method.

Input: X^S: Source domain data, Y^S: Source domain labels; X^T: Target domain data; T: Total training iterations; K: Number of prototypes per category.

Output: Teacher segmentation network $\tilde{f}_\phi$;

Initialize teacher and student model with pretrained weights and initialize h_{cls} , h_{proto} randomly;
Initialize prototype $\mathcal{P}$ randomly.
for *iter* $t = 1$ ***to*** T; **do**
  Sample source labels, souce images and target images: $x^S, y^S \sim X^S, Y^S$; $x^T \sim X^T$;
  Compute feature embeddings $\mathcal{F}$;
  Compute confidence weight w^T in Eq. (16);
  Perform multiple-prototypes online clustering to get Q_c^* in Eq. (14);
  Update prototypes statistic μ or Σ according to clustering assignments Q_c^*;
  Compute pixel-to-prototype comtrastive loss $\mathcal{L}_{Contrast}^{M}$ in Eq. (17);
  Compute Total Loss L_{tot} in Eq. (19);
  Update student model f_θ with backpropagation;
  Momentum update teacher model $\tilde{f}_\phi$;

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

matrix between prototypes and pixel feature embeddings, denoted as $S_c = sim(\mathcal{P}^{(c)}, \mathcal{F}^{(c)})$.

As indicated in [23], the clustering procedure can be perceived as maximizing the similarity between feature vectors and their corresponding prototypes under the equipartition constraint. We introduce the prior assumption that samples are evenly distributed among prototypes, with each sample assigned to only one prototype. Thus, we can formulate it as an optimization problem:

$$\max_{Q_c \in \tilde{\mathcal{Q}}_c} \left(\sum_{i,j}^{K,L} Q_c \odot S_c \right)$$
$$\tilde{\mathcal{Q}}_c = \{Q_c \in \{0,1\}^{K \times L}, Q_c^\top \mathbf{1}^K = \mathbf{1}^L, Q_c \mathbf{1}^L = \frac{L}{K}\mathbf{1}^K\} \tag{13}$$

and the constraint specified in $\tilde{\mathcal{Q}}_c$ aligns with all the desired prior assumption. Here, $\mathbf{1}^L$ represents a vector of ones with L dimensions. If we consider similarity as the negative transportation cost, maximizing overall similarities is equivalent to minimizing the cost of association between prototypes and pixel embeddings. Consequently, the clustering process can be transformed into solving an optimal transport problem [56]. In practice, the assignment matrix in (13) can be relaxed to $Q_c \in \mathbb{R}_+^{K \times L}$, meanwhile, by introducing an extra entropy $H(Q_c) = \Sigma_{L,k} - q_{i_L,k} \log(q_{i_L,k})$ the optimal solution is then determined:

$$Q_c^* = \mathbf{D}(u) \exp\left(\frac{S_c^\top}{\epsilon}\right) \mathbf{D}(v) \tag{14}$$

where $\epsilon > 0$ is the weight that regularizes the prototypes distribution, $\mathbf{D}(x)$ denotes the diagonal matrix with $(\mathbf{D}(x))_{ii} = x_i$, $u \in \mathbb{R}^K$ and $v \in \mathbb{R}^L$ are scaling vectors which are obtained through several iterations of the Greenkhorn algorithm [57] on modern GPU. In this way, both the centroid-based and Gaussian-based clustering can be efficiently solved, even when dealing with large-scale pixel-level datasets. It's worth noting that [27] also formulate clustering process as an optimal transport problem. However, we still differs in two aspects. First, [27] and our work have different inputs for clustering. In [27], different clusters correspond to different categories, while in our work, the input of features belong to the same category, and the purpose of clustering is to find the subspace in each category. Second, [27] and our work differ in similarity computation method. In [27], the similarity are computed with self-labeling head. However, in our work, we compute the similarity of all the prototypes in one category and then take the largest as the similarity for clustering.

Pixel-to-Prototype Contrast Learning: Rather than following the traditional pixel-to-pixel contrast, we conduct contrastive learning between pixels and prototypes.

Specifically, we start by aligning the original pixel feature to the prototype feature space through projection layer h_{proto}, yielding in $\mathcal{F}$. Then, given the optimal assignment matrix Q_c^*, the pixel-to-prototype assignments can be defined as $\{(\mathcal{F}_i, p_{k_i}^{(c_i)})\}$, where $c_i \in \{1, 2, \cdots, C\}, k_i \in \{1, 2, \cdots, K\}$. In this way, each pixel $\mathcal{F}_i$ is now associated with $p_{k_i}^{(c_i)}$, the k_i-th prototype of the c_i-th class.

The assignments allow us to establish positive and negative sample pairs. In the contrastive learning module, $p^+ = p_{k_i}^{(c_i)}$ serves as the only positive prototype sample for pixel $\mathcal{F}_i$. Meanwhile, the negative prototypes consist of the other $C \times K - 1$ prototypes, denoted as $\mathcal{P}^- = \mathcal{P}/p_{k_i}^{(c_i)}$.

In the source domain, we can define the contrastive loss as:

$$\mathcal{L}_{Contrast} = \frac{1}{H \times W} \sum_{i=1}^{H \times W} -log \frac{\exp(sim(\mathcal{F}_i, p^+)/T)}{\exp(sim(\mathcal{F}_i, p^+)/\tau) + \sum_{p^- \in \mathcal{P}^-} \exp(sim(\mathcal{F}_i, p^-)/\tau)} \tag{15}$$

where τ is the temperature coefficient, and the similarity metric sim is defined differently depending on the type of prototype being used. (15) could be interpreted as a mechanism to encourage that the pixel features $\mathcal{F}_i$ be similar to its associated prototype $p_{k_i}^{(c_i)}$, and dissimilar to the unassociated prototypes $\mathcal{P}^-$, thus preserving the intra-class feature variance. In this context, only $C \times K$ prototypes are needed to calculate pixel-to-prototype contrastive loss, which requires significantly less computation and memory for pair sampling compared to pixel-to-pixel contrast learning.

At the early stages of training, pseudo-labels contain a considerable amount of noise. For training stability, prior methods like [48] introduced a warm-up stage to start modeling prototypes after specific iterations. Instead, we employ a confidence-weighted approach based on the confidence estimation of pseudo labels. Following [16], we calculate it as the proportion of pixels with a maximum probability prediction above a threshold θ. Thus, θ is able to control the number of pixels for mix training

$$w^T = \frac{\sum_{i=1}^{H \times W} [\max_c(h_{cls}(\tilde{f}_\phi(x_i^T)_c)) > \theta]}{H \times W} \tag{16}$$

which allows for the adaptive introduction of the prototype contrastive loss from the very beginning of training. In mixed-domain images, the adaptive mask is formulated as $w^M = w^T \odot (\mathbf{1} - M) + \mathbf{1} \odot M$, with M representing the binary mask indicating the pixels to be copied.

In the mixed-domain, the contrastive learning is conducted through the calculation of:

$$\mathcal{L}_{Contrast}^M = \frac{1}{H \times W} \sum_{i=1}^{H \times W} -w_i^M log \frac{\exp(sim(\mathcal{F}_i^M, p_{k_i}^{(c_i)})/\tau)}{\sum_{c,j=1} \exp(sim(\mathcal{F}_i^M, p_j^{(c)})/\tau)} \tag{17}$$

Here, each prototype $p_j^{(c)} \in \mathcal{P}$ is shared by features across domains.

Multi-resolution Prototype Learning and Total Loss: HRDA [31] is currently one of the most widely adopted UDA methods, utilizing both high-resolution image crops and low-resolution crops as inputs to enable the model to make more accurate segmentation predictions based on information captured at different resolutions.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

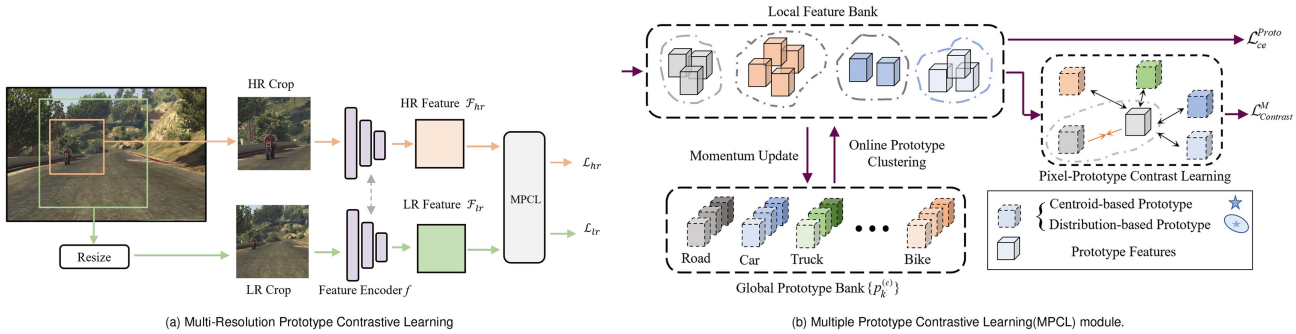


Fig. 5. Illustration of multi-resolution prototype contrastive learning method. The subfigure(a) shows the pipeline of which use cropped images of different scales and MPCL module in training, which enable the model to process image of different scales. The subfigure(b) introduces the detail of the MPCL Module, which consists of global and local feature banks and adopt contrastive learning for training.

In the context of multi-prototype clustering, our goal is to enhance prototype feature diversity within the same category. Accordingly, we extend upon HRDA [31] by integrating multi-resolution features into prototype clustering and contrastive learning. Specifically, we randomly extract high-resolution crops from low-resolution ones. As illustrated in Fig. 5(a), the detailed features from the HR crop, as well as the broader contextual features from the original LR crop, can complement each other, resulting in increased diversity and robustness among the prototypes generated through clustering.

Specifically, for x_{hr} and x_{lr}, the extracted features $\mathcal{F}_{hr}$ and $\mathcal{F}_{lr}$ are individually fed into the Multiple Prototype Contrastive Learning (MPCL) module, as shown in Fig. 5(b), where separate clustering processes are performed, and prototypes are updated accordingly. Then, following (17) and (12), prototype losses $\mathcal{L}^{proto}(x_{hr})$ and $\mathcal{L}^{proto}(x_{lr})$ are obtained for different resolution features. The combined multi-resolution prototype loss $\mathcal{L}_{tot}^{proto}$ can then be given as:

$$\begin{aligned}\mathcal{L}_{tot}^{proto} &= (1-\lambda_d)\mathcal{L}^{proto}(x_{lr}) + \lambda_d \mathcal{L}^{proto}(x_{hr}) \\ \mathcal{L}^{proto} &= \lambda_1 \mathcal{L}_{Contrast}^{M} + \lambda_2 \mathcal{L}_{ce}^{Proto}\end{aligned} \tag{18}$$

where λ_1 and λ_2 are weights of contrastive loss $\mathcal{L}_{Contrast}^{M}$ and consistency loss $\mathcal{L}_{ce}^{Proto}$. Similarly, λ_d weights the importance of x_{hr} and x_{lr}. Combined with the self-training framework, the total training objective of our method is defined as:

$$\mathcal{L}_{tot} = \mathcal{L}_{ce}^{S} + \mathcal{L}_{ce}^{T} + \mathcal{L}_{tot}^{proto} \tag{19}$$

Prototypes Initialization and Update: The initialization of prototypes for each category involves clustering on randomly sampled pixel features. At the early stage of training, a feature bank is set up to gather features from different categories. Then, K clusters are initialized within each category subspace via K-Means [40]. The initial prototype statistics are calculated as the mean and covariance of category features.

In our model, the non-learnable prototypes are dynamic updates based on assignments obtained from prototype learning rather than learning through gradient descent. To be more precise, the centroid-based prototypes are updated in each iteration following:

$$p_k^{(c)} \leftarrow \alpha p_k^{(c)} + (1-\alpha)\bar{\mathcal{F}}_k^{(c)} \tag{20}$$

where α is the momentum parameter for EMA update. and $\bar{\mathcal{F}}_k^{(c)}$ represents the mean vector of pixel feature embeddings with each embedding $\mathcal{F}$ assigned to prototype $p_k^{(c)}$.

As for the distribution-based prototypes $p_k^{(c)}(\mu_k, \Sigma_k)$, it is updated according to:

$$\begin{aligned}\mu_k^{(c)} &\leftarrow \alpha \mu_k^{(c)} + (1-\alpha)\bar{\mu}_k^{(c)} \\ \Sigma_k^{(c)} &\leftarrow \alpha \Sigma_k^{(c)} + (1-\alpha)\bar{\Sigma}_k^{(c)}\end{aligned} \tag{21}$$

where $\bar{\Sigma}_k^{(c)}, \bar{\mu}_k^{(c)}$ is the estimated mean and covariance of features embeddings within the cluster of Gaussian-based prototype $p_k^{(c)}$.

IV. Experiments

A. Datasets

Firstly, our method is evaluated on four commonly adopted UDA semantic segmentation benchmarks, GTA $\rightarrow$ Cityscapes, Synthia $\rightarrow$ Cityscapes, Cityscapes $\rightarrow$ DarkZurich and Cityscapes $\rightarrow$ACDC.

Cityscapes [62] consists of 2,975 and 500 images for the training and validation set, respectively, all captured at a resolution of $2,048 \times 1,024$. It focuses on European city scenes and offers pixel-level annotations for 19 semantic classes, making it a crucial resource for urban scene analysis in computer vision.

GTA [5] consists of 24966 synthetic images of American-style streets created by a video game engine. These images depict virtual American-style city streets and include 19 semantic classes consistent with the categories within the Cityscapes dataset.

SYNTHIA [6] is a urban scene dataset consisting of synthetic images. It comprises a collection of 9400 images meticulously crafted to resemble urban environments. These images offer pixel-level annotations for 16 semantic classes, making them a valuable resource for training and assessing models in the context of synthetic urban scenes.

DarkZurich [63]. is a real-world dataset comprising 2416 nighttime images, 2920 twilight images, and 3041 daytime images, all with a resolution of 1920×1080. In our analysis, we utilize 2416 day-night image pairs as the primary training data and an additional 151 images for testing purposes.

ACDC [4] dataset encompasses the same semantic classes as the Cityscapes dataset and is collected under four distinct

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON OF UDA SEGMENTATION PERFORMANCE (MIOU,%) ON GTA → CITYSCAPES

Method	Road	S.walk	Build.	Wall	Fence	Pole	T.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
Source Only	71.5	18.0	84.2	34.4	30.9	33.4	44.3	23.5	87.4	41.3	86.6	64.0	22.5	88.3	44.5	39.1	2.3	35.2	31.6	46.5
AdaptSegNet [11]	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
AdvEnt [25]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5
CyCADA [7]	86.7	35.6	80.1	19.8	17.5	38	39.9	41.5	82.7	27.9	73.6	64.9	19.0	65	12	28.6	4.5	31.1	42.0	42.7
CLAN [58]	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
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
Uncertainty [18]	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 [8]	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
Adaboost [59]	90.7	35.9	85.7	40.1	27.8	39.0	49.0	48.4	85.9	35.1	85.1	63.1	34.4	86.8	38.3	49.5	0.2	26.5	45.3	50.9
DACS [14]	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	52.1
BAPA [60]	94.4	61.0	88.0	26.8	39.9	38.3	46.1	55.3	87.8	46.1	89.4	68.8	40.0	90.2	60.4	59.0	0.0	45.1	54.2	57.4
ProDA [19]	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
ProCA [17]	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
CAlign [20]	94.8	66.6	87.7	41.5	26.1	45.2	48.1	53.5	85.7	32.7	88.2	70.9	36.0	87.9	41.9	60.8	33.5	44.6	62.5	58.3
DAFormer [16]	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	65.1	55.9	61.8	68.3
DAFormer+Centroid	96.8	75.8	90.1	56.8	53.0	54.1	58.0	64.5	89.3	43.8	93.4	72.3	40.3	93.8	80.9	87.3	78.3	58.4	67.2	71.3
DAFormer+Gaussian	96.9	77.1	90.0	56.9	53.3	52.9	61.4	62.8	90.1	48.2	91.8	75.8	47.3	93.2	79.4	84.6	76.0	61.1	62.2	71.6
HRDA [31]	96.4	74.4	91.0	61.6	51.5	57.1	63.9	69.3	91.3	48.4	94.2	79.0	52.9	93.9	84.1	85.7	75.9	63.9	67.5	73.8
HRDA+Centroid	97.4	80.1	90.9	58.8	55.1	56.2	66.1	70.2	91.6	50.1	94.4	81.1	**59.6**	94.8	85.1	87.4	75.7	**66.3**	66.5	75.1
HRDA+Gaussian	97.2	79.3	91.6	61.5	57.7	60.4	61.5	72.5	91.3	47.8	94.9	79.8	54.7	94.5	84.0	89.2	80.7	64.1	68.4	75.3
MIC [35]	97.4	80.1	91.7	61.2	56.9	59.7	66.0	71.3	91.7	51.4	94.3	79.8	56.1	94.6	85.4	90.3	80.4	64.5	68.5	75.9
MIC+Centroid	**97.4**	**80.1**	91.7	**64.3**	57.9	62.1	64.8	70.9	**91.7**	**51.5**	94.5	80.2	55.6	94.9	87.6	91.3	80.2	65.5	68.7	76.4
MIC+Gaussian	97.2	79.0	**92.1**	61.7	**60.7**	**62.2**	**66.9**	**74.2**	91.5	46.2	**95.1**	**81.4**	59.5	**95.1**	**87.6**	**92.1**	**82.0**	65.0	**70.2**	**76.8**

The best result is marked in bold.

adverse visual conditions: Fog, Night, Rain, and Snow. As a real-world dataset, it comprises a total of 1600 training images, 406 validation images, and 2000 test images, all of which have a resolution of 1920×1080 pixels.

In line with prior works [16], [31], [35], we evaluate segmentation performance on the Cityscapes validation set. Specifically, we report the mean Intersection over Union (mIoU) for all 19 classes in GTA → Cityscapes, Cityscapes → DarkZurich and Cityscapes → ACDC and for 16 categories in SYNTHIA → Cityscapes.

B. Implementation Details

Architectures Settings: We follow the established UDA settings [14], [16], [31], [35], employing the MiT-B5 model pretrained on ImageNet-1k as our backbone. In the prototype head, a 1×1 conventional layer projects pixel embeddings to prototype features space, where $D_{proto} = 256$, then build $K = 5$ prototypes within each category. We follow the design of context-aware fusion decoder [16] in the prototype branch. The feature bank size is defined as 20000. Our approach introduces an auxiliary prototype head that works harmoniously with the majority of existing UDA methods. The prototype head is exclusively employed during the training phase and is deactivated during testing.

Training details: Following the standard UDA protocol [16], we employ AdamW optimizer [64] with a learning rate of 6×10^{-5} for the encoder and 6×10^{-4} for the decoder. Additionally, we apply weight decay of 0.01 and adopted linear warmup of the learning rate [65] during the initial 1,500 iterations. The images in GTA and Synthia are resized to $1,280 \times 720$ and $1,280 \times 760$, respectively. As for mixed-resolution training, the crop size is set to 640×640 for the single-resolution model and $1,024 \times 1,024$ for the multi-resolution model. The parameter $\alpha = 0.999$ is used for EMA updating. We set temperature coefficient $T = 0.1$ and pseudo confidence threshold $\theta = 0.968$. Following HRDA [31], the weight on low-resolution branch $\lambda_d = 0.1$. For loss weighs, we empirically choose the setting of $\lambda_1 = 0.1$ and $\lambda_2 = 0.1$. We also employ additional training strategies, including domain mixing [14], rare class sampling, and ImageNet feature distance [16]. We trained the framework a batch size of 2 on a single TESLA-V100(32G) GPU for 60,000 iterations.

C. Comparison to the State-of-the-art Methods

We compare MPCL to state-of-the-art UDA approaches on GTA → Cityscapes and Synthia → Cityscapes. As shown in Tables I and II. Our centroid-based and distribution-based prototype methods were integrated into three widely adopted frameworks: DAFormer [16], HRDA [31], and MIC [35].

GTA → Cityscapes: Table I shows that our centroid-based and distribution-based methods exhibit segmentation performance improvements of $+3.0\%, +1.3\%$ and $+3.3\%, +1.5\%$ over DAFormer and HRDA, respectively, indicating better feature representation learning from the distribution-based prototypes in this benchmarks.

Combined with the latest MIC [35], our centroid-based and distribution-based versions achieved additional segmentation improvements of $+0.5\%$ and $+0.9\%$ over the baseline, highlighting the complementarity of our approach with existing UDA methods. Particularly, the integration of MIC with the distribution-based method has led to a remarkable mIoU of 76.8%.

Additionally, the distribution-based approach exhibits superior performance in comparison to its counterparts across 12 out of the 19 categories, especially in segmenting large objects like *building, bus, car*, and in long-tailed classes like *train, traffic sign, traffic light*. These results imply that the multi-prototype approach can identify patterns despite substantial intra-class variations.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

TABLE II
COMPARISON OF UDA SEGMENTATION PERFORMANCE (MIOU,%) ON SYNTHIA → CITYSCAPES

Method	Road	S.walk	Build.	Wall*	Fence*	Pole*	Tr.Light	Sign	Veget.	Sky	Person	Rider	Car	Bus	M.bike	Bike	mIoU*	mIoU
Source Only	51.5	20.3	79.2	19.3	1.8	40.9	29.9	22.7	79.1	82.4	63.0	24.9	75.8	33.7	18.9	24.9	46.6	35.2
PatchAlign [28]	82.4	38.0	78.6	8.7	0.6	26.0	3.9	11.1	75.5	84.6	53.5	21.6	71.4	32.6	19.3	31.7	46.5	40.0
AdvEnt [25]	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	48.0	41.2
CBST [13]	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	48.9	42.6
DADA [61]	89.2	44.8	81.4	6.8	0.3	26.2	8.6	11.1	81.8	84.0	54.7	19.3	79.7	40.7	14.0	38.8	49.8	42.6
DACS [14]	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24.0	83.7	90.8	67.6	38.3	82.9	38.9	28.5	47.6	54.8	48.3
Uncertainty [18]	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	54.9	47.9
Adaboost [59]	85.6	43.9	83.9	19.2	1.7	38	37.9	19.6	85.5	88.4	64.1	25.7	86.6	43.9	31.2	51.3	57.5	50.4
ProDA [19]	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	62.0	55.5
CAlign [20]	89.1	51.5	89.7	53.3	6.8	54.6	66.5	61.1	88.0	94.7	79.7	56.9	90.6	67.9	63.7	65.1	74.2	67.5
DAFormer [16]	84.5	40.6	88.4	41.5	6.5	50.0	55.0	54.6	86.0	89.8	73.2	48.2	87.2	53.2	53.9	61.7	67.4	60.9
DAFormer+Centroid	86.8	43.4	88.7	42.3	6.9	55.3	59.7	57.1	86.5	92.7	75.0	47.9	88.7	61.5	57.9	61.6	69.8	63.3
DAFormer+Gaussian	87.0	46.5	88.5	43.3	5.1	54.7	61.0	57.8	88.0	92.9	74.5	48.8	89.2	54.8	57.5	63.2	70.0	63.3
HRDA [31]	85.2	47.7	88.8	49.5	4.8	57.2	65.7	60.9	85.3	92.9	79.4	52.8	89.0	64.7	63.9	64.9	72.4	65.8
HRDA+Centroid	88.7	**53.5**	89.4	51.5	7.5	57.3	66.2	58.8	87.0	94.1	80.4	57.9	90.2	66.8	63.5	64.9	74.0	67.4
HRDA+Gaussian	89.1	51.5	89.7	**53.3**	6.8	54.6	66.5	61.1	88.0	94.7	79.7	56.9	**90.6**	**67.9**	63.7	65.1	74.2	67.5
MIC [35]	86.6	50.5	89.3	47.9	7.8	59.4	66.7	63.4	87.1	94.6	81.0	58.9	90.1	61.9	67.1	64.3	74.0	67.3
MIC+Centroid	86.6	50.3	89.7	41.8	**8.8**	**62.6**	**70.6**	**65.3**	89.2	**95.1**	**81.8**	**60.6**	86.7	60.3	**69.2**	65.5	74.7	67.8
MIC+Gaussian	**87.4**	52.8	**89.7**	49.9	8.5	57.7	69.5	64.6	**89.5**	94.9	80.4	60.5	89.7	64.4	68.6	**66.3**	**75.3**	**68.4**

mIoU* is calculated for 16 classes, with the excluded classes masked with *.
The bold values mark the best result in each column.

TABLE III
SEMANTIC SEGMENTATION PERFORMANCE (MIOU, %) ON TWO REAL-WORLD UDA BENCHMARKS

Method	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
Day-to-Nighttime: Cityscapes→DarkZurich (Test)																				
AdvEnt [25]	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
MGCDA [66]	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	**66.0**	0.0	37.5	29.1	22.7	42.5
DANNet [67]	90.0	54.0	74.8	41.0	21.1	25.0	26.8	30.2	**72.0**	26.2	**84.0**	47.0	33.9	68.2	19.0	0.3	66.4	38.3	23.6	44.3
DAFormer [16]	93.5	65.5	73.3	39.4	19.2	53.3	**44.1**	**44.0**	59.5	34.5	66.6	53.4	52.7	**82.1**	52.7	**9.5**	89.3	**50.5**	**38.5**	53.8
DAFormer+Gaussian	**94.5**	**73.2**	**78.6**	**50.2**	**23.2**	**53.8**	38.8	42.8	57.9	**39.6**	65.5	**55.1**	52.3	81.2	44.9	6.6	**90.0**	49.7	38.0	**54.5**
Clear-to-Adverse-Weather: Cityscapes→ACDC (Test)																				
AdvEnt [25]	72.9	14.3	40.5	16.6	21.2	9.3	17.4	21.2	63.8	23.8	18.3	32.6	19.5	69.5	36.2	34.5	46.2	26.9	36.1	32.7
MGCDA [66]	73.4	28.7	69.9	19.3	26.3	36.8	**53.0**	53.3	**75.4**	32.0	84.6	51.0	26.1	77.6	43.2	45.9	53.9	32.7	41.5	48.7
DANNet [67]	**84.3**	**54.2**	77.6	38.0	30.0	18.9	41.6	35.2	71.3	39.4	**86.6**	48.7	29.2	76.2	41.6	43.0	58.6	32.6	43.9	50.0
DAFormer [16]	58.4	51.3	**84.0**	**42.7**	**35.1**	50.7	30.0	**57.0**	74.8	52.8	51.3	58.3	32.6	**82.7**	58.3	54.9	**82.4**	44.1	50.7	55.4
DAFormer+Gaussian	62.8	51.6	83.0	34.7	35.0	**52.1**	30.1	56.4	73.0	**55.9**	60.9	**62.7**	**33.8**	80.2	**59.5**	**58.5**	81.8	**47.5**	**52.3**	**56.4**

The bold values mark the best result in each column.

Synthia → Cityscapes: Table II illustrates the competitive performance of our approach on the Synthia → Cityscapes benchmark. Notably, both the centroid-based and distribution-based methods outperform the previous UDA method, DAFormer [16] and HRDA [31] by a large margin. Also, when combined with current state-of-the-art MIC [35], we see an improvement of $+0.5$ and $+1.1$ in mIoU, respectively., resulting in an overall performance of 68.4%. It is noteworthy that our method achieves the highest performance across a wide range of categories, demonstrating its effectiveness.

Realworld benchmarks: Table III illustrates the performance of our method on two real-world datasets. Unlike synthetic datasets GTA and Synthia, real-world scenes are more complex, posing significant challenges to algorithm generalization. As shown in Table III, our method obatains 54.5% and 56.4% mIoUs on Cityscapes → DarkZurich and Cityscapes → ACDC benchmarks separately. Our method demonstrates improvements over the latest methods based on the baseline, surpassing other approaches. This highlights the effectiveness and generalizability of the method we propose.

D. Visualization

Qualitative results: We have conducted a comparative analysis of our segmentation visualization results with those of the previous state-of-the-art methods, DAFormer [16] and MIC [35], using both GTA → Cityscapes and SYNTHIA → Cityscapes benchmarks. As depicted in Fig. 6, our approach excels in segmenting challenging classes, including *traffic signs, sidewalks* and *poles*, which is inalign with the results in Table I. In addition, our approach performs exceptionally well in the context of rare classes, such as *rider*. It accurately separates the rider's head instead of misclassifying it as *person*. This demonstrates that the utilization of multiple prototypes allows it to focus on distinctive parts within objects, ultimately enhancing classification accuracy.

Visualization on pixel-to-prototype similarity: To gain a more intuitive insight into the role of prototypes in the domain adaptation process, we visualized the activation of distinct pixel features with respect to multiple prototypes within their corresponding category subspaces in Fig. 7. It can be observed that, for class *car* and *person*, their respective prototypes exhibit similarities during the exploration of distribution patterns. Each prototype corresponds to different components of the objects, with some emphasizing the overall object context while others concentrate on capturing segments near the contour. In the scenes of *sky*, *road* and *wall*, the coverage of different prototype heatmaps varies, collectively encompassing the entire pixel range of their respective categories. Thus, these complementary

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

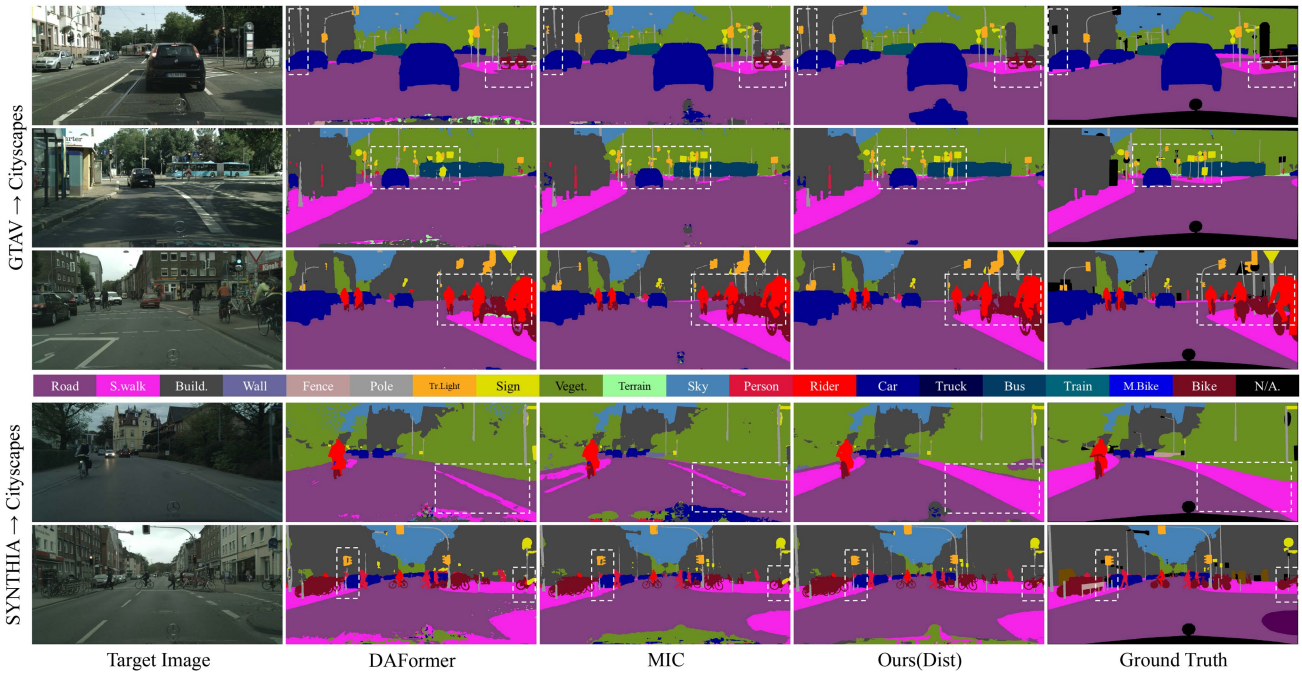


Fig. 6. Qualitative Comparison on GTA → Cityscapes. Better segmentation results are highlighted in dash boxes(zoom in for finer details).

TABLE IV
ABLATION STUDY ON DIFFERENT COMPONENTS

Method	$\mathcal{L}^{M}_{Contrast}$	$\mathcal{L}^{proto}_{tot}$	mIoU
MPCL(Centroid)	✓		74.8
		✓	74.5
	✓	✓	75.1
MPCL(Gaussian)	✓		75.0
		✓	74.7
	✓	✓	75.3
Baseline [31]			73.8

TABLE V
AVERAGE TIME COST COMPARISON(S/ITER) WITH [16] AS THE BASELINE

Dataset	DAFormer	DAFormer+Gaussian
GTA→Cityscapes	1.4615	1.6134
Synthia→Cityscapes	1.4405	1.5734

prototypes better cover the dispersed feature distributions across categories, thereby enabling accurate classification of samples located near the decision boundary.

T-SNE visualization: In Fig. 8, we visualize the feature space via T-SNE [68]. Compared to the baseline, both versions of prototypes exhibit the capability to learn a more compact structure, with the Gaussian-based prototype being particularly discriminative. This visualization demonstrates the model's capacity to attain a well-structured feature space via prototypical contrastive learning. Moreover, Fig. 8(d) shows that multiple prototypes can capture distinctive characteristics within different categorical spaces.

E. Ablation Study

Effect of different components: We conduct ablation studies on different components in the prototype contrastive module, specifically, contrastive loss $\mathcal{L}^{M}_{Contrast}$, consistency loss $\mathcal{L}^{Proto}_{ce}$ using the GTA → Cityscapes benchmark with HRDA as the baseline.

As depicted in Table IV, for Gaussian-based prototypes, the inclusion of $\mathcal{L}^{M}_{Contrast}$ in the centroid version results in a performance improvement of $+1.2\%$, while $\mathcal{L}^{Proto}_{ce}$ contributes to a $+0.9\%$ enhancement. Both losses serve to regularize the feature distribution across domains, promoting the generation of more reliable pseudo-labels. When we combine the two components, a mIoU of 75.3% is achieved. As for the centroid version, the centroid contrastive learning module performs better than the baseline, with an improvement of $+1.3\%$.

Table V illustrates the time cost comparison of baseline and our proposed method. It can be observed that our method achieves slight improvements of 0.1519 $s/iter$ and 0.1329 $s/iter$ over the baseline on the GTA → Cityscapes and Synthia → Cityscapes benchmarks, respectively. Considering the performance enhancement analysis discussed above, it is evident that our method achieves a significant performance boost at a minimal time cost. Furthermore, during the testing phase, the plug-and-play modules do not participate in inference, resulting in inference times that are identical to those of the baseline method.

Effect of in-class contrastive learning: Our method treat all prototypes, except for the nearest one, as negative samples when computing $\mathcal{L}^{M}_{Contrast}$, including those from the same category. This approach aims to help multiple prototypes of the same category represent different features. We compared our method with

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

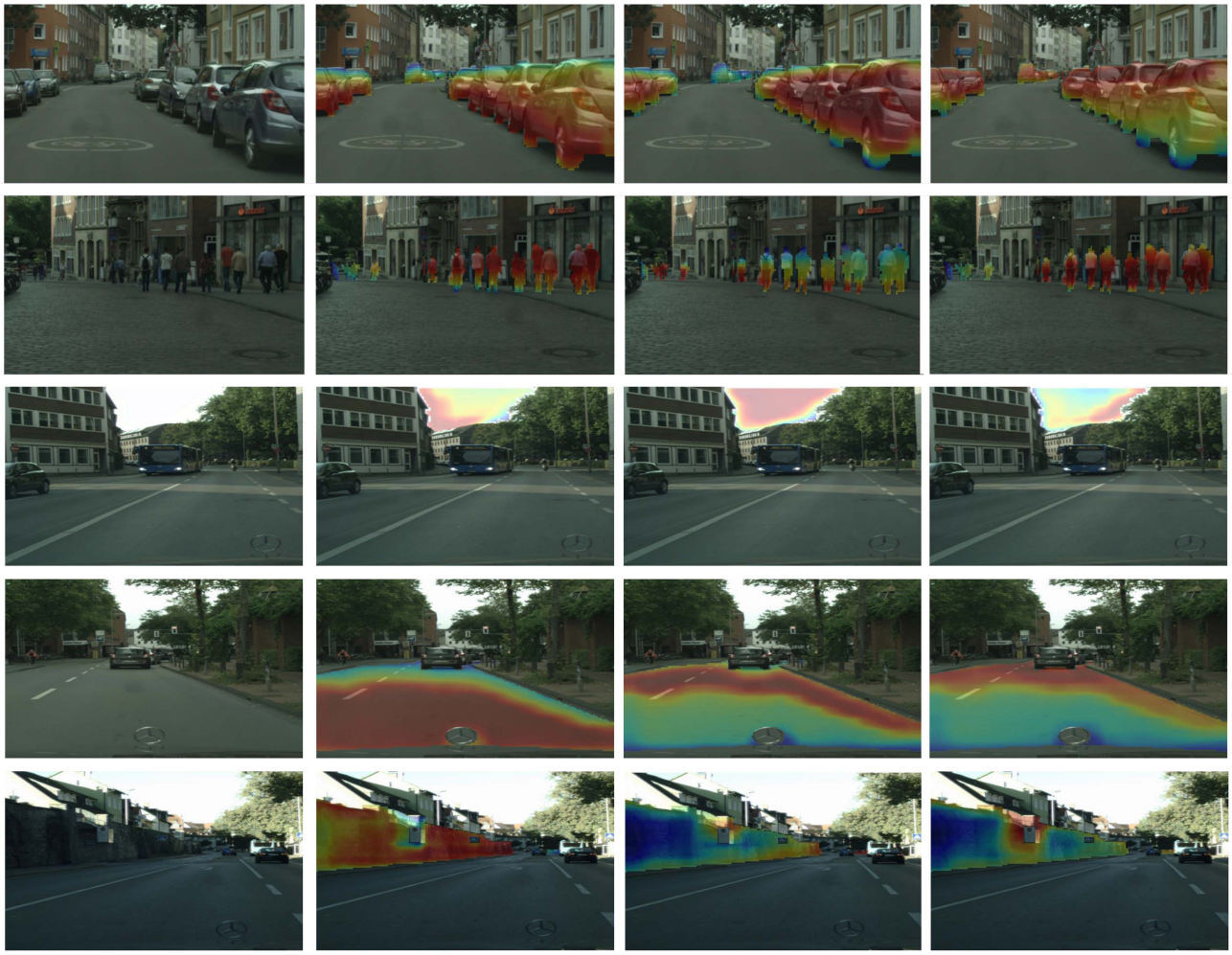
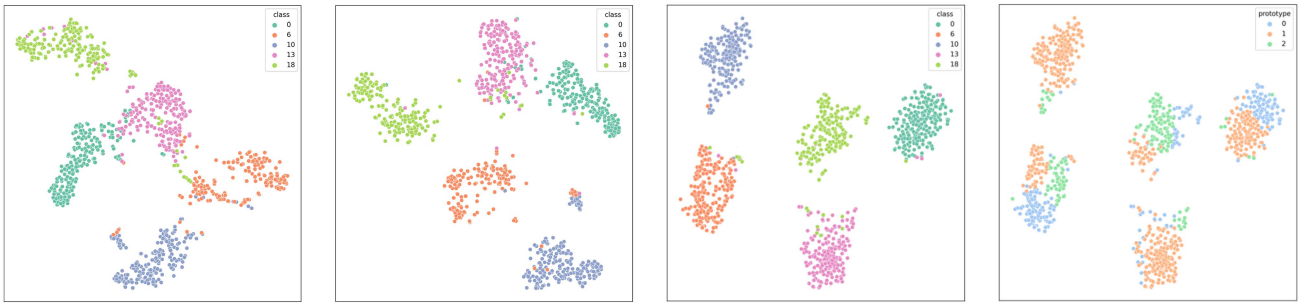


Fig. 7. Activation map of multi-prototypes. The leftmost column is the input image, and the images in columns 2 to 4 correspond to three different prototypes of the category, respectively. The categories shown from the top to the bottom are *car*, *person*, *sky*, *road* and *wall*.



(a) Baseline (b) Category feature space (centroid) (c) Category feature space (Gaussian) (d) Prototype feature space(Gaussian)

Fig. 8. T-SNE visualization of features in under different prototype learning methods with 5 classes in view.

an alternative approach(w/o In-class Contrast in Table VI) that only considers prototypes from other categories as negative ones, with the results presented in Table VI. Compared to this method, our approach achieve improvements of 2.6% and 0.6% on the GTA $\rightarrow$ Cityscapes and Synthia $\rightarrow$ Cityscapes benchmarks, respectively. The results indicate that our method outperforms the approach that only uses out-of-class prototypes as negative samples in terms of performance.

Effect of contrastive learning on different domains: We conduct a comprehensive performance assessment of our method across various domains. Table VII shows that the use of centroid-based and distribution-based prototypes in either the source or target domain results in performance improvements. The most significant enhancement, with an increase of $+1.3\%$ and $+1.5\%$ on GTA $\rightarrow$ Cityscapes, is achieved when features from both domains are utilized for prototype contrastive

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
SEMANTIC SEGMENTATION PERFORMANCE (MIOU, %) UNDER DIFFERENT MODEL SETTINGS

Method	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
Synthetic-to-Real: GTA→Cityscapes (Val.)																				
DAFormer [16]	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	65.1	55.9	61.8	68.3
DAFormer+Gaussian	96.9	77.1	**90.0**	**56.9**	**53.3**	**52.9**	**61.4**	**62.8**	**90.1**	48.2	91.8	**75.8**	47.3	93.2	**79.4**	**84.6**	**76.0**	**61.1**	62.2	**71.6**
Prototype-based Output	**97.2**	**77.4**	89.1	47.4	44.4	48.1	58.8	62.3	89.6	**48.3**	**93.2**	73.1	**49.7**	92.2	65.1	80.8	73.7	57.6	63.8	69.1
w/o In-class Contrast	96.8	75.6	89.3	50.9	46.3	48.0	60.4	59.9	89.8	47.1	93.1	75.2	43.0	**93.4**	73.1	82.4	63.0	59.1	**64.5**	69.0
Synthetic-to-Real: Synthia→Cityscapes (Val.)																				
DAFormer [16]	84.5	40.6	88.4	41.5	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
DAFormer+Gaussian	87.0	46.5	88.5	43.3	5.1	**54.7**	**61.0**	57.8	**88.0**	-	**92.9**	**74.5**	**48.8**	**89.2**	-	54.8	-	**57.5**	**63.2**	**63.3**
Prototype-based Output	86.7	43.5	88.6	43.9	6.9	50.0	57.1	54.4	84.0	-	76.8	73.0	45.5	88.3	-	**62.4**	-	47.8	48.7	59.9
w/o In-class Contrast	**88.9**	**47.7**	**89.0**	**48.4**	**8.4**	53.7	58.7	**58.8**	85.1	-	81.3	71.7	41.1	**89.2**	-	61.7	-	57.1	61.8	62.7

The bold values mark the best result in each column.

TABLE VII
ABLATION STUDY ON DIFFERENT PROTOTYPE DOMAINS

Method	Target	Source	mIoU
MPCL(Centroid)	✓		74.8
		✓	74.6
	✓	✓	75.1
MPCL(Gaussian)	✓		74.9
		✓	74.4
	✓	✓	75.3
Baseline [31]			73.8

TABLE VIII
ABLATION STUDY ON DIFFERENT RESOLUTION OF PROTOTYPES FEATURES WITH [31] AS THE BASELINE

Method	LR	LR+HR
MPCL(Centroid)	75.0	75.1
MPCL(Gaussian)	75.1	75.3

TABLE IX
ANALYSIS ON THE PROTOTYPE NUMBER FOR EACH CLASS WITH [31] AS THE BASELINE

K	1	3	5	10	20
MPCL(centroid)	74.2	74.5	75.1	74.8	73.9
MPCL(Gaussian)	74.5	74.8	75.3	74.3	73.5

TABLE X
ANALYSIS ON HYPERPARAMETES WITH [16] AS BASELINE λ_1: WEIGHT OF $\mathcal{L}^{M}_{Contrast}$, λ_2 WEIGHT OF $\mathcal{L}^{Proto}_{ce}$

λ_1	mIoU
0.05	71.1
0.1	**71.6**
0.2	71.2
0.5	70.9

λ_2	mIoU
0.05	71.3
0.1	**71.6**
0.5	71.4
1.0	71.1

The bold values mark the best result in each column.

learning. This highlights our model's ability to capture distinct feature patterns across mixed domains.

Effect of multi-resolution prototype features: To verify the effectiveness of the multi-resolution prototype learning, we test our method based on HRDA [31]. When only low-resolution feature is used for centroid-based prototype modeling, the model is similar to the previous multi-prototype method [21]. As listed in Table VIII, we see a boost in performance when the high-resolution features are considered during prototypical contrastive learning, indicating that features from HR-corps improve the diversity of intra-class prototypes.

Effect of prototype-based output: The plug-and-play scheme we designed is based on prototypes, where different prototypes encode and represent distinct category features. This allows for the possibility of direct classification using prototypes. As shown in the Table VI, the use of prototypes can also achieve high segmentation performance with 69.1% and 59.9% on GTA → Cityscapes and Synthia → Cityscapes benchmarks, respectively, which is only slightly lower than the output of our proposed method. This demonstrates that the prototypes have learned effective features.

F. Parameter Study

Analysis on prototype number: The hyperparameter K determines the number of distinct prototypes adopted to represent features in each category. Thus, an insufficient number of prototypes can lead to inadequate feature representation by the model. Notably, when $K = 1$, our model aligns with the previous single-category centroid-based method [17]. However, the richness of features is also limited; an excessive number of prototypes may introduce noise during training. Thus, Selecting an appropriate number of prototypes per category is crucial to ensure the representativeness of prototypes. We assess our model's performance across different values of K in the GTA → Cityscapes benchmark. All the experiments are conducted on the centroid-based version with HRDA [31]. Table IX shows that improvements are observed with an increased number of prototypes, demonstrating the effectiveness of multiple prototypes. However, an excessive number of prototypes leads to a performance decrease, indicating a sparse feature space. Empirically, we select K as 5 during experiments.

Additionally, we conduct a hyperparameter analysis on the weight associated with contrastive loss ($\mathcal{L}^{M}_{Contrast}$) and prototype-based prediction loss ($\mathcal{L}^{Proto}_{ce}$) with centroid-based prototypes. The results are presented in Table X. These experiments were carried out on the GTA → Cityscapes under DAFormer [16] with Gaussian-based prototypes. Empirically, we select $\lambda_1 = 0.1$ and $\lambda_2 = 0.1$.

Analysis on training parameters: During the training process, three parameters play a crucial role. θ, as the confidence threshold, determines which target domain pixels can be selected for mixed training. The feature bank size influences the speed of prototype updates, while the momentum parameter α dictates

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

TABLE XI
ABLATION STUDY OF FEATURE BANK SIZE, CONFIDENCE THRESHOLD θ AND MOMENTUM PARAMETER α WITH [16] AS THE BASELINE

	Feature Bank Size				Confidence Threshold θ				Momentum Parameter α		
	12k	16k	20k	24k	0.956	0.968	0.980	0.992	0.900	0.990	0.999
GTA→Cityscapes	69.5	69.5	**71.6**	70.4	69.3	**71.6**	70.4	69.5	66.6	68.1	**71.6**
Synthia→Cityscapes	59.5	61.4	**63.3**	61.1	63.0	**63.3**	62.3	61.7	60.6	62.2	**63.3**

the magnitude of model parameter updates. We conducted a further investigation into the values of these three parameters, with the results presented in the Table XI. An excessively large feature bank size can slow down the prototype update rate, while a value that is too small can lead to instability in training. Similarly, a large θ can reduce the number of effective features, whereas a smaller T may introduce more noise. Additionally, if α is too large, the model parameter updates will be slow, while a value that is too small can cause instability in model training. Therefore, we selected 20k, 0.968, and 0.999 as the values for feature bank size, θ, and α to achieve a balance in the training process.

V. CONCLUSION

In this article, we explore the reason behind noisy pseudo labels and present a contrastive learning framework with different types of prototypes for UDA semantic segmentation. To denoise pseudo labels, we use online non-parametric clustering to establish multiple prototypes within category feature spaces. Then, we leverage contrastive learning between pixels and prototypes to regularize feature space. Both centroid-based and distribution-based prototypes are utilized to capture distinctive intra-class characteristics and encourage better separation between categories, resulting in improved category-specific discrimination. Furthermore, the introduction of multi-resolution prototype learning enhances the diversity and robustness of prototype representations. Experimental results demonstrate the effectiveness of the proposed method, surpassing previous state-of-the-arts with a strong performance of 76.8% mIoU on GTA → Cityscapes and 68.4% mIoU on Synthia → Cityscapes. Moreover, we conduct experiments on two real-world benchmarks, i.e. Cityscapes → ACDC and Cityscapes → DarkZurichand obtains the favorable results of 56.4% and 54.5%, respectively. Moving forward, we plan to further expand the applicability of our method to related fields, such as medical image segmentation and video segmentation.

REFERENCES

[1] L.-C. 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, Apr. 2018.

[2] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "FCNs in the wild: Pixel-level adversarial and constraint-based adaptation," 2016, *arXiv:1612.02649*.

[3] Z. Liu et al., "SWIN transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[4] C. Sakaridis, D. Dai, and L. Van Gool, "ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10765–10775.

[5] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 102–118.

[6] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Computer Vis. Pattern Recognit.*, 2016, pp. 3234–3243.

[7] J. Hoffman et al., "CYCADA: Cycle-consistent adversarial domain adaptation," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1989–1998.

[8] Y. Yang and S. Soatto, "FDA: Fourier domain adaptation for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4085–4095.

[9] H. Huang, Q. Huang, and P. Krahenbuhl, "Domain transfer through deep activation matching," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 590–605.

[10] T. Chen et al., "Enhanced feature alignment for unsupervised domain adaptation of semantic segmentation," *IEEE Trans. Multimedia*, vol. 24, pp. 1042–1054, 2022.

[11] Y.-H. Tsai et al., "Learning to adapt structured output space for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7472–7481.

[12] Y. Xu, F. He, B. Du, D. Tao, and L. Zhang, "Self-ensembling GAN for cross-domain semantic segmentation," *IEEE Trans. Multimedia*, vol. 25, pp. 7837–7850, 2022.

[13] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 289–305.

[14] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson, "DACS: Domain adaptation via cross-domain mixed sampling," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1379–1389.

[15] F. Yu et al., "DAST: Unsupervised domain adaptation in semantic segmentation based on discriminator attention and self-training," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 12, pp. 10754–10762.

[16] L. Hoyer, D. Dai, and L. Van Gool, "DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9924–9935.

[17] Z. Jiang et al., "Prototypical contrast adaptation for domain adaptive semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 36–54.

[18] 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.

[19] P. Zhang et al., "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12414–12424.

[20] S. Wang et al., "Cluster alignment with target knowledge mining for unsupervised domain adaptation semantic segmentation," *IEEE Trans. Image Process.*, vol. 31, pp. 7403–7418, 2022.

[21] Q. Liu et al., "Prototypical contrastive learning for domain adaptive semantic segmentation," in *Proc. 2023 Int. Joint Conf. Neural Netw.*, 2023, pp. 1–9.

[22] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[23] S. Bond, A. Hoeffler, and J. Temple, "GMM estimation of empirical growth models," Economics Group, Nuffield College, Univ. Oxford, Tech. Rep., 2001.

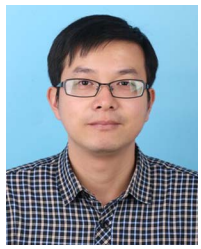
[24] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[25] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2517–2526.

[26] H. Wang, T. Shen, W. Zhang, L.-Y. Duan, and T. Mei, "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 642–659.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

[27] R. Li et al., "Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11593–11603.
[28] Y.-H. Tsai et al., "Domain adaptation for structured output via discriminative patch representations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1456–1465.
[29] R. Gong, W. Li, Y. Chen, and L. V. Gool, "DLOW: Domain flow for adaptation and generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2477–2486.
[30] F. Pizzati, R. d. Charette, M. Zaccaria, and P. Cerri, "Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2990–2998.
[31] L. Hoyer, D. Dai, and L. Van Gool, "HRDA: Context-aware high-resolution domain-adaptive semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 372–391.
[32] D. Dai, C. Sakaridis, S. Hecker, and L. Van Gool, "Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding," *Int. J. Comput. Vis.*, vol. 128, pp. 1182–1204, 2020.
[33] Q. Xu, Y. Ma, J. Wu, C. Long, and X. Huang, "CDAda: A curriculum domain adaptation for nighttime semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2962–2971.
[34] N. Araslanov and S. Roth, "Self-supervised augmentation consistency for adapting semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15384–15394.
[35] L. Hoyer, D. Dai, H. Wang, and L. Van Gool, "MIC: Masked image consistency for context-enhanced domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 11721–11732.
[36] M. Chen, H. Xue, and D. Cai, "Domain adaptation for semantic segmentation with maximum squares loss," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2090–2099.
[37] Q. Wang, D. Dai, L. Hoyer, L. Van Gool, and O. Fink, "Domain adaptive semantic segmentation with self-supervised depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8515–8525.
[38] Q. Zhang, J. Zhang, W. Liu, and D. Tao, "Category anchor-guided unsupervised domain adaptation for semantic segmentation," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 435–445.
[39] Q. Ren, Q. Mao, and S. Lu, "Prototypical bidirectional adaptation and learning for cross-domain semantic segmentation," *IEEE Trans. Multimedia*, vol. 26, pp. 501–513, 2024.
[40] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *J. Roy. Stat. Soc. Ser. C (Appl. Statist.)*, vol. 28, no. 1, pp. 100–108, 1979.
[41] D. A. Reynolds, "Gaussian mixture models," in *Encyclopedia Biometrics*, vol. 741. Boston, MA, USA: Springer, 2009, pp. 659–663.
[42] K. Tanwisuth et al., "A prototype-oriented framework for unsupervised domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 17194–17208. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/8edd72158ccd2a879f79cb2538568fdc-Paper.pdf
[43] Y. Pan et al., "Transferrable prototypical networks for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2239–2247.
[44] J. Zhang, J. Huang, X. Jiang, and S. Lu, "Black-box unsupervised domain adaptation with bi-directional Atkinson-Shiffrin memory," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 11771–11782.
[45] Y. Du, D. Zhou, Y. Xie, Y. Lei, and J. Shi, "Prototype-guided feature learning for unsupervised domain adaptation," *Pattern Recognit.*, vol. 135, 2023, Art. no. 109154. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320322006331
[46] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4080–4090.
[47] S. Jetley, B. Romera-Paredes, S. Jayasumana, and P. Torr, "Prototypical priors: From improving classification to zero-shot learning," 2015, *arXiv:1512.01192*.
[48] T. Zhou, W. Wang, E. Konukoglu, and L. Van Gool, "Rethinking semantic segmentation: A prototype view," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2572–2583.
[49] Z. Chen and Z. Lian, "Semi-supervised semantic segmentation via prototypical contrastive learning," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 6696–6705.
[50] G. Lee, C. Eom, W. Lee, H. Park, and B. Ham, "Bi-directional contrastive learning for domain adaptive semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* 2022, pp. 38–55.
[51] J. Li, P. Zhou, C. Xiong, and S. Hoi, "Prototypical contrastive learning of unsupervised representations," in *Proc. Int. Conf. Learn. Representations*, 2021.
[52] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.
[53] R. D. Hjelm et al., "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Representations*, 2019.
[54] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6210–6219.
[55] S. Liu, S. Zhi, E. Johns, and A. Davison, "Bootstrapping semantic segmentation with regional contrast," in *Proc. Int. Conf. Learn. Representations*, 2022.
[56] Y. Asano, C. Rupprecht, and A. Vedaldi, "Self-labelling via simultaneous clustering and representation learning," in *Proc. Int. Conf. Learn. Representations*, 2020.
[57] J. Altschuler, J. Niles-Weed, and P. Rigollet, "Near-linear time approximation algorithms for optimal transport via sinkhorn iteration," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1961–1971.
[58] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2507–2516.
[59] Z. Zheng and Y. Yang, "Adaptive boosting for domain adaptation: Toward robust predictions in scene segmentation," *IEEE Trans. Image Process.*, vol. 31, pp. 5371–5382, 2022.
[60] Y. Liu, J. Deng, X. Gao, W. Li, and L. Duan, "BAPA-Net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8801–8811.
[61] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "DADA: Depth-aware domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7363–7372.
[62] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput,. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.
[63] C. Sakaridis, D. Dai, and L. Van 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, Jun. 2022.
[64] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2017.
[65] P. Goyal et al., "Accurate, large minibatch SGD: Training Imagenet in 1 hour," 2017, *arXiv:1706.02677*.
[66] C. Sakaridis, D. Dai, and L. V. Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7374–7383.
[67] X. Wu, Z. Wu, H. Guo, L. Ju, and S. Wang, "DANNet: A. one-stage domain adaptation network for unsupervised nighttime semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15769–15778.
[68] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.



Jun Yu is currently an Associate Professor and the Laboratory Director with the Department of Automation and the Institute of Advanced Technology, University of Science and Technology of China, Hefei, China. He has authored or coauthored more than 180 journal articles and conference papers in IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, TOMM, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, ACL, CVPR, ICCV, ICML, ICLR, MM, SIGGRAPH, VR, AAAI, and IJCAI. His research interests include multimedia computing and intelligent robot. He was the recipient of six Best Paper awards from premier conferences, including CVPR PBVS, ICCV MFR, ICME, FG, and was the recipient of more than 50 champions from Grand Challenges held in NeurIPS, CVPR, ICCV, MM, ECCV, IJCAI, AAAI.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.

Guochen Xie received the B.E. degree from Shandong University, Jinan, China. He is currently working toward the Ph.D. degree with the Department of Automation, School of Information and Technology, University of Science and Technology of China, Hefei, China. His research interests include debiasing learning, image generation, and representation learning. He has participated in many challenges with top conferences such as CVPR, ICCV, and ACM MM, and was the recipient of more than ten championships.

Quansheng Liu received the B.E. degree from Jiangnan University, Wuxi, China. He is currently working toward the Undergraduate degree with the Department of Automation, School of Information and Technology, University of Science and Technology, Hefei, China. His research interests include computer vision and transfer learning.

Zhen Kan (Senior Member) received the Ph.D. degree from the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA, in 2011. He was a Postdoctoral Research Fellow with Air Force Research Laboratory (AFRL), Eglin AFB, and was also with the University of Florida REEF, from 2012 to 2016. From 2016 to 2019, he was an Assistant Professor with the Department of Mechanical Engineering, University of Iowa, Iowa City, IA, USA. He is currently a Professor with the Department of Automation, University of Science and Technology of China, Hefei, China. His research interests include controls, robotics, and formal methods. He is also on Program Committees of several internationally recognized scientific and engineering conferences and is an Associate Editor for IEEE TRANSACTIONS ON AUTOMATIC CONTROL and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

Lei Wang received the Ph.D. degree from the University of Science and Technology of China, Hefei, China. He is currently an Associate Professor and the Deputy Head with the Department of Automation, School of Information and Technology, University of Science and Technology, China. His research interests include machine learning and simulation of complex systems.

Tianyu Liu received the B.S. degree from the Harbin Institute of Technology, Harbin, China, in 2018, and the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China, in 2023. She is currently a Research Assistant Professor with the Jianghuai Advance Technology Center, Hefei, China. Her research interests include robotics and motion control.

Qiang Ling (Senior Member, IEEE) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 1997, the M.E. degree from Tsinghua University, Beijing, China, in 2000, and the Ph.D. degree from the University of Notre Dame, Notre Dame, IN, USA, in 2005. From 2005 to 2008, he was a Research Staff Member with Seagate Technology. In 2008, he joined the University of Science and Technology of China, where he is currently a Professor with the Department of Automation. His research interests include networked control systems, signal processing and machine learning. From 2017 to 2022, he was an Associate Editor on the IEEE Control Systems Society Conference Editorial Board.

Wei Xu received the Undergraduate degree and the master's degree in orthopedics from Anhui Medical University, Hefei, China, in 2003 and 2009, respectively. He was a Surgical Resident with Anhui Provincial Hospital, China, for the next three years. Since 2009, he has been with Anhui Provincial Hospital, and was appointed Deputy Chief Physician in 2015. His research interests include treating severe fractures and injuries. He is also the Member of External Fixation Group, Chinese Association of Orthopaedic Surgeons (CAOS), and Young Members of the 8th and 9th Committees of the Orthopedics Branch, Anhui Branch of the Chinese Medical Association.

Fang Gao (Member, IEEE) received the B.S. and Ph.D. degrees in chemical physics from the University of Science and Technology of China, Hefei, China, in 2004 and 2010, respectively. He was an Assistant Professor and an Associate Professor with the Institute of Intelligent Machines, Chinese Academy of Sciences. He is currently a Professor with the College of Electrical Engineering, Guangxi University, Nanning, Guangxi, China. His research interests include deep learning, computer vision, embodied artificial intelligence, and quantum machine learning.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 29,2025 at 02:21:32 UTC from IEEE Xplore. Restrictions apply.