Informative Class-Conditioned Feature Alignment for Unsupervised Domain Adaptation

Wanxia Deng
National University of Defense Technology
Changsha, Hunan, China

Yawen Cui
University of Oulu
Oulu, Finland

Zhen Liu
National University of Defense Technology
Changsha, Hunan, China

Gangyao Kuang
National University of Defense Technology
Changsha, Hunan, China

Dewen Hu
National University of Defense Technology
Changsha, Hunan, China

Matti Pietikäinen
University of Oulu
Oulu, Finland

Li Liu*
National University of Defense Technology
Changsha, Hunan, China

ABSTRACT

The goal of unsupervised domain adaptation is to learn a task classifier that performs well for the unlabeled target domain by borrowing rich knowledge from a well-labeled source domain. Although remarkable breakthroughs have been achieved in learning transferable representation across domains, two bottlenecks remain to be further explored. First, many existing approaches focus primarily on the adaptation of the entire image, ignoring the limitation that not all features are transferable and informative for the object classification task. Second, the features of the two domains are typically aligned without considering the class labels; this can lead the resulting representations to be domain-invariant but non-discriminative to the category. To overcome the two issues, we present a novel Informative Class-Conditioned Feature Alignment (IC^{2}FA) approach for UDA, which utilizes a twofold method: informative feature disentanglement and class-conditioned feature alignment, designed to address the above two challenges, respectively. More specifically, to surmount the first drawback, we cooperatively disentangle the two domains to obtain informative transferable features; here, Variational Information Bottleneck (VIB) is employed to encourage the learning of task-related semantic representations and suppress task-unrelated information. With regard to the second bottleneck, we optimize a new metric, termed Conditional Sliced Wasserstein Distance (CSWD), which explicitly estimates the intra-class discrepancy and the inter-class margin. The intra-class and inter-class CSWDs are minimized and maximized, respectively, to yield the domain-invariant discriminative features. IC^{2}FA equips class-conditioned feature alignment with informative feature disentanglement and causes the two procedures to work cooperatively, which facilitates informative discriminative features adaptation. Extensive experimental results on three domain adaptation datasets confirm the superiority of IC^{2}FA.

CCS CONCEPTS

• **Computing methodologies → Computer vision**; **Image representations**; **Neural networks**.

KEYWORDS

Domain adaptation, disentanglement, sliced Wasserstein distance

ACM Reference Format:
Wanxia Deng, Yawen Cui, Zhen Liu, Gangyao Kuang, Dewen Hu, Matti Pietikäinen, and Li Liu. 2021. Informative Class-Conditioned Feature Alignment for Unsupervised Domain Adaptation. In *MM '21: ACM International Conference on Multimedia, October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475579

1 INTRODUCTION

Deep Neural Networks (DNNs) have ushered in significant advances in various tasks, such as image classification [49], object detection [32], image segmentation [15, 41], face recognition [56], and many others. However, these impressive gains depend on the strict assumption that large quantities of well-labeled data in the interested domain are accessible for model learning. Manually labeling often turns out to be both costly and labor-intensive; in particular, for data-sensitive domains such as medical imagery and industrial inspection, labeled samples are even impossible to obtain.

A general strategy (*e.g.*, transfer learning) operates by recycling off-the-shelf learnt knowledge/models in an available related domain (dubbed *source domain*) for the domain of interest (dubbed *target domain*) [42]. Unfortunately, this learning paradigm often results in significant performance degradation, a phenomenon known

*Corresponding author: Li.Liu@oulu.fi

MM '21, October 20–24, 2021, Virtual Event, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475579

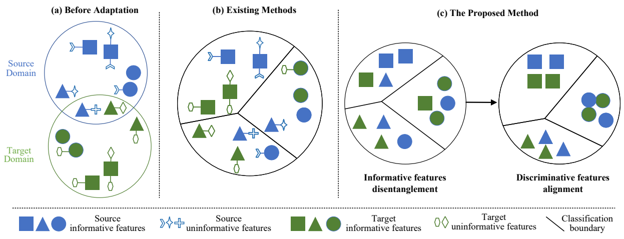

(a) Before Adaptation
(b) Existing Methods
(c) The Proposed Method
Source Domain
Target Domain
Informative features disentanglement
Discriminative features alignment
Source informative features
Source uninformative features
Target informative features
Target uninformative features
Classification boundary

Figure 1: Comparison of existing methods and the proposed method. (a) The feature distributions before adaptation for the source and target domains. We can see some samples include informative and uninformative features; here, the informative features related to the object classification task are shared by these two domains, and the uninformative features are domain-specific. (b) Existing methods directly adapt the two domains with all features of each sample, where forcefully aligning the uninformative features leads to the learned feature distribution discrete. Moreover, discriminative feature adaptation is not considered, which cause some misclassifications. (c) The proposed method first suppresses the uninformative features so that only the informative features are retained, after which, the informative discriminative features are aligned.

as domain shift [58]: this refers to the difference in data distributions between the source and target domains. One practical tactic to address this problem is Domain Adaptation (DA) [3, 4], which bridges the distribution gap to generalize a target model.

In the paper, we focus on unsupervised domain adaptation (UDA), where the source domain contains abundant labeled data while the target domain is fully unlabeled. The main objective of UDA is to learn domain-invariant features that are immune to the domain shift, enabling a classifier trained on the source domain to perform well on target samples [9, 25, 61]. Inspired by this, pioneering works of this kind either explicitly minimize the distribution discrepancy between the source and target domains with the metric paradigm [16, 18, 19, 29, 33–36, 55, 66, 67] or implicitly align the source and target domain distributions via domain adversarial learning [13, 14, 28, 37, 44, 47, 51, 53, 59]. Despite the significant success that has been achieved in this domain, a challenging problem has been neglected: namely, that some uninformative encoded representations may be compulsorily learned and adapted. The source and target domains essentially overlap in task-related information, while the redundant information from the task-unrelated factors (*e.g.*, background, color and context) might be different in nature, and forcefully aligning the uninformative features can impair the adaptation performance. In addition, in some existing UDA approaches, there is another bottleneck: namely, the class-level distribution discrepancy is not fully exploited. Adapting the distribution discrepancy at the domain level only, without encoding the difference of class-level information, will render the learned features domain-invariant but indistinguishable for the category.

To address the two challenges outlined above, we propose a new approach, named Informative Class-Conditioned Feature Alignment (IC^{2}FA), which comprises two key components: informative feature disentanglement and class-conditioned feature alignment, designed to address each of the two bottlenecks, respectively. The motivation of IC^{2}FA is illustrated in Figure 1 [1].

For the informative feature disentanglement paradigm, we attempt to suppress the task-unrelated information, which may include certain domain-specific variations, while retaining the informative task-related information for both domains. Inspired by the Information Bottleneck (IB) principle [57], which dictates that the learned latent representation needs to be maximally informative about the object classification task while being maximally compressive about the original input, we directly apply an adversarial excitation and inhibition mechanism in order to encourage the disentanglement of the latent representations via Variational Information Bottleneck (VIB) [1] to disentangle the labeled source domain. The mutual information maximization of the learned representation and the object classification task is excited, while the mutual information maximization of the learned representation and original input is inhibited. Accordingly, only the features of the source domain that are most conducive to the down-stream classification task can pass through and be retained. Undoubtedly, VIB belongs to the scope of supervised disentanglement; thus, how to utilize VIB for the disentanglement of the unlabeled domain remains a problem. With this goal in mind, we first apply spherical K-means to cluster the source and target samples, then assign the pseudo label to each target data. We next develop the task-related feature disentanglement for the target data using the obtained pseudo label.

As for the class-conditioned feature alignment paradigm, our goal is to search for a metric into which the label information can be easily embedded to explicitly reduce the class-level distribution discrepancy between source and the target domains. The classic metric in UDA, *i.e.*, Maximum Mean Distance [2] widely utilized by some works [16, 18], is difficult to compute due to the presence of kernel functions and their affiliated hyperparameters. Moreover, it has been argued that MMD-based methods fail to adapt once no significant overlap exists between the domain distributions [47, 63]. The sliced Wasserstein distance [48] slices the high-dimensional data distribution via random projections into a one-dimensional distribution, which achieves excellent performance in generative models [12, 64] and has been further proven to be an efficient and reliable discrepancy metric between probability distributions [5, 24]. Enlightened by the excellent properties of the sliced Wasserstein distance (*e.g.*, non-negativity, identity of indiscernible, symmetry, and subadditivity) [12, 23], we extend the metric to our work and propose Conditional Sliced Wasserstein Distance (CSWD), which is defined with reference to the class-level information. We optimize the CSWD between the projected one-dimensional distributions of the two domains to obtain the domain-invariant discriminative features. The optimization is conducted by minimizing the intra-class CSWD and maximizing inter-class CSWD across domains.

The proposed IC^{2}FA unifies the informative feature disentanglement and cross-domain adaptation with class-distinguishable features preserved into one framework, enabling them to benefit from each other and facilitating the adaptation process. The main contributions of our work can be summarized as follows: (1) We propose a novel approach named IC^{2}FA, which addresses the informative feature disentanglement and class-conditioned feature

[1] The depiction of uninformative features is inspired by the work [39].

alignment simultaneously to facilitate better domain adaptation. To the best of our knowledge, this is the first work for combining feature disentanglement with discriminative feature adaptation. (2) By means of our ingenious design, the disentanglement of both domains can be implemented to filter out the task-unrelated features using VIB, an approach that has not been explored by existing UDA works. (3) We embed the class-level information into the sliced Wasserstein distance via the pseudo labels learned and construct a new metric, CSWD. (4) Comprehensive experiments on the Office31, Office-Home and VisDA-C datasets are conducted to demonstrate that the proposed method outperforms existing methods.

2 RELATED WORK

Domain-level adaptation One classical approach directly minimizes the distribution discrepancy of domains via the metric paradigm. Representative metric methods include Maximum Mean Discrepancy (MMD) [34, 36], CORrelation (CORAL) alignment [55], optimal transport distance [10, 29] and Central Moment Discrepancy (CMD) [67]. In [34] and [36], the distribution discrepancy is minimized via Multi-Kernel MMD (MK-MMD) and Joint Maximum Mean Discrepancy (JMMD), respectively. Another popular branch of UDA is based on adversarial learning, inspired by the Generative Adversarial Network (GAN) [20]. DANN [14] and Conditional Domain Adversarial Network (CDAN) [37] both utilize a domain discriminator to represent the domain discrepancy; however, the domain discriminator is confused in a two-player minimax game. Wasserstein Distance Guided Representation Learning (WDGRL) [53] and Re-weighted Adversarial Adaptation Network (RAAN) [47] estimate the distribution distance between the source and target samples in a domain-critical network and optimize the feature extractor network in an adversarial manner. Maximum Classifier Discrepancy (MCD) [51] utilizes task-specific classifiers as discriminators and is used to align the target and source distributions using adversarial learning. Sliced Wasserstein Discrepancy (SWD) [28] adopts the Wasserstein metric by minimizing the cost of moving the marginal distributions between task-specific classifiers.

Class-level adaptation Moving Semantic Transfer Network (MSTN) [65] aligns the labeled source centroid and the pseudo-labeled target centroid to learn domain-invariant semantic representations. SimNet [46] and Transferable Prototypical Networks (TPN) [43] learn categorical prototype representations by computing the similarity between prototype representations of each category. Contrastive Adaptation Network (CAN) [16] explicitly models the intra-class and inter-class domain discrepancies based on the MMD metric. Jiang et al. [21] present a sampling-based implicit alignment approach to promote the class-conditioned adaptation. Progressive Feature Alignment Network (PFAN) [8] adapts the discriminative features progressively, via exploiting the intra-class variation in the target domain.

Feature disentanglement-based methods Domain Separation Network (DSN) [6] proposes separating the feature into shared and private features. These two features are encouraged to be orthogonal, while can also be decoded back to images. The Transferable Attention for Domain Adaptation (TADA) [62] and CADA [26] propose to apply the attention mechanism for UDA, which present transferable attention, focusing the adaptation model on transferable regions but not all regions of an image. Domain-Specific Batch Normalization (DSBN) [7] is proposed to separate domain-specific information for UDA using two branches of batch normalization, each of which is exclusively in charge of a single domain. The goal of these four works is to develop a framework in which the domain-specific variations can be filtered out, consistent with our proposed informative feature disentanglement. The difference is that our proposed disentanglement does not add network module and introduce extra trained parameters.

3 PROPOSED METHODOLOGY

In this section, we first present the UDA problem formulation, then introduce the proposed IC^{2}FA approach, with a focus on disentangling informative features and the adaptation of the class-conditioned feature via VIB and sliced Wasserstein distance. The overall framework of IC^{2}FA is illustrated in Figure 2.

3.1 Problem Formulation and Pseudo-label Definition

Our objective is to predict the labels of samples drawn from a target domain as accurately as possible, given N_s labeled samples $\{\boldsymbol{x}_i^s, y_i^s\}_{i=1}^{N_s}$ drawn from a source domain and N_t unlabeled samples $\{\boldsymbol{x}_i^t\}_{i=1}^{N_t}$ sampled from the target domain, and we have $y_i^s \in 1, 2, ..., K$. We define the feature extractor as f with parameters θ and the embedding classifier as g with parameters ϕ. We denote the entire network as $h = f \circ g$.

In the following discussion, we need to use pseudo labels of the target domain for disentanglement and adaption. Thus, we first present the formulation of pseudo label, which is implemented in the first stage of our proposed approach. The feature extractor is utilized to obtain the features of the source and target domains. We then employ spherical K-means to cluster the source features; this will enable us to find the centroids that are as the initial centroids of the target domain. Finally, we revisit spherical K-means to cluster target features and attach corresponding pseudo labels $\{\hat{y}_i^t\}_{i=1}^{N_t}$.

3.2 Informative Feature Disentanglement

To extract the general high-level features for UDA, we choose the Variational Information Bottleneck (VIB), built upon recently developed information theories for deep learning [1], to disentangle the source and target domains, respectively.

To facilitate this discussion, we denote X^s as the input images from the source domain. Let Y^s denote the corresponding output variables (*e.g.*, desired label), the information of which we want to preserve. We regard the internal representation of certain intermediate layer as a stochastic encoding Z^s of the input images X^s, defined by the parametric encoder $p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s)$. For clarity, we denote $\boldsymbol{x}^s$, $\boldsymbol{y}^s$ and $\boldsymbol{z}^s$ as the instances of X^s, Y^s and Z^s, respectively. Our goal is to learn an encoding that is maximally informative regarding our output variables Y^s, measured by the mutual information between our encoding Z^s and the output variables $I(Z^s, Y^s; \boldsymbol{\theta})$, while the mutual information $I(X^s, Z^s; \boldsymbol{\theta})$ between the input images X^s and the encoding Z^s is minimized. We therefore assume the following Markov chain constraint introduced in the Information Bottleneck (IB) theory [57]: $Y^s \leftrightarrow X^s \leftrightarrow Z^s$; moreover, the objective function

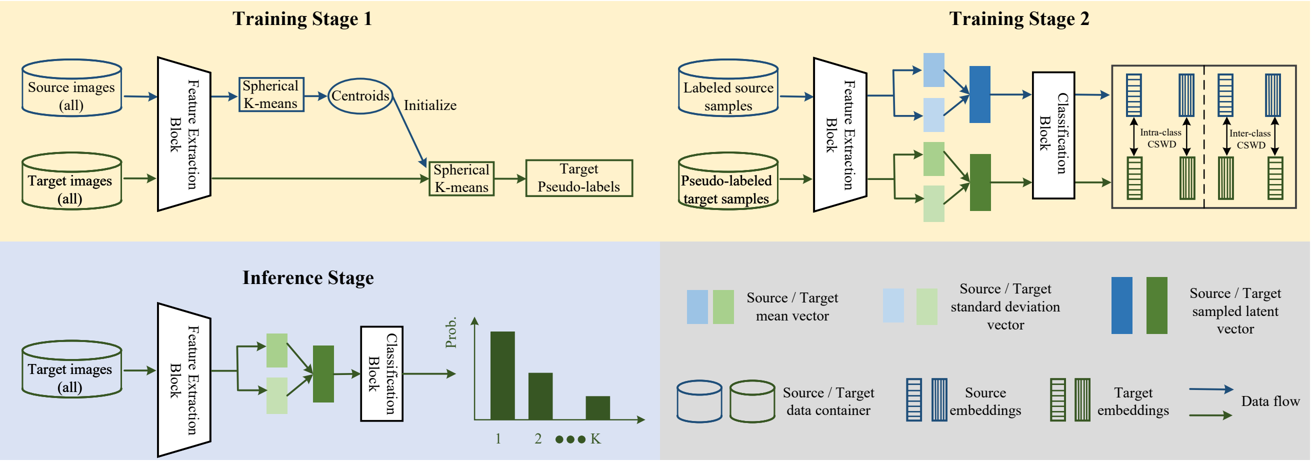


Figure 2: The training and inference stages of the proposed IC^{2}FA. The training of IC^{2}FA includes two stages, which work alternately. At the first training stage, all source images and target images are applied to calculate the target pseudo labels. The second stage comprises two main components, *i.e.*, informative feature disentanglement and discriminative features alignment, both of which are integrated into a single framework and work cooperatively. Notably, the colors of blocks and lines represent their corresponding data flows.

that is maximized is defined as follows:

$$I(Z^s, Y^s; \boldsymbol{\theta}) - \beta_s I(X^s, Z^s; \boldsymbol{\theta}), \tag{1}$$

$$I(Z^t, \hat{Y}^t; \boldsymbol{\theta}) - \beta_t I(X^t, Z^t; \boldsymbol{\theta}), \tag{2}$$

where β_s denotes the Lagrange multiplier. The first term $I(Z^s, Y^s; \boldsymbol{\theta}) = \int d_{\boldsymbol{z}^s} d_{\boldsymbol{y}^s} p_{\boldsymbol{\theta}}(\boldsymbol{z}^s, \boldsymbol{y}^s) log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}^s, \boldsymbol{y}^s)}{p_{\boldsymbol{\theta}}(\boldsymbol{z}^s) p_{\boldsymbol{\theta}}(\boldsymbol{y}^s)}$ encourages Z^s to be predictive of Y^s. The second term $I(X^s, Z^s; \boldsymbol{\theta}) = \int d_{\boldsymbol{z}^s} d_{\boldsymbol{x}^s} p_{\boldsymbol{\theta}}(\boldsymbol{z}^s, \boldsymbol{x}^s) log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s)}{p_{\boldsymbol{\theta}}(\boldsymbol{z}^s)}$ encourages Z^s to inhibit as many details of X^s as possible.

However, it is computationally challenging to compute mutual information. We write the first term out in full; this becomes:

$$\begin{aligned} I(Z^s, Y^s; \boldsymbol{\theta}) &= \int d_{\boldsymbol{z}^s} d_{\boldsymbol{y}^s} p_{\boldsymbol{\theta}}(\boldsymbol{z}^s, \boldsymbol{y}^s) log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}^s, \boldsymbol{y}^s)}{p_{\boldsymbol{\theta}}(\boldsymbol{z}^s) p_{\boldsymbol{\theta}}(\boldsymbol{y}^s)} \\ &= \int d_{\boldsymbol{z}^s} d_{\boldsymbol{y}^s} p_{\boldsymbol{\theta}}(\boldsymbol{z}^s, \boldsymbol{y}^s) log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{y}^s|\boldsymbol{z}^s)}{p_{\boldsymbol{\theta}}(\boldsymbol{y}^s)}. \end{aligned} \tag{3}$$

Since $p_{\boldsymbol{\theta}}(\boldsymbol{y}^s|\boldsymbol{z}^s)$ is intractable, we apply $q_{\boldsymbol{\phi}}(\boldsymbol{y}^s|\boldsymbol{z}^s)$ to be a variational approximation to $p_{\boldsymbol{\theta}}(\boldsymbol{y}^s|\boldsymbol{z}^s)$. The $q_{\boldsymbol{\phi}}(\boldsymbol{y}^s|\boldsymbol{z}^s)$ is the defined decoder of VIB, which we will take as the classification block g with its own set of parameters $\boldsymbol{\phi}$. According to the Kullback Leibler divergence $KL[p_\theta(Y^s|Z^s), q_\phi(Y^s|Z^s)] \geqslant 0$, we have the following inequality: $\int d_{\boldsymbol{y}^s} log p_{\boldsymbol{\theta}}(\boldsymbol{y}^s|\boldsymbol{z}^s) \geqslant \int d_{\boldsymbol{y}^s} log q_{\boldsymbol{\phi}}(\boldsymbol{y}^s|\boldsymbol{z}^s)$. Thus, Equation 3 can be rewritten as follows:

$$\begin{aligned} I(Z^s, Y^s; \boldsymbol{\theta}, \boldsymbol{\phi}) &\geqslant \int d_{\boldsymbol{z}^s} d_{\boldsymbol{y}^s} p_{\boldsymbol{\theta}}(\boldsymbol{z}^s, \boldsymbol{y}^s) log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{y}^s|\boldsymbol{z}^s)}{p_{\boldsymbol{\theta}}(\boldsymbol{y}^s)} \\ &= \int d_{\boldsymbol{z}^s} d_{\boldsymbol{y}^s} p_{\boldsymbol{\theta}}(\boldsymbol{z}^s, \boldsymbol{y}^s) log q_{\boldsymbol{\phi}}(\boldsymbol{y}^s|\boldsymbol{z}^s) + H(Y^s), \end{aligned} \tag{4}$$

where $H(Y^s)$ is the entropy of our labels, which is independent of the optimization procedure and can thus be ignored. $I(Z^s, Y^s; \boldsymbol{\theta})$ can obtain a new lower bound:

$$I(Z^s, Y^s; \boldsymbol{\theta}, \boldsymbol{\phi}) \geqslant \int d_{\boldsymbol{x}^s} d_{\boldsymbol{z}^s} d_{\boldsymbol{y}^s} p_{\boldsymbol{\theta}}(\boldsymbol{x}^s) p_{\boldsymbol{\theta}}(\boldsymbol{y}^s|\boldsymbol{x}^s) p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s) log q_{\boldsymbol{\phi}}(\boldsymbol{y}^s|\boldsymbol{z}^s). \tag{5}$$

We now consider the second term $I(X^s, Z^s; \boldsymbol{\theta})$ of Equation 2. $I(X^s, Z^s; \boldsymbol{\theta})$ can be further computed as follows:

$$I(X^s, Z^s; \boldsymbol{\theta}) = \int d_{\boldsymbol{x}^s} d_{\boldsymbol{z}^s} p_{\boldsymbol{\theta}}(\boldsymbol{z}^s, \boldsymbol{x}^s) log p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s) - \int d_{\boldsymbol{z}^s} p_{\boldsymbol{\theta}}(\boldsymbol{z}^s) log p_{\boldsymbol{\theta}}(\boldsymbol{z}^s). \tag{6}$$

However, it may be intractable to directly compute the marginal distribution of the $\boldsymbol{z}^s$, since $p_{\boldsymbol{\theta}}(\boldsymbol{z}^s) = \int d_{\boldsymbol{x}^s} p_{\boldsymbol{\theta}}(\boldsymbol{x}^s) log p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s)$ requires integral to be solved over latent feature space. We apply an alternative way $r(\boldsymbol{z}^s)$, to represent the variational approximation of the $p_{\boldsymbol{\theta}}(\boldsymbol{z}^s)$. The $r(\boldsymbol{z}^s)$ denotes the prior distribution of the latent features $\boldsymbol{z}^s$. We choose $r(\boldsymbol{z}^s)$ as a standard Gaussian distribution $\mathcal{N}(0, I)$. Since $KL[p_{\boldsymbol{\theta}}(\boldsymbol{z}^s), r(\boldsymbol{z}^s)] \geqslant 0$, we can obtain the following inequality: $\int d_{\boldsymbol{z}^s} p_{\boldsymbol{\theta}}(\boldsymbol{z}^s) log p_{\boldsymbol{\theta}}(\boldsymbol{z}^s) \geqslant \int d_{\boldsymbol{z}^s} p_{\boldsymbol{\theta}}(\boldsymbol{z}^s) log r(\boldsymbol{z}^s)$. Thus, the $I(X^s, Z^s; \boldsymbol{\theta})$ can get the following upper bound:

$$I(X^s, Z^s; \boldsymbol{\theta}) \leqslant \int d_{\boldsymbol{x}^s} d_{\boldsymbol{z}^s} p_{\boldsymbol{\theta}}(\boldsymbol{x}^s) p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s) \frac{log p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s)}{r(\boldsymbol{z}^s)}. \tag{7}$$

Combining $I(Z^s, Y^s; \boldsymbol{\theta}, \boldsymbol{\phi})$ and $I(X^s, Z^s; \boldsymbol{\theta})$, we can obtain the resulting evidence lower bound (ELBO):

$$\begin{aligned} & I(Z^s, Y^s; \boldsymbol{\theta}, \boldsymbol{\phi}) - \beta_s I(X^s, Z^s; \boldsymbol{\theta}) \geqslant \\ & \int d_{\boldsymbol{x}^s} d_{\boldsymbol{z}^s} d_{\boldsymbol{y}^s} p_{\boldsymbol{\theta}}(\boldsymbol{x}^s) p_{\boldsymbol{\theta}}(\boldsymbol{y}^s|\boldsymbol{x}^s) p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s) log q_{\boldsymbol{\phi}}(\boldsymbol{y}^s|\boldsymbol{z}^s) \\ & - \beta_s \int d_{\boldsymbol{x}^s} d_{\boldsymbol{z}^s} p_{\boldsymbol{\theta}}(\boldsymbol{x}^s) p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s) \frac{log p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s)}{r(\boldsymbol{z}^s)} \end{aligned} \tag{8}$$

Following the VAE [1], we determine that the $p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s)$ is realized as a Gaussian distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s) = \mathcal{N}(\boldsymbol{z}^s|f^{\mu}(\boldsymbol{x}^s), f^{\sigma}(\boldsymbol{x}^s))$, where f outputs the mean μ and the variance σ of the latent features $\boldsymbol{z}^s$. We can then use the reparameterization trick outlined in [22] to write $p_{\boldsymbol{\theta}}(\boldsymbol{z}^s|\boldsymbol{x}^s)dz = p_{\boldsymbol{\theta}}(\epsilon)d\epsilon$, where $\boldsymbol{z}^s = f(\boldsymbol{x}^s, \epsilon)$ denotes the deterministic function of $\boldsymbol{x}^s$ and the Gaussian random variable ϵ. We can therefore obtain the following loss function:

$$\begin{aligned}\mathcal{L}^s =&\frac{1}{N_s}\sum_{i=1}^{N_s}\mathbb{E}_{\epsilon\sim p_\theta(\epsilon)}[-logq_\phi(\boldsymbol{y}_i^s|f(\boldsymbol{x}_i^s,\epsilon))]\\ &+\beta_s KL[p_\theta(\boldsymbol{z}^s|\boldsymbol{x}_i^s),r(\boldsymbol{z}^s)],\end{aligned} \tag{9}$$

where the first term is a form of the classification loss of the source domain, while the second denotes the information bottleneck loss, which is minimized to filter out the irrelevant part of X^s.

Since the disentanglement carried out by VIB operates only under the supervised condition, we use pseudo labels of target domain as supervised information to forcefully disentangle. Similarly, the disentanglement loss function of the target domain is written as:

$$\begin{aligned}\mathcal{L}^t =&\frac{1}{N_t}\sum_{i=1}^{N_t}\mathbb{E}_{\epsilon\sim p_\theta(\epsilon)}[-logq_\phi(\hat{\boldsymbol{y}}_i^t|f(\boldsymbol{x}_i^t,\epsilon))]\\ &+\beta_t KL[p_\theta(\boldsymbol{z}^t|\boldsymbol{x}_i^t),r(\boldsymbol{z}^t)],\end{aligned} \tag{10}$$

where the first term acts as the classification of the target domain to capture the informative features. We utilize the assignments obtained by the clustering as supervision for updating the network weights and capturing the informative features. Similar to the disentanglement of the source domain, the second term indicates that we try to filter out task-unrelated information of the target domain.

3.3 Class-Conditioned Feature Alignment

We take class-level information into account to explicitly measure the distribution discrepancy, which is defined using sliced Wasserstein distance. To develop our framework, we first review the preliminary formulations of sliced Wasserstein distance.

3.3.1 Sliced Wasserstein distance revisit. The Wasserstein distance is induced by the optimal transport theory [52]. Formally, the Wasserstein distance is defined by

$$W_p(\rho,\nu)=(\inf_{\gamma\in\Pi(\rho,\nu)}\int_{X\times X'}C^p(\boldsymbol{x},\boldsymbol{x'})d\gamma(\boldsymbol{x},\boldsymbol{x'}))^{\frac{1}{p}}, \tag{11}$$

where ρ and ν are two probability measures defined on $X, X' \subseteq \Omega$, and $\boldsymbol{x}$, $\boldsymbol{x'}$ are random variables. $\gamma\in\Pi(\rho,\nu)$ denotes the set of all joint distributions $\gamma(\boldsymbol{x},\boldsymbol{x'})$, whose marginal distributions are ρ and ν respectively. C is a metric, and $p > 0$; thus, the Equation 11 is called the p-Wasserstein distance. Due to the computational burden of the Wasserstein distance for high-dimensional distributions, the sliced Wasserstein distance is as a potential alternative [5].

The conceptual underpinning of sliced Wasserstein distance involves first factorizing the higher-dimensional probability distribution into a set of one-dimensional distributions via linear projections. The distance between the two distributions is calculated in the form of the Wasserstein distance of one-dimensional distribution. In this way, the computation of distance can be translated into the solving of several one-dimensional optimal transport problems, which have closed-form solutions. More formally, the Sliced p-Wasserstein distance between distributions can be defined as:

$$SW_p(\rho,\nu)=\left(\int_{\omega\in\Omega}W_p^p(\rho^\omega,\nu^\omega)d_\omega\right)^{\frac{1}{p}}, \tag{12}$$

where ρ^ω and ν^ω denote the linear projections of ρ and ν onto the direction ω, while Ω is the unit sphere. Following [12], we set p to 2; moreover, the quadratic Wasserstein distance can be approximated using the samples $\boldsymbol{x}\in\mathcal{D}$ and $\boldsymbol{x'}\in\mathcal{D}'$:

$$W_2^2(\mathcal{D},\mathcal{D}')=\frac{1}{\mathcal{D}'}\sum_{i=1}^{|\mathcal{D}|}\left\|\mathcal{D}^\omega_{\sigma_{\mathcal{D}(i)}}-\mathcal{D}'^\omega_{\sigma_{\mathcal{D}'(i)}}\right\|_2^2, \tag{13}$$

here, this assumes $|\mathcal{D}|=|\mathcal{D}'|$ for simplicity, which is not a strict restriction. $\sigma_{\mathcal{D}(i)}$ and $\sigma_{\mathcal{D}'(i)}$ denote the permutations such that:

$$\begin{aligned}\mathcal{D}^\omega_{\sigma_{\mathcal{D}(i)}}&\leqslant\mathcal{D}^\omega_{\sigma_{\mathcal{D}(i+1)}}, &\forall i\in\{1\leqslant i<|\mathcal{D}|\},\\ \mathcal{D}'^\omega_{\sigma_{\mathcal{D}'(i)}}&\leqslant\mathcal{D}'^\omega_{\sigma_{\mathcal{D}'(i+1)}}, &\forall i\in\left\{1\leqslant i<\left|\mathcal{D}'\right|\right\}.\end{aligned} \tag{14}$$

In combination with Equation 13, Equation 12 can be rewritten as:

$$SW_2(\mathcal{D},\mathcal{D}')=\frac{1}{M}\frac{1}{|\mathcal{D}|}\sum_{m=1}^{M}\sum_{i=1}^{|\mathcal{D}|}\left\|\mathcal{D}^{\omega_m}_{\sigma_{\mathcal{D}(i)}}-\mathcal{D}'^{\omega_m}_{\sigma_{\mathcal{D}'(i)}}\right\|_2^2, \tag{15}$$

where M is the number of one-dimensional random projection directions ω_m.

3.3.2 Conditional sliced Wasserstein distance. First, we introduce the sliced 2-Wasserstein distance into our UDA formulation:

$$SW_2(X^s,X^t)=\frac{1}{M}\frac{1}{N_s}\frac{1}{N_t}\sum_{m=1}^{M}\sum_{i=1}^{N_s}\sum_{j=1}^{N_t}\left\|(\boldsymbol{p}^s)^{\omega_m}_{\sigma_{p^s}(i)}-(\boldsymbol{p}^t)^{\omega_m}_{\sigma_{p^t}(j)}\right\|_2^2, \tag{16}$$

where X^s and X^t are the input images sampled from the source and target domains, respectively. $\boldsymbol{p}^s=h(\boldsymbol{x}^s)$ and $\boldsymbol{p}^t=h(\boldsymbol{x}^t)$ are the classifier outputs. Directly employing Equation 16 to align the distributions of the two domains only achieves domain-level adaptation, but does not guarantee semantic consistency. We integrate the class-level information into Equation 16 and accordingly propose Conditional Sliced Wasserstein Distance (CSWD). For notational simplicity, we denote $C^{\omega_m}(\boldsymbol{p}_i^s,\boldsymbol{p}_j^t)=\left\|(\boldsymbol{p}^s)^{\omega_m}_{\sigma_{p^s}(i)}-(\boldsymbol{p}^t)^{\omega_m}_{\sigma_{p^t}(j)}\right\|_2^2$, and the intra-class CSWD can be given:

$$\begin{aligned}CSW_2^{intra}(X^s,X^t)=&\frac{1}{M}\frac{1}{K}\sum_{m=1}^{M}\sum_{k=1}^{K}(\frac{1}{N_s^k}\frac{1}{N_t^k}\sum_{i=1}^{N_s^k}\\ &\sum_{j=1}^{N_t^k}\mathbf{1}_{y_i^s=\hat{y}_j^t=k}C^{\omega_m}(\boldsymbol{p}_i^s,\boldsymbol{p}_j^t)),\end{aligned} \tag{17}$$

where y_i^s denotes the true label of source sample x_i^s, and $\hat{y}_i^t$ refers to the pseudo label computed via spherical K-means for the target sample x_i^t. N_s^k and N_t^k represent the total number of source and target images that have the same label k, respectively. $\mathbf{1}_{y_i^s=\hat{y}_j^t=k}$ is defined as: $\mathbf{1}_{y_i^s=\hat{y}_j^t=k}=\begin{cases}1 & if\ y_i^s=\hat{y}_j^t=k\\ 0 & otherwise\end{cases}$. The intra-class CSWD is minimized while we maximize the inter-class CSWD:

$$CSW_2^{inter}(X^s, X^t) = \frac{1}{M}\frac{1}{K(K-1)}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{k'=1,k\neq k'}^{K}(\frac{1}{N_s^k}\frac{1}{N_t^{k'}}\sum_{i=1}^{N_s^k}\sum_{j=1}^{N_t^{k'}}\mathbf{1}_{y_i^s=k,\hat{y}_j^t=k'}C^{\omega_m}(\boldsymbol{p}_i^s,\boldsymbol{p}_j^t)), \tag{18}$$

where $\mathbf{1}_{y_i^s=k,\hat{y}_j^t=k'} = \begin{cases} 1 & if\ y_i^s = k, \hat{y}_j^t = k' \\ 0 & otherwise \end{cases}$. By combining Equations 17 and 18, we try to minimize

$$CSW_2(X^s, X^t) = CSW_2^{intra}(X^s, X^t) - \lambda_0 CSW_2^{inter}(X^s, X^t), \tag{19}$$

where λ_0 is utilized to balance the two terms. To fully adapt the discriminative features, we minimize $\mathcal{L}^{ada}$ over all multiple FC layers of the classifier block g; thus, the adaptation objective is:

$$\mathcal{L}^{ada} = \sum_{l=1}^{L} CSW_2^l(X^s, X^t), \tag{20}$$

where L denotes the number of FC layers in the classifier block g.

In our work, the proposed IC^{2}FA unifies the informative feature disentanglement and CSWD into a single framework in which these two components work cooperatively. The overall objective is formulated as

$$\mathcal{L} = \mathcal{L}^s + \beta\mathcal{L}^t + \lambda\mathcal{L}^{ada}, \tag{21}$$

where β and λ are applied to regularize the loss function.

4 EXPERIMENTS

4.1 Datasets

We evaluate our proposed IC^{2}FA on three UDA datasets: Office31 [50], Office-Home [60] and VisDA-C [45].

Office-31 which is a standard dataset used to evaluate different DA methods for object recognition, comprises three different domains: Amazon (A), Dslr (D), and Webcam (W), and includes 4,652 images in 31 classes. Amazon images are collected from *amazon.com*, while Webcam and Dslr images are taken using a webcam and a high-quality camera, respectively.

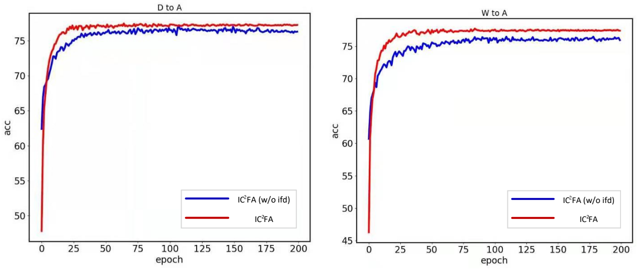
Office-Home is a large dataset containing approximately 15,500 images divided into 65 classes. The dataset comprises four domains: Artistic (Ar), Clip Art (Cl), Product (Pr) and Real-World (Rw).

VisDA-C is a highly challenging dataset featuring domain shift from synthetic data to real imagery. It has two domains and 12 classes in common: the Synthetic one, consisting of 152,397 synthetic 2D renderings of 3D objects, and the Real one, consisting of 55,388 real images cropped from the MS-COCO [31] dataset.

4.2 Implementation Details

We applied ResNet-50 and ResNet-101 [17], pretrained on ImageNet [11], as the feature extractor branch, and replaced the last FC layer with the task-specific FC layer.

The network was trained using the mini-batch stochastic gradient descent (SGD) optimizer with a momentum of 0.9. The learning rate annealing strategy is based on the cosine function [38]. For the Office-31 and Office-Home datasets, the initial learning rate is 1e-3 for the convolutional layers and 1e-2 for the task-specific FC layer. For VisDA-C, the initial learning rate is 3e-5 for the convolutional layers and 3e-4 for the task-specific FC layer. We selected the hyperparameters following the importance-weighted cross-validation (IWCV) [54]. λ and λ_0 are set to 3.0 and 0.5, respectively. β can be selected from (0.01,0.1). β_s and β_t are set to 1e-5. M is set to 32 in our experiments



(a) D⟶A

(b) W⟶A

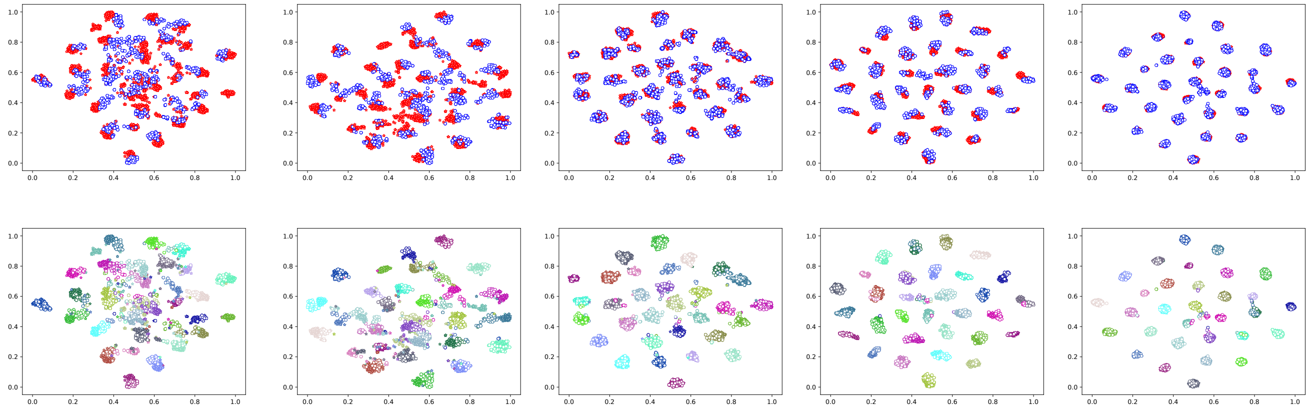
Figure 3: (a)-(b) Accuracy curves of IC^{2}FA and IC^{2}FA (w/o ifd) on the task D⟶A and W⟶A.

More specifically, for the target domain, we concentrated only on data with high reliability; *i.e.*, we filtered out ambiguous data points located far away from the corresponding cluster centroid. Similar to CAN [16], the point-to-centroid threshold is defined as 0.05 for Office-31 tasks A ⟶ W, D ⟶ W, W ⟶ D and A ⟶ D. For other tasks, this threshold is defined as 1.

We compare the proposed IC^{2}FA model with several state-of-the-art domain adaptation methods, some of which related to our work are focused on: (1) **ResNet-50** [17] acts as the lower bound; (2) **DAN** [34] minimizes the MMD distance of two domains; (3) **CDAN** [37] develops a conditional alignment network based on adversarial learning; (4) **MCD** [51] adapts distributions utilizing the task-specific decision boundaries in an adversarial manner; (5) **DeepJDOT** [10] adapts optimal transport distance in the deep network; (6) **SWD** [28] applies the Wasserstein distance in adversarial task-classifier learning; (7) **SimNet** [46] learns the similarity between prototype representations of each category; (8) **ETD** [29] builds an attention-aware optimal transport distance to measure the domain discrepancy; (9) **TADA** [62] and (10) **CADA** [26] utilizes the attention mechanism to learn domain-shared features; (11) **MDD** [21] presents sampling-based implicit domain alignment to address within-domain class imbalance and between-domain class distribution shift; (12) **DCAN** [30] explore the domain-wise convolutional channel activation; (13) **CAN** [16] optimizes the intra-class and inter-class MMD distances to obtain the class-level adaptation.

4.3 Comparison Results

The unsupervised adaptation results on Office-31 are reported in Table 1. To facilitate fair comparison, the results for most comparison methods are quoted from their original papers. Through this comparison of results, we can observe that our proposed method defeats the state-of-the-art method CAN [16] on the whole, which strongly confirms the effectiveness of IC^{2}FA. Although the improvement is slight, IC^{2}FA does perform with more efficiency owing to the simple nature of the computation required for one-dimensional distributions. Moreover, on the complex task W⟶A, IC^{2}FA outperforms many methods by a large margin.



(a) ResNet-50

(b) DAN

(c) CDAN+E

(d) CAN

(e) IC^{2}FA

Figure 4: (a)-(e) The t-SNE visualization of the embedded features generated by ResNet-50, DAN, CDAN+E, CAN and IC^{2}FA on the task A→W, respectively. The "*" and "o" represent the source and target domain, respectively. In the first row, different colors represent different domains; in the second row, different colors represent different classes.

Table 1: Classification accuracies (%) on the Office31 dataset for UDA. All models utilize ResNet-50 as the base architecture. The bold numbers denote the best results for each column.

	A→W	D→W	W→D	A→D	D→A	W→A	Average
ResNet-50 [17]	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DAN [34]	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
DeepJDOT [10]	88.9±0.3	88.2±0.1	98.5±0.1	99.6±0.2	72.1±0.4	70.1±0.4	86.2
SimNet [46]	88.6±0.5	98.2±0.2	99.7±0.2	85.3±0.3	73.4±0.8	71.8±0.6	86.2
ETD [29]	92.1	**100.0**	**100.0**	88.0	71.0	67.8	86.2
CDAN+E [37]	94.1±0.1	98.6±0.1	**100.0**±0.0	92.9±0.2	71.0±0.3	69.3±0.3	87.7
TADA [62]	94.3±0.3	98.7±0.1	99.8±0.2	91.6±0.3	72.9±0.2	73.0±0.3	88.4
MDD [21]	90.3±0.2	98.7±0.1	99.8±.0	92.1±0.5	75.3±0.2	74.9±0.3	88.8
CADA [26]	**97.0**±0.2	99.3±0.1	**100.0**±0.0	**95.6**±0.1	71.5±0.2	73.1±0.3	89.5
DCAN [30]	95.0	97.5	100.0	92.6	77.2	74.9	89.5
CAN [16]	94.5±0.3	99.1±0.2	99.8±0.2	95.0±0.3	**78.0**±0.3	77.0±0.3	90.6
IC^{2}FA	94.6 ±0.2	99.2 ±0.2	**100.0**±0.0	95.4 ±0.3	77.3 ±0.3	**77.6** ±0.2	**90.7**

Table 2: Classification results (%) on the Office-Home dataset for UDA. All models utilize ResNet-50 as the base architecture. The bold numbers denote the best results for each column.

	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Ave.
ResNet-50 [17]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [34]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DeepJDOT [10]	48.2	69.2	74.5	58.5	69.1	71.1	56.3	46.0	76.5	68.0	52.7	80.9	64.3
CDAN+E [37]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
ETD [29]	51.3	71.9	85.7	57.6	69.2	73.7	57.8	51.2	79.3	70.2	57.5	82.1	67.3
TADA [62]	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
MDD [21]	56.2	77.9	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
CADA [26]	**56.9**	76.4	80.7	61.3	**75.2**	**75.2**	63.2	54.5	80.7	73.9	**61.5**	84.1	70.2
IC^{2}FA	56.7	**78.6**	**81.0**	**64.8**	73.7	74.9	**65.5**	53.9	**81.7**	**74.1**	59.8	**84.5**	**70.8**

Table 2 illustrates the classification accuracies of experimental methods on the Office-Home dataset. As we can observe, as desired, IC^{2}FA dramatically outperform all comparison methods on most tasks. Notably, moreover, the best and second best performances are obtained by IC^{2}FA and CADA [26], respectively, which verifies that focusing on task-related feature adaptation without considering domain-specific variations truly improves the performance.

The results on VisDA-C are presented in Table 3. Due to the large domain shift between the source and target distributions, the comparison methods achieve poor performance in some classes.

Table 3: Classification accuracy (%) of each category on VisDA-C dataset for UDA. All models utilize ResNet-101 as base architecture, except for SimNet [46] which uses ResNet-152. The bold numbers denote the best results for each column.

	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Average
ResNet-50 [17]	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DAN [34]	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
MCD [51]	87.0	60.9	**83.7**	64.0	88.9	**79.6**	84.7	76.9	88.6	40.3	83.0	25.8	71.9
SimNet [46]	**94.3**	82.3	73.5	47.2	87.9	49.2	75.1	**79.7**	85.3	**68.5**	81.1	**50.3**	72.9
SWD [28]	90.8	**82.5**	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
IC^{2}FA	89.7	70.6	79.8	**84.3**	**96.5**	72.1	**90.4**	65.3	**92.7**	63.3	**86.5**	36.0	**77.3**

Table 4: Ablation experiments on Office31 and Office-Home dataset. Bold numbers denote the best results for each column.

	A→W	D→W	W→D	A→D	D→A	W→A	Ave.
IC^{2}FA (w/o ifd)	93.2±0.3	99.1±0.1	99.9±0.1	92.4±0.2	76.7±0.3	76.6±0.2	89.7
IC^{2}FA	**94.6** ±0.2	**99.2** ±0.2	**100.0**±0.0	**95.4** ±0.3	**77.3** ±0.3	**77.6** ±0.2	**90.7**

	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Ave.
IC^{2}FA (w/o ifd)	56.5	76.8	79.4	63.6	72.4	72.0	64.6	51.9	80.5	73.9	59.4	84.0	69.6
IC^{2}FA	**56.7**	**78.6**	**81.0**	**64.8**	**73.7**	**74.9**	**65.5**	**53.9**	**81.7**	**74.1**	**59.8**	**84.5**	**70.8**

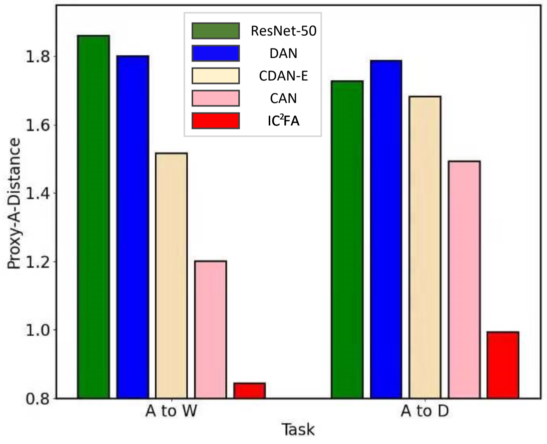


Figure 5: Empirical analysis: Proxy $\mathcal{A}$-Distance of different features on A→W and A→D.

The IC^{2}FA achieves a performance boost overall, demonstrating that our IC^{2}FA is efficient for the large-gap adaptation task.

4.4 Ablation Studies

We conducted ablation experiments on the Office-31 and Office-Home datasets to determine the effects of the informative feature disentanglement in our IC^{2}FA. We first eliminated the disentanglement component; the resulting method is denoted as IC^{2}FA (w/o ifd). Table 4 presents the results of our ablation studies. IC^{2}FA outperforms IC^{2}FA (w/o ifd); this demonstrates that the informative feature disentanglement plays an essential role in adapting the discriminative features across domains.

To further explore the functionality of informative feature disentanglement, we depict the accuracy curves on the D→A and W→A tasks in the Figure 3. We can observe that IC^{2}FA achieves optimal performance more quickly compared to IC^{2}FA (w/o ifd) on these two tasks, verifying that informative feature disentanglement can accelerate the training process.

4.5 Further Remarks

Feature Visualization A popular method of visualizing high dimensional data in 2D is t-SNE [27]. We visualize embedded features from ResNet-50, DAN, CDAN+E, CAN and IC^{2}FA on the source and target domains for the adaptation task A→W and illustrate the results in Figure 4. From the figure, we can observe that the feature distributions of ResNet-50 are disordered. DAN can alleviate this problem to a certain extent; however, there are still large discrepancies between the distributions of the two domains. Although CDAN+E can improve the marginal distribution adaptation, a mismatch of category-level features occurs. CAN yields fine class-level distribution alignment; however, compared to our proposed IC^{2}FA, the marginal distribution is somewhat more discrete. IC^{2}FA achieves the best adaptation results; that is the class-level distributions are better aligned and more compact.

Discrepancy Distance The theory of DA [3, 40] denotes the $\mathcal{A}$-distance as a measure of the cross-domain discrepancy, which will bound the target risk together with the source risk. The way in which the proxy $\mathcal{A}$-distance (PAD) is estimated can be defined as $\hat{d}_{\mathcal{A}} = 2(1 - 2\epsilon)$, where ϵ is the generalization error of a binary classifier of a discriminating source and target. We applied a kernel SVM to estimate the $\mathcal{A}$-distance. Figure 5 illustrates PADs on tasks A→W and A→D with features of ResNet-50, DAN, CDAN+E, CAN and IC^{2}FA. We observe that the PAD of IC^{2}FA is much smaller than comparison methods on the two tasks; this demonstrates that our features can reduce the cross-domain gap more effectively.

5 CONCLUSION

In this paper, we develop a new approach, named IC^{2}FA, to address the problem of UDA. It incorporates two main components—specifically, disentangling the informative features and adapting the class-level features— which they work cooperatively. The VIB is delicately applied to disentangle these two domains. The sliced Wasserstein distance is extended into a new metric, CSWD, which is employed to explicitly measure the class-level discrepancy. We equip the discriminative features alignment with the informative feature disentanglement, facilitating the adaption process and easing the adaptation process. Extensive experimental evaluations clearly demonstrate the effectiveness of IC^{2}FA.

ACKNOWLEDGMENTS

This work was partially supported by the Academy of Finland under grant 331883, and the National Natural Science Foundation of China under Grant 61872379, 62022091 and 71701205.

REFERENCES

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2017. Deep variational information bottleneck. (2017).

[2] Gretton Arthur, Borgwardt Karsten, Rasch Malte, Schoelkopf Bernhard, and Smola Alex. 2012. A Kernel Two-Sample Test. *Journal of Machine Learning Research* 13 (2012), 723–773.

[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79, 1-2 (2010), 151–175.

[4] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2007. Analysis of representations for domain adaptation. In *NIPS*. 137–144.

[5] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. 2015. Sliced and Radon Wasserstein Barycenters of Measures. *J. Math. Imaging Vis.* 51, 1 (2015), 22–45.

[6] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *NIPS*. 343–351.

[7] Woong Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. 2019. Domain-Specific Batch Normalization for Unsupervised Domain Adaptation. In *CVPR*. 7354–7362.

[8] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. 2019. Progressive Feature Alignment for Unsupervised Domain Adaptation. In *CVPR*. 627–636.

[9] Gabriela Csurka. 2017. A Comprehensive Survey on Domain Adaptation for Visual Applications. In *Domain Adaptation in Computer Vision Applications*. Springer, 1–35.

[10] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. 2018. DeepJDOT: Deep Joint Distribution Optimal Transport for Unsupervised Domain Adaptation. In *ECCV*, Vol. 11208. Springer, 467–483.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*. 248–255.

[12] Ishan Deshpande, Ziyu Zhang, and Alexander G. Schwing. 2018. Generative Modeling Using the Sliced Wasserstein Distance. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 3483–3491.

[13] Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. In *ICML*.

[14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *JMLR* 17, 1 (2016), 2096–2030.

[15] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, Pablo Martinez-Gonzalez, and Jose Garcia-Rodriguez. 2018. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing* 70 (2018), 41–65.

[16] Kang Guoliang, Jiang Lu, Yang Yi, and Hauptmann Alexander, G. 2019. Contrastive Adaptation Network for Unsupervised Domain Adaptation. In *CVPR*.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[18] Yan Hongliang, Ding Yukang, Li Peihua, Wang Qilong, Xu Yong, and Zuo Wangmeng. 2017. Mind the Class Weight Bias: Weighted Maximum Mean Discrepancy for Unsupervised Domain Adaptation. In *CVPR*. 2272–2281.

[19] Yan Hongliang, Li Zhetao, Wang Qilong, Li Peihua, Xu Yong, and Zuo Wangmeng. 2019. Weighted and Class-specific Maximum Mean Discrepancy for Unsupervised Domain Adaptation. *IEEE Transactions on Multimedia* (2019).

[20] Goodfellow Ian, Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherjil, Courville Aaron, and Bengio Yoshua. 2014. Generative Adversarial Nets. In *NeurIPS*. 2672–2680.

[21] Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. 2020. Implicit Class-Conditioned Domain Alignment for Unsupervised Domain Adaptation. In *ICML (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 4816–4827.

[22] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. (2014).

[23] Soheil Kolouri, Se Rim Park, and Gustavo K. Rohde. 2016. The Radon Cumulative Distribution Transform and Its Application to Image Classification. *IEEE Trans. Image Process.* 25, 2 (2016), 920–934.

[24] Soheil Kolouri, Phillip E. Pope, Charles E. Martin, and Gustavo K. Rohde. 2019. Sliced Wasserstein Auto-Encoders. In *ICLR*.

[25] Wouter M Kouw and Marco Loog. 2020. A review of domain adaptation without target labels. *IEEE Transactions on pattern analysis and machine intelligence* (2020).

[26] Vinod Kumar Kurmi, Shanu Kumar, and Vinay P. Namboodiri. 2019. Attending to Discriminative Certainty for Domain Adaptation. In *CVPR*. Computer Vision Foundation / IEEE, 491–500.

[27] Van Der Maaten Laurens and Geoffrey Hinton. 2008. Visualizing Data Using t-SNE. *JMLR* 9 (2008), 2579–2605.

[28] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. 2019. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*. 10285–10295.

[29] Mengxue Li, Yiming Zhai, You-Wei Luo, Pengfei Ge, and Chuan-Xian Ren. 2020. Enhanced Transport Distance for Unsupervised Domain Adaptation. In *CVPR*. IEEE, 13933–13941.

[30] Shuang Li, Chi Harold Liu, Qiuxia Lin, Binhui Xie, Zhengming Ding, Gao Huang, and Jian Tang. 2020. Domain Conditioned Adaptation Network. In *AAAI*. AAAI Press, 11386–11393.

[31] Tsung Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*. 740–755.

[32] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. 2020. Deep learning for generic object detection: A survey. *International Journal of Computer Vision* 128, 2 (2020), 261–318.

[33] Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Transferable representation learning with deep adaptation networks. *IEEE transactions on pattern analysis and machine intelligence* (2018).

[34] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. 2015. Learning transferable features with deep adaptation networks. In *ICML*.

[35] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2016. Unsupervised domain adaptation with residual transfer networks. In *NIPS*. 136–144.

[36] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep transfer learning with joint adaptation networks. In *ICML*, Vol. 70. JMLR, 2208–2217.

[37] Mingsheng Long†, Zhangjie Cao†, Jianmin Wang†, and Michael I. Jordan. 2017. Conditional Adversarial Domain Adaptation. *NIPS* (2017).

[38] Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. In *ICLR*.

[39] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. 2019. Significance-Aware Information Bottleneck for Domain Adaptive Semantic Segmentation. In *ICCV*. IEEE, 6777–6786.

[40] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain Adaptation: Learning Bounds and Algorithms. In *COLT*.

[41] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. 2020. Image Segmentation Using Deep Learning: A Survey. *arXiv preprint arXiv:1702.05374* (2020).

[42] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* 22, 10 (2010), 1345–1359.

[43] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong Wah Ngo, and Tao Mei. 2019. Transferrable Prototypical Networks for Unsupervised Domain Adaptation. In *CVPR*. 2239–2247.

[44] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. 2018. Multi-adversarial domain adaptation. In *AAAI*.

[45] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. 2017. VisDA: The Visual Domain Adaptation Challenge. *arXiv preprint arXiv:1710.06924* (2017).

[46] Pedro O Pinheiro. 2018. Unsupervised Domain Adaptation with Similarity Learning. In *CVPR*. 8004–8013.

[47] Chen Qingchao, Liu Yang, Wang Zhaowen, Wassell Ian, and Chetty Kevin. 2018. Re-Weighted Adversarial Adaptation Network for Unsupervised Domain Adaptation. In *CVPR*. 7976–7985.

[48] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. 2011. Wasserstein Barycenter and Its Application to Texture Mixing. In *Scale Space and Variational Methods in Computer Vision - Third International Conference, SSVM*, Vol. 6667. Springer, 435–446.

[49] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. [n.d.]. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 ([n. d.]), 211–252.

[50] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting visual category models to new domains. In *ECCV*. 213–226.

[51] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*. 3723–3732.

[52] F. Santambrogio. 2015. *Optimal Transport for Applied Mathematicians*. Optimal Transport for Applied Mathematicians.

[53] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *AAAI*.

[54] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. 2007. Covariate Shift Adaptation by Importance Weighted Cross Validation. *J. Mach. Learn. Res.* 8 (2007), 985–1005.

[55] Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*. Springer, 443–450.

[56] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*. 1701–1708.

[57] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. (2000), 368–377.

[58] Antonio Torralba and Alexei A. Efros. 2011. Unbiased look at dataset bias. In *Computer Vision Pattern Recognition*.

[59] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *CVPR*. 7167–7176.

[60] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep hashing network for unsupervised domain adaptation. In *CVPR*. 5018–5027.

[61] Mei Wang and Weihong Deng. 2018. Deep visual domain adaptation: A survey. *Neurocomputing* 312 (2018), 135–153.

[62] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. 2019. Transferable attention for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5345–5352.

[63] Hanrui Wu, Yuguang Yan, Michael K. Ng, and Qingyao Wu. 2020. Domain-attention Conditional Wasserstein Distance for Multi-source Domain Adaptation. *ACM Trans. Intell. Syst. Technol.* 11, 4 (2020), 44:1–44:19.

[64] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. 2019. Sliced Wasserstein Generative Models. In *CVPR*. 3713–3722.

[65] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. 2018. Learning semantic representations for unsupervised domain adaptation. In *ICML*. 5423–5432.

[66] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. 2020. Reliable Weighted Optimal Transport for Unsupervised Domain Adaptation. In *CVPR*. IEEE, 4393–4402.

[67] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. 2017. Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning. (2017).