SCIENCE CHINA
Information Sciences

• Supplementary File •

Prototype-Guided Diffusion Alignment for Few-Shot Unsupervised Domain Adaptation

Heyang Sun[1], Chuanxing Geng[1] & Songcan Chen[1*]

[1]*MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing 211106, China*

Appendix A Abstract

Few-shot unsupervised domain adaptation (FUDA) aims to conduct accurate predictions on the unsupervised target domain with shared label space in case only a few source samples are labeled while the rest are unlabeled. Given the scarcity of supervision, existing methods strive to derive relatively reliable representations to implicitly align distributions in the latent space through various techniques such as self-supervised learning. Although these approaches have shown some promising results, they often overlook a critical issue: the *latent* spaces they construct in such a spare-label setting can hardly sufficiently capture task-relevant semantic information and, consequently, result in inaccurate predictions. To address this challenge, we argue for aligning feature representations in the *original* feature space with more comprehensive semantic information, thereby mitigating the negative transfer caused by inadequate latent representations. Specifically, we propose a prototype-guided diffusion alignment method (PGDA) for FUDA, where we transfer the target domain distribution to the training distribution in the original space via a diffusion alignment module, with the prototypes learned by a representation learning module aiding the knowledge transfer. In addition, a confidence-based noise diffusion strategy is designed to adaptively adjust the alignment strength for each sample, promoting more precise knowledge transfer. Extensive experiments on various benchmark datasets, including Office-31, Office-Home, VisDA-C, and DomainNet, demonstrate the superiority of the proposed method.

Key words Few-shot unsupervised domain adaptation (FUDA), Prototype learning, Diffusion model

Appendix B Introduction

Deep neural network models achieve great and impressive successes on a variety of machine learning tasks [1]. However, their performances always become frustrating once applied in the real-world scenarios where the test and training set no longer meet the independent and identically distributed (i.i.d.) hypothesis [2, 3]. To cope with this problem, unsupervised domain adaptation (UDA) [4–7] as a popular paradigm is proposed to boost the prediction of the unlabeled target domain by transferring the knowledge from the labeled source domain.

With promising results, UDA methods have been widely used in object detection [8, 9], medical image analysis [10, 11], age estimation [12, 13] and so on. However, existing UDA methods usually rely on access to fully-annotated source data, which can hardly be guaranteed in some practical applications with sparse annotations due to data privacy protection [14,15] or expensive labeling costs [16]. For example, in the financial sector [17], apart from a small fraction of publicly traded companies that disclose data, most non-public companies do not share the data with private institutions or irrelevant people. Similarly, in the medical field [18], certain data may require annotations from specific expert professionals, making it challenging to obtain an adequate amount of annotations. In these resource-constrained scenarios, it is difficult to train effective UDA models, let alone some large-scale models [19, 20] for domain adaptation. Therefore, such so-called few-shot unsupervised domain adaptation (FUDA) issues are gradually attracting attention and research [21–23]. Here to avoid confusion, we further differentiate the FUDA scenario we focused on from UDA and other few-shot UDA setups [24–30], as shown in Table B1. The FUDA scenario we aim to solve refers to a setting where only an extremely small fraction of source samples are labeled, while the remaining source and all target samples are unlabeled.

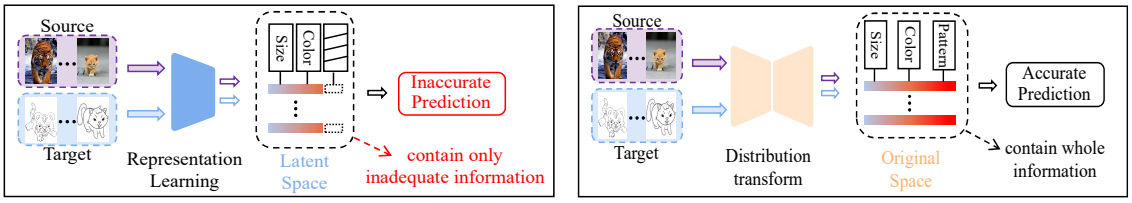
Compared to UDA, the FUDA scenario is more challenging as it must leverage very limited supervisory information and remaining unsupervised information to realize downstream task-related knowledge transfer. To address this issue, existing FUDA methods persistently strive to learn an effective discriminative space through various approaches such as Self-supervised Learning [31], thereby implicitly achieving distribution alignment between source and target domains. Specifically, CDS [21] designs an in-domain self-supervised task to capture similarity and an across-domain self-supervised task to align latent features. Similarly, PCS [22] captures the category-wise semantic structures of the data through intra-class prototype contrastive learning and aligns distributions via cross-domain prototype self-supervision. Recently, C-VisDiT [23] proposes a confidence-based transfer framework that selects high-confidence samples from both the source

* Corresponding author (email: s.chen@nuaa.edu.cn)

Table B1 Comparison of different few-shot settings in unsupervised domain adaptation. "full" and "**few**" respectively denote whether the sample is scarce or not, "none" means there is no label.

Setting	UDA	FSDA	FS-UDA	FUDA (ours)
Source label	full	full	full	**few**
Source sample	full	full	**few**	full
Target label	none	none	none	none
Target sample	full	**few**	full	full
Referance	[4–7]	[24–26]	[27–30]	[21–23]

and target domain to construct separate self-supervised tasks for cross-domain alignment and in-domain discriminative learning.



(a) Previous Methods (b) Proposed Method

Figure B1 Comparison of (a) previous latent space alignment methods and (b) our proposed method. Previous methods align in the learned latent representation space without considering the loss of discriminative information, often resulting in unreliable alignment and erroneous prediction outcomes. The proposed method conducts distribution alignment in the original space, aiming to preserve as much discriminative information as possible for accurate predictions.

Although these approaches demonstrate promising progress by employing diverse auxiliary representation tasks to learn discriminative latent spaces for cross-domain knowledge transfer, they overlook a critical limitation: latent space learning is inherently an information selection process. In few-shot settings, such methods can hardly ensure that the latent space retains sufficient task-relevant semantic information for effective discrimination. In other words, due to the absence of sufficient supervisory information, the network prioritizes learning simple latent patterns, often at the cost of losing important discriminative semantics, which can lead to disastrous negative transfer [32]. As illustrated in Figure B1(a), the source domain consists of real images and the target domain contains sketches, due to sparse annotations, the network may prioritize learning simple latent features such as "color" or "size" from the source domain. However, these features provide no discriminative value for the target domain, leading to unreliable feature alignment and ultimately resulting in incorrect predictions.

To address this challenge, drawing inspiration from diffusion models [33], we propose a Prototype-Guided Diffusion Alignment (PGDA) method for FUDA, which aligns the source and target domain distributions in the original feature space while leveraging richer semantic information, as illustrated in Figure B1(b). By doing so, PGDA effectively mitigates negative transfer induced by unreliable latent representations. Specifically, we construct a diffusion alignment module that realizes domain alignment by transferring inputs into the noise distribution and then transforming the noise distribution back into the training distribution, where we embed the class prototypes as condition guidance to ensure the precision of the distribution transformation and the consistency of discriminative information. For the purpose of learning such effective guidance, we employ a representation learning module that selects high-confidence source domain samples to enrich the discriminative representations for prototype learning and performs coarse-grained cross-domain contrastive learning on the clustered target domain prototypes to bring the class prototypes closer together while avoiding negative transfer. Additionally, during the prediction phase, we design a confidence-based strategy to dynamically adjust the scale of the noise alignment, further ensuring accurate knowledge transfer. The overall model framework is shown in Figure C1 and the main contributions of this paper are as follows:

1. We propose a novel prototype-guided diffusion alignment method to address the issue of discriminative semantic loss caused by existing methods solely learning in the latent space.

2. We develop an effective representation learning scheme to extract class prototypes for guiding alignment and design a dynamic adjustment scheme of transfer strength to ensure accurate knowledge transfer.

3. We conduct extensive experiments on various benchmarks, which validate the effectiveness and superiority of the proposed method.

Appendix C Related work

In this section, we focus on unsupervised domain adaptation (UDA) and few-shot domain adaptation (FUDA) which are relevant to the method proposed in this paper.

Appendix C.1 Unsupervised Domain Adaptation

UDA methods aim to transfer knowledge from a fully labeled source domain to make predictions on an unlabeled target domain with distribution discrepancy. Existing mainstream domain adaptation methods can be divided into metric-based and generation-based approaches. Metric-based methods focus on reducing domain discrepancies by minimizing various statistical criteria. For example, DAN [34] explicitly measures domain differences using Maximum Mean Discrepancy (MMD), CORAL [35] aligns the second-order statistics by matching covariance matrices, CMD [36] introduces higher-order alignment by matching central moments, and MDD [37] measures the difference in domain distributions by comparing their density functions, among others. Recently, MLRGL [67] effectively captures the contextual relationships in the target domain through multi-view low-rank high-order graph learning, significantly enhancing the model's adaptability.

Differently, generation-based methods encourage the network to learn domain-invariant representations, thus enabling knowledge transfer. Representative approaches include DANN [38], which aligns the marginal distributions through adversarial training, CDAN [39], which introduces conditional adversarial training to align the conditional distributions further, ADDA [40], which incorporates an additional discriminator loss to enhance the discriminative power of adversarial training, and so on. Recently, with advances in diffusion models (DMs), several approaches have applied DMs to unsupervised domain adaptation. For instance, DAD [41] progressively mitigates domain discrepancies through a step-wise approach and enhances model adaptability via a mutual learning strategy. Meanwhile, DACDM [42] facilitates knowledge transfer by transforming the source domain distribution to align with the target domain in a domain-guided manner. These UDA methods leverage abundant supervised information from the source domain to accurately measure distributions in the latent space or learn invariant latent representations. However, in cases where the labels in the source domain are sparse, the effectiveness of these methods significantly decreases.

Appendix C.2 Few-Shot Unsupervised Domain Adaptation

FUDA methods are constrained to utilizing extremely limited supervised and remaining unsupervised information from the source domain to facilitate knowledge transfer. Due to the inherently challenging nature of this task, existing approaches remain relatively limited, with current research primarily focusing on learning reliable discriminative representations and implicitly aligning domain distributions through various self-supervised learning tasks. Specifically, CDS [21] constructs intra-domain self-supervised tasks to learn similarity-based discriminative representations while using inter-domain self-supervised tasks for feature alignment. PCS [22] employs prototype-based self-supervised learning to obtain robust representations while simultaneously achieving semantic structure alignment and feature distribution alignment. Recently, C-VisDiT [23] introduces confidence learning to separate samples into low-confidence and high-confidence groups for self-supervised learning, enabling intra-domain discriminative learning and inter-domain alignment. Recently, MASA [68] enhances few-shot image classification in complex backgrounds by adaptively aligning class-specific subspaces across multiple views to leverage complementary information. These methods have achieved some progress in FUDA tasks, benefiting from the significant advantages of self-supervised learning in few-shot or unsupervised scenarios. However, their introduction of self-supervised learning tasks essentially relies on custom-designed tasks to learn potentially universal representations to address the lack of sufficient supervised information in the source domain. This approach lacks a genuine understanding of the 65actual domain alignment and classification tasks at hand. Such task-agnostic representation learning cannot prevent the inherent loss of discriminative semantic information in the latent space learning under label-sparse conditions. Therefore, we propose to approach the problem from an original-space modeling perspective, fundamentally preventing negative transfer caused by the loss of discriminative semantic information in latent space learning when supervised information is scarce.

Appendix D Proposed method

In this section, we first revisit the preliminaries including the setting of few-shot unsupervised domain adaptation and diffusion models. Then we introduce the prototype-guided diffusion alignment (PGDA) framework and demonstrate the whole training algorithm, which is composed of class-prototype learning and diffusion alignment. And finally, we offer a theoretical explanation of our approach.

Appendix D.1 Preliminaries

Problem Definition In FUDA scenario, the source domain $\mathcal{D}_s = \{\mathcal{D}_{ls}, \mathcal{D}_{us}\}$ is consisted of the few labeled source data $\mathcal{D}_{ls} = \left\{(\boldsymbol{x}_i^{ls}, \boldsymbol{y}_i^{ls})\right\}_{i=1}^{n_{ls}}$ and some unlabeled source data $\mathcal{D}_{us} = \left\{(\boldsymbol{x}_i^{us})\right\}_{i=1}^{n_{us}}$, where $n_{ls} << n_{us}$. The remaining settings are still consistence with the vanilla UDA task, a completely unlabeled target domain $\mathcal{D}_t = \left\{(\boldsymbol{x}_i^{t})\right\}_{i=1}^{n_t}$ is given, the source domain and target domain share the same feature space $\mathcal{X}_S = \mathcal{X}_T$ and label space $\mathcal{Y}_S = \mathcal{Y}_T$, their distributions follow the Covariate Shift [43] assumption, i.e., the marginal distribution $\mathcal{P}_S(\boldsymbol{x}_S) \neq \mathcal{P}_T(\boldsymbol{x}_T)$ and the conditional distribution $\mathcal{P}_S(\boldsymbol{y}_S \mid \boldsymbol{x}_S) \neq \mathcal{P}_T(\boldsymbol{y}_T \mid \boldsymbol{x}_T)$. The goal of FUDA is to leverage $\mathcal{D}_{ls}$, $\mathcal{D}_{us}$, and $\mathcal{D}_t$ simultaneously during model training to achieve accurate predictions on the target task. Following previous works, here we focus on the classification task.

Diffusion Model Diffusion Models (DMs) [44] have achieved great performance in various generative tasks, which acquire the image distribution from the Gaussian distribution by learning an image denoising task. Given a original image $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$, DMs obtain the corresponding noisy images $\{\boldsymbol{x}_t\}_{t=1}^T$ by adding the random Gaussian noises $\{\epsilon_t\}_{t=1}^T \sim \mathcal{N}(0,1)$

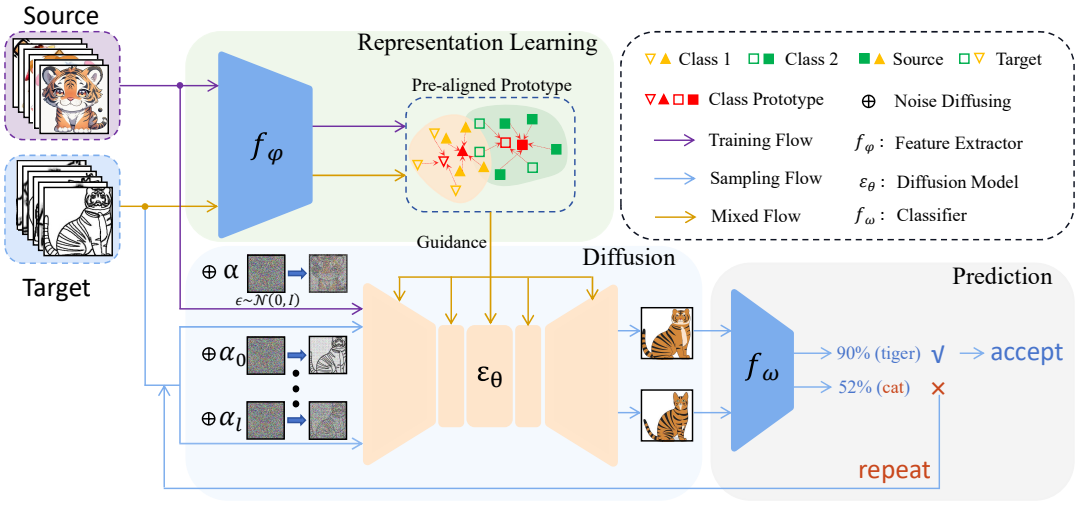


Figure C1 Illustration of PGDA. The feature extractor f_ϕ is trained by the prototype learning to guide the knowledge transfer of the alignment module. The diffusion model ϵ_θ trained on the source domain is responsible for transforming the noise-added target domain distribution into the source domain distribution under the guidance of class prototypes, which is supervised by the classifier f_ω pre-trained on the source domain, accepting high-confidence samples while resampling low-confidence samples with an increased noise scale. Additionally, gray flow, blue flow and yellow flow represent the source data, target data, and both of them respectively. The diffusion model adopts a U-Net architecture, where the light orange blocks on the left and right correspond to the downsampling and upsampling modules, respectively, while the three blocks in the middle denote the bottleneck modules. These modules are specific layers within the diffusion model.

gradually. The image denoising task is to fit the noise by training a parameterized neural network $\epsilon_\theta(\cdot, t)$, i.e. the U-Net [45]. Resorting to the maximization of the evidence lower bound [46], the objective function is trained by:

$$\mathcal{L}_{DM} = \mathbb{E}_{\boldsymbol{x}_0,\epsilon}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right] \tag{D1}$$

where $\alpha_t = \prod_{i=1}^{t}(1-\beta_i)$ and $\beta_i \in (0,1)$ measures the rate of noising. According to DDPM [44], the images can be generated by iteratively predicting noises until $t = 1$:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t, t\right)\right) + \sigma_t\boldsymbol{\epsilon}_t \tag{D2}$$

Here, $\mathbf{x}_T$ sampled from the random Gaussian noise. Further, a faster sampling process DDIM [47] is proposed:

$$\begin{aligned}\boldsymbol{x}_{t-1} = &\sqrt{\alpha_{t-1}}\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t)}{\sqrt{\alpha_t}}\right)\\ &+\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t,t\right) + \sigma_t\boldsymbol{\epsilon}_t\end{aligned} \tag{D3}$$

where $\sigma_t = \zeta\sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{1-\alpha_t/\alpha_{t-1}}$, when $\zeta = 1$, the generative process is equal to a DDPM and when $\zeta = 0$, the generative process becomes a deterministic process without the random noise ϵ_t. Here, we choose the deterministic DDIM sampling as the framework to conduct the reconstruction task.

Appendix D.2 Guided Class-Prototype Learning

We aim to learn class prototypes [48] to guide the distribution transformation of diffusion models. In consideration of the FUDA scenario, where the source domain lacks sufficient supervision, we adopt a progressive strategy to select high-confidence samples to ensure the reliability of class prototypes. Specifically, we utilize a shared convolutional neural network (CNN) $f_\varphi(\cdot)$ to extract features $f_i = f_\varphi(\boldsymbol{x}_i)$ from both the source and target domains. The stored semantic feature $\hat{\boldsymbol{f}}_i$ of sample $\boldsymbol{x}_i$, is initialized with $\boldsymbol{f}_i$ and updated as $\hat{\boldsymbol{f}}_i = m\hat{\boldsymbol{f}}_i + (1-m)\boldsymbol{f}_i$ with a momentum m after each batch. The source domain class prototypes are initialized as the average of labeled source samples. For each class cluster $c_j^s \in c^s = \{c_i^s\}_{i=1}^k$, the prototype of class j is computed as the normalized mean of its corresponding features, i.e, $\mu_j^s = \frac{\mathbf{u}_j^s}{\left\|\mathbf{u}_j^s\right\|}$, where $\mathbf{u}_j^s = \frac{1}{\left|c_j^s\right|}\sum_{\hat{\boldsymbol{f}}_i^s \in c_j^s}\hat{\boldsymbol{f}}_i^s$, $\hat{\boldsymbol{f}}_i^s$ is the stored semantic feature of source sample $\boldsymbol{x}_i^s$. To select the high-confidence samples, we calculate the cosine similarity between the source samples $\{\boldsymbol{f}_i^s\}$ and the source class-prototype $\{\mathbf{u}_j^s\}_{j=1}^k$ as $P_i^s = \left\{P_{i,j}^s\right\}_{j=1}^k$. So the confidence gap of the i-th sample can be defined as

$$g_i = \mathbf{max}(P_i^s) - \mathbf{secmax}(P_i^s) \tag{D4}$$

where the $\mathbf{max}(\cdot)$ and $\mathbf{secmax}(\cdot)$ return the maximum and the second maximum value respectively. Then the mean similarity ms_j and mean gap mg_j of the j-th class is calculate as

$$\begin{aligned} ms_j &= \frac{1}{\left|c_j^s\right|} \sum_{C(i)=\mathbf{u}_j^s} \mathbf{max}\left(P_i\right), \\ mg_j &= \frac{1}{\left|c_j^s\right|} \sum_{C(i)=\mathbf{u}_j^s} g_i, j = 1, 2, \ldots, k \end{aligned} \tag{D5}$$

where $C(i)$ represents the corresponding class-prototype of the i-th sample. We select the high confidence samples which have high cosine similarity and confidence gap by the condition:

$$\mathbf{max}\left(P_i\right) > ms_j \; and \; g_i > mg_j, j = \mathbf{argmax}\left(P_i\right) \tag{D6}$$

Through selecting high-confidence samples based on Eq.D6 at each iteration, the j-th class-prototype for the current r-th round can be progressively updated as follows:

$$\mathbf{u}_j^{s(r)} = -\sum_{\hat{\boldsymbol{f}}_i^s \in c_j^s} \frac{\exp(\hat{\boldsymbol{f}}_i^s \cdot \mathbf{u}_j^{s(r-1)}/\phi)}{\sum_{\hat{\boldsymbol{f}}^s \in c_j^s} \exp(\hat{\boldsymbol{f}}^s \cdot \mathbf{u}_j^{s(r-1)}/\phi)} \hat{\boldsymbol{f}}_i^s \tag{D7}$$

where ϕ is the temperature value. This strategy aims to assign different update weights for selected samples based on the similarity of these samples to class prototypes which could mitigate the impact of misclassified samples. Similarly, we utilize the updated source domain prototype to initialize the target domain prototype, and then update the target domain prototype $\{\mathbf{u}_j^t\}_{j=1}^k$ by selecting high-confidence samples using the same strategy.

Utilizing the confidence-based progressive strategy can effectively compute class prototypes, yet we believe that an effective guiding prototype should meet the following criteria: 1. Semantic Invariance: the class prototype should ensure that the category of a sample remains unchanged after the distribution transformation; 2. Spatial Proximity: the class prototypes of the source and target domains should be as spatially close as possible in their distributions, thereby facilitating effective distribution transformation.

Semantic Invariance Learning. The semantic invariance learning is essentially resorting to the guiding prototypes of the corresponding categories as supervisory conditions which guarantee that the semantics of the samples remain unchanged during the distribution alignment process. To achieve this purpose, the class prototypes should sufficiently support the sample set of their respective categories, meaning that each class cluster is as compact as possible. Hence, we employ contrastive learning on intra-domain samples, encouraging the feature f_i of each sample outputted by the extractor f_φ to move closer to the prototype of its respective class cluster:

$$\mathcal{L}_{SIL} = -\sum_{f_i \in D_{ls} \cup D_{us} \cup D_t} \log \frac{\exp(C(f_i) \cdot f_i/\tau)}{\sum_{j=1}^k \exp\left(\mu_j \cdot f_i/\tau\right)} \tag{D8}$$

where μ_j denotes the j-th class prototype of the domain corresponding to vector f_i and temperature parameter τ controls the concentration level of the similarity distribution.

Spatial Proximity Learning. Considering the lack of adequate supervision in FUDA scenarios, which makes feature alignment more challenging, we introduce spatial proximity learning instead of direct feature alignment. The key difference lies in the fact that spatial proximity learning can be viewed as domain-level pre-alignment-it coarsely reduces the distance between domain distributions without performing fine-grained, sample-level alignment. On the one hand, such coarse alignment is easy and can be effectively implemented even in sparsely labeled scenarios, thereby reducing the complexity of distribution transformation. On the other hand, given the inevitable issue of noisy labels in sparsely labeled settings, this method helps mitigate the risk of negative transfer caused by sample-level alignment. Specifically, we utilize the spatial proximity learning loss that minimizes the distance between the corresponding class prototype of each sample and the class prototype in the reverse domain:

$$\mathcal{L}_{\mathrm{SPL}} = -\sum_{f_i \in D_{ls} \cup D_{us} \cup D_t} \log \frac{\exp\left(C(f_i) \cdot C^*(f_i)/\tau\right)}{\sum_{j=1}^k \exp\left(\mu_j^* \cdot C(f_i)/\tau\right)} \tag{D9}$$

where $C^*(f_i)$ represents the corresponding class-prototype of the f_i in the reverse domain and similarly, μ_j^* denote the i-th class-prototype in the reverse domain of f_i.

Appendix D.3 Prototype-Guided Diffusion Alignment

As discussed above, to avoid the negative transfer issue caused by latent space learning in sparse-label UDA tasks, we consider modeling the distribution alignment problem in the full data space. While existing generative models [49] are inherently capable of learning and aligning distributions in the full space, DMs exhibit distinct advantages in terms of generation quality, distribution fitting capability, and the smoothness of distribution transformation [50, 51]. Most importantly, we aim to embed class prototypes as guiding conditions during the modeling process to ensure semantic invariance [52] in distribution transformation and boost alignment. The progressive learning process of DMs inherently decouples the data distribution from conditional guidance while explicitly modeling the underlying data probability distribution. This property enables us to directly represent class prototypes as conditional probabilities and seamlessly integrate them in a classifier-free manner [53], providing greater flexibility and convenience. Therefore, we choose DMs to model the distribution alignment

problem after considering the overall situation. Specifically, we train the diffusion model using source domain samples while embedding source domain class prototypes as guiding conditions:

$$\mathcal{L}_{DA} = \mathbb{E}_{\boldsymbol{x}_0,\epsilon}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t, C(\boldsymbol{x}_0), t\right)\|^2\right] \tag{D10}$$

where the parameterized neural network $\boldsymbol{\epsilon}_\theta$ take the class prototype c^s of source domain as the guidance which will be randomly discard as a $\varnothing$ condition in the training process with probability p_{uncond}.

Once the diffusion model adequately learns the source domain distribution, we perform resampling on the target domain samples to transform the target domain distribution into the source domain distribution, thereby achieving alignment between the target and source domain distributions. Specifically, given the standard Gaussian noise ϵ, we first diffuse the target domain sample $\boldsymbol{x}$ into a noise space by constructing the linear combine $\hat{\boldsymbol{x}}_{s_l} = s_l\boldsymbol{x} + \sqrt{1-s_l^2}\epsilon$ at the noise scale s_l, and then sample them into the source domain space by their corresponding target domain class prototypes:

$$\begin{aligned}\hat{\boldsymbol{x}}_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{\hat{\boldsymbol{x}}_t - \sqrt{1-\alpha_t}\tilde{\boldsymbol{\epsilon}}_\theta(\hat{\boldsymbol{x}}_t, C(\boldsymbol{x}), t)}{\sqrt{\alpha_t}}\right) \\ + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\tilde{\boldsymbol{\epsilon}}_\theta\left(\hat{\boldsymbol{x}}_t, C(\boldsymbol{x}), t\right)\end{aligned} \tag{D11}$$

where $t \in [1, ..., s_l]$ and the noise $\tilde{\boldsymbol{\epsilon}}_\theta$ is predicted as a linear combination of conditional and unconditional scores:

$$\tilde{\boldsymbol{\epsilon}}_\theta\left(\hat{\boldsymbol{x}}_t, C(\boldsymbol{x}), t\right) = (1+w)\boldsymbol{\epsilon}_\theta\left(\hat{\boldsymbol{x}}_t, C(\boldsymbol{x}), t\right) - w\boldsymbol{\epsilon}_\theta\left(\hat{\boldsymbol{x}}_t, t\right). \tag{D12}$$

The unconditional score $\boldsymbol{\epsilon}_\theta\left(\hat{\boldsymbol{x}}_t, t\right) = \boldsymbol{\epsilon}_\theta\left(\hat{\boldsymbol{x}}_t, \varnothing, t\right)$ is the conditional score replacing the condition $C(\boldsymbol{x})$ with a null token $\varnothing$.

Here, we focus on the design of the noise scale s_l, which determines the level of noise used to sample target domain data. As discussed in the image editing task [54], s_l controls the reality and faithfulness of the generated images. Analogously, in the context of the proposed distribution alignment task, it governs the performance of alignment and the extent to the preserved semantic information, where an inappropriate choice of s_l can significantly hinder the adaptation process. For example, if the noise scale is too small, the alignment effect will be weak, and the preserved semantic information will include a large amount of redundancy, making classification difficult. On the contrary, if the noise scale is too large, the alignment will be overly aggressive and the discriminative semantic features will be missing, which can lead to class misalignment (i.e. negative transfer). Additionally, in practical tasks, different samples exhibit varying distances from the training distribution and contain differing levels of semantic information, which further complicates the choice of an appropriate noise scale. Therefore, we design a confidence-based adaptive noise diffusion strategy to progressively learn the optimal noise scale for each sample. Specifically, the noise scale is predefined as a sequence $\{s_l\}_{l=0}^L$ and a source domain classifier f_ω is trained as

$$\mathcal{L}_{CLS} = \frac{1}{n^s}\sum_{i=1}^{n^s}\mathbf{CE}\left(\delta(f_\omega(\boldsymbol{x}_i^s), \hat{\boldsymbol{y}}_i^s\right) \tag{D13}$$

where $\mathbf{CE}(\cdot)$ is the cross-entropy loss, $\hat{\boldsymbol{y}}_i^s$ is the pseudo label obtain by the class prototypes and $\delta(\cdot)$ is the soft-max function which returns a C-dimensional vector.

For our diffusion alignment task, a key indication that target domain samples are well-aligned to the source domain is their outstanding classification performance on the source domain classifier. Therefore, we sample each target domain sample starting from the initial noise scale s_0 and use the classifier f_ω to predict the maximum probability of the corresponding sample as a measure of confidence. If the confidence exceeds the predefined threshold η, it indicates that the current noise scale is appropriate, and we accept the classification result for that sample. Conversely, if the confidence falls below the threshold, it suggests that the sample requires alignment with a stronger noise scale. In this case, we reject the current prediction and resample this sample using a larger noise scale sequentially selected from the noise schedule. If, after iterating through the final noise scale s_L, the confidence still does not meet the threshold, we accept the prediction with the highest probability among all iterations. This confidence-based noise scale strategy essentially assigns smaller noise scales to samples closer to the source domain distribution and larger noise scales to samples farther from the source domain distribution. On the one hand, this approach enables a more flexible and adaptive noise scale assignment for each individual sample. On the other hand, it significantly reduces the computational cost of denoising. We use the averaged maximum prediction probabilities across all source domain samples as the threshold η, thereby adaptively adjusting the noise scale based on the resampling confidence of each sample.

Appendix D.4 Overall Objective Function

Eventually, we integrate all the components presented in the previous subsections, including the feature extractor f_φ, the noise predictor ϵ_θ, and the classifier f_ω, and the overall optimization objective of the final PGDA model as follows:

$$\min_{f_\varphi, f_\omega}\ \mathcal{L}_{CLS} + \lambda_1\mathcal{L}_{SIL} + \lambda_2\mathcal{L}_{SPL} \tag{D14}$$

$$\min_{\boldsymbol{\epsilon}_\theta}\mathcal{L}_{DA} \tag{D15}$$

where λ_1 and λ_2 are the balancing hyper-parameters. The complete training procedure is summarized in Algorithm D1.

Appendix D.5 Theoretical Explanation of Alignment

Here, we provide a theoretical explanation of the motivation for using diffusion models for distribution alignment, where we demonstrate that the process of adding noise and subsequently denoising can effectively transform the target domain distribution into the source domain distribution, thereby achieving distribution alignment.

Theorem 1. Given a diffusion model trained on the source distribution $p(\boldsymbol{x})$, let p_t denote the distribution at time t in the forward transformation, let $\bar{p}(\boldsymbol{x})$ denote the output distribution when the input of the backward process is standard Gaussian noise ϵ whose distribution is denoted by $\rho(\boldsymbol{x})$, let $\omega(\boldsymbol{x})$ denote the output distribution when the input of backward process is a convex combination $\hat{\boldsymbol{X}} = (1-\alpha)\boldsymbol{X}' + \alpha\boldsymbol{\epsilon}$, where random variable $\boldsymbol{X}'$ is sampled following the target distribution $q(\boldsymbol{x})$ and $\alpha \in (0,1)$. Under some regularity conditions listed below, we have

$$KL(p\|\omega) \leqslant \mathcal{J}_{SM} + KL\left(p_T\|\rho\right) + \mathcal{F}(\alpha) \tag{D16}$$

To prove Theorem 1, we make the following assumptions. Assumptions (i) to (xii) are required for implementing Theorem 1 in [33]. Specifically, assumptions (i) & (ii) require the source distribution and noise distribution to be differentiable and have finite variance, assumptions (iii)-(iv) require f or the difference of f in diffusion sampling (SDE form [54]) to be bounded corresponding to its input or the difference of its inputs, assumption (iv) requires g in diffusion sampling (SDE form [54]) to be non-zero, assumption (vi) requires p_t and its derivative to be bounded. assumptions (vii)-(viii) require the score function of p_t or the difference of it to be bounded corresponding to its input or the difference of its inputs, assumptions (ix)-(x) require that the estimated score function or the difference of it to be bounded corresponding to its input or the difference of its inputs, assumption (x) requires the estimation error is not infinitely large, assumption (xii) requires the value of $\boldsymbol{X}$ to be bounded, e.g., bounding the values to $[0, 255]$ for images. Assumption (xiii) is used to constrain that $\mathcal{F}$ has a finite first-order derivative.

(i) $p(\boldsymbol{X}) \in \mathcal{C}^2$ and $\mathbb{E}_{\boldsymbol{X}\sim p}\left[\|\boldsymbol{X}\|_2^2\right] < \infty$.
(ii) $\omega_T(\boldsymbol{X}) \in \mathcal{C}^2$ and $\mathbb{E}_{\boldsymbol{X}\sim\omega_T}\left[\|\boldsymbol{X}\|_2^2\right] < \infty$.
(iii) $\forall t \in [0,T] : f(\cdot,t) \in \mathcal{C}^1, \exists C > 0 \forall \boldsymbol{X} \in \mathbb{R}^D, t \in [0,T] : \|f(\boldsymbol{X},t)\|_2 \leqslant C\left(1 + \|\boldsymbol{X}\|_2\right)$.
(iv) $\exists C > 0, \forall \boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^D : \|f(\boldsymbol{X},t) - f(\boldsymbol{Y},t)\|_2 \leqslant C\|\boldsymbol{X} - \boldsymbol{Y}\|_2$.
(v) $g \in \mathcal{C}$ and $\forall t \in [0,T], |g(t)| > 0$.
(vi) For any open bounded set $\mathcal{O}$, $\int_0^T \int_{\mathcal{O}} \|p_t(\boldsymbol{X})\|_2^2 + Dg(t)^2 \left\|\nabla_{\boldsymbol{X}} p_t(\boldsymbol{X})\right\|_2^2 \, \mathrm{d}\boldsymbol{X} \, \mathrm{d}t < \infty$.
(vii) $\exists C > 0 \forall \boldsymbol{X} \in \mathbb{R}^D, t \in [0,T] : \|\nabla_{\boldsymbol{X}} \log p_t(\boldsymbol{X})\|_2 \leqslant C\left(1 + \|\boldsymbol{X}\|_2\right)$.
(viii) $\exists C > 0, \forall \boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^D : \|\nabla_{\boldsymbol{X}} \log p_t(\boldsymbol{X}) - \nabla_{\boldsymbol{Y}} \log p_t(\boldsymbol{Y})\|_2 \leqslant C\|\boldsymbol{X} - \boldsymbol{Y}\|_2$.
(ix) $\exists C > 0 \forall \boldsymbol{X} \in \mathbb{R}^D, t \in [0,T] : \|s_{\boldsymbol{\theta}}(\boldsymbol{X},t)\|_2 \leqslant C\left(1 + \|\boldsymbol{X}\|_2\right)$.
(x) $\exists C > 0, \forall \boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^D : \|s_{\boldsymbol{\theta}}(\boldsymbol{X},t) - s_{\boldsymbol{\theta}}(\boldsymbol{Y},t)\|_2 \leqslant C\|\boldsymbol{X} - \boldsymbol{Y}\|_2$.
(xi) Novikov's condition: $\mathbb{E}\left[\exp\left(\frac{1}{2}\int_0^T \|\nabla_{\boldsymbol{X}} \log p_t(\boldsymbol{X}) - s_{\boldsymbol{\theta}}(\boldsymbol{X},t)\|_2^2 \, \mathrm{d}t\right)\right] < \infty$.
(xii) $\forall t \in [0,T] \exists k > 0 : p_t(\boldsymbol{X}) = O\left(e^{-\|\boldsymbol{X}\|_2^k}\right)$ as $\|\boldsymbol{X}\|_2 \to \infty$.
(xiii) $\exists C_1 > 0$ and $C_2 > 0 : \left|\mathbb{E}_{\boldsymbol{X}\sim q}[\boldsymbol{X}]\right| < C_1$ and $\left|\mathbb{E}_{\boldsymbol{X}\sim p_T}[\boldsymbol{X}]\right| < C_2$.

Proof 1. Here we can bound the KL-divergence: Sampling a variable $\boldsymbol{x}^t$ from the target domain, which can be diffused as $\boldsymbol{x} = (1-\alpha)\boldsymbol{x}^t + \alpha\epsilon$. Since that $\boldsymbol{x}^t$ and the noise ϵ are independent, the generated distribution can be expressed in an convolution form, hence we can derive $\omega_T(\boldsymbol{x})$ as:

$$\begin{aligned}
\omega_T(\boldsymbol{x}) &= \int \frac{1}{\alpha(1-\alpha)} \rho\left(\frac{\boldsymbol{x}-\boldsymbol{\tau}}{\alpha}\right) q\left(\frac{\boldsymbol{\tau}}{1-\alpha}\right) d\boldsymbol{\tau} \\
&= \int \frac{1}{\alpha(1-\alpha)} \frac{1}{\sqrt{2\pi}} e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\tau}\|_2^2}{2\alpha^2}} q\left(\frac{\boldsymbol{\tau}}{1-\alpha}\right) d\boldsymbol{\tau} \\
&= \int \frac{1}{\alpha(1-\alpha)} \frac{1}{\sqrt{2\pi}} e^{-\frac{\|\boldsymbol{x}\|_2^2}{2\alpha^2}} e^{-\left(\frac{\|\boldsymbol{\tau}\|_2^2}{2\alpha^2} - \frac{\boldsymbol{x}^T\boldsymbol{\tau}}{\alpha^2}\right)} q\left(\frac{\boldsymbol{\tau}}{1-\alpha}\right) d\boldsymbol{\tau} \\
&= \frac{1}{\alpha}\rho\left(\frac{\boldsymbol{x}}{\alpha}\right) \int \frac{1}{1-\alpha} e^{-\left(\frac{\|\boldsymbol{\tau}\|_2^2}{2\alpha^2} - \frac{\boldsymbol{x}^T\boldsymbol{\tau}}{\alpha^2}\right)} q\left(\frac{\boldsymbol{\tau}}{1-\alpha}\right) d\boldsymbol{\tau} \\
&= \frac{1}{\alpha}\rho\left(\frac{\boldsymbol{x}}{\alpha}\right) \int e^{-\frac{1}{2\alpha^2}\left[(1-\alpha)^2\|\boldsymbol{\nu}\|_2^2 - 2(1-\alpha)\boldsymbol{x}^T\boldsymbol{\nu}\right]} q(\boldsymbol{\nu}) d\boldsymbol{\nu} \\
&= \rho(\boldsymbol{x}) \int e^{-\frac{1}{2\alpha^2}\left[(1-\alpha)^2\|\boldsymbol{\nu}\|_2^2 - 2(1-\alpha)\boldsymbol{x}^T\boldsymbol{\nu}\right]} q(\boldsymbol{\nu}) d\boldsymbol{\nu} \\
&= \rho(\boldsymbol{x})\mathcal{H}(\alpha,\boldsymbol{x}),
\end{aligned} \tag{D17}$$

Further, the KL-divergence between the aligned sample distribution $\omega_T(\boldsymbol{x})$ and the noisy sample distribution p_T can be calculated as:

$$\begin{aligned}
KL\left(p_T\|\omega_T\right) &= \int p_T(\boldsymbol{x}) \log \frac{p_T(\boldsymbol{x})}{\omega_T(\boldsymbol{x})} d\boldsymbol{x} \\
&= \int p_T(\boldsymbol{x}) \left[\log \frac{p_T(\boldsymbol{x})}{\rho(\boldsymbol{x})} - \log \mathcal{H}(\alpha,\boldsymbol{x})\right] d\boldsymbol{x} \\
&= KL\left(p_T\|\rho\right) - \int p_T(\boldsymbol{x}) \log \mathcal{H}(\alpha,\boldsymbol{x}) d\boldsymbol{x} \\
&= KL\left(p_T\|\rho\right) + \mathcal{F}(\alpha)
\end{aligned} \tag{D18}$$

Table D1 Statistics of the benchmarks

Dateset	#Sample	#Class	#Sub-domain
Office-31	4110	31	Amazon(A), Webcam(W), DSLR(D)
Office-Home	15500	65	Art(Ar), Clipart(Cl), Product(Pr), Real(Rw)
VisDA-C	207785	12	Real(R), Synthesis(S)
DomianNet	145145	126	Clipart(C), Real(R), Painting(P), Sketch(S)

where it is easy to calculate that $\mathcal{F}(1)=0$ and $\mathcal{F}'(1)=\mathbb{E}_{\boldsymbol{X}\sim q}[X]\mathbb{E}_{\boldsymbol{X}\sim p_T}[X]$, then $\mathcal{F}(\alpha)$ can be rewritten as

$$\mathcal{F}(\alpha)=(\alpha-1)\mathbb{E}_{\boldsymbol{X}\sim q}[\boldsymbol{X}]^T\mathbb{E}_{\boldsymbol{X}\sim p_T}[\boldsymbol{X}]+o\left((\alpha-1)^2\right) \tag{D19}$$

Take Eq. D18 and Eq. D19 in to the Theorem 1 of the reference [33], the KL-divergence between the source domain distribution p and aligned sample distribution ω can be written as:

$$KL(p\|\omega)\leqslant \mathcal{J}_{DA}+KL\left(p_T\|\rho\right)+\mathcal{F}(\alpha) \tag{D20}$$

where the distribution discrepancy is bounded by three terms. The first term is the intrinsic loss of the diffusion model, which decreases gradually as the diffusion model is well-trained. The second term, the KL divergence, measures the distributional difference between the noisy samples and Gaussian noise, which will approach zero when the distributions p_T and ρ become identical as the noise step increases to T. The third term is controlled by the noise coefficient α, as shown in Eq. D19, when the noise step $t\to T$, having $\alpha\to 1$, this term also converges to zero. From this, it can concluded that we can align the distributions between the source and target domain by DMs, which is jointly controlled by the training of the diffusion model and the noise steps. Therefore, we leverage a conditional diffusion model to further reduce the training loss and adaptively select the noise scale to control the alignment strength.

Algorithm D1 Optimization Algorithm for PGDA

Initialize: Feature extractor $f_\varphi(\cdot)$, Source domain classifier $f_\omega(\cdot)$, Noise predictor $\boldsymbol{\epsilon}_\theta(\cdot)$, Noise sequence $\{s_l\}_{l=0}^{L}$, Confidence threshold η

 # training:

repeat
  Optimize f_φ, f_ω based on Eq. (D14); *(Train the Extractor and Classifier)*
until Convergence;
repeat
  Optimize $\boldsymbol{\epsilon}_\theta$ based on Eq. (D15); *(Train the Diffusion Model)*
until Convergence;

 # sampling:

$i=0$;
for $\boldsymbol{x}\leftarrow \boldsymbol{x}_1$ to $\boldsymbol{x}_{nt}$ **do**
  $i++$;
  for $s_l\leftarrow s_1$ to s_L **do**
    Sample Diffusion $\hat{\boldsymbol{x}}_{s_l}=s_l\boldsymbol{x}+\sqrt{1-s_l^2}\epsilon$;
    repeat
      Caculate $\hat{\boldsymbol{x}}_{t-1}$ based on Eq. (D10); *(Sample Denoising)*
    until The step $t=1$;
    if $\max(\delta(f_\omega(\hat{\boldsymbol{x}}_0)))\geqslant\eta$; *(Select the confident sample)*
    then
      $\boldsymbol{z}_i^l\leftarrow\hat{\boldsymbol{x}}_0$, break; *(Accept this sampling)*
    else
      $\boldsymbol{z}_i^l\leftarrow\hat{\boldsymbol{x}}_0$, continue; *(Reject this sampling)*
    end if
  end for
end for
$\boldsymbol{z}_i=\arg\max_{\boldsymbol{z}_i^l}\left\{\max(\delta(f_\omega(\boldsymbol{z}_i^l)))\right\}$; *(Select the most confident sampling)*
$\arg\max_{p_i}(\delta(f_\omega(\boldsymbol{z}_i)))$; *(Caculate the prediction of each sample)*
return Target domian prediction $\{p_i\}_{i=1}^{nt}$.

Table E1 Adaptation accuracy (%) comparison on 1-shot / 3-shots labeled source per class on the Office-31 dataset

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg
Source Only	31.3/49.0	19.6/43.3	41.3/55.7	58.7/81.8	39.9/49.8	61.9/81.7	42.1/60.2
MME	21.5/51.0	12.2/54.6	23.1/60.2	60.9/89.7	14.0/52.3	62.4/91.4	32.3/66.5
CDAN	11.2/43.7	6.2/50.1	9.1/65.1	54.8/91.6	10.4/57.0	41.6/89.8	22.2/66.2
CAN	25.3/48.6	26.4/45.3	23.9/41.2	69.4/78.2	21.2/39.3	67.3/82.3	38.9/55.8
MDDIA	45.0/62.9	54.5/65.4	55.6/67.9	84.4/93.3	53.4/70.3	79.5/93.2	62.1/75.5
DAD	51.5/66.6	55.2/68.0	64.2/70.7	81.6/90.5	57.9/71.8	84.4/91.0	65.8/76.4
DACDM	49.5/63.8	52.6/66.7	61.9/73.4	78.5/88.6	56.1/70.2	79.9/88.5	63.1/75.2
CDS	52.6/65.1	55.2/68.8	65.7/71.2	76.6/88.1	59.7/71.0	73.3/87.3	63.9/75.3
PCS	60.2/78.2	69.8/82.9	76.1/76.4	90.6/94.1	71.2/76.3	91.8/96.0	76.6/84.0
C-VisDiT	74.1/**82.9**	72.3/86.0	75.7/76.5	**93.2**/**95.0**	76.4/76.9	94.2/**97.0**	81.0/85.7
Ours	**75.9**/82.3	**76.3**/**88.2**	**77.7**/**79.3**	91.7/94.9	**79.1**/**79.9**	**94.7**/96.6	**82.8**/**86.6**

Appendix E Experiment

In this section, we evaluate the effectiveness of our method on four widely used datasets and compare them with state-of-the-art FUDA and UDA methods. Firstly, we introduce the relevant datasets and compared methods, and then, we present and analyze the quantitative comparison results between our method and the baseline methods on these datasets. Finally, we conduct a thorough evaluation of the proposed method through ablation studies, visualizations, and sensitivity analyses to provide a comprehensive understanding of its performance.

Appendix E.1 Setups

Datasets We evaluate the performance of PGDA on datasets with various scales including the small-sized Office-31 [55], the medium-sized Office-Home [56], and the large-sized VisDA-C [57] and DomainNet [58], where the statistics of these datasets are listed in Table D1.

Office-31 is a classic benchmark which contains 4110 images with 31 categories and three domains: *Amazon* (A) with 2817 images, *Dslr* (D) with 498 images, and *Webcam* (W) with 795 images. We construct six transfer tasks such as "A → D" with 1-shot and 3-shots settings.

Office-Home is a more challenging benchmark that consists of 15500 images with 65 categories from four domains: *Art* (Ar), *Clipart* (Cl), *Product* (Pr) and *Real-World* (Rw). Similarly, twelve transfer tasks are conducted for evaluation such as "Ar → Cl, where 3% and 6% labeled source images are employed.

VisDA-C is a large-scale benchmark for synthetic-to-real domain adaptation. It contains 12 categories of two distinct domains: *synthetic* domain with 152397 images and $real-world$ domain with 55388 images. We focus on the synthetic-to-real transfer task with 0.1% and 1% labeled source images.

Domain-Net is currently the largest domain adaptation benchmark. Since the presence of noise in certain domains and categories, we follow the previous work [22] and select a subset of the dataset, which includes over 140000 images with 126 categories and four domains: *Clipart* (C), *Real* (R), *Painting* (P), *Sketch* (S). We construct six transfer tasks such as "R → C" with 1-shot and 3-shots settings.

Table E2 Adaptation accuracy (%) comparison on 3% and 6% labeled source samples per class on the Offce-Home dataset

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
3% labeled source													
Source Only	22.5	36.5	41.1	18.5	29.7	28.6	27.2	25.9	38.4	33.5	20.3	41.4	30.3
MME	4.5	15.4	25.0	28.7	34.1	37.0	25.6	25.4	44.9	39.3	29.0	52.0	30.1
CDAN	5.0	8.4	11.8	20.6	26.1	27.5	26.6	27.0	40.3	38.7	25.5	44.9	25.2
CAN	17.1	30.5	33.2	22.5	34.5	36.0	18.5	19.4	41.3	28.7	18.6	43.2	28.6
MDDIA	21.7	37.3	42.8	29.4	43.9	44.2	37.7	29.5	51.0	47.1	29.2	56.4	39.1
DAD	42.7	54.3	58.9	50.1	54.9	58.0	49.7	42.6	62.8	58.2	46.1	70.7	54.1
DACDM	39.4	47.3	55.6	46.5	49.7	53.2	48.6	39.2	60.0	56.5	43.0	64.9	50.3
CDS	43.8	55.5	60.2	51.5	56.4	59.6	51.3	46.4	64.5	62.2	52.4	70.2	56.2
PCS	42.1	61.5	63.9	52.3	61.5	61.4	58.0	47.6	73.9	66.0	52.5	75.6	59.7
C-VisDiT	44.1	66.8	67.0	54.9	66.4	66.8	60.5	47.9	75.7	67.2	51.6	78.8	62.3
Ours	**47.2**	**69.7**	**69.5**	**59.6**	**69.1**	**67.3**	**63.9**	**50.9**	**78.3**	**68.1**	**56.3**	**80.9**	**65.1**
6% labeled source													
Source Only	26.5	41.3	46.7	29.3	40.4	37.9	35.5	31.6	57.2	46.2	32.7	59.2	40.4
MME	27.6	43.2	49.5	41.1	46.6	49.5	43.7	30.5	61.3	54.9	37.3	66.8	46.0
CDAN	26.2	33.7	44.5	34.8	42.9	44.7	42.9	36.0	59.3	54.9	40.1	63.6	43.6
CAN	20.4	34.7	44.7	29.0	40.4	38.6	33.3	21.1	53.4	36.8	19.1	58.0	35.8
MDDIA	25.1	44.5	51.9	35.6	46.7	50.3	48.3	37.1	64.5	58.2	36.9	68.4	50.3
DAD	43.4	58.4	63.5	52.8	57.0	61.5	53.0	46.5	69.6	63.9	51.3	75.5	58.0
DACDM	41.1	51.6	59.3	49.9	51.2	57.0	55.8	42.6	66.9	60.8	47.9	70.7	54.6
CDS	45.4	60.4	65.5	54.9	59.2	63.8	55.4	49.0	71.6	66.6	54.1	75.4	60.1
PCS	46.1	65.7	69.2	57.1	64.7	66.2	61.4	47.9	75.2	67.0	53.9	76.6	62.6
C-VisDiT	46.5	69.8	**72.5**	60.2	71.2	**71.2**	64.1	49.0	78.5	69.1	52.8	80.1	65.4
Ours	**47.5**	**70.3**	71.9	**61.7**	**72.5**	69.8	**65.6**	**51.0**	**79.8**	**69.7**	**56.0**	**81.3**	**66.4**

Table E3 Adaptation accuracy (%) comparison on 0.1% and 1% labeled samples per class on the VisDA-2017 dataset

Method	Source Only	MME	CDAN	CAN	MDDIA	DAD	DACDM	CDS	PCS	C-VisDiT	Ours
S → R (0.1%)	47.9	55.6	58.0	51.3	68.9	69.5	65.2	34.2	<u>78.0</u>	76.4	**81.7**
S → R (1%)	51.4	69.4	61.5	57.2	71.3	78.7	76.5	67.5	79.0	<u>80.5</u>	**82.5**

Compared Methods We conduct the source only method as the baseline which is training on the labeled source images and classifies the target images. Then we compare our method with the related UDA and FUDA methods. The compared UDA methods include Minimax Entroy (MME) [59], Conditional Domain Adversarial Networks (CDAN) [39], Contrastive Adaptation Network (CAN) [60], Maximum Mean Discrepancy Implicit Alignment (MMDIA) [61], Domain-Adaptive Diffusion (DAD) [41], Domain-Guided Conditional Diffusion Model (DACDM) [42]. The FUDA methods include Cross-Domain Self-supervised Learning (CDS) [21], Prototypical Cross-domain Self-supervised Learning (PCS) [22] and Confdence-based Visual Dispersal Transfer Learning (C-VisDiT) [23]. (We follow the experimental setup of Reference [23]. The results of baseline methods are compared against the standards reported in that paper. For methods not covered there, we report results from our reproduction under the same setting.)

Implementation Details During the class-prototype learning phase, the feature extractor utilizes the ResNet-101 [62] (for DomainNet) and ResNet-50 (for other datasets) pre-trained on ImageNet [63] as the backbone, which is fixed and combined with a classify head as the classifier for prediction, where the optimizer is SGD with a momentum of 0.9, a weight decay of 0.0005, a learning rate of 0.01 and a batch size of 64. The smooth momentum m is set as a fixed value 0.5 and the temperature value is adaptively calculated according to [64]. During the diffusion alignment phase, we utilize the standard U-Net structure [53] pre-trained on ImageNet as the backbone, with 100000 epochs for VisDA-C and DomainNet, and 20000 epochs for other datasets. We follow the Classifier-Free Guidance approach [53] and employ the U-Net architecture to embed class prototypes as conditional information. Specifically, the input class prototype is first mapped into an embedding space compatible with the intermediate feature dimensions of the U-Net via a projection layer. It is then incorporated into the denoising process of the U-Net through a cross-attention mechanism. In this mechanism, the Query is derived from the U-Net's feature maps, while the Key and Value are obtained via linear transformation of the conditional embedding. The implementation is available on GitHub [1)].

Appendix E.2 Experimental Results and Analysis

The quantitative experimental results on Office-31, Office-Home, VisDA-C and DomainNet are shown in Table E1, Table E2, Table E3 and Table E4 respectively, with classification accuracy used as the evaluation metric. It can be observed that specialized FUDA methods significantly outperform general UDA methods. Moreover, our method achieves the best or second-best results on most tasks across the four tested datasets and achieves the best average performance. Specifically, on Office-31, our method outperforms the second-best method by an average of 1.8% in the 1-shot and 0.9% in the 3-shots. Notably, for the transfer tasks between W and D, which are relatively simple with limited room for improvement, our method achieves slightly lower results than the SOTA due to the inherent randomness introduced by the noise injection property of diffusion models. On the larger Office-Home, our method achieves an average improvement of 2.8% with 3% labeled source domain data and 1% with 6% labeled data. On the more challenging VisDA dataset, with only 0.1% labeled source domain data, our method surpasses the SOTA method by over 5%, where remarkably, it even outperforms the SOTA method with 1% labeled data. Similarly, on DomainNet, our method achieves an average improvement of over 3% in both the 1-shot and 3-shots settings. An interesting observation is that the Source Only method, which trains solely on labeled source domain samples without applying domain adaptation, outperforms general UDA methods in certain tasks(i.e. on Office-31 with 1-shot and DomainNet with 3-shots). This suggests that when labeled data is extremely scarce, it is difficult to learn robust discriminative representations, and blindly aligning distributions may have a negative impact. Due to the primary motivation of our method being to address this issue, our method achieves better performance, especially in tasks with fewer labels and greater difficulty our method exhibits even greater advantages. For example, on the 1-shot DomainNet task, our method achieves over x% of improvement, highlighting its superiority in low-resource scenarios.

Appendix E.3 Empirical Analysis

Ablation Study To evaluate the contribution of each component, we conduct the ablation study to investigate their impact on target domain classification performance. The proposed model consists of three main components: the prototype guidance (PG) module, the diffusion alignment (DA) module, and the confidence-based alignment strategy (CS), where the PG module further incorporates Semantic Invariance Learning (SIL) and Spatial Proximity Learning (SPL). Here, the key components are not independent but exhibit a hierarchical dependency: the DA relies on both PG and the CS, while the PG module further includes SIL and SPL. Hence we performed the ablation experiment on these five components and compared them with the complete model. Since the DA module serves as the backbone of the proposed method and cannot be directly removed, we utilize the pre-trained model without any fine-tuning for distribution transformation as its ablation. Without loss of generality, we conduct the experiment on the $A \rightarrow D$ task of Office-31 dataset and report the results in Table E5 (where "w/o" indicates the removal of the corresponding module).

1) https://github.com/sunhy228/PGDA-FUDA

Table E5 Classification accuracy (%) of Ablation study on "A → D" task of Office-31 dataset

Method	w/o PG	w/o SIL	w/o SPL	w/o DA	w/o CS	w/o PG & CS	Ours
1-shot	70.8	71.6	73.2	56.1	67.7	60.5	75.9
3-shots	77.7	79.3	80.5	70.4	76.2	72.1	82.3

Table E6 comparation of classification accuracy (%) in the different alignment space

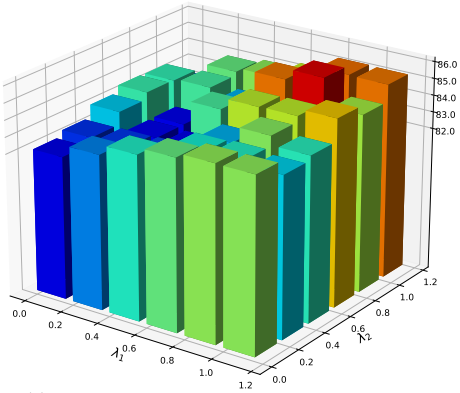
Dataset	Office-Home			VisDA	Office-Home			VisDA
Method	Ar → Cl	Cl → Pr	Pr → Rw	S → R	Ar → Cl	Cl → Pr	Pr → Rw	S → R
Labeled source	3%			0.1%	6%			1%
Latent-4	37.9	55.8	81.7	73.9	41.2	59.5	72.1	77.6
Latent-2	44.6	63.2	73.1	77.2	45.3	69.4	75.7	79.7
Original(Ours)	47.2	69.1	81.7	64.9	47.5	72.5	79.8	82.5

Table E4 Adaptation accuracy (%) comparison on 1-shot and 3-shots per class on the DomainNet dataset

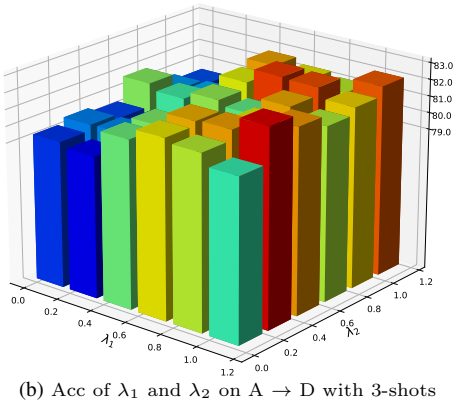
Method	R → C	R→ P	R→ S	P→ C	P→ R	C→ S	S→ P	Avg
1-shot labeled source								
Source Only	18.4	30.6	16.7	16.2	28.9	12.7	10.5	19.1
MME	13.8	29.2	9.7	16.0	26.0	13.4	14.4	17.5
CDAN	16.0	25.7	12.9	12.6	19.5	7.2	8.0	14.6
CAN	18.3	22.1	16.7	13.2	23.9	11.1	12.1	16.8
MDDIA	18.0	30.6	15.9	15.4	27.4	9.3	10.2	18.1
DAD	23.4	30.4	19.7	18.5	27.5	19.0	21.7	22.9
DACDM	21.7	28.8	19.2	17.8	25.6	18.6	20.9	21.8
CDS	16.7	24.4	11.1	14.1	15.9	13.4	19.0	16.4
PCS	39.0	51.7	39.8	26.4	38.8	23.7	23.6	34.7
C-VisDiT	40.1	50.9	43.5	26.9	38.5	26.6	25.1	35.9
Ours	**43.6**	**56.2**	**43.9**	**32.4**	**43.9**	**30.1**	**32.3**	**40.3**
3-shots labeled source								
Source Only	30.2	44.2	25.7	24.6	49.8	24.2	23.2	31.7
MME	22.8	46.5	14.5	25.1	50.0	20.1	24.9	29.1
CDAN	30.0	40.1	21.7	21.4	40.8	17.1	19.7	27.3
CAN	28.1	33.5	25.0	24.7	46.9	23.3	20.1	28.8
MDDIA	41.4	50.7	37.4	31.4	52.9	23.1	24.1	37.3
DAD	44.3	54.9	37.7	38.1	49.6	31.9	35.1	41.7
DACDM	43.8	52.8	35.1	32.9	45.9	28.2	30.6	38.5
CDS	35.0	43.8	36.7	34.1	36.8	31.1	34.5	36.0
PCS	45.2	59.1	41.9	41.0	66.6	31.9	37.4	46.1
C-VisDiT	46.3	59.7	43.8	45.6	**67.9**	34.1	38.2	47.9
Ours	**49.8**	**60.6**	**44.9**	**47.2**	65.5	**37.2**	**38.7**	**49.1**

It can be found that each module contributes significantly to the performance of the proposed model, with the DA module, CS module, and PG module showing the most substantial impact. Specifically, removing the DA module leads to accuracy drops exceeding 10% points in both 1-shot and 3-shots settings, which is understandable as the diffusion model can only transform the target distribution to a pre-trained distribution that significantly differs from the source domain distribution, making it challenging for the source domain classifier to perform effectively. The CS module also considerably influences the method's performance, primarily because samples exhibit inherent discrepancy, and applying the same alignment strength across all samples makes it difficult to achieve optimal alignment results, thereby affecting the final classification. Regarding the PG module, the quality of class prototypes proves crucial, where class prototypes that maintain semantic consistency and distributional proximity effectively enhance both the final alignment and classification performance. Furthermore, it can be observed that removing both PG and CS leads to a more significant performance drop than removing either one individually. This further validates the synergistic role of these components in the diffusion alignment process, particularly the complementary effects of PG and CS in guiding alignment strength and preserving semantic consistency.

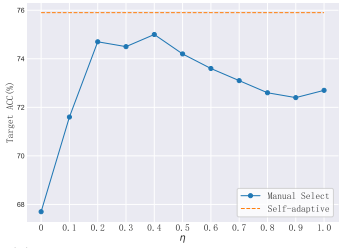
Parameter Sensitivity Analysis As shown in Figure E1, we analyze the sensitivity of various parameters in the $A \rightarrow D$ task of Office-31, including the hyperparameters λ_1 and λ_2 in the prototype learning loss, the noise scale threshold η, and the number of labeled samples per class in the source domain. Figure E1(a) and E1(b) demonstrate the results for hyperparameters λ_1 and λ_2 under 1-shot and 3-shots settings, respectively, where the model achieves better performance



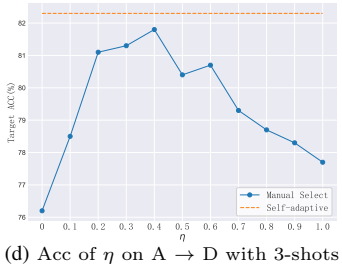
(a) Acc of λ_1 and λ_2 on A $\rightarrow$ D with 1-shot



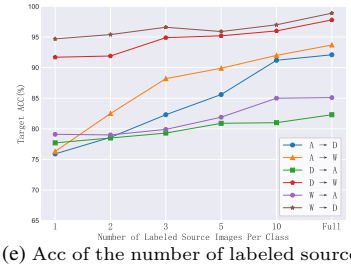
(b) Acc of λ_1 and λ_2 on A $\rightarrow$ D with 3-shots



(c) Acc of η on A $\rightarrow$ D with 1-shot



(d) Acc of η on A $\rightarrow$ D with 3-shots



(e) Acc of the number of labeled source images per class

Figure E1 Parameter sensitivity. (a)(b): Accuracy of target samples varying with parameters λ_1 and λ_1 on task A $\rightarrow$ D (Office-31) with 1-shot (a) or 3-shots (b). (c)(d): Accuracy of target samples varying with the threshold η on task A $\rightarrow$ D (Office-31) with 1-shot (c) or 3-shots (d). (e): Accuracy of target samples varying with the number of labeled source images per class on Office-31.

Figure E2 Demonstration of the alignment performance on Office-Home. The top row shows the source domain samples, the middle row shows the target domain samples, and the bottom row shows the aligned target samples.

when $\lambda_1 \in \{0.6, 0.8, 1.0\}$ and $\lambda_2 \in \{0.2, 0.4, 0.6, 0.8\}$. The effects of noise scale threshold η under 1-shot and 3-shots settings are illustrated in Figure E1(c) and E1(d). The model shows significant sensitivity to the manually selected noise scale, which yields better results within $\{0.2, 0.3, 0.4, 0.5\}$, though still not matching our proposed adaptive confidence-based threshold selection strategy. In terms of the manually selected threshold, when the scale is 0, equivalent to not using the noise scale strategy, the performance is poorest. On a scale of 1, the noise gradually increases to a maximum as almost all samples are rejected, ultimately selecting samples with the highest probability. However, the large noise scale introduces randomness, also resulting in frustrating model performance. Additionally, Figure E1(e) shows the sensitivity analysis of the number of labeled samples per class in the source domain. The performance of the proposed method consistently improves as the number of labeled samples increases, reaching optimal results when all samples are labeled. However, such extensive labeling is impractical in Few-shot Unsupervised Domain Adaptation (FUDA) scenarios, where unlabeled data predominates. Effectively leveraging this unlabeled data remains challenging, particularly for large-scale pre-trained models [65], as fine-tuning rudely on unlabeled data may even degrade performance which explains the scarcity of related research in this area. As shown in Table E7, we also conducted sensitivity experiments varying $w \in \{0, 1, 2, 3, 4\}$. It can be observed that the best results occur at $w = 1, 2$. When the guidance weight exceeds 3, the model tends to overfit the prototypes, slightly reducing target diversity and domain alignment smoothness, which consequently leads to a decline in performance. The sensitivity of the momentum m is shown in Table E8. It can be observed that the model exhibits low sensitivity to the momentum m, achieving good performance within the range of approximately $[0.3, 0.7]$. This is intuitively understandable: due to the nature of few-shot learning, the prototype needs to both retain original supervised information and learn information from unsupervised samples. Therefore, selecting a moderate value contributes to the stability of the model. Furthermore, unlike CDS and PCS, which utilize category prototypes for discrimination, in our approach, category

Table E7 Classification accuracy (%) of Ablation study on "A → D" task of Office-31 dataset

Method	0	1	2	3	4
1-shot	70.8	76.1	75.9	74.7	72.3
3-shots	77.7	81.2	82.3	81.5	79.6

Table E8 Classification accuracy (%) of Ablation study on "A → D" task of Office-31 dataset

Method	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1-shot	72.6	73.5	74.9	75.8	75.9	75.9	75.3	75.7	75.2	74.7	74.1
3-shot	78.2	80.7	81.3	82.2	82.0	82.3	82.4	82.2	82.1	81.6	80.9

prototypes serve only as conditional guidance for diffusion and are balanced through a dynamic noise strategy. As a result, the model's sensitivity to momentum is further reduced.

Time Cost Computational complexity is indeed an important practical concern for diffusion-based methods. Here we compare the proposed method with other diffusion-based method on training and inference time. As shown in Table E9, the proposed not only achieves superior performance but also demonstrates significant advantages in terms of time cost. Specifically, in terms of training time, DAD requires multi-stage diffusion training, and DACDM necessitates simultaneous training on both source and target domains, whereas our method only requires source domain training, thus reducing overall cost. Regarding inference time, while DAD accelerates by reducing inference modules, our method employs an early-exit mechanism, which still maintains a competitive advantage. We would like to note that the computational cost of diffusion-based domain adaptation inherently reflects a trade-off between efficiency and alignment quality. Our method is built upon a relatively standard diffusion framework, and fast diffusion sampling is an active and rapidly evolving research direction. Techniques such as pretrained diffusion models, higher-order or consistency-based samplers, and knowledge distillation are expected to further reduce the computational overhead in future work, potentially pushing the efficiency boundary while preserving the advantages of diffusion-based alignment.

Class-wise Gains As shown in Figure E3, when no confidence threshold is applied, we can find that although the class prototypes are relatively well-aligned, the sample-level alignment appears looser. In contrast, with the confidence threshold applied, although some failing samples are completely aligned to other classes, the overall alignment is improved, where the confidence threshold acts more as an alignment constraint: it selects samples closer to the source domain (high-confidence samples) by rejecting low-confidence ones (those farther from the source domain), thereby facilitating alignment. Additionally, as shown in Table E10, it can be observed that classes such as binders, printers, and trash cans, which initially had lower classification accuracy, exhibit higher accuracy gains under confidence-based sampling. Additionally, with the exception of classes that were already fully correctly classified, all other categories show improvement when the confidence threshold strategy is applied.

Comparison of the Alignment Space To further investigate the impact of alignment spaces on the proposed method, following LDA [66], we conduct experiments by mapping samples into latent spaces before applying PGDA alignment. Without loss of generality, we select three tasks (Ar→ Cl, Cl → Pr and Pr→ Rw) from the Office-Home dataset and a task (S→ R) from the VisDA dataset where we compare alignment results in double-scaled space (Latent-2), quadruple-scaled space (Latent-4), and the original space, as shown in Table E6. The results demonstrate that model performance gradually deteriorates as the alignment space becomes more compressed, with the gap widening particularly under the 3% and 0.1% labeled conditions, which aligns with our hypothesis that in label-constrained scenarios, it becomes challenging to capture a latent semantic space with sufficient discriminative information, consequently affecting both alignment performance and final classification.

Backbone To further verify the generalization ability of our method across different backbones, we have conducted additional experiments using ViT. As shown in Table E11, our method achieves further performance improvement when a stronger backbone is employed. We attribute this improvement to two main factors: on one hand, a more powerful backbone helps extract more discriminative class prototypes, thereby enhancing the accuracy of prototype guidance; on the other hand, as the supervising classifier in the diffusion alignment process, ViT also encourages the model to adopt more effective noise sampling strategies, which in turn strengthen the alignment effect.

Visual Performance As illustrated in Figure E2 and Figure E3, we showcase the visual results of the proposed method from both semantic and space performance. In terms of semantic performance, it can be found that target domain samples become semantically very close to the source domain after distribution alignment, enabling effective classification by the source domain classifier. PGDA successfully complements and corrects background details and textures from the source

Table E9 Comparison of time cost on "A → D" task (1-shot) of Office-31 dataset (with a single 4090 GPU)

Method	Training Time (hour)	Inference Time (min)	Acc (%)
DAD [41]	103.9	7.3	51.5
DACDM [42]	87.6	8.7	49.5
Ours	71.1	5.7	75.9

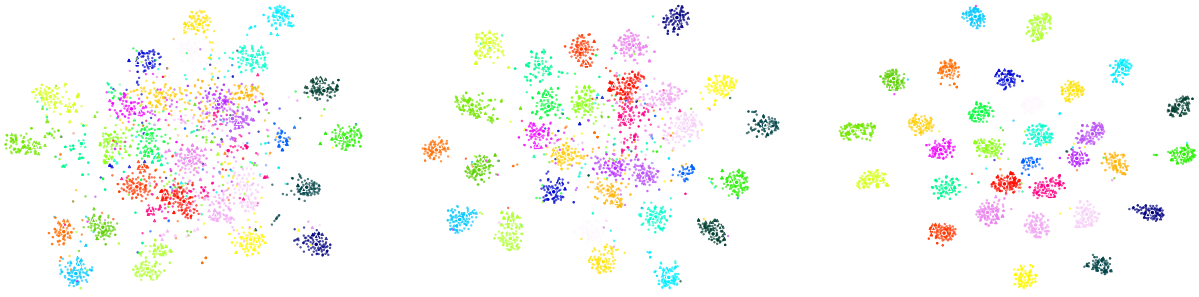
Table E10 Per-class accuracy on task A → D (Office-31)

Class	back_pack	bike	bike_helmet	bookcase	bottle	calculator	desk_chair	desk_lamp
Acc (%)	91.7% → 100%	95.24% →100%	100% →100%	75% →83.3%	100% →100%	100% → 100%	100% →100%	85.71%→ 92.9%
Class	desktop_computer	file_cabinet	headphones	keyboard	laptop_computer	letter_tray	mobile_phone	monitor
Acc (%)	66.67% →73.3%	66.76% →86.7%	100% →100%	70% →90%	66.67% →83.3%	75%→ 87.5%	83.9% →93.3%	95.5% →100%
Class	mouse	mug	paper_notebook	pen	phone	printer	projector	punchers
Acc (%)	100% →100%	100% →100%	100% →100%	90% →100%	76.9% →92.3%	**33.3% →66.7%**	90.9% →100%	33.3% →55.6%
Class	ring_binder	ruler	scissors	speaker	stapler	tape_dispenser	trash_can	
Acc (%)	**40% →80%**	28.6% →57.1%	83.3% →94.4%	84.6% →92.3%	71.4% →85.7%	77.3% →86.4%	**33.3% →66.7%**	

Table E11 Classification accuracy (%) of Ablation study on "R → C" task of DomainNet dataset

Backbone	ResNet101		ViT	
Shot	1-shot	3-shot	1-shot	3-shot
Acc	43.6	49.8	52.6	61.2

domain during the alignment process, making them more consistent with the source domain distribution. Furthermore, resorting to prototype guidance and noise scale control, PGDA effectively preserves the semantic content of target domain subjects, thereby minimizing impact on downstream classification tasks. In terms of space performance, it can be observed that after diffusion alignment, both the target domain samples and prototypes are well-aligned. Moreover, the target domain samples are effectively clustered around the prototypes, which further verifies that the prototypes indeed guide the distribution transformation effectively and promote semantic consistency.



(a) Before alignment (b) After alignment without CG (c) After alignment

Figure E3 t-SNE visualization of (a) baseline, (b) ours without confidence gate and (c) ours on task A → D (Office-31). Source domain samples are represented by circles, target domain samples by triangles, and different classes are distinguished by colors. The class prototypes are marked with white-bordered symbols for clarity.

Appendix F Conclusion

In this paper, we propose a prototype-guided diffusion alignment (PGDA) method for FUDA, aiming to address the negative transfer problem caused by the loss of task-related semantic information during the process of latent space learning under sparsely labeled scenarios. The proposed PGDA method leverages class prototypes to guide the diffusion model in aligning the distribution discrepancy between the source and target domains across the entire space. In addition, PGDA employs a confidence-based strategy to adaptively select the noise scale, ensuring that each sample is aligned while preserving semantic consistency and reducing computational overhead. And we also offer a theoretical upper bound on the alignment error, offering insights into the design motivation of our method. Finally, extensive experiments on various datasets demonstrate the superiority of the proposed method.

Acknowledgements This work was supported by the National Natural Science Foundation of China (62376126, 62076124) and the Fundamental Research Funds for the Central Universities (NS2024058).

References

1 Jordan M I, Mitchell T M. Machine learning: Trends, perspectives, and prospects. Science, 2015, 349(6245): 255-260
2 Pan J, Yang Q. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10): 1345-1359
3 Zhuang F, Qi Z, Duan K, et al. A comprehensive survey on transfer learning. In: Proceedings of the IEEE, 2020, 109(1): 43-76
4 Long M, Zhu H, Wang J, et al. Unsupervised domain adaptation with residual transfer networks. Advances in neural information processing systems, 2016, 29
5 Saito K, Watanabe K, Ushiku Y, et al. Maximum classifier discrepancy for unsupervised domain adaptation. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018. 3723-3732

6 Wilson G, Cook D J. A survey of unsupervised deep domain adaptation. ACM Transactions on Intelligent Systems and Technology, 2020, 11(5): 1-46
7 Jing M, Meng L, Li J, et al. Adversarial mixup ratio confusion for unsupervised domain adaptation. IEEE Transactions on Multimedia, 2022, 25: 2559-2572
8 You F, Li J, Zhu L, et al. Domain adaptive semantic segmentation without source data. In: Proceedings of the 29th ACM international conference on multimedia, 2021. 3293-3302
9 Li G, Ji Z, Qu X. Stepwise domain adaptation (SDA) for object detection in autonomous vehicles using an adaptive CenterNet. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(10): 17729-17743
10 Zhang Y, Wei Y, Wu Q, et al. Collaborative unsupervised domain adaptation for medical image diagnosis. IEEE Transactions on Image Processing, 2020, 29: 7834-7844
11 Guan H, Liu M. Domain adaptation for medical image analysis: a survey. IEEE Transactions on Biomedical Engineering, 2021, 69(3): 1173-1185
12 Zhao S, Fu H, Gong M, et al. Geometry-aware symmetric domain adaptation for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 9788-9798
13 Liu X, Ge Y, Ye P, et al. Recursively conditional gaussian for ordinal unsupervised domain adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision, 2021. 764-773
14 Ye Y, Pan T, Luo T, et al. Learning mlatent representations for generalized zero-shot learning. IEEE Transactions on Multimedia, 2022, 25: 2252-2265
15 Xu Q, Zhang R, Zhang Y, et al. Federated adversarial domain hallucination for privacy-preserving domain generalization. IEEE Transactions on Multimedia, 2023, 26: 1-14
16 Chen Z, Luo Y, Wang S, et al. GSMFlow: Generation shifts mitigating flow for generalized zero-shot learning. IEEE Transactions on Multimedia, 2022, 25: 5374-5385
17 Pejic Bach M, Krstic Z, Seljan S, et al. Text mining for big data analysis in financial sector: A literature review. Sustainability, 2019, 11(5): 1277
18 Javaid M, Haleem A. Industry 4.0 applications in medical field: A brief review. Current Medicine Research and Practice, 2019, 9(3): 102-109
19 Ge C, Huang R, Xie M, et al. Domain adaptation via prompt learning. IEEE Transactions on Neural Networks and Learning Systems, 2025, 1(36): 1160-1170
20 Tang S, Su W, Ye M, et al. Source-free domain adaptation with frozen multimodal foundation model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 23711-23720
21 Kim D, Saito K, Oh T H, et al. Cross-domain self-supervised learning for domain adaptation with few source labels. arXiv preprint arXiv:2003.08264, 2020
22 Yue X, Zheng Z, Zhang S, et al. Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021. 13834-13844
23 Xiong Y, Chen H, Lin Z, et al. Confidence-based visual dispersal for few-shot unsupervised domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 11621-11631
24 Motiian S, Jones Q, Iranmanesh S, et al. Few-shot adversarial domain adaptation. Advances in neural information processing systems, 2017. 30
25 Motiian S, Piccirilli M, Adjeroh D A, et al. Unified deep supervised domain adaptation and generalization. In: Proceedings of the IEEE international conference on computer vision, 2017. 5715-5725
26 Jing T, Xia H, Hamm J, et al. Marginalized augmented few-shot domain adaptation. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(9): 12459 - 12469
27 Huang S, Yang W, Wang L, et al. Few-shot unsupervised domain adaptation with image-to-class sparse similarity encoding. In: Proceedings of the 29th ACM International Conference on Multimedia, 2021. 677-685
28 Yu L, Yang W, Huang S, et al. High-level semantic feature matters few-shot unsupervised domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 37(9): 11025-11033
29 Yang W, Yang C, Huang S, et al. Few-shot unsupervised domain adaptation via meta learning. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), 2022. 1-6
30 Li Y, Long S, Wang S, et al. Prompt-induced prototype alignment for few-shot unsupervised domain adaptation. Expert Systems with Applications, 2025, 269: 126400
31 Jaiswal A, Babu A R, Zadeh M Z, et al. A survey on contrastive self-supervised learning. Technologies, 2020, 9(1): 2
32 Zhang C, Bengio S, Hardt M, et al. Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 2021, 64(3): 107-115
33 Song Y, Durkan C, Murray I, et al. Maximum likelihood training of score-based diffusion models. Advances in neural information processing systems, 2021. 34: 1415-1428
34 Long M, Cao Y, Wang J, et al. Learning transferable features with deep adaptation networks. In: Proceedings of the International conference on machine learning, PMLR, 2015. 97-105
35 Sun B, Saenko K. Deep coral: Correlation alignment for deep domain adaptation. In: Proceedings of the Computer vision-ECCV 2016 workshops: Amsterdam, the Netherlands, October 8-10 and 15-16, 2016, proceedings, part III 14. Springer International Publishing, 2016: 443-450
36 Zellinger W, Grubinger T, Lughofer E, et al. Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning. In: Proceedings of the International Conference on Learning Representations, 2017
37 Li J, Chen E, Ding Z, et al. Maximum density divergence for domain adaptation. IEEE transactions on pattern analysis and machine intelligence, 2020, 43(11): 3918-3930
38 Ganin Y, Ustinova E, Ajakan H, et al. Domain-adversarial training of neural networks. Journal of machine learning research, 2016, 17(59): 1-35
39 Long M, Cao Z, Wang J, et al. Conditional adversarial domain adaptation[J]. Advances in neural information processing systems, 2018. 31
40 Tzeng E, Hoffman J, Saenko K, et al. Adversarial discriminative domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. 7167-7176
41 Peng D, Ke Q, Ambikapathi A M, et al. Unsupervised domain adaptation via domain-adaptive diffusion. IEEE Transactions on Image Processing, 2024, 33: 4245-4260
42 Zhang Y, Chen S, Jiang W, et al. Domain-guided conditional diffusion model for unsupervised domain adaptation. Neural Networks, 2025, 184: 107031
43 Bickel S, Bruckner M, Scheffer T. Discriminative learning under covariate shift. Journal of Machine Learning Research, 2009, 10(9)
44 Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. Advances in neural information processing systems, 2020. 33: 6840-6851
45 Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: Proceedings of the Medical image computing and computer-assisted intervention-MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer international publishing, 2015. 234-241

46 Hoffman M D, Johnson M J. Elbo surgery: yet another way to carve up the variational evidence lower bound. In: Proceedings of the Workshop in advances in approximate Bayesian inference, NIPS, 2016. 1(2)
47 Song J, Meng C, Ermon S. Denoising Diffusion Implicit Models. In: Proceedings of the International Conference on Learning Representations, 2020
48 Zhu Y, Ai J, Wu L, et al. An active multi-target domain adaptation strategy: Progressive class prototype rectification. IEEE Transactions on Multimedia, 2024
49 Harshvardhan G M, Gourisaria M K, Pandey M, et al. A comprehensive survey and analysis of generative models in machine learning. Computer Science Review, 2020, 38: 100285
50 Croitoru F A, Hondru V, Ionescu R T, et al. Diffusion models in vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(9): 10850-10869
51 Yang L, Zhang Z, Song Y, et al. Diffusion models: A comprehensive survey of methods and applications. ACM Computing Surveys, 2023, 56(4): 1-39
52 Niu Z, Yuan J, Ma X, et al. Knowledge distillation-based domain-invariant representation learning for domain generalization. IEEE Transactions on Multimedia, 2023, 26: 245-255
53 Ho J, Salimans T. Classifier-Free Diffusion Guidance. In: Proceedings of the NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications, 2021
54 Meng C, He Y, Song Y, et al. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. International Conference on Learning Representations, 2022
55 Saenko K, Kulis B, Fritz M, et al. Adapting visual category models to new domains. In: Proceedings of the Computer vision-ECCV 2010: 11th European conference on computer vision, Heraklion, Crete, Greece, September 5-11, 2010, proceedings, part iV 11. Springer Berlin Heidelberg, 2010. 213-226
56 Venkateswara H, Eusebio J, Chakraborty S, et al. Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. 5018-5027
57 Peng X, Usman B, Kaushik N, et al. Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924, 2017
58 Peng X, Bai Q, Xia X, et al. Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision, 2019. 1406-1415
59 Saito K, Kim D, Sclaroff S, et al. Semi-supervised domain adaptation via minimax entropy. In: Proceedings of the IEEE/CVF international conference on computer vision, 2019. 8050-8058
60 Kang G, Jiang L, Yang Y, et al. Contrastive adaptation network for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019. 4893-4902
61 Jiang X, Lao Q, Matwin S, et al. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In: Proceedings of the International conference on machine learning, PMLR, 2020. 4816-4827
62 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. 770-778
63 Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2009. 248-255
64 Li J, Zhou P, Xiong C, et al. Prototypical Contrastive Learning of Unsupervised Representations. In: Proceedings of the International Conference on Learning Representations, 2020
65 Tang Y, Wan Y, Qi L, et al. DPStyler: dynamic promptstyler for source-free domain generalization. IEEE Transactions on Multimedia, 2025
66 Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022. 10684-10695
67 Zhu C, Zhang L, Luo W, et al. Tensorial multiview low-rank high-order graph learning for context-enhanced domain adaptation. Neural Networks, 2025. 181: 106859
68 Wang X, Liu J, Li L. MASA: Multi-view adaptive subspace alignment for enhanced few-shot learning. Knowledge-Based Systems, 2025. 114148