Informative Feature Disentanglement for Unsupervised Domain Adaptation

Wanxia Deng, Lingjun Zhao, Qing Liao, Deke Guo, Gangyao Kuang, Dewen Hu, Matti Pietikäinen, and Li Liu

***Abstract*—Unsupervised Domain Adaptation (UDA) aims at learning a classifier for an unlabeled target domain by transferring knowledge from a labeled source domain with a related but different distribution. The strategy of aligning the two domains in latent feature space via metric discrepancy or adversarial learning has achieved considerable progress. However, these existing approaches mainly focus on adapting the entire image and ignore the bottleneck that occurs when forced adaptation of uninformative domain-specific variations undermines the effectiveness of learned features. To address this problem, we propose a novel component called Informative Feature Disentanglement (IFD), which is equipped with the adversarial network or the metric discrepancy model, respectively. Accordingly, the new network architectures, named IFDAN and IFDMN, enable informative feature refinement before the adaptation. The proposed IFD is designed to disentangle informative features from the uninformative domain-specific variations, which are produced by a Variational Autoencoder (VAE) with lateral connections from the encoder to the decoder. We cooperatively apply the IFD to conduct supervised disentanglement for the source domain and unsupervised disentanglement for the target domain. In this way, informative features are disentangled from the domain-specific details before the adaptation. Extensive experimental results on three gold-standard domain adaptation datasets, e.g., Office31, Office-Home and VisDA-C, demonstrate the effectiveness of the proposed IFDAN and IFDMN models for UDA.**

Manuscript received December 24, 2020; revised March 27, 2021 and May 6, 2021; accepted May 10, 2021. Date of publication May 20, 2021; date of current version May 11, 2022. This work was supported in part by the National Natural Science Foundation of China under Grants 62022091, 61872379, 71701205, 61701508, and 62036013, in part by the Academy of Finland under Grant 331883, and in part by Hunan Provincial Natural Science Foundation of China under Grant 2018JJ3613. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xin Geng. *(Corresponding author: Li Liu.)*

Wanxia Deng, Lingjun Zhao, and Gangyao Kuang are with the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics, and Information System, College of Electronic Science, National University of Defense technology, Changsha 410073, China (e-mail: wanxiadeng@163.com; nudtzlj@163.com; kuanggangyao@nudt.edu.cn).

Qing Liao is with the Department of Computer Science, and Technology, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: liaoqing@hit.edu.cn).

Deke Guo is with the College of System Engineering, National University of Defense Technology, Changsha 410073, China (e-mail: guodeke@gmail.com).

Dewen Hu is with the College of Intelligent Science, National University of Defense Technology, Changsha 410073, China (e-mail: dwhu@nudt.edu.cn).

Matti Pietikäinen is with the Center for Machine Vision, and Signal analysis, University of Oulu, 90570 Oulu, Finland (e-mail: matti.pietikainen@oulu.fi).

Li Liu is with the College of System Engineering, National University of Defense Technology, Changsha 410073, China, and is also with the Center for Machine Vision, and Signal analysis, University of Oulu, 90570 Oulu, Finland (e-mail: dreamliu2010@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3080516.

Digital Object Identifier 10.1109/TMM.2021.3080516

***Index Terms*—Domain Adaptation, deep learning, deep convolutional neural network, autoencoder, transfer learning, unsupervised learning.**

I. Introduction

RECENT advancements in Deep Neural Networks (DNNs), have brought a huge success in a broad range of computer vision tasks, such as image classification [1], object detection [2], image segmentation [3], [4], and face recognition [5]. Nevertheless, the impressive progress depends on strong supervision, i.e., massive amounts of annotated data which are painstakingly labeled by numerous workers or specialists. Manual labels are often difficult or expensive to obtain, and especially for data-sensitive domains such as medical imagery and industrial inspection, labeled samples may be even impossible.

To address the aforementioned problems, an alternative approach (e.g., transfer learning) is to leverage the massively available labeled data on the related domain (dubbed *source domain*), to improve the model for the interested domain (dubbed *target domain*) [6]. However, the recent evidence [7] indicates that DNNs have a strong dependency on the dataset with which they are originally trained, and the learned features cannot be easily transferred to a different domain without adjusting [8], [9]. This difficulty in transferring is caused by domain shift [10]; i.e., predictors trained on a source domain undergo a drastic drop in performance when applied to the target domain. Illustratively, the domain shift refers to the difference in data distributions between source and target domains and is caused by many factors, such as different backgrounds, changes in viewing angles, occlusions and volatile illumination conditions. To tackle the above domain shift problem, Domain Adaptation (DA), as a subfield of transfer learning, has been proposed. The objective of DA is to leverage labeled data from one or more similar domains (source domain) to improve the learning of the interested domain (target domain) that has a distribution different from but related to the source distribution.

In this paper, we address one category of DA, *i.e.*, the problem of Unsupervised DA (UDA), where the source domain contains abundant labeled data while the target domain is fully unlabeled. The general idea of UDA is to make extracted features similar between the two domains [11], [12]. Therefore, most existing UDA approaches are devoted to embedding adaptation modules in deep architectures to extract transferable features [13]–[16]. The state-of-the-art methods realize this goal by either

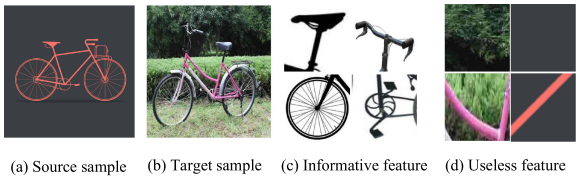


Fig. 1. (a) and (b) denote the "bike" samples of source and target domain, respectively. (c) denotes the informative features, e.g., handlebar, saddle, wheel and paddle, which are more transferable. (d) shows some useless information, e.g., background and colors, for the domain adaptation and the object task.

conducting adversarial learning or minimizing a metric that defines the distribution discrepancy.

Metric discrepancy-based methods [17]–[23] explicitly decrease the distribution discrepancy by measuring the dissimilarity between distributions and are widely used due to their efficient implementation. Recently, adversarial network-based methods [24]–[31] have emerged as some of the most prominent UDA methods. They train a domain discriminator to distinguish whether the features are from the source or target domain and then borrow adversarial ideas from Generative Adversarial Networks (GANs) [32] to deceive the domain discriminator in order to obtain indistinguishable features. Despite their general simplicity and efficacy for UDA, these methods may still be constrained by the bottleneck caused by application of the whole image to conduct the feature adaptation. Intuitively, not all image regions are transferable, and forcefully aligning the domain-specific variations may lead to negative transfer. There are domain-specific details in an image that may be unsuitable for the adaptation and useless for improving the classification, as illustrated in Fig 1. From Fig. 1(c), we can see that important object parts are informative for adaptation and object classification tasks, while some parts, as shown in Fig. 1 (d), representing domain-specific variations such as background and object color, are uninformative. To address this issue, we propose Informative Feature Disentanglement (IFD) module to select regions that can be adapted, which is integrated into the adversarial network and metric discrepancy module, leading to the novel architectures named IFDAN and IFDMN, respectively.

The proposed IFD realizes two complementary disentanglement strategies: supervised representation disentanglement of the source domain by focusing on transferable regions, and unsupervised disentanglement of the target domain by suppressing the less useful domain-specific details of an image. In the supervised strategy, inspired by the Information Bottleneck (IB) principle [33], we use an adversarial excitation and inhibition mechanism to encourage the disentanglement of the latent variables via Variational Information Bottleneck (VIB) [34]. Our goal is to learn an encoding that is maximally informative about the object classification task while being maximally compressive about the original input, as shown in Fig. 2(a). 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 the original input is inhibited. By this means, only the features most conducive to the downstream classification task can pass through, which reveals that the learned representation is more generalizable and transferable.

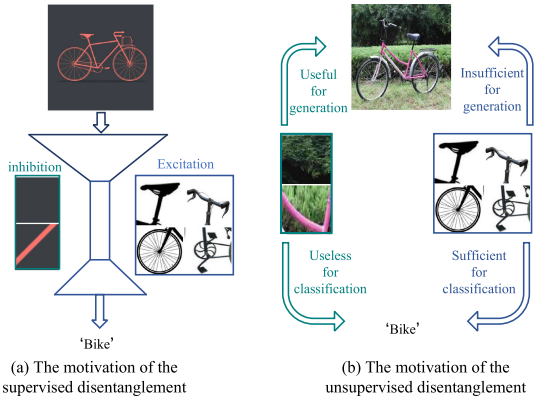


Fig. 2. The motivation of the proposed IFD module. (a) The encoding only encourages the most informative features, e.g., handlebar, saddle, wheel and paddle, for the classification task to pass through and inhibits the irrelevant domain-specific information, e.g., background and colors, so that the downstream task only obtains the most representative, high-level discriminative features. (b) The informative features, e.g., handlebar, saddle, wheel and paddle, which are what we care about, are enough for classification but not enough for reconstruction. For example, if we learn these object parts such as saddle and wheel, we can infer that the target is a bicycle. But we do not know the compositional information and other details of the seat and wheel, so we can not reconstruct it well. Some useless information, e.g., background and colors, can not contribute to the classification task, but can help generate images. Inspired by this, we are committed to gradually disentangle informative features for the object classification task in the process of unsupervised reconstruction.

In the unsupervised paradigm, since the target domain is unlabeled, the information bottleneck principle is not applicable. We intend to introduce an alternative way to learn disentangled representations by crafting a network architecture that prefers to hierarchically learn high-level features from certain parts of the latent code. In this way, we can disentangle the high-level abstract features only, which are applied to the adaptation. Figure 2 (b) shows that high-level salient features are often sufficient for classification but are not sufficient for the good image generation task, which requires preservation of low-level image details. Motivated by this, we are devoted to gradually separating useful features for the downstream task in the process of the unsupervised reconstruction of the target domain. Recent research on the Variational Ladder Autoencoder (VLAE) [35] discloses that the Variational Autoencoder (VAE) with lateral connections can extract high-level representations that are helpful for discriminative tasks by the encoder pipeline, and can achieve the goal of image generation. Inspired by this insight, we introduce VLAE for UDA. We apply VLAE to disentangle the target domain and extract the transferable latent representations hierarchically. We combine the supervised disentanglement of the source domain and the unsupervised disentanglement of the target domain. In this way, the learned encoding is transferable high-level semantic features. We equip the IFD with the representative adversarial network, enabling an informative feature purification before adversarial adaptation. In summary, the main contributions of this paper are as follows.

- We propose an IFD module to extract more transferable high-level semantic features combining the supervised disentanglement of the source domain and the unsupervised disentanglement of the target domain. The two disentanglement strategies are achieved via IB theory and VLAE,

respectively. To the best of our knowledge, we are the first to successfully disentangle the two domains via IB and VLAE for UDA.

- We propose incorporating the proposed IFD into the adversarial network and the metric discrepancy model for UDA. The new networks, named IFDAN and IFDMN, inherit the advantages and overcome the drawbacks (i.e., applying the entire image to implement adaptation) of existing adversarial learning and metric discrepancy-based UDA methods, thus leading to significant performance improvement.
- Experiments on the Office31, Office-Home and VisDA-C datasets are provided to demonstrate the proposed IFDAN and IFDMN outperform existing methods. In addition, we conduct careful ablation studies on benchmark UDA datasets, verifying the efficacy of the proposed IFD module.

The remainder of the paper is organized as follows. In Section II, we review the related work. Section III introduces the formulation of our network. Section IV reports the experimental results and analysis on Office-31, Office-Home and VisDA-C datasets. Our conclusion and future work are presented in Section V.

II. Related Works

Research in UDA is so vast that only closely related work is discussed in this section. In general, UDA can be coarsely classified into two categories, i.e., the feature-level and the pixel-level.

A. Feature-Level UDA

Most existing feature-level UDA methods align source and target domains by minimizing discrepancy metrics or adversarial learning of domains.

Metric discrepancy-based methods Minimizing the domain distribution discrepancy with the metric paradigm is one more classical approach for UDA. Some representative metric methods include Maximum Mean Discrepancy (MMD) [17]–[23], [36], [37], CORrelation (CORAL) ALignment [38] and Central Moment Discrepancy (CMD) [39]. In [17], [19], the DAN is proposed to explore the multi-kernel MMD (MK-MMD) [40] metric to minimize marginal distributions of two domains. The JAN [18] aligns the joint distributions of multiple domain-specific layers via a Joint Maximum Mean Discrepancy (JMMD) criterion. The paper [21] exploits the class prior probability on source and target domains via a weighted MMD model which introduces class-specific auxiliary weights into the original MMD. The Contrastive Adaptation Network (CAN) [22] explicitly models the intra-class domain discrepancy and the inter-class domain discrepancy based on MMD metric. The D-CORAL [41] extends CORAL to learn a nonlinear transformation that aligns correlations of layer activations in deep neural network by adding the adaptation layer. The Central Moment Discrepancy (CMD) [39] is suggested to match the higher-order central moments of probability distributions.

Adversarial learning-based methods An alternative branch of UDA is based on the adversarial learning of domains, which is inspired by the Generative Adversarial Network (GAN) [32]. Adversarial learning has been widely applied in UDA to deceive the domain discriminator. RevGrad [24], DANN [26], Adversarial Discriminative Domain Adaptation (ADDA) [25] and Conditional Domain Adversarial Network (CDAN) [30] utilize a domain discriminator to represent the domain discrepancy. The domain discriminator is confused the domain discriminator in a two-player minimax game. MADA [29] trains multiple class-wise domain discriminators to capture multi-mode structures to enable fine-grained alignment of different data distributions. Domain symmetric Networks (SymNets) [42] constructs an additional classifier that shares with source and target classifiers for DA. The Wasserstein Distance Guided Representation Learning (WDGRL) [27] estimates empirical Wasserstein distance between the source and target samples in domain critic network and optimizes the feature extractor network to minimize the estimated Wasserstein distance in an adversarial manner. Similar to the motivation of WDGRL, Re-weighted Adversarial Adaptation Network (RAAN) [43] minimizes the optimal transport (OT) based Earth Mover (EM) distance and reformulates it to a minimax objective function. Unlike the above domain adversarial learning methods, Maximum Classifier Discrepancy (MCD) [44] defines a new adversarial standard in developing generic DA frameworks. The MCD utilizes task-specific classifiers as discriminators and aligns distributions of source and target by the adversarial learning of two task-specific classifiers. Similar to the adaptation standard of the MCD, Sliced Wasserstein Discrepancy (SWD) [45] adopts the Wasserstein metric to minimize the cost of moving the marginal distributions between the task-specific classifiers. Many other adversarial DA methods are extensions of the representative DANN, such as Moving Semantic Transfer Network (MSTN) [46], Transferable Adversarial Training (TAT) [47] and so on.

In addition, there are other feature-level UDA approaches, e.g., the work [48] combines the adversarial learning with the metric discrepancy, which adjusts the weight of the adversarial loss to confuse the domains and employs the triplet loss to achieve the class-level alignment. Minimum Centroid Shift (MCS) [49] is built upon the subspace learning, aiming to seek a subspace where the centroids in the target domain are shifted from those in the source domain. Structurally Regularized Deep Clustering (SRDC) [50] applies the clustering to conduct the discriminative features assignment which mines the target discrimination using the clustering learning of intermediate network features. Stepwise Adaptive Feature Norm (SAFN) [51] achieves more transferable features via progressively adapting the feature norms of the two domains to a large range of values.

B. Pixel-Level UDA

Pixel-level UDA methods generate a new version of images. The generated images and original images are applied together to learn domain-invariant features.

Generative adversarial networks (GAN) based methods There are more popular DA approaches which incorporate the generative model into the feature learning process using Generative Adversarial Networks (GAN). The pixel-level domain

adaptation (PixelDA) [52] learns to generate a new version of the source images in the style of the target domain, so that one shared classifier can accommodate both domains. Couple GAN (CoGAN) [53] trains coupled generators and discriminators by weight-sharing to learn the joint distribution across the two domains. The Domain Transfer Network (DTN) [54] synthesizes source domain samples that resemble target domain ones by the extra consistency constraint that the same asymmetric transformation should keep the target domain samples identical. Generate To Adapt (GTA) [55] proposes an adversarial image generation approach to learn the feature embedding using a combination of generated source-like images classification loss and an image generation procedure. The Cycle-consistent Adversarial Domain Adaptation (CyCADA) [56], the UNsupervised Image-to-image Translation (UNIT) [57], Deep Adversarial Attention Alignment (DAAA) [58] and SBADA-GAN [59] constrain the mapping to be well covered across two domains by imposing cycle consistency: the mapping in one direction (source-to-target or target-to-source) should get back where it started. The Image to Image translation (I2I) [60] combines domain-agnostic feature extraction, domain generation and cycle consistency into a single unified framework for domain knowledge.

Autoencoder reconstruction based methods The DA approaches based on autoencoder reconstruction typically learn the domain-invariant feature with a shared encoder and a reconstruction loss of the autoencoder. DRCN [61] is proposed to jointly learn common encoding representation combining the supervised classification of source domain and the unsupervised reconstruction of the target domain. Domain Separation Network (DSN) [62] proposes to separate the feature into the shared feature and the private feature. These two features are encouraged to be orthogonal while both the features can be decoded back to images. The shared feature is adopted for classification. On the basis of the DSN, the work [63] proposes that orthogonal regularization is applied between private features across domains, in addition to private features and shared features in each domain.

The DSN and the extension of the DSN have the same motivation as ours. They explicitly disentangle the domain-specific and domain-shared features. Our proposed method implicitly disentangle the task-related high-level features and task-unrelated details. Apart from the above two methods, there are other types of newly proposed methods similar to our intention. The Transferable Attention for Domain Adaptation (TADA) [64] proposes to apply the attention mechanism for UDA, which present transferable attention, focusing the adaptation model on transferable regions not all regions of an image. The Domain-Specific Batch Normalization (DSBN) [65] is proposed to separate domain-specific information for UDA using two branches of batch normalization, each of which is in charge of a single domain exclusively.

III. Proposed Methodology

In this section, we first give the UDA problem formulation and then introduce the proposed Information Features Disentanglement (IFD) module, mainly including supervised disentanglement of the source domain and unsupervised disentanglement of the target domain. Finally, we show how to train the IFDAN, which is a combination of adversarial network and the proposed IFD, and the IFDMN, which integrates the metric discrepancy model with IFD. The proposed IFDAN and IFDMN are depicted in Fig. 3.

A. The UDA Problem Formulation

Consider an input space (or a feature representation space) $\mathcal{X}$, and an output space (or a label space) $\mathcal{Y}$. We focus on the problem of UDA in image classification, where we consider two different domains defined with different but related probability distributions $p(x, y)$ over the input-label space pair $\mathcal{X} \times \mathcal{Y}$. Specifically, the domain of interest is denoted the target domain with the distribution $p_t(x, y)$, and the available domain with labeled data is called the source domain with the distribution $p_s(x, y)$. Let $|\mathcal{Y}| = C$; then we have $y \in 1, 2, \ldots, C$. The goal of the UDA task is to predict the labels of samples drawn from a target domain as accurate as possible, given N_s labeled samples $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ drawn from a source domain and N_t unlabeled samples $D_t = \{(x_i^t)\}_{i=1}^{N_t}$ sampled from the target domain itself. Our goal is to learn feature transformations to map the input space $\mathcal{X}$ to a feature space $\mathcal{F}$, where distributions of the two domains are well aligned in order to obtain the domain-invariant feature so that the target domain images can be classified into $\mathcal{Y}$ correctly.

Formally, our proposed IFDAN can be formulated as follows. We define the shared encoder of IFDAN as $\mathbb{F}$ with parameters $\boldsymbol{\theta}$, the decoder as $\mathbb{G}$ with parameters $\boldsymbol{\alpha}$, and the classification block as $\mathbb{H}$ with parameters $\boldsymbol{\phi}$. $\mathbb{F}$ consists of convolutional, ReLU, and downsampling layers. $\mathbb{G}$ consists of convolutional, ReLU, and upsampling layers.

B. IFD: Informative Feature Disentanglement

In this section, we introduce how to conduct the supervised disentanglement of the source domain and the unsupervised disentanglement of the target domain, respectively.

1) Supervised Disentanglement of the Source Domain: To extract the general high-level features for UDA, we choose the Variational Information Bottleneck (VIB) built upon the recently developed information theories for deep learning [34] to disentangle the source domain. To facilitate the discussion, we denote X^s as the input images from the source domain. Let Y^s denote the corresponding output variables (e.g., the desired label), whose information we want to preserve. We regard the internal representation of some intermediate layer as a stochastic encoding Z^s of the input images X^s, defined by the shared parametric encoder $p_{\boldsymbol{\theta}}(z^s|x^s)$. For clarity, we denote x^s, y^s and 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 about 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. Thus, we assume the following Markov chain constraint introduced in the Information Bottleneck (IB) theory [33]: $Y^s \leftrightarrow X^s \leftrightarrow Z^s$, and the objective function that is maximized is defined as follows:

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

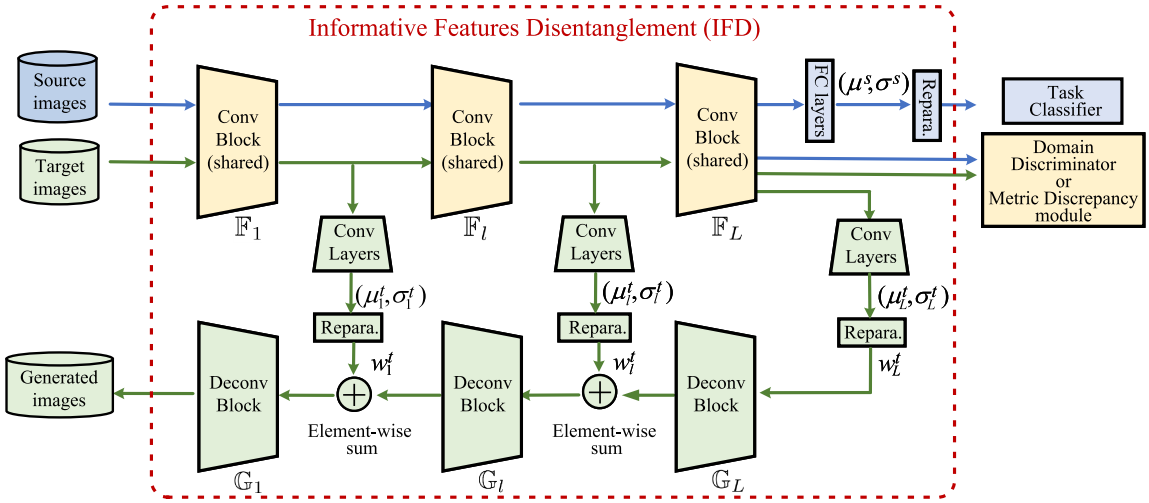


Fig. 3. Framework of the proposed IFDAN and IFDMN. The IFDAN is the combination of adversarial network and the proposed IFD, and similarly, IFDMN combines the metric discrepancy module with IFD. The IFD module is constructed with Variational Information Bottleneck (VIB) and variational ladder autoencoder (VLAE). The encoder is shared by the source images and target images. The IFD module mainly includes two strategies: supervised disentanglement of the source domain and unsupervised disentanglement of the target domain. The source domain is disentangled by VIB, and the target domain is disentangled by the VLAE.

where β denotes Lagrange multiplier. The first term $I(Z^s, Y^s; \boldsymbol{\theta}) = \int d_{z^s} d_{y^s} p_{\boldsymbol{\theta}}(z^s, y^s) log \frac{p_{\boldsymbol{\theta}}(z^s, y^s)}{p_{\boldsymbol{\theta}}(z^s) p_{\boldsymbol{\theta}}(y^s)}$[1] encourages Z^s to be predictive of Y^s. The second term $I(X^s, Z^s; \boldsymbol{\theta}) = \int d_{z^s} d_{x^s} p_{\boldsymbol{\theta}}(z^s, x^s) log \frac{p_{\boldsymbol{\theta}}(z^s|x^s)}{p_{\boldsymbol{\theta}}(z^s)}$ encourages Z^s to inhibit as many details of X^s as possible.

However, computing mutual information is computationally challenging. We write the first term out in full, as follows:

$$\begin{aligned} I(Z^s, Y^s; \boldsymbol{\theta}) &= \int d_{z^s} d_{y^s} p_{\boldsymbol{\theta}}(z^s, y^s) log \frac{p_{\boldsymbol{\theta}}(z^s, y^s)}{p_{\boldsymbol{\theta}}(z^s) p_{\boldsymbol{\theta}}(y^s)} \\ &= \int d_{z^s} d_{y^s} p_{\boldsymbol{\theta}}(z^s, y^s) log \frac{p_{\boldsymbol{\theta}}(y^s|z^s)}{p_{\boldsymbol{\theta}}(y^s)}. \end{aligned} \tag{2}$$

Since the $p_{\boldsymbol{\theta}}(y^s|z^s)$ is intractable, we apply $q_{\boldsymbol{\phi}}(y^s|z^s)$ to be a variational approximation to $p_{\boldsymbol{\theta}}(y^s|z^s)$. The $q_{\boldsymbol{\phi}}(y^s|z^s)$ is the defined decoder of VIB, which we will take to be the classification block $\mathbb{H}_{\boldsymbol{\phi}}$ 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: $\int d_{y^s} log p_{\boldsymbol{\theta}}(y^s|z^s) \geqslant \int d_{y^s} log q_{\boldsymbol{\phi}}(y^s|z^s)$. Thus 2 can be rewritten as:

$$\begin{aligned} I(Z^s, Y^s; \boldsymbol{\theta}, \boldsymbol{\phi}) &\geqslant \int d_{z^s} d_{y^s} p_{\boldsymbol{\theta}}(z^s, y^s) log \frac{q_{\boldsymbol{\phi}}(y^s|z^s)}{p_{\boldsymbol{\theta}}(y^s)} \\ &= \int d_{z^s} d_{y^s} p_{\boldsymbol{\theta}}(z^s, y^s) log q_{\boldsymbol{\phi}}(y^s|z^s) - \\ &\quad \int d_{y^s} p_{\boldsymbol{\theta}}(y^s) log p_{\boldsymbol{\theta}}(y^s) \\ &= \int d_{z^s} d_{y^s} p_{\boldsymbol{\theta}}(z^s, y^s) log q_{\boldsymbol{\phi}}(y^s|z^s) + H(Y^s), \end{aligned} \tag{3}$$

where $H(Y^s)$ is the entropy of our labels, which is independent of the optimization procedure and so can be ignored. Recalling the Markov chain constraint, $I(Z^s, Y^s; \boldsymbol{\theta})$ can achieve a new lower bound, and hence we can write:

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

We now consider the second term $I(X^s, Z^s; \boldsymbol{\theta})$ of 1. The $I(X^s, Z^s; \boldsymbol{\theta})$ can be further computed as follows:

$$\begin{aligned} I(X^s, Z^s; \boldsymbol{\theta}) &= \int d_{x^s} d_{z^s} p_{\boldsymbol{\theta}}(z^s, x^s) log p_{\boldsymbol{\theta}}(z^s|x^s) \\ &\quad - \int d_{z^s} p_{\boldsymbol{\theta}}(z^s) log p_{\boldsymbol{\theta}}(z^s). \end{aligned} \tag{5}$$

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

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

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

$$\begin{aligned} &I(Z^s, Y^s; \boldsymbol{\theta}, \boldsymbol{\phi}) - \beta I(X^s, Z^s; \boldsymbol{\theta}) \\ &\geqslant \int d_{x^s} d_{z^s} d_{y^s} p_{\boldsymbol{\theta}}(x^s) p_{\boldsymbol{\theta}}(y^s|x^s) p_{\boldsymbol{\theta}}(z^s|x^s) log q_{\boldsymbol{\phi}}(y^s|z^s) \\ &\quad - \beta \int d_{x^s} d_{z^s} p_\theta(x^s) p_\theta(z^s|x^s) \frac{log p_\theta(z^s|x^s)}{r(z^s)} \\ &= \mathcal{L}^{sup}_{ELBO}(\boldsymbol{\theta}, \boldsymbol{\phi}) \end{aligned} \tag{7}$$

[1]Note that in the present discussion, Y^s is the ground truth label which is independent of our parameters $\boldsymbol{\theta}$, so $p_{\boldsymbol{\theta}}(y^s) = p(y^s)$.

Following the VAE [34], we define the $p_\theta(z^s|x^s)$ as a Gaussian distribution $p_\theta(z^s|x^s) = \mathcal{N}(z^s|\mathbb{F}^\mu_{IB}(x^s), \mathbb{F}^\sigma_{IB}(x^s))$, where $\mathbb{F}_{IB}$ denotes the information bottleneck layers (Fc layers in Fig. 3), which outputs the mean μ and the variance σ of latent features z^s. Then, we can use the reparameterization trick [66] to write $p_\theta(z^s|x^s)dz = p_\theta(\epsilon)d\epsilon$, where $z^s = \mathbb{F}(x^s, \epsilon)$ denotes the deterministic function of x and the Gaussian random variable ϵ. Thus, we can obtain the following loss function, which we try to minimize:

$$\mathcal{L}_{sup} = \frac{1}{N_s}\sum_{i=1}^{N_s}\mathbb{E}_{\epsilon\sim p_\theta(\epsilon)}[-logq_\phi(y_i^s|\mathbb{F}(x_i^s,\epsilon))] + \beta_{sup}KL[p_\theta(z^s|x_i^s), r(z^s)], \tag{8}$$

where the first term is a form of the classification loss of the source domain and the second term denotes the information bottleneck loss, which is minimized to filter out the irrelevant part of the input image X^s. Interestingly, maximizing a variational lower bound of the mutual information between learned representations Z^s and the labels Y^s in the source domain is equivalent to minimizing the classification loss, and minimizing the mutual information of learned representations Z^s and the input X^s corresponds to finding the transferable features. β_{sup} is used to replace the β for easy understanding.

2) Unsupervised Disentanglement of the Target Domain: In this section, we introduce the unsupervised disentanglement of the target domain. Since there is no label in the target domain, the supervised disentanglement is impossible. We apply an alternative way to disentangle the target domain, e.g., Variational Ladder Autoencoder (VLAE), which can skillfully reconstruct the target domain to mine the features of the target domain, at the same time, it can ensure that the encoder can only capture the task-related part.

We denote the X^t as the input images from the target domain, and for simplicity, let the x^t be the instance of X^t. The input image undergoes the encoder to generate hierarchical feature representations:

$$z_l^t = \begin{cases} \mathbb{F}_l(x^t), & l = 1 \\ \mathbb{F}_l(z_{l-1}^t), & 2 \leqslant l \leqslant L, \end{cases} \tag{9}$$

where L represents the total number of layers of the encoder, and l denotes the variable of layers. z_l^t denotes the extracted encoding of the target domain at layer l. We decompose the latent representations into subparts $z^t = \{z_1^t, \ldots z_l^t, \ldots, z_L^t\}$, where z_1^t relates to x^t with a shallow network $\mathbb{F}_1$, and increase network depth up to z_L^t, which relates to x^t with a deep network $\mathbb{F}_L$.

The latent representations z^t from the encoder are delivered into the variational ladder module (Conv layers in Figure 3) to generate the new hierarchical representations $w^t = \{w_1^t, \ldots w_l^t, \ldots, w_L^t\}$ for the decoder. The hierarchical representations mainly pay attention to the details of the images. In this way, the shared encoder can gradually capture the task-related features and suppress the task-related details via the variational ladder module. The new hierarchical representations $w^t = \{w_1^t, \ldots w_l^t, \ldots, w_L^t\}$ are defined as follows:

$$w_l^t \sim \mathcal{N}(\mu_l^t, \sigma_l^t), \tag{10}$$

where μ_l^t and σ_l^t are the mean and variance of z_l^t. w_l^t is subject to the Gaussian distribution with mean μ_l^t and variance σ_l^t. The latent representation z^t from the ladder module undergoes the decoder to reconstruct $\hat{x^t}$ via the following:

$$\hat{w}_l^t = \begin{cases} \mathbb{G}_l(\textit{Repara.}(\mu_l^t, \sigma_l^t)), & l = L \\ \mathbb{G}_l(w_l^t \oplus \hat{w}_{l+1}^t), & L > l \geq 1 \end{cases}, \tag{11}$$

where $\hat{w}_l^t$ indicates the l^{th} layer reconstruction feature of the target domain in the decoder. $\hat{w}_1^t$ is the reconstructed output $\hat{x^t}$. $\hat{x^t}$ is the output of the decoder, i.e., it is the reconstruction of x^t. Here, $\oplus$ indicates the element-wise sum operator. *Repara.* denotes the reparameterization operator [66] using the mean μ_l^t and the variance σ_l^t.

Next, the disentanglement learning of the target domain is presented. Formally, in VLAE, the main task is to generate images similar to the original images. Thus, we would like to maximize:

$$logv_{\boldsymbol{\alpha}}(X^t) = \sum_{i=1}^{N_t} logv_{\boldsymbol{\alpha}}(x_i^t), \tag{12}$$

where we hope to discover a meaningful representation for the data x^t by maximizing 12. We introduce the shared encoder with ladder connections $p_{\boldsymbol{\theta}}(w^t|x^t)$ as the inference model. We can rewrite 12 as:

$$\begin{aligned} logv_{\boldsymbol{\alpha}}(x^t) &= \int d_{w^t} p_{\boldsymbol{\theta}}(w^t|x^t) logv_{\boldsymbol{\alpha}}(x^t) \\ &= \int d_{w^t} p_{\boldsymbol{\theta}}(w^t|x^t) log\left(\frac{v_{\boldsymbol{\alpha}}(w^t, x^t)}{p_{\boldsymbol{\theta}}(w^t|x^t)}\right) \\ &\quad + \int d_{w^t} p_{\boldsymbol{\theta}}(w^t|x^t) log\left(\frac{p_{\boldsymbol{\theta}}(w^t|x^t)}{v_{\boldsymbol{\alpha}}(w^t|x^t)}\right) \\ &= \int d_{w^t} p_{\boldsymbol{\theta}}(w^t|x^t) log\left(\frac{v_{\boldsymbol{\alpha}}(w^t, x^t)}{p_{\boldsymbol{\theta}}(w^t|x^t)}\right) \\ &\quad + KL[p_{\boldsymbol{\theta}}(w^t|x^t), v_{\boldsymbol{\alpha}}(w^t|x^t)], \end{aligned} \tag{13}$$

where $KL[p_{\boldsymbol{\theta}}(w^t|x^t), v_{\boldsymbol{\alpha}}(w^t|x^t)] \geqslant 0$. Thus, we have the evidence lower bound (ELBO):

$$\begin{aligned} logv_{\boldsymbol{\alpha}}(x^t) &\geqslant \int d_{w^t} p_{\boldsymbol{\theta}}(w^t|x^t) log\left(\frac{v_{\boldsymbol{\alpha}}(w^t, x^t)}{p_{\boldsymbol{\theta}}(w^t|x^t)}\right) \\ &= L_{ELBO}^{unsup}(\boldsymbol{\theta}, \boldsymbol{\phi}). \end{aligned} \tag{14}$$

The ELBO of the target domain can be further deduced as:

$$\begin{aligned} \mathcal{L}_{ELBO}^{unsup}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\alpha}) &= \int d_{w^t} p_{\boldsymbol{\theta}}(w^t|x^t) log(p_{\boldsymbol{\theta}}(w^t|x^t)) \\ &\quad + \int d_{w^t} p_{\boldsymbol{\theta}}(w^t|x^t) log\left(\frac{v_{\boldsymbol{\alpha}}(w^t)}{p_{\boldsymbol{\theta}}(w^t|x^t)}\right)\Big) \\ &= \mathbb{E}_{p_{\boldsymbol{\theta}}(w^t|x^t)}[logv_{\boldsymbol{\alpha}}(x^t|w^t)] \\ &\quad - KL[p_{\boldsymbol{\theta}}(w^t|x^t), v_{\boldsymbol{\alpha}}(w^t)]. \end{aligned} \tag{15}$$

We define $v_{\boldsymbol{\alpha}}(w^t) = v_{\boldsymbol{\alpha}}(w_1^t, \ldots, w_L^t)$ as the simple prior on all latent variables. Therefore, we choose $v_{\boldsymbol{\alpha}}(w^t)$ to be a standard Gaussian distribution $v_{\boldsymbol{\alpha}}(w^t) = \mathcal{N}(0, I)$. Then, we can obtain the final objective function of the unsupervised disentanglement

for the target domain:

$$\mathcal{L}_{unsup} = \frac{1}{N_t}\sum_{i=1}^{N_t}[-\mathbb{E}_{p_{\boldsymbol{\theta}}(w^t|x_i^t)}[log v_{\boldsymbol{\alpha}}(x_i^t|w^t)] + \beta_{unsup}\sum_{l=1}^{L} KL[p_{\boldsymbol{\theta}}(w_l^t|x_i^t), v_{\boldsymbol{\alpha}}(w_l^t)]], \quad (16)$$

where the first term is the pixel-level reconstruction loss and the second term is to make the conditional distribution of ladder latent features closer to the prior distribution for each variational layer. Thus, we call the second term the variational loss. The β_{unsup} is used to balance the losses.

C. IFDAN: Informative Feature Disentanglement Adversarial Network

In this section, we will introduce the learning of Informative Feature Disentanglement Adversarial Network (IFDAN) which combines the proposed IFD with the popular adversarial network. We apply IFD to a well-known adversarial model, e.g., Domain Adversarial Neural Network (DANN) [26] and a state-of-the-art adversarial domain adaptation model, e.g., Conditional Domain Adversarial Network (CDAN) [30], respectively. Correspondingly, we name the integrated methods as IFDAN-1 and IFDAN-2 for clarity, respectively.

1) IFDAN-1: Naturally, we embed the domain discriminator $\mathbb{D}$ with the gradient reversal layer in the IFD. The domain discriminator $\mathbb{D}$ is parameterized with $\boldsymbol{\omega}$.

2) IFDAN-2: CDAN exploits the cross-covariance between feature representations and classifier predictions to improve the discriminability. We denote the input image x from the source domain or target domain. The adversarial model conditions domain discriminator $\mathbb{D}$ on the classifier prediction $\mathbb{H}$ through the multilinear map:

$$\begin{aligned} z &= \mathbb{F}(x) \\ \tilde{y} &= \mathbb{H}(z) \\ \mathbf{T}_{\otimes}(z,\tilde{y}) &= z \otimes \tilde{y}, \end{aligned} \quad (17)$$

where $\otimes$ denotes multilinear map. $\mathbb{D}(z,\tilde{y}) = \mathbb{D}(z \otimes \tilde{y})$. For simplicity, we use the joint variable $o = (z,\tilde{y})$.

As such, the adversarial loss is defined as:

$$\mathcal{L}_{ad} = \frac{1}{N_s+N_t}\left[\sum_{i=1}^{N_s}\mathbb{E}_{x_i^s\sim D_s} log(\mathbb{D}(v_i^s)) + \sum_{i=1}^{N_t}\mathbb{E}_{x_i^t\sim D_t} log(\mathbb{D}(v_i^t))\right], \quad (18)$$

where $v_i^t = z_i^t$, $\mathcal{L}_{ad}$ points to DANN which is employed for IFDAN-1, and when $v_i^t = \mathbf{T}_{\otimes}(o_i^t)$, $\mathcal{L}_{ad}$ is for CDAN which constructs IFDAN-2.

D. IFDMN: Informative Feature Disentanglement Metric Discrepancy Network

In this section, we will introduce the learning of Informative Feature Disentanglement Metric Discrepancy Network (IFDMN) which embeds the proposed IFD into the metric discrepancy based method. We integrate IFD into a state-of-the-art domain adaptation model based metric discrepancy, e.g., Contrastive Adaptation Network (CAN) [22].

CAN aims to minimize the intra-class domain discrepancy and maximize the inter-class domain discrepancy via Maximum Mean Discrepancy (MMD) [40]. The metric discrepancy loss can be calculated:

$$\mathcal{L'}_{md} = \frac{1}{C}\sum_{c=1}^{C}\mathcal{L}^{cc} - \frac{1}{C(C-1)}\sum_{c=1}^{C}\sum_{c'=1,c'\neq c}^{C}\mathcal{L}^{cc'}, \quad (19)$$

where $\mathcal{L}^{cc}$ and $\mathcal{L}^{cc'}$ can be defined according to the following formation:

$$\begin{aligned} \mathcal{L}^{c_1c_2} &= e_1 + e_2 - 2e_3, \\ e_1 &= \sum_{i=1}^{N_s}\sum_{j=1}^{N_s}\frac{\mathbf{1}_{c_1c_1}(y_i^s,y_j^s)k(z_i^s,z_j^s)}{\sum_{i=1}^{N_s}\sum_{j=1}^{N_s}\mathbf{1}_{c_1c_1}(y_i^s,y_j^s)}, \\ e_2 &= \sum_{i=1}^{N_t}\sum_{j=1}^{N_t}\frac{\mathbf{1}_{c_2c_2}(\hat{y}_i^t,\hat{y}_j^t)k(z_i^t,z_j^t)}{\sum_{i=1}^{N_t}\sum_{j=1}^{N_t}\mathbf{1}_{c_2c_2}(\hat{y}_i^t,\hat{y}_j^t)}, \\ e_3 &= \sum_{i=1}^{N_s}\sum_{j=1}^{N_t}\frac{\mathbf{1}_{c_1c_2}(y_i^s,\hat{y}_j^t)k(z_i^s,z_j^t)}{\sum_{i=1}^{N_s}\sum_{j=1}^{N_t}\mathbf{1}_{c_1c_2}(y_i^s,\hat{y}_j^t)}, \end{aligned} \quad (20)$$

where k represents the kernel function [17]. y^s denotes the true label of the source domain, and $\hat{y}^t$ means the pseudo label of the target domain, which is predicted via the spherical K-means. The element of $\mathbf{1}_{c_1c_2}(y_i,y_j)$ is defined as: $\mathbf{1}_{c_1c_2}(y_i,y_j) = 1$, if $y_i = c_1, y_j = c_2; \mathbf{1}_{c_1c_2}(y_i,y_j) = 0$, otherwise. When $c_1 = c_2 = c$, $\mathcal{L}^{c_1c_2}$ changes to $\mathcal{L}^{cc}$, which is the intra-class domain discrepancy. When $c_1 = c, c_2 = c'$ and $c \neq c'$, $\mathcal{L}^{c_1c_2}$ becomes to $\mathcal{L}^{cc'}$, which is the inter-class domain discrepancy. In the deep neural network, CAN minimizes the discrepancy loss over multiple FC layers, i.e., minimizing the total discrepancy loss:

$$\mathcal{L}_{md} = \sum_{L'}\mathcal{L'}_{md}, \quad (21)$$

where L' denotes the total number of FC layers of the classification block.

Combine the supervised disentanglement loss, unsupervised disentanglement loss and the adaptation loss (*i.e.*, adversarial loss and metric discrepancy loss), the overall objective can be formulated as:

$$\mathcal{L} = \mathcal{L}_{sup} + \mathcal{L}_{unsup} + \beta_{ada}\mathcal{L}_{ada}, \quad (22)$$

here, when $\mathcal{L}_{ada} = \mathcal{L}_{ad}$, $\mathcal{L}$ denotes for IFDAN. When $\mathcal{L}_{ada} = \mathcal{L}_{md}$, $\mathcal{L}$ denotes for IFDMN. β_{ada} can balance the losses. The objective function is minimized to train the IFDAN or IFDMN. The transferable informative representations across domains can be learned by embedding the proposed IFD into adversarial learning modules.

By combining the supervised disentanglement of the source domain and the unsupervised disentanglement of the target domain, the shared encoder is constrained by the classification task of the source domain, so that the shared encoder can only capture the high-level semantic features. By introducing the adversarial network or metric discrepancy model, IFDAN and IFDMN

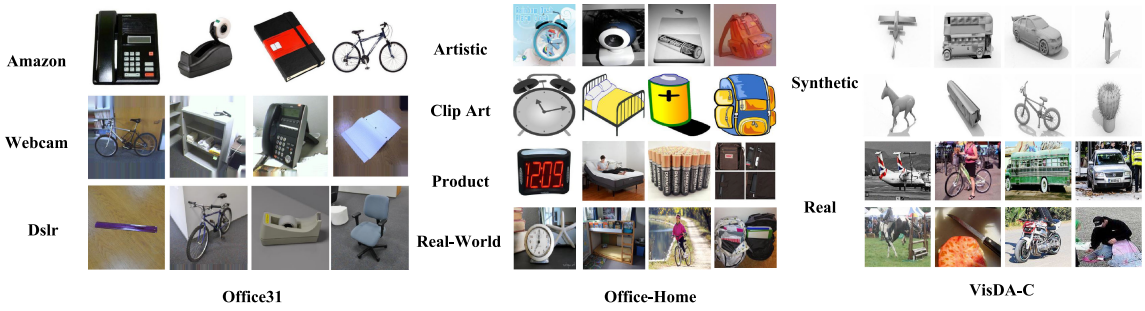


Fig. 4. Example Images from the Office31, Office-Home and VisDA-C datasets used in our experiments.

can impose additional representation structures on the high-level abstract features to achieve the adaptation of the informative features, respectively.

E. Discussion

The proposed IFD disentangles the informative features via VIB and VLAE, which is not explored by existing methods. The disentanglement is based on the hypothesis that the informative feature is a high-level semantic feature that can be shared and adapted by two domains. Thus unlike the works [62], [63], we implicitly disentangle the high-level semantic feature from task-unrelated details.

There is a claim [67] that completely unsupervised disentangling is impossible for arbitrary generative models. Inductive biases, e.g., some form of supervision or constraints on the latent space, are necessary to find ways that match the real generative model. Consequently, IFD uses the supervised disentanglement of the source domain to add constraint and guidance for the unsupervised disentanglement of the target domain.

IV. Experiments

A. Datasets

To evaluate the effectiveness of the proposed IFDAN (including IFDAN-1 and IFDAN-2) and IFDMN approaches, we conducted extensive experiments on three datasets: Office31 [68], Office-Home [69] and VisDA-C [70]. Fig. 4 gives examples of these three datasets.

Office-31 is a benchmark dataset for evaluating different DA methods for object recognition, which consists of three different domains: Amazon (A), Dslr (D), and Webcam (W), including 4652 images in 31 classes. Amazon images are collected from $amazon.com$, Webcam and Dslr images were manually gathered in an office environment. **Office-Home** is a large benchmark dataset with around 15500 images and contains images of 65 classes. The dataset contains four domains: Artistic (Ar), Clip Art (Cl), Product (Pr) and Real-World (Rw), and there are 12 DA tasks. **VisDA-C** is a very challenging dataset with the domain shift from synthetic data to real imagery. In this experiment, we validated our method on its classification task. It has two domains where the Synthetic one consists of 152397 synthetic 2D renderings of 3D objects, and the Real one consists of 55388 real images from real-world images from MS-COCO [71] dataset. The two domains have 12 classes in common.

B. Network Setting

We applied a standard DCNN trained on source domain, e.g., ResNet-50 [72] that has been pretrained on ImageNet as the encoder branch of the IFDAN and IFDMN. The final classification branch contains a linear layer after global average pooling. For the reconstruction pipeline, we applied 5 3×3 convolutional layers with feature dimensions of 2048, 1024, 512, 256, 64 in the decoder of the IFDAN and IFDMN. Each convolutional layer is followed by a Leaky-ReLU [73] nonlinearity and an upsampling layer except the layer before the output layer. For the information bottleneck layer, we apply a linear layer with the number of neurons 256 to compute the mean of latent features, and a linear layer with the number of neurons 256, followed by a Softplus to produce the variance. For the ladder variational connections, there are two convolution layers with kernel 3×3 at each connected layer to compute the mean and variance of each intermediate representation. The information bottleneck and ladder variational connections are stochastic structures in the model, and we applied the reparameterization trick introduced in [66] to back-propagate unbiased estimated gradients. For the adversarial network, we adopted a similar structure with [30], which consists of three linear layers. Each linear layer is followed by ReLU nonlinearity and Dropout layers except the final output layer.

C. Implementation Details

We implemented all experiments using Pytorch [74]. The network was trained using the mini-batch stochastic gradient descent (SGD) optimizer with a momentum of 0.9. We used a batch size of 28 samples for IFDAN, and a batch size of 30 samples for IFDMN. The learning rate annealing strategy was adopted as [22], [30]: $\eta_p = \eta_0(1 + \alpha p)^{-\beta}$, where p was the training progress changing from 0 to 1, $\alpha = 10$, $\beta = 0.75$ for Office31 and Office-Home and $\beta = 2.25$ for VisDA-C. The η_0 was the initial learning rate i.e., 1e-3 for the convolutional layers, and 1e-2 for the task-specific FC layer and the adversarial discriminator. In our model IFDAN, we set the hyper-parameters $\beta_{unsup} = 1e-4$ and $\beta_{ada} = 1$, and we empirically vary $\beta_{sup} \in [1e-4, 1e-3]$. For IFDMN, β_{unsup} is set to 1e-4, and β_{ada} is set to 0.3, and β_{sup} varies from 1e-4 to 1e-3 gradually.

D. Comparative Studies

In this section, we conducted extensive experiments to demonstrate the performance of the proposed IFDAN and IFDMN. For

TABLE I
CLASSIFICATION ACCURACIES (%) ON OFFICE31 DATASET FOR UDA. ALL MODELS UTILIZE RESNET-50 AS 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 [72]	77.9±0.1	97.4±0.2	99.5±0.0	81.7±0.1	63.3±0.2	60.6±0.3	80.1
DAN [17]	81.3±0.3	97.2±0.0	99.8±0.0	83.1±0.2	66.3±0.0	66.3±0.1	82.3
DANN [26]	81.7±0.2	98.0±0.2	99.8±0.0	83.9±0.7	66.4±0.2	66.0±0.3	82.6
JAN [18]	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
MADA [29]	90.0±0.1	97.4±0.1	99.6±0.1	87.8±0.2	70.3±0.3	66.4±0.3	85.2
VADA [75]	86.5±0.5	98.2±0.4	99.7±0.2	86.7±0.4	70.1±0.4	70.5±0.4	85.4
SimNet [76]	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
GTA [55]	89.5±0.5	97.9±0.3	99.8±0.4	87.7±0.5	72.8±0.3	71.4±0.4	86.5
MCD [44]	88.6±0.2	98.5±0.1	**100.0**±0.0	92.2±0.2	69.5±0.1	69.7±0.3	86.5
MSTN [46]	91.3	98.9	**100.0**	90.4	72.7	65.6	86.5
CDAN [30]	93.1±0.2	98.2±0.2	**100.0**±0.0	89.8±0.3	70.1±0.4	68.0±0.4	86.6
DAAA [58]	86.8±0.2	99.3±0.1	**100.0**±0.0	88.8±0.4	74.3±0.2	73.9±0.2	87.2
iCAN [77]	92.5	98.8	**100.0**	90.1	72.1	69.9	87.2
SAFN+ENT* [51]	90.3	98.7	100.0	92.1	73.4	71.2	87.6
CDAN+E [30]	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
MSTN+DSBN [65]	92.7	99.0	**100.0**	92.2	71.7	74.4	88.3
TAT [47]	92.5±0.3	99.3±0.1	**100.0**±0.0	93.2±0.2	73.1±0.3	72.1±0.3	88.4
TADA [64]	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
BSP+CDAN [78]	93.3±0.2	98.2±0.2	**100.0**±0.0	93.0±0.2	73.6±0.3	72.6±0.3	88.5
CAN [22]	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
SRDC [50]	**95.7**±0.2	99.2±0.1	**100.0**±0.0	95.8±0.2	76.7±0.3	77.1±0.1	90.8
IFDAN-1	85.3 ±0.1	97.4 ±0.2	99.8±0.0	80.1 ±0.1	69.2 ±0.1	69.4 ±0.2	83.5
IFDAN-2	**95.5** ±0.3	**99.6** ±0.2	**100.0**±0.0	94.2 ±0.3	72.5 ±0.1	74.6 ±0.2	89.4
IFDMN	95.4 ±0.1	99.3 ±0.0	**100.0**±0.0	**95.9** ±0.2	**79.3** ±0.2	**77.8** ±0.2	**91.3**

Office31 and Office-Home datasets, the results of most comparison methods are directly quoted from their original papers to facilitate fair comparison. For VisDA-C, we use the released code of DAN,[2] DANN,[3] CDAN,[4] BSP+CDAN,[5] MCD[6] and CAN[7] to conduct comparison experiments with the ResNet-50 backbone. In addition, we further choose the IFDMN to conduct experiments using ResNet-101 backbone network to more fairly compare with existing algorithms. For these three datasets, we use ResNet-50 as the backbone following the same protocol of our methods and only apply the source domain to train the network without any adaptation and process, which acts as the lower bound.

Results on Office31 Table I shows the detailed comparison results of the proposed approaches and many existing approaches in 6 transfer tasks. Table I shows that the proposed IFDMN achieves the highest accuracy on four tasks, which is superior to the recently proposed SRDC [50] by 0.5% in all. Compared to the baselines DANN and CDAN, IFDAN-1 and IFDAN-2 improve the performance by incorporating the IFD module. For the two complex tasks D → A and W → A, the improvement of the integrated method IFDMN over existing excellent methods is by 1.3% and 0.7%, respectively. The images from the background of Amazon (A) domain are relatively clear, while the images taken in real world from Webcam (W) and Dslr (D) domains posses a certain degree of complexity. When transferring from the Webcam (W) or Dslr (D) domains to Amazon (A) domain, the proposed IFDMN enforces the shared encoder to concentrate on the task-related features of complex domains via the VIB component. Besides, IFDMN reconstructs the target domain to mine the structure of the target domain, and further uses the ladder variational connections to suppress the redundant information of the target domain. In this way, the network architecture not only achieves the potential characteristics of the target domain, but also suppresses domain-specific information. From the results of these two tasks D → A and W → A, we can see the proposed IFDMN has a considerable advantage in dealing with the adaptation from a complex domain to a simple domain.

Results on Office-Home Table II lists the accuracy of the proposed approaches and existing several representative methods over 12 tasks on the Office-Home dataset. The Office-Home dataset is a very challenging dataset for DA due to the large categories in each domain. Not surprisingly, our method IFDMN outperforms the other baseline methods. The mean accuracy of the proposed IFDMN (73.7%) outperforms the state-of-the-art SRDC [50] by 2.4%. From Table II, we can see the proposed method dramatically improves the performance on most adaptation tasks, demonstrating the efficiency of the proposed IFDMN in disentangling the informative features. In addition, the integrated approach IFDAN-2 is also superior to many methods, which further verifies the efficiency of the proposed IFD. From

[2][Online]. Available: https://github.com/thuml/Xlearn
[3][Online]. Available: http://sites.skoltech.ru/compvision/projects/grl/
[4][Online]. Available: https://github.com/thuml/CDAN
[5][Online]. Available: https://github.com/thuml/Batch-Spectral-Penalization
[6][Online]. Available: https://github.com/mil-tokyo/MCD_DA
[7][Online]. Available: https://github.com/kgl-prml/Contrastive-Adaptation-Network-for-Unsupervised-Domain-Adaptation

TABLE II
CLASSIFICATION RESULTS (%) ON OFFICE-HOME DATASET FOR UDA. ALL MODELS UTILIZE RESNET-50 AS 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	Average
ResNet-50 [72]	38.9	54.6	58.4	42.4	54.2	54.5	42.4	41.6	65.3	58.9	43.0	71.9	52.2
DAN [17]	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
DANN [26]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [18]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [30]	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
DWT-MEC [79]	50.3	72.1	77.0	59.6	69.3	70.2	58.3	48.1	77.3	69.3	53.6	82.0	65.6
CDAN+E [30]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	69.3	53.6	82.0	65.8
TAT [47]	51.6	69.5	75.4	59.4	69.5	68.6	59.5	50.5	76.8	70.9	56.6	81.6	65.8
BSP+CDAN [78]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
MCS [49]	55.9	73.8	79.0	57.5	69.9	71.3	58.4	50.3	78.2	65.9	53.2	82.2	66.3
TADA [64]	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
SAFN* [51]	54.4	73.3	77.9	65.2	71.5	73.2	63.6	52.6	78.2	72.3	58.0	82.1	68.5
CAN [22]	60.3	77.2	79.4	69.2	74.9	72.9	68.3	58.4	79.4	74.8	57.5	82.2	71.2
SRDC [50]	52.3	76.3	81.0	69.5	76.2	**78.0**	68.7	53.8	81.7	**76.3**	57.1	85.0	71.3
IFDAN-1	46.3 ±0.1	59.8 ±0.1	71.2±0.2	50.2 ±0.1	61.9 ±0.1	61.4 ±0.2	47.8±0.2	44.3 ±0.3	69.7 ±0.4	64.2±0.2	52.7 ±0.1	77.3 ±0.1	58.9
IFDAN-2	53.4±0.2	71.8±0.3	77.6±0.1	59.8±0.3	71.6±0.2	72.7±0.4	60.9±0.2	52.8 ±0.1	78.7±0.2	74.3±0.1	60.3±0.2	83.2±0.3	68.1
IFDMN	**61.2** ±0.2	**80.4** ±0.1	**82.7**±0.1	**69.8** ±0.1	**76.5** ±0.2	**78.0** ±0.2	**69.2** ±0.5	**59.2** ±0.1	**84.1** ±0.2	75.3±0.1	**61.9** ±0.2	**86.2** ±0.3	**73.7**

TABLE III
CLASSIFICATION ACCURACIES (%) ON THE VISDA-C DATASET FOR UDA. ALL MODELS UTILIZE RESNET-50 AS THE BASE ARCHITECTURE EXCEPT THE METHODS MARKED WITH *, WHICH USES RESNET-101. THE BOLD NUMBERS DENOTE THE BEST RESULT

Methods	ResNet-50 [72]	DAN [17]	DANN [26]	JAN [18]	CDAN [30]	BSP+CDAN [78]	MCD [44]	GTA [55]	CAN [22]	BSP+CDAN* [78]	MCD* [44]	CAN* [22]	IFDAN-1	IFDAN-2	IFDMN	IFDMN*
Average	58.6	63.1	63.7	64.8	66.8	68.2	69.2	69.5	72.7	74.1	73.0	78.1	65.4±0.3	69.9±0.2	74.1±0.3	**79.2**±0.3

the results of IFDAN-1, IFDAN-2 and IFDMN, we can see the proposed module can be applied to many current DA algorithms successfully and dramatically improves performance.

Results on VisDA-C Table III presents the accuracy over 12 classes on VisDA-C with the validation set as the target domain. Slightly superior results are achieved by our algorithm IFDMN. We can see that our proposed methods yield increasingly good results compared with their corresponding baseline models e.g., DANN [26], CDAN [30] and CAN [22], verifying the efficiency of the proposed IFD. Notably, the performance of IFDMN outperforms BSP+CDAN [78] and MCD [44] which use the ResNet-101 as the backbone verifying the effectiveness of the proposed IFD.

E. Ablation Studies

To demonstrate the efficiency of the supervised disentanglement and unsupervised disentanglement, we choose IFDAN-2 and IFDMN to conduct a series of ablation experiments on Office31, Office-Home and Visda-C. For IFDAN-2 and IFDMN, the proposed IFD is embedded in CDAN [30] and CAN [22], respectively; thus we regard the results of CDAN and CAN as our benchmark where there is no disentanglement. In addition, we also explored the component IFD where the features are disentangled but not adapted. Importantly, we explored two other main factors: IFDAN-2 (w/o sup) and IFDMN (w/o sup) representing the lack of the supervised disentanglement, IFDAN-2 (w/o unsup) and IFDMN (w/o unsup) denoting that the unsupervised disentanglement is not considered.

Table IV shows the experimental results on the above two datasets. IFD can not achieve better performance, because it is devoted to extracting informative features without concentrating on the features adaptation. It is worth noting that the experimental methods with the only supervised disentanglement, e.g., IFDAN-2 (w/o unsup) and IFDMN (w/o unsup), or with only unsupervised disentanglement, e.g., IFDAN-2 (w/o sup) and IFDMN (w/o sup), achieve some improvements in the performance, which demonstrates the importance of the informative feature disentanglement. Combining the two types of disentanglement, IFDAN-2 and IFDMN can further improve the performance.

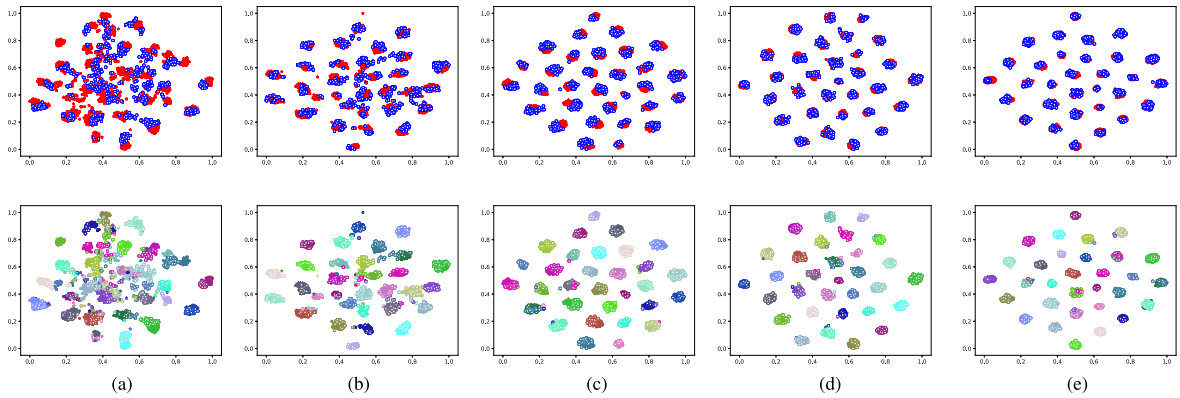
F. Further Remarks

Feature Visualization A popular method to visualize high-dimensional data in 2D is t-SNE [80]. We are interested in the distribution of embeddings for target domain when we employ our training scheme. We chose the A → W DA task of Office31 dataset and plotted it in Fig. 5 to visualize the learned feature representations. The first row shows the features of domain-level distributions. The second row represents the features of class-level distributions. Fig. 5(a) depicts the learned feature of source and target domains by ResNet-50. We can see that the samples of source and target are rarely aligned. Fig. 5(b) shows the visualizations of features learned by CDAN. Some of the same classes can gather together, but the feature distribution is still discrete. The feature representations of IFDAN-2 (w/o sup) and IFDAN-2 (w/o unsup) are described in Fig. 5(c) and (d). We can see that the clusters become compact, but some categories have been mixed up in the feature space. The feature representation of IFDAN-2 is shown in Fig. 5(e). We can see the domain distributions align well from the first row. Moreover, the features of the two domains can be discriminated very well from the second row. By contrast, combining supervised disentanglement and unsupervised disentanglement can learn more informative transferable representations.

TABLE IV
CLASSIFICATION ACCURACIES (%) ON OFFICE31 AND OFFICE-HOME DATASETS FOR UDA. ALL MODELS UTILIZE RESNET-50 AS BASE ARCHITECTURE. THE BOLD NUMBERS DENOTE THE BEST RESULTS FOR EACH COLUMN IN EACH BOX

	A→W	D→W	W→D	A→D	D→A	W→A	Average
IFD	74.1±0.1	98.4±0.2	**100.0**±0.0	77.9±0.1	59.6±0.2	63.5±0.1	78.9
CDAN [30]	93.1±0.2	98.2±0.2	**100.0**±0.0	89.8±0.3	70.1±0.4	68.0±0.4	86.6
IFDAN-2 (w/o sup)	94.7 ±0.2	98.6±0.1	**100.0**±0.0	91.2±0.1	72.0±0.2	72.2±0.2	88.1
IFDAN-2 (w/o unsup)	95.0±0.1	99.1 ±0.1	**100.0**±0.0	93.5±0.1	71.9 ±0.2	72.3 ±0.1	88.6
IFDAN-2	**95.5** ±0.3	**99.6** ±0.2	**100.0**±0.0	**94.2** ±0.3	**72.5** ±0.1	**74.6** ±0.2	**89.4**
CAN [22]	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
IFDMN (w/o sup)	95.1 ±0.2	**99.3** ±0.0	100.0±0.0	95.4 ±0.2	78.4 ±0.1	77.4 ±0.1	90.9
IFDMN (w/o unsup)	94.8 ±0.1	99.1 ±0.1	100.0±0.0	95.8 ±0.3	78.3 ±0.2	77.3 ±0.1	90.9
IFDMN	**95.4** ±0.1	**99.3** ±0.0	**100.0**±0.0	**95.9** ±0.2	**79.3** ±0.2	**77.8** ±0.2	**91.3**

	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Average
IFD	41.8	62.3	70.7	49.6	58.6	61.3	47.5	37.6	68.9	63.2	44.6	76.1	56.9
CDAN [30]	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
IFDAN-2 (w/o sup)	51.2	70.4	75.2	57.6	68.8	69.2	59.6	52.8	77.8	73.5	58.1	82.6	66.4
IFDAN-2 (w/o unsup)	52.3	70.6	76.1	58.6	70.1	70.8	58.1	50.6	77.2	71.5	58.3	81.5	66.3
IFDAN-2	**53.4**	**71.8**	**77.6**	**59.8**	**71.6**	**72.7**	**60.9**	**52.8**	**78.7**	**74.3**	**60.3**	**83.2**	**68.1**
CAN [22]	60.3	77.2	79.4	69.2	74.9	72.9	68.3	58.4	79.4	74.8	57.5	82.2	71.2
IFDMN (w/o sup)	60.8	79.3	81.6	69.4	75.9	77.1	68.4	58.6	83.2	75.0	59.9	85.8	72.9
IFDMN (w/o unsup)	60.4	79.8	82.0	69.6	76.3	77.8	68.6	58.9	83.4	75.1	61.4	85.4	73.2
IFDMN	**61.2**	**80.4**	**82.7**	**69.8**	**76.5**	**78.0**	**69.2**	**59.2**	**84.1**	**75.3**	**61.9**	**86.2**	**73.7**

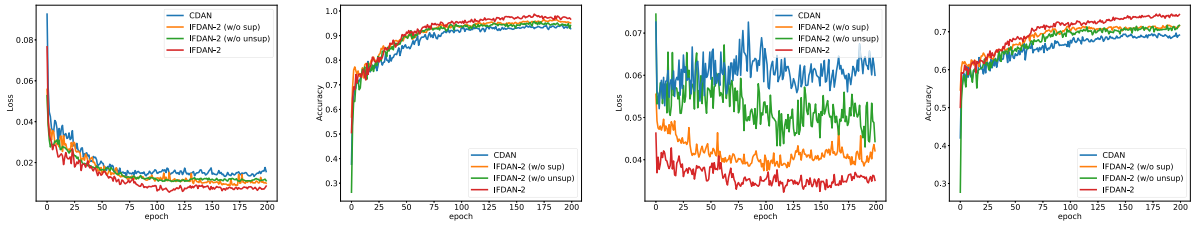


(a) (b) (c) (d) (e)

Fig. 5. (a)-(e) The t-SNE visualization of network activations on the source and target domains generated by ResNet-50, CDAN, IFDAN-2 (w/o sup), IFDAN-2 (w/o unsup) and IFDAN-2, respectively. Different shapes denote different domains. 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. (a) ResNet-50. (b) CDAN. (c) IFDAN-2 (w/o sup). (d) IFDAN-2 (w/o unsup). (e) IFDAN-2.

Convergence We verified the performance of the experimental method IFDAN-2 on the convergence from the testing loss and accuracy as illustrated in Fig 6 (a–d). From the loss curve, we can see that our proposed model IFDAN-2 converges faster than the baseline method. As the training proceeds, the loss of IFDAN-2 has always been at a lower level. From the Fig. 6 (c), there is an interesting scenario that the convergence speed of CDAN is not stable. However, when CDAN is integrated with IFD, i.e., IFDAN-2, converges stably and fast. Furthermore, from the accuracy curve illustrated in Figure 6, we see that disentangling informative features leads to notable accuracy improvement of IFDAN-2, compared to CDAN.

Reconstruction analysis To further analyze the disentanglement structure, we reconstructed the source and target domains with the trained model IFDAN-2 as shown in Figure 7. Figures 7 (a) and (b) denote images of the source domain (Amazon) and the target domain (Webcam), respectively. Figure 7 (c) and (d) show the reconstruction result of source and target domains, respectively. We can see some objects of images can be reconstructed, but some details (e.g., background) are lost, which verifies the disentanglement of informative features. The reconstruction results show that IFDAN-2 is at the expense of reconstructing complete images to disentangle the informative transferable features in the learning process.



(a) (b) (c) (d)

Fig. 6. (a)-(b) The curve of loss and accuracy during training on the task of A → W. (c)-(d) The curve of loss and accuracy during training on the task of W → A.

(a) (b) (c) (d)

Fig. 7. Data reconstruction after training from Amazon to Webcam via IFDAN. (a)-(b) samples from source (Amazon) and target (Webcam), (c) the reconstruction images of source domain (Amazon), (d) the reconstruction images of target domain (Webcam).

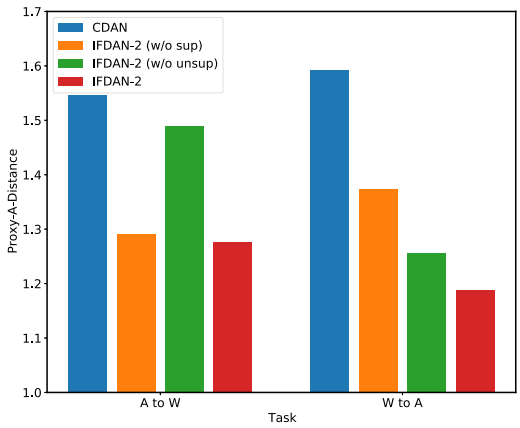


Fig. 8. Empirical analysis: Proxy $\mathcal{A}$-Distance of different features on A → W and W → A.

Discrepancy Distance The theory of DA [81], [12] 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 to estimate the proxy $\mathcal{A}$-distance (PAD) is defined as $\hat{d}_{\mathcal{A}} = 2(1 - 2\epsilon)$, where ϵ is the generalization error of a binary classifier of discriminating source and target. We applied a kernel SVM to estimate the $\mathcal{A}$-distance. Figure 8 shows PADs on tasks A → W and W → A with features of CDAN, IFDAN-2 (w/o sup), IFDAN-2 (w/o unsup) and IFDAN-2. We observe that PAD of IFDAN-2 is much smaller than the baseline method, which suggests that our features can reduce the cross-domain gap more effectively.

V. Conclusion

In this paper, we propose a novel module named Informative Feature Disentanglement (IFD) aiming at only applying the high-level semantic features to adapt and filtering out the redundant information. Technically, we conduct the supervised disentanglement of the source domain and the unsupervised disentanglement of the target domain via VIB and VLAE, respectively. We combine the proposed IFD with the adversarial networks and a metric discrepancy network, respectively; thus the Informative Feature Disentanglement Adversarial Network (IFDAN) and Informative Feature Disentanglement Metric Discrepancy Network (IFDMN) emerge as required. By conducting an informative features purification before the features adaptation, IFDAN and IFDMN ease the following features alignment, thus improving the adaptation and accelerates convergence. Comparative experiments on three gold DA datasets demonstrate that IFDMN yields leading result compared with many existing methods in classification accuracy. Besides, extensive ablation experiments clearly show the effectiveness of IFD in significantly improving the performance of the popular domain adaptation network.

Despite the effectiveness of the IFD, there are still some aspects that can be further improved. The proposed IFD aims at solving the closed-set UDA problem, i.e., the source domain and target domain share the identical label space, where there is only one feature-level difference between domains. Recently, partial-set [82]–[84], open-set [85]–[87] and universal-set [88] domain adaptation problems which are more inclined to real-world applications are put forward, where there are not only differences in the feature level, but also variances in the category level, e.g., target labels are only a subset of source labels in partial-set scenarios. IFD is designed to disentangle the feature-level divergence for closed-set UDA without considering the category-level difference of several extended UDA settings. A better understanding of informative category information disentanglement would play a great role in this field of research. Thus, we will extend the IFD to category information disentanglement for open-set, partial-set and universal-set domain adaptation problems in our future work.

REFERENCES

[1] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252.

[2] L. Liu *et al.*, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, 2020.

[3] S. Minaee *et al.*, "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021, *arXiv:1702.05374*.

[4] A. Garcia-Garcia *et al.*, "A survey on deep learning techniques for image and video semantic segmentation," *Appl. Soft Comput.*, vol. 70, pp. 41–65, 2018.

[5] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.

[6] L. Zhang *et al.*, "Manifold criterion guided transfer learning via intermediate domain generation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3759–3773, Dec. 2019.

[7] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, "A deeper look at dataset bias," in *Proc. GCPR*, vol. 9358, pp. 504–516, 2015.

[8] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[9] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[10] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. Comput. Vis. Pattern Recognit.*, 2011, pp. 1521–1528.

[11] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," *in Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 137–144.

[12] S. Ben-David *et al.*, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1-2, pp. 151–175, 2010.

[13] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[14] G. Csurka, "A comprehensive survey on domain adaptation for visual applications," in *Domain Adapt. Computer Vision Appl.*, ser. Advances in Computer Vision and Pattern Recognition. Springer, pp. 1–35, 2017, *arXiv:1702.05374*.

[15] W. M. Kouw and M. Loog, "A review of domain adaptation without target labels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 766–785, Mar. 2021.

[16] Z. Lei, "Transfer adaptation learning: A decade survey," 2019, *arXiv:1903.04687*.

[17] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[18] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.. JMLR. org*, 2017, pp. 2208–2217.

[19] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3071–3085, Dec. 2019.

[20] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[21] Y. Hongliang *et al.*, "Mind the class weight bias: weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2272–2281.

[22] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4888–4897.

[23] H. Yan *et al.*, "Weighted and class-specific maximum mean discrepancy for unsupervised domain adaptation," *IEEE Trans. Multimedia*, vol. 22, no. 9, pp. 2420–2433, Sep. 2020.

[24] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[25] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7167–7176.

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

[27] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4058–4065.

[28] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4068–4076.

[29] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3934–3941.

[30] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1647–1657.

[31] X. Ma, T. Zhang, and C. Xu, "Deep multi-modality adversarial networks for unsupervised domain adaptation," *IEEE Trans. Multimedia*, vol. 21, no. 9, pp. 2419–2431, Sep. 2019.

[32] G. Ian *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, p. 2672–2680.

[33] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. Allerton*, 2000, pp. 368–377.

[34] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. ICLR*, 2017, pp. 1–19.

[35] S. Zhao, J. Song, and S. Ermon, "Learning hierarchical features from generative models," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 4091–4099.

[36] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *Proc. Pacific Rim Int. Conf. Artif. Intell.*, 2014, pp. 898–904.

[37] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*.

[38] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2058–2065.

[39] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (CMD) for domain-invariant representation learning," in *Proc. ICLR*, 2017, pp. 1–13.

[40] G. Arthur, B. Karsten, R. Malte, S. Bernhard, and S. Alex, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, 2012.

[41] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 443–450.

[42] Y. Zhang, H. Tang, K. Jia, and M. Tan, "Domain-symmetric networks for adversarial domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5031–5040.

[43] C. Qingchao, L. Yang, W. Zhaowen, W. Ian, and C. Kevin, "Re-weighted adversarial adaptation network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7976–7985.

[44] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3723–3732.

[45] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 285–10 295.

[46] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5423–5432.

[47] H. Liu, M. Long, J. Wang, and M. Jordan, "Transferable adversarial training: A general approach to adapting deep classifiers," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4013–4022.

[48] S. Wang and L. Zhang, "Self-adaptive re-weighted adversarial domain adaptation," in *Proc. IJCAI*, 2020, pp. 3181–3187.

[49] J. Liang, R. He, Z. Sun, and T. Tan, "Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2975–2984.

[50] H. Tang, K. Chen, and K. Jia, "Unsupervised domain adaptation via structurally regularized deep clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8722–8732.

[51] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1426–1435.

[52] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3722–3731.

[53] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 469–477.

[54] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," in *Proc. ICLR*, 2016, pp. 1–14.

[55] S. Sankaranarayanan, Y. Balaji, C. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8503–8512.

[56] J. Hoffman *et al.*, "Cycada: Cycle-consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1989–1998.
[57] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 700–708.
[58] G. Kang, L. Zheng, Y. Yan, and Y. Yang, "Deep adversarial attention alignment for unsupervised domain adaptation: The benefit of target expectation maximization," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 401–416.
[59] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo, "From source to target and back: Symmetric bi-directional adaptive gan," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8099–8108.
[60] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4500–4509.
[61] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.
[62] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 343–351.
[63] Y. Liu, X. Tian, Y. Li, Z. Xiong, and F. Wu, "Compact feature learning for multi-domain image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7193–7201.
[64] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, "Transferable attention for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5345–5352.
[65] W. G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7354–7362.
[66] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. ICLR*, 2014, pp. 1–14.
[67] F. Locatello *et al.*, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4114–4124.
[68] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.*. Springer, 2010, pp. 213–226.
[69] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5018–5027.
[70] X. Peng *et al.*, "Visda: The visual domain adaptation challenge," 2017, *arXiv:1710.06924*.
[71] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*. Springer, 2014, pp. 740–755.
[72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[73] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn. Workshop Deep Learn. Audio, Speech Lang. Process.*, 2013, pp. 1–6.
[74] P. Adam *et al.*, "Automatic differentiation in pytorch," in *Proc. NeurIPS Workshop J. Mach. Learn. Res.*, 2017, pp 1–4.
[75] R. Shu, H. H. Bui, H. Narui, and S. Ermon, "A DIRT-T approach to unsupervised domain adaptation," in *Proc. ICLR*, 2018, pp. 1–19.
[76] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, p. 8004–8013.
[77] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3801–3809.
[78] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in Proc. Int. Conf. Mach. Learn., 2019, pp. 1081–1090.
[79] S. Roy *et al.*, "Unsupervised domain adaptation using feature-whitening and consensus loss," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9471–9480.
[80] V. D. M. Laurens and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
[81] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," in *Proc. COLT*, 2009, pp. 1–11.
[82] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, vol. 11212, 2018, pp. 139–155.
[83] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2985–2994.
[84] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8156–8164.
[85] P. P. Busto and J. Gall, "Open set domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 754–763.
[86] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada, "Open set domain adaptation by backpropagation," in *Proc. Eur. Conf. Comput. Vis.*, vol. 11209, 2018, pp. 156–171.
[87] M. Baktashmotlagh, M. Faraki, T. Drummond, and M. Salzmann, "Learning factorized representations for open-set domain adaptation," in *Proc. ICLR*, 2019, pp. 1–11.
[88] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Universal domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2720–2729.

Wanxia Deng received the university B.E. degree, in 2014 in electronic information science and technology from Xiamen University, Xiamen, China, and the M.S. degree, in 2016 in information and communication engineering from the National University of Defense Technology, Changsha, China, where she is currently working toward the Ph.D. degree in information and communication engineering. Her research interests include domain adaptation, transfer learning, deep learning, and image processing.

Lingjun Zhao received the B.S., M.S., and Ph.D. degrees in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2003, 2004, and 2009, respectively. She is currently an Associate Professor with the School of Electronic Science, National University of Defense Technology. Her research interests include remote sensing information processing, SAR automatic target recognition, and deep learning.

Qing Liao received the B.Sc. degree in software technology and application from the Macau University of Science and Technology, Taipa, Macau, in 2010, the M.Phil. degree in computer science and technology from the Fok Ying Tung Graduate School, Hong Kong University of Science and Technology, Hong Kong, in 2013, and the Ph.D. degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, in 2016. She is currently an Assistant Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. Her research interests include artificial intelligence and bioinformatics.

Deke Guo received the B.S. degree in industry engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001, and the Ph.D. degree in management science and engineering from the National University of Defense Technology, Changsha, China, in 2008. He is currently a Professor with the College of System Engineering, National University of Defense Technology, and is also with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests include distributed systems, software-defined networking, data center networking, wireless and mobile systems, and interconnection networks.

Gangyao Kuang received the B.S. and M.S. degrees from the Central South University of Technology, Changsha, China, in 1988 and 1991, respectively, and the Ph.D. degree from the National University of Defense Technology, Changsha, China, in 1995. He is currently a Professor at School of Electronic Science, National University of Defense Technology. From 2009 to 2010, he was a Visiting Scholar with the University of Waterloo, Waterloo, ON, Canada. His current interests include remote sensing, SAR image processing, change detection, SAR ground moving target indication, and the classification of polarimetric SAR images.

Dewen Hu received the B.S. and M.S. degrees from Xi'an Jiaotong University, Xi'an, China, in 1983 and 1986, respectively, and the Ph.D. degree from the National University of Defense Technology, Changsha, China, in 1999. He is currently a Professor with the School of Intelligent Science, National University of Defense Technology. From 1995 to 1996, he was a Visiting Scholar with The University of Sheffield, Sheffield, U.K. His research interests include image processing, system identification and control, neural networks, and cognitive science.

Matti Pietikäinen received the Doctor of Science degree in technology from the University of Oulu, Oulu, Finland. He is currently an emeritus Professor with the Center for Machine Vision and Signal Analysis, University of Oulu. From 1980 to 1981 and from 1984 to 1985, he visited the Computer Vision Laboratory, University of Maryland, College Park, MD, USA. He has made fundamental contributions to local binary pattern methodology, texture based image and video analysis, and facial image analysis. He has authored more than 350 refereed papers in international journals, books, and conferences. His papers have about 57000 citations in Google Scholar (hindex 82). He was an Associate Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Pattern Recognition*, IEEE TRANSACTIONS ON FORENSICS AND SECURITY, and *Image and Vision Computing journals*. He is currently an Associate Editor for the IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR AND IDENTITY SCIENCE, and the Guest Editor of special issues of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and *International Journal of Computer Vision*. From 1989 to 1992, he was the President of the Pattern Recognition Society of Finland and in 2014, was named its honorary member. From 1989 to 2007, he was a Member of the Governing Board of International Association for Pattern Recognition (IAPR), and in 1994, he became one of the founding fellows of the IAPR. In 2014, his research on LBP-based face description was awarded the Koenderink Prize for fundamental contributions in computer vision. He was the recipient of the IAPR King-Sun Fu Prize 2018 for fundamental contributions to texture analysis and facial image analysis. In 2018, he was named a highly cited Researcher by Clarivate Analytics, by producing multiple highly cited papers in 2006 and 2016 that rank in the top 1% by citation for his field in web of science. He is a Fellow of the IEEE for contributions to texture and facial image analysis for machine vision.

Li Liu received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2012. She is currently a Professor with the College of System Engineering. During the Ph.D. degree, she was a Visiting Student with the University of Waterloo, Waterloo, ON, Canada, from 2008 to 2010. From 2015 to 2016, she was visiting the Multimedia Laboratory, Chinese University of Hong Kong, Hong Kong, for ten months. From December 2016 to November 2018, she was a Senior Researcher with Machine Vision Group, University of Oulu, Oulu, Finland. Her papers have currently more than 3500 citations in Google Scholar. Her current research interests include computer vision, pattern recognition, and machine learning.