TransVQA: Transferable Vector Quantization Alignment for Unsupervised Domain Adaptation

Yulin Sun, *Member, IEEE*, Weisheng Dong, *Member, IEEE*, Xin Li, *Fellow, IEEE*, Le Dong, *Member, IEEE*, Guangming Shi, *Fellow, IEEE*, and Xuemei Xie, *Senior Member, IEEE*

***Abstract*— Unsupervised Domain adaptation (UDA) aims to transfer knowledge from the labeled source domain to the unlabeled target domain. Most existing domain adaptation methods are based on convolutional neural networks (CNNs) to learn cross-domain invariant features. Inspired by the success of transformer architectures and their superiority to CNNs, we propose to combine the transformer with UDA to improve their generalization properties. In this paper, we present a novel model named *Trans*ferable *V*ector *Q*uantization *A*lignment for Unsupervised Domain Adaptation (*TransVQA*), which integrates the Transferable transformer-based feature extractor (Trans), vector quantization domain alignment (VQA), and mutual information weighted maximization confusion matrix (MIMC) of intra-class discrimination into a unified domain adaptation framework. First, TransVQA uses the transformer to extract more accurate features in different domains for classification. Second, TransVQA, based on the vector quantization alignment module, uses a two-step alignment method to align the extracted cross-domain features and solve the domain shift problem. The two-step alignment includes global alignment via vector quantization and intra-class local alignment via pseudo-labels. Third, for intra-class feature discrimination problem caused by the fuzzy alignment of different domains, we use the MIMC module to constrain the target domain output and increase the accuracy of pseudo-labels. The experiments on several datasets of domain adaptation show that TransVQA can achieve excellent performance and outperform existing state-of-the-art methods.**

***Index Terms*— Transferable transformer feature extractor, vector quantization alignment, two-step alignment, mutual information weighted, unsupervised domain adaptation.**

I. Introduction

THE convolutional neural networks (CNNs) [1], [2], [3] and transformer architectures [4], [5] have achieved high accuracy in supervised image classification. However, collecting label samples on a large scale is often expensive [6], [7]. Meanwhile, the effects of light, color, background, location, and style can lead to domain shift problems, affecting the generalization property of deep models [8], [9] and resulting in decreased accuracy [6], [8]. To solve the above problem, Unsupervised Domain Adaptation (UDA) methods are proposed to move the knowledge learned from the labeled source domain to the unlabeled target domain [10], [11], [12], [13], [14] based on the assumption that both domains share the same label set [15].

Existing UDA methods can be roughly classified into aligned discrepancy-based methods [12], [16], [17], [18] and adversarial learning-based methods [11], [19], [20]. The former (discrepancy-based methods) seek to address the domain shift issue by carefully designing the alignment loss - e.g., the loss of Correlation Alignment (CORAL) [21], the maximum mean discrepancy (MMD) [16], local maximum mean discrepancy (LMMD) [22], etc. The latter (adversarial learning-based methods) use a domain discriminator to distinguish features in the source and target domains. With a deep network trained to extract features that the domain discriminator cannot recognize [14], [19], the domain shift problem is solved under the adversarial learning framework. In addition, some UDA methods extract useful information from the unlabeled target domain, such as Minimum Class Confusion (MCC) [15], which uses hidden knowledge to minimize the confusion matrix of the target domain; Bi-Classifier Determinacy Maximization (BCDM) [23] exploits the dark knowledge of bi-classifiers for domain adaptation.

Regardless of the type of UDA methods, feature extraction is at the foundation of these methods. Existing feature extraction modules for UDA can be divided into CNN-based and transformer-based. CNN-based methods usually use pre-trained ResNet [2] networks as feature extractors [15], [18], [19], [23], [24], [25]. Transformer-based methods usually use pre-trained Vision Transformer (ViT) [4] and Swin transformer (Swin) [5] networks as feature extractors [26], [27], [28]. Transformer-based on the self-attention mechanism can build long-range dependencies between visual features in the image. Thus, Transformers can often perform better than CNNs on domain-adaptation tasks [28]. Most domain adaptive methods use CNNs to learn cross-domain invariant features, even though transformer architectures have shown better performance [4], [5].

Several transformer-based UDA methods have been proposed in the open literature in recent years [12], [15], [18], [27], [28], [29]. However, they suffer from the following limitations. First, most are multi-branch structures, requiring more training resources [27], [28] than the single-branch

Manuscript received 6 July 2023; revised 21 November 2023; accepted 30 December 2023. Date of publication 17 January 2024; date of current version 19 January 2024. This work was supported in part by the Natural Science Foundation of China under Grant 61991451 and Grant 62302369 and in part by the China Postdoctoral Science Foundation under Grant 2023M742737. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ran He. *(Corresponding author: Guangming Shi.)*

Yulin Sun, Weisheng Dong, Le Dong, Guangming Shi, and Xuemei Xie are with the School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: daitusun@gmail.com; wsdong@mail.xidian.edu.cn; dongle@xidian.edu.cn; gmshi@xidian.edu.cn; xmxie@mail.xidian.edu.cn).

Xin Li is with the Department of Computer Science, University at Albany, Albany, NY 12222 USA (e-mail: xli48@albany.edu).

Digital Object Identifier 10.1109/TIP.2024.3352392

counterpart. Second, only the global distribution alignment of cross-domain features is considered, while the benefit of local alignment is ignored [12], [18]. Last but not least, the method of using target domain pseudo-labeling usually introduces more confusing information by ignoring the post-processing of the target domain [15], [29]. In summary, the overall performance of these existing transformer-based UDA methods has not been optimized.

To solve the above problems, we propose a method named **Trans**ferable **V**ector **Q**uantization **A**lignment for Unsupervised Domain Adaptation (**TransVQA**) in this work. First, Our TransVQA method uses the visual transformer for cross-domain feature extraction with the self-attention mechanism. Moreover, our transformer is easier to train without the multi-branch structure. Second, our method introduces a latent space (codebook) that provides a priori information through vector quantization (VQ) and uses this codebook for two-step cross-domain feature alignment based on the MSE metric. In the first step, we use the global alignment in the latent space, which can remove the feature distribution gaps across different domains. In the second step, we propose a local alignment strategy using codebook and target domain pseudo-labels, which removes the domain shift within the same class. Third, to solve the feature confusion problem caused by incorrect pseudo-labels, we proposed a Mutual Information-weighted Maximization Confusion (MIMC) matrix module to enhance the pseudo-label confidence of the target domain. Mutual information weighting can bring the pseudo-labels in the target domain closer to the one-hot code. Our contributions are summarized as follows:

- A novel Transferable Vector Quantization Alignment for Unsupervised Domain Adaptation (TransVQA) method was proposed. Our TransVQA method learns deep transformer features and introduces a latent space (codebook) to provide a priori information for two-step domain feature alignment based on MSE metrics.
- The TransVQA method performs global and local features alignment of the cross-domain by a two-step alignment method based on vector quantization and transformer feature. Thus, we can obtain more discriminative features and remove the cross-domain gap.
- The TransVQA method uses MIMC matrix loss in the target domain to enhance the intra-class discriminability of alignment features and pseudo-label confidence. Mutual information weight makes the classifier output closer to one-hot encoding, improving the discriminative performance.
- We experiment on DomainNet, VisDA-2017, Office-Home, and Office-31 datasets, and show that Our TransVQA method outperforms the state-of-the-art methods in different UDA scenarios.

II. Related Work

A. Domain Adaptation

In the literature, many UDA methods have been proposed. The DANN [11] and CDAN [19] propose domain discriminators for networks that use domain discriminators to discriminate source and target samples. Maximum classifier discrepancy (MCD) [13] employs adversarial between feature extractors and classifiers. Collaborative and adversarial network (CAN) [30] uses pseudo-labels directly as regularization. Semantic concentration for domain adaptation (SCDA) [14] based on pseudo-labels encourages models to focus on the most dominant features by predicting pairwise adversarial alignments of distributions. Deep adaptation network (DAN) [16] minimizes the maximum mean discrepancy (MMD) in selected layers to reduce cross-domain gaps. Joint adaptation networks (JAN) [12] uses the joint MMD and pseudo-labels to enhance the alignment of the joint distribution across domains. Margin disparity discrepancy (MDD) [18] introduced margin disparity discrepancy that reduces distributional discrepancy by strict generalization bounds. Source HypOthesis Transfer (SHOT) [31] learns the features of the target domain by fitting the frozen source classification module. FixBi [32] adds multiple intermediate domains between the source and target domains by introducing a fixed ratio-based mixup.

B. Transformer for Vision

The Transformer [33] is proposed for the first time in the NLP application. Vision transformer (ViT) [4] first applied Transformer to image classification. Then, several variants of ViT [5], [34] were proposed and achieved comparable performance over CNN on computer vision tasks. Swin Transformer [5] proposes to use a shift window to compute the feature representation. Based on the excellent performance of the transformer in vision tasks, some vision transformer-based UDA schemes have been presented. Transferable vision transformer (TVT) [26] is based on the ViT network, which captures transferable and discriminative features via transferable self-attention blocks. Cross-domain transformer (CDtrans) [27] combines pseudo-labels to propose a weight-shared three-branch transformer framework for aligning cross-attention across domains. Bidirectional cross-attention transformer (BCAT) [28] is a weight-shared quadruple-branch transformer network. It uses the mechanism of bidirectional cross-attention to learn domain-invariant representations. Domain-oriented transformer (DOT) [35] proposed two individual classifiers based on ViT to maintain domain-wise discriminability.

C. Vector Quantization

The Variational Autoencoder (VAE) is an unsupervised learning method that contains an encoder and a decoder [36]. The encoder projects a high-dimensional input x into a low-dimensional latent variable z. The decoder reconstructs z to x. Many training discrete VAE methods have been proposed [37], [38]. Vector Quantized-Variational AutoEncoder (VQ-VAE) [37] integrates the VAE scheme with discrete latent representations via a codebook of a given prior distribution. Li et al. [39] proposed VAE, adversarial learning, and pseudo-label are applied to domain adaptation tasks. JVA^{2}E [40] proposed a Joint Adversarial Variational AutoEncoder for unsupervised domain adaptation tasks. In contrast, our model introduces vector quantization alignment techniques to domain

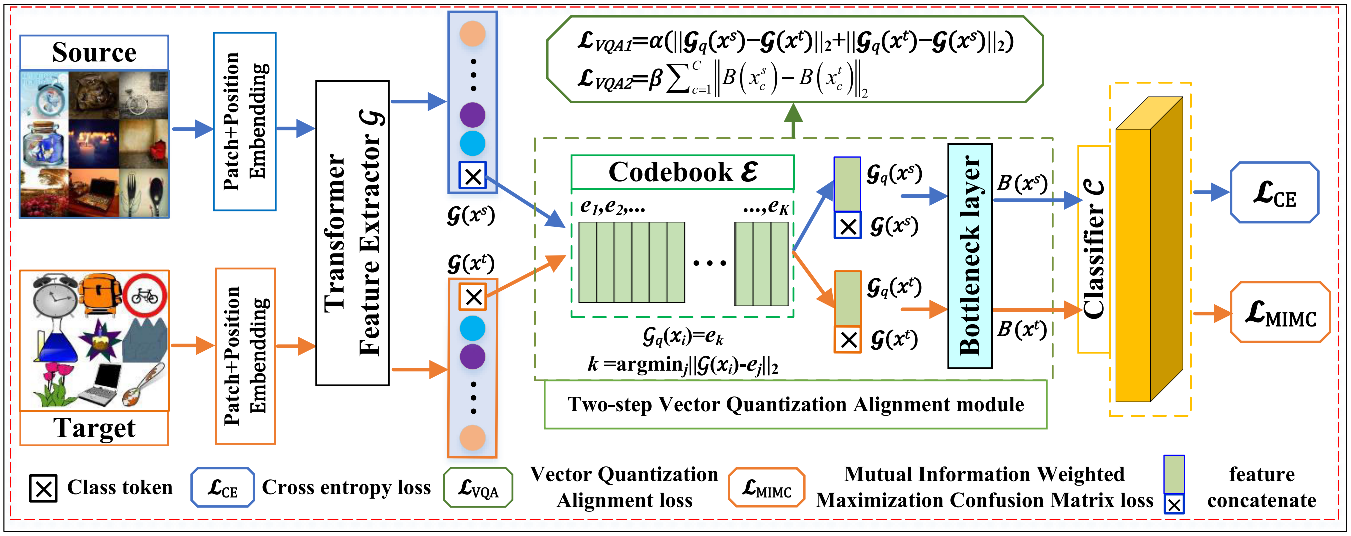

Source
Patch+Position Embendding
Transformer Feature Extractor $\mathcal{G}$
$\mathcal{G}(x^s)$
$\mathcal{G}(x^t)$
Target
Patch+Position Embendding
$\mathcal{L}_{VQA1}=\alpha(\|\mathcal{G}_q(x^s)-\mathcal{G}(x^t)\|_2+\|\mathcal{G}_q(x^t)-\mathcal{G}(x^s)\|_2)$
$\mathcal{L}_{VQA2}=\beta\sum_{c=1}^{C}\left\|B\left(x_c^s\right)-B\left(x_c^t\right)\right\|_2$
Codebook $\mathcal{E}$
$e_1,e_2,\ldots$
$\ldots,e_K$
$\mathcal{G}_q(x_i)=e_k$
$k=\mathrm{argmin}_j\|\mathcal{G}(x_i)-e_j\|_2$
$\mathcal{G}_q(x^s)$
$\mathcal{G}(x^s)$
$\mathcal{G}_q(x^t)$
$\mathcal{G}(x^t)$
Bottleneck layer
$B(x^s)$
$B(x^t)$
Classifier $\mathcal{C}$
$\mathcal{L}_{CE}$
$\mathcal{L}_{MIMC}$
Two-step Vector Quantization Alignment module
☒ Class token
$\mathcal{L}_{CE}$ Cross entropy loss
$\mathcal{L}_{VQA}$ Vector Quantization Alignment loss
$\mathcal{L}_{MIMC}$ Mutual Information Weighted Maximization Confusion Matrix loss
feature concatenate

Fig. 1. Overview of TransVQA for unsupervised domain adaptation (UDA). The transformer feature extractor $\mathcal{G}$ is used for feature extraction. The two-step vector quantization alignment module for cross-domain feature global and local alignment. Classifier $\mathcal{C}$ for classification. $\mathcal{L}_{CE}$ is the cross-entropy loss on the source domain. $\mathcal{L}_{MIMC}$ is the mutual information weighted maximization confusion matrix loss on the target domain. $\mathcal{L}_{VQA}$ is the two-step alignment method to align the extracted cross-domain features. $\mathcal{E}$ is the codebook with K items for vector quantization (VQ).

adaptation and uses the codebook to achieve global and local alignment of features across different domains.

III. The TransVQA Method

This section describes the proposed Transferable Transformer-based Vector Quantization Alignment method (TransVQA). In UDA, a labeled source domain data with N_s samples are denoted by $\mathcal{S} = \left\{\left(x_i^s, y_i^s\right)\right\}_{i=1}^{N_s}$, where $y_i^s \in \{1, 2, \ldots, C\}$ is the class label of a sample x_i^s. An unlabeled target domain data with N_t samples are denoted by $\mathcal{T} = \left\{\left(x_j^t\right)\right\}_{j=1}^{N_t}$. Due to the domain shift issue, the source and target domains have different data probability distributions i.e., $p_s\left(x^s\right) \neq p_t\left(x^t\right)$, but share the same class label space, i.e., $\mathcal{Y}_s = \mathcal{Y}_t$. Domain adaptation aims to solve the domain shift issue and improve the recognition accuracy of unlabeled target domain data using labeled source domain data.

A. Motivation and Preliminaries

The UDA method mainly solves domain shift issues for different domains [6], [8], [9]. Most existing domain adaptation methods are based on CNN networks and pseudo-label supervision for data distribution alignment across-domain, expecting to solve the domain shift problem [15], [18], [19], [23], [24], [25]. However, with the excellent performance of the transformer network in various tasks, it can be found that the transformer network is more efficient than CNN in extracting features as a backbone network [26], [27], [28], [41]. To take full advantage of the transformer network and pseudo-label, we proposed a domain adaptation method (TransVQA) as shown in Fig. 1. Our TransVQA method uses a transformer as the backbone network to extract more accurate features. Then the VQ technique is applied to the Two-step Vector Quantization Alignment module [36], [37]. Finally, TransVQA uses the mutual information weighted maximization confusion matrix loss module [14], [15] to enhance the confidence of the target domain pseudo label and solve the problem of confusion of cross-domain aligned features.

In this paper, we represent our TransVQA framework by $\mathcal{F}$, which consists of a deep transformer feature extractor $\mathcal{G}$, a vector quantization alignment module with codebook $\mathcal{E}$ and bottleneck layer $\mathcal{B}$, and a head classifier $\mathcal{C}$. Our TransVQA method will input both labeled source domain data and unlabeled target domain data, thus moving the source knowledge to the target and enhancing the recognition performance of the target domain.

B. Transformer Feature Extraction Module

Deep transformer networks with self-attention mechanisms have surpassed the performance of CNN networks in image classification, detection, and segmentation [4], [5]. Therefore, deep transformer networks are quickly becoming the backbone for feature extraction in domain adaptation [26], [27], [28], [41]. In our TransVQA framework, the backbone $\mathcal{G}$ for feature extraction will use ViT-transformer [4] or Swin-transformer [5]. For batch samples (x^s, y^s) and $\left(x^t\right)$ in the source domain and target domain. First, build the input images via the patch + position embedding module (where Swin-transformer has no class token) to retain information about content and position and get z_0^s and z_0^t. Second, the extracted features $\mathcal{G}\left(x^s\right)$, $\mathcal{G}\left(x^t\right)$ can be obtained via the backbone $\mathcal{G}$ consisting of a series of Multi-Head self Attention (MHSA), MLP, layer normalization (LN), and residual connections [4], [5].

We can obtain the correspondent features $\mathcal{G}\left(x^s\right) \in \mathcal{R}^{b\times D}$ and $\mathcal{G}\left(x^t\right) \in \mathcal{R}^{b\times D}$ through the feature extraction network $\mathcal{G}$ (for ViT network only use the class token feature). Where D is the dimension of the feature, and b is the batch size of the network.

C. Two-Step Vector Quantization Alignment Module

After obtaining the source domain feature $\mathcal{G}\left(x^s\right)$ and the target domain feature $\mathcal{G}\left(x^t\right)$, we want to remove the domain

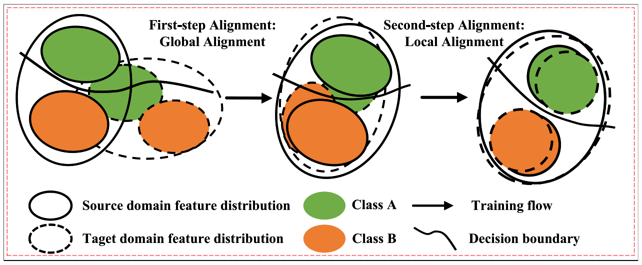


Fig. 2. Overview of Two-step Vector Quantization Alignment. Global alignment is used to align the overall distribution of features across domains. Local alignment is used for cross-domain feature alignment within the same class.

shift issue between the source and target domains. To achieve this, First, we use the Select Vector from the Codebook procedure for updating and selecting vector items. Then we use the selection and learned codebook in two steps: Global Alignment via Vector Quantization (VQ) and Local Alignment via Pseudo label. Vector quantization uses nearest neighbors to select the items in the codebook that are closest to the feature. Global alignment is used to align the overall distribution of features across domains. Local alignment is used for cross-domain feature alignment of data of the same class. This two-step alignment process is illustrated in Fig. 2.

Vector Select From the Codebook: As shown in overview Fig. 1, the codebook $\mathcal{E} \in \mathcal{R}^{K\times D}$ is compared with features from both the source and target domains and then updates the items in the codebook. Therefore codebook provided a cross-domain shared latent space that contains the shared information in the source and target domains. For example, domain shift features such as size, illumination, angle, background, location, etc. are ignored, and discriminable semantic features such as the contour and shape of the object are included. Therefore the learnable codebook $\mathcal{E} \in \mathcal{R}^{K\times D}$ provides more prior information for cross-domain alignment. Where D is the dimension of the codebook item vector e_i, and K is the number of the codebook items. The codebook item of $\mathcal{G}(x_i)$ is using the nearest neighbor item in the shared embedding space $\mathcal{E}$, as shown in Eq. (1).

$$\mathcal{G}_q(x_i) = e_k, \quad k = \mathrm{argmin}_j \left\|\mathcal{G}(x_i) - e_j\right\|_2 \qquad (1)$$

where $\mathcal{G}(x_i)$ is the output of the feature extractor $\mathcal{G}$. We approximate the gradient by copying the gradient from the classifier input $\mathcal{G}_q(x)$ to the feature extractor output $\mathcal{G}(x)$ [37]. e_i is initialized using uniform distribution. We can find that the process of selecting items from the codebook is similar to the VQ-VAE method, but the direct use of item $\mathcal{G}_q(x)$ instead of $\mathcal{G}(x)$ will lead to information distortion issues. Therefore, we apply the a priori information provided by the codebook by concatenating $\mathcal{G}(x)$ and $\mathcal{G}_q(x)$. The $\mathcal{G}(x)$ similar elements selected from the codebook will help the features to perform a better cross-domain alignment.

First-Step Alignment: Global Alignment via Vector Quantization: For features $\mathcal{G}(x^s)$ and $\mathcal{G}\left(x^t\right)$ of the source and target domains, the corresponding items $\mathcal{G}_q(x^s)$ and $\mathcal{G}_q\left(x^t\right)$ can be obtained after the vector select from Codebook process. To reduce the domain shift of the source and target domains, we can do it by reducing the distance between $\mathcal{G}(x^s)$ and $\mathcal{G}_q\left(x^t\right)$, $\mathcal{G}\left(x^t\right)$ and $\mathcal{G}_q(x^s)$ at the same time. This process is a global alignment for cross-domain, and the loss function can be expressed as:

$$\begin{aligned}\mathcal{L}_{VQA1} = \alpha\Big(&\left\|\mathcal{G}_q\left(x^s\right) - sg\left\lfloor\mathcal{G}\left(x^t\right)\right\rfloor\right\|_2 \\ &+ \left\|\mathcal{G}_q\left(x^t\right) - sg\left\lfloor\mathcal{G}\left(x^s\right)\right\rfloor\right\|_2\Big)\end{aligned} \qquad (2)$$

where $sg\lfloor\cdot\rfloor$ denotes the stop gradient operation, $\|\cdot\|_2$ denotes the Mean Squared Error (MSE, or Euclidean Distance), and α is a scalar parameter. $\left\|\mathcal{G}_q(x^s) - sg\left\lfloor\mathcal{G}\left(x^t\right)\right\rfloor\right\|_2$ denotes aligning the target domain feature with the prior information of the source domain in the codebook, and $\left\|\mathcal{G}_q\left(x^t\right) - sg\left\lfloor\mathcal{G}(x^s)\right\rfloor\right\|_2$ denotes aligning the source domain feature with the prior information of the target domain in the codebook. In this indirect cross-domain global alignment with the help of a priori codebook, the learnable hidden space can be used as a bridge between the source and target domains, reducing the alignment difficulty. For example, for an "Art" scene and a "Painting" scene, the learnable codebook in the hidden space can contain information for both scenes. For cross-domain global alignment, we only need to minimize $\mathcal{L}_{VQA1}$ to align the global feature distribution.

Second-Step Alignment: Local Alignment via Pseudo Label: Since the First-step Alignment focuses more on the overall distribution and ignores inter-class differences, this may lead to confusion about the features of different classes. To solve the problem, Second-step Alignment is used for cross-domain feature alignment within the same class. After the vector selects from the codebook, the concatenating $\mathcal{G}(x)$ and $\mathcal{G}_q(x)$ will go through a bottleneck layer (a linear layer, an activation layer, and a batch-norm layer) to obtain new features, with source and target domain bottleneck features $B(x^s)$ and $B\left(x^t\right)$, respectively. These can be expressed as:

$$\begin{aligned}B\left(x^s\right) &= bottleneck\left(\left[\mathcal{G}\left(x^s\right), \mathcal{G}_q\left(x^s\right)\right]\right) \\ B\left(x^t\right) &= bottleneck\left(\left[\mathcal{G}\left(x^t\right), \mathcal{G}_q\left(x^t\right)\right]\right)\end{aligned} \qquad (3)$$

In Eq. (3), the parameters of *bottleneck* are shared for the source and target domains. As seen on overview Fig. 1, different domain samples with the same class label can form sample pairs. For a sample pair, we can make local cross-domain alignment by reducing the distance between the source domain samples and the target domain samples. Since the class label of the target domain is unknown, a pseudo-label is needed. For the C classes cross-domain bottleneck features, the Second-step Alignment: Local Alignment via Pseudo label loss can be expressed as follows.

$$\mathcal{L}_{VQA2} = \beta\textstyle\sum_{c=1}^{C}\left\|B\left(x_c^s\right) - B\left(x_c^t\right)\right\|_2 \qquad (4)$$

where β is a scalar parameter, c is the class label for source and target domains. $\left\|B\left(x_c^s\right) - B\left(x_c^t\right)\right\|_2$ denotes the Euclidean distance of the cross-domain sample pairs of class c. We only need to minimize $\mathcal{L}_{VQA2}$ to align the cross-domain feature representation of the same class. The local alignment loss needs to use the pseudo label of the target domain, but the pseudo label obtained by cross-entropy is inefficient. Therefore, for the output of classifier $\mathcal{C}$ on the target domain, a post-processing method combining mutual information and confusion matrix is proposed to obtain the pseudo label of the target domain.

In summary, the Two-step vector quantization alignment loss can be expressed as follows:

$$\begin{aligned}\mathcal{L}_{VQA} &= \alpha\mathcal{L}_{VQA1} + \beta\mathcal{L}_{VQA2}\\ &= \alpha\left(\left\|\mathcal{G}_q\left(x^s\right) - sg\left\lfloor\mathcal{G}\left(x^t\right)\right\rfloor\right\|_2\right.\\ &\quad \left.+\left\|\mathcal{G}_q\left(x^t\right) - sg\left\lfloor\mathcal{G}\left(x^s\right)\right\rfloor\right\|_2\right)\\ &\quad +\beta\textstyle\sum_{c=1}^{C}\left\|B\left(x_c^s\right) - B\left(x_c^t\right)\right\|_2 \end{aligned} \quad (5)$$

where α and β are positive trade-off parameters for global and local alignment.

D. Mutual Information Weighted Maximization Confusion Matrix Module

Addressing the domain shift issue to classify unlabeled target domain data is a challenging task. If only cross-domain feature alignment is used, it will instead reduce the ability of the classifier to discriminate the target domain. To solve this problem, we propose the Mutual Information weighted Maximization Confusion matrix (MIMC) module obtain more accurate pseudo labels in the target domain. In the MIMC module, we can use $\mathcal{L}_{MIMC}$ loss to constrain the target domain output $Z^t = \mathcal{F}\left(x^t\right) \in \mathcal{R}^{b\times C}$ of the classifier. Where C is the number of classes for the source domain, b is the batch size of the target domain. To enhance the robustness of $\mathcal{L}_{MIMC}$ loss, First, we normalize the probability for Z^t via Eq. (6).

$$Y_{i,j}^t = \frac{\exp\left(Z_{i,j}^t/T\right)}{\sum_{j=1}^{C}\exp\left(Z_{i,j}^t/T\right)} \quad (6)$$

where $Y_{i,j}^t$ is the i-th sample belongs to the j-th class on target source, $Z_{i,j}^t$ is the output of classifier $\mathcal{C}$, and T is the probability normalized scale parameter. We use $T = 2.5$ in all our experiments [15].

To retain more variation between samples and to enforce the prediction of the target domain close to the one-hot code [26]. We introduce the mutual information $\mathcal{I}\left(Y^t; x_i^t\right)$ of each sample x_i^t, which defined as:

$$\mathcal{I}\left(Y^t; x_i^t\right) = H\left(\overline{Y}^t\right) - H\left(Y_i^t\right) \quad (7)$$

where, $\overline{Y}^t = \mathbb{E}_{x^t}\left[Y^t\right]$, H denotes entropy. In addition, the larger value of $\mathcal{I}\left(Y^t; x_i^t\right)$ denotes that sample x_i^t is more important for the classification of the target domain. Therefore, the mutual information weight of sample i can be defined as.

$$W_{i,i} = \frac{\mathcal{I}\left(Y^t; x_i^t\right)}{\sum_{i=1}^{b}\mathcal{I}\left(Y^t; x_i^t\right)} \quad (8)$$

where W is a diagonal matrix to denote the importance of each sample in a batch for the target domain. For the probability normalized Y^t and mutual information weight, we can compute the class weighted confusion matrix by Eq. (9):

$$R_{j,j'}^t = {Y_{.,j}^t}^T W Y_{.,j'}^t \quad (9)$$

where, $R_{j,j'}^t$ denotes the correlation between class j and class j' on the target domain. For C class data $R_{j,j'}^t$ on the target domain, we need to normalize it as $\widetilde{R}_{j,j'} = R_{j,j'}/\sum_{j''=1}^{C} R_{j,j''}$. We only need to maximize the confusion matrix trace [15]. Finally, the mutual information weighted maximization confusion matrix (MIMC) loss is denoted as:

$$\mathcal{L}_{MIMC} = \frac{1}{C}\sum_{j=1}^{C}\sum_{j'\neq j}^{C}\left|\widetilde{R}_{j,j'}\right| \quad (10)$$

Note that the loss $\mathcal{L}_{MIMC}$ only works on the target domain. We only need to maximize $\mathcal{L}_{MIMC}$ to enhance the confidence of the pseudo labels. In addition, we set a $threshold = 0.8$ to select the target domain samples which are relatively correctly classified, i.e., only $\left\{x_j^t | Z_{j,(c)}^t > 0.8\right\}$ are involved in the calculation of local alignment loss $\mathcal{L}_{VQA2}$.

E. Overall Formulation

As shown in Fig. 1, the loss function has three terms, the standard cross-entropy loss $\mathcal{L}_{CE}$ only works on the source domain, the mutual information weighted maximization confusion matrix loss $\mathcal{L}_{MIMC}$ only works on the target domain, and the Two-step vector quantization alignment loss $\mathcal{L}_{VQA}$. The loss $\mathcal{L}_{VQA}$ consists of two parts, global alignment loss $\mathcal{L}_{VQA1}$ and local alignment loss $\mathcal{L}_{VQA2}$. The aggregate loss function is defined as follows:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{CE} + \mathcal{L}_{VQA} - \gamma\mathcal{L}_{MIMC}\\ &= \mathcal{L}_{CE} + \alpha\mathcal{L}_{VQA1} + \beta\mathcal{L}_{VQA2} - \gamma\mathcal{L}_{MIMC}\end{aligned} \quad (11)$$

where α, β, and γ are positive trade-off parameter. With this joint loss, feature extractor $\mathcal{G}$, codebook $\mathcal{E}$ and the classifier $\mathcal{C}$ of the deep domain adaption network $\mathcal{F}$ can be trained end-to-end by back-propagation.

IV. Experiment

A. Experimental Setting

a) Datasets. We use four classic datasets: **DomainNet** [42], **VisDA-2017** [43], **Office-Home** [44], and **Office-31** [45]. Some sample images of each domain in these datasets are shown in Fig. 3.

(1) DomainNet is the most challenging UDA large dataset. It contains 0.6 million images in 345 classes and six domains: Clipart (**clp**), Infograph (**inf**), Painting (**pnt**), Quickdraw (**qdr**), Real (**rel**), and Sketches (**skt**). We build 30 UDA tasks to evaluate our model, i.e., **clp→inf**, ..., **skt→rel**.

(2) VisDA-2017 includes 12 classes and two domains: synthetic and realistic. We build adaptation task is **synthetic** (155K images) → **realistic** (55K images).

(3) Office-Home contains 15,500 images in 65 classes and four domains: Art (**Ar**), Clipart (**Cl**), Product(**Pr**), and Real World (**Rw**). We build 12 UDA tasks, i.e., **Ar** → **Cl**, ..., **Rw** → **Pr**.

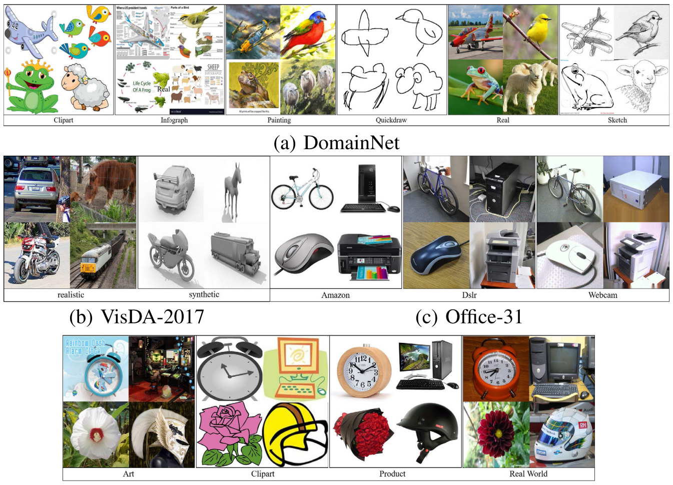
(4) Office-31 contains 4,110 images in 31 classes and 3 domains: Amazon (**A**), Webcam (**W**) and Dslr (**D**). We build 6 adaptation tasks, i.e., **A** → **W**, ..., **D** → **W**.

b) Implementation details. For a fair comparison with the existing UDA methods, we use the same backbone network (ViT-B and Swin-B) pre-trained on ImageNet [46] (ImageNet 1K and ImageNet 21K). We use source and target domains trained for all UDA tasks and predict the unlabeled target domain. The input image size in experiments is resized to 224 × 224. For the ViT-B-based TransVQA method, we use

TABLE I

ACCURACY (%) ON DOMAINNET FOR UDA. COLUMNS ARE THE SOURCE DOMAIN AND ROWS ARE THE TARGET DOMAIN

ViT-B [4]	clp	inf	pnt	dqr	rel	skt	Avg	CDTrans-B [27]	clp	inf	pnt	dqr	rel	skt	Avg
clp	-	20.1	46.2	13.0	62.3	48.8	38.1	clp	-	27.9	57.6	27.9	73.0	58.5	49.0
inf	46.4	-	45.2	5.1	62.3	37.5	39.3	inf	58.6	-	53.4	9.6	71.1	47.6	48.1
pnt	48.1	19.1	-	4.4	62.5	41.8	35.2	pnt	60.7	24.0	-	13.0	69.8	49.6	43.4
qdr	28.2	5.2	14.4	-	21.9	17.7	17.5	qdr	2.9	0.4	0.3	-	0.7	4.7	1.8
rel	53.2	19.3	53.5	7.2	-	41.6	35.0	rel	49.3	18.7	47.8	9.4	-	33.5	31.7
skt	58.0	18.5	46.5	15.7	58.7	-	39.5	skt	66.8	23.7	54.6	27.5	68.0	-	48.1
Avg	46.8	16.4	41.2	9.1	53.5	37.5	34.1	Avg	47.7	18.9	42.7	17.5	56.5	38.8	37.0
DOT-B [35]	clp	inf	pnt	dqr	rel	skt	Avg	TransVQA-B	clp	inf	pnt	dqr	rel	skt	Avg
clp	-	20.2	53.6	26.7	71.2	55.2	45.4	clp	-	26.7	56.9	26.8	72.9	59.2	48.5
inf	63.0	-	54.6	12.3	73.1	50.7	50.7	inf	60.9	-	52.8	11.5	74.6	53.1	50.6
pnt	61.8	20.3	-	11.4	72.2	50.5	43.2	pnt	62.9	25.2	-	13.4	74.3	50.8	45.3
qdr	47.3	7.4	30.3	-	44.6	33.7	32.7	qdr	46.4	8.5	31.0	-	45.9	35.1	33.4
rel	62.9	20.0	56.9	17.3	-	49.3	41.3	rel	69.6	21.3	58.0	16.4	-	51.2	43.3
skt	67.3	18.7	52.9	27.8	69.8	-	47.3	skt	68.8	34.4	53.3	27.2	70.9	-	50.9
Avg	60.5	17.3	49.7	19.1	66.2	47.9	43.3	Avg	61.7	23.2	50.4	19.0	67.7	49.9	**45.3**



(a) DomainNet

(b) VisDA-2017

(c) Office-31

(d) Office-Home

Fig. 3. Some sample images of each domain in Domainnet, VisDA-2017, Office-Home, and Office-31 datasets, respectively.

the SGD optimizer [47] with a momentum of 0.9 and weight decay of 5×10^{-4}. For all databases, the batch size is 100. For DomainNet, the hyper-parameter $K = 4096$, $\alpha = 1$, $\beta = 1$, and $\gamma = 10$ for all UDA tasks. For VisDA-2017, OfficeHome, and Office-31, the hyper-parameter $K = 2048$, $\alpha = 1$, $\beta = 1$, and $\gamma = 1$ for all UDA tasks. For the TransVQA method built on the Swin-B, we employ AdamW [48] optimizer with a momentum of 0.9 and weight decay of 5×10^{-3}. For all databases, the batch size is 64. For VisDA-2017, OfficeHome, and Office-31, the hyper-parameter $K = 2048$, $\alpha = 0.8$, $\beta = 1.2$, and $\gamma = 1$ for all UDA tasks. We implemented all simulations on a PC with Intel Xeon (R) CPU E5-2620 v4 @2.1 GHz×32 128G and four NVIDIA 2080Ti.

B. Over All Results

We compare the recognition accuracy of the TransVQA method with existing UDA methods, especially transformer-based methods. The Transformer-based methods are Transferable vision transformer (TVT) [26], Cross-domain transformer (CDtrans) [27], Bidirectional cross-attention transformer (BCAT) [28], and Domain-oriented transformer (DOT) [35]. “Source only” means the baseline model trained on the source data using the backbone network.

The results on DomainNet dataset are shown in Table I. Using the same backbone network (ViT-B) our TransVQA method achieves higher performance. The average accuracy of our TransVQA method is 11.2% higher than ViT-B and 2.0% higher than DOT-B. Our method achieves high accuracy in most UDA tasks, e.g., clp → rel, skt → clp, pnt → rel, inf → pnt, rel → skt, and inf → skt, etc.

The results on VisDA-2017 dataset are shown in Table II. Compared with other UDA methods, our TransVQA obtains a higher average accuracy with the same backbone network and pre-trained weights. Specifically, TransVQA (ViT-B, IN-1K) outperforms the DOT (ViT-B, IN-1K) by 1.0% average accuracy, TransVQA (ViT-B, IN-21K) outperforms the TVT (ViT-B, IN-21K) by 5.5% average accuracy, and TransVQA (Swin-B, IN-1K) outperforms BCAT (Swin-B, IN-1K) by 3.3% average accuracy. In difficult categories, such as knife and truck, our method outperforms other existing methods by achieving an accuracy of 98.8% and 76.4%.

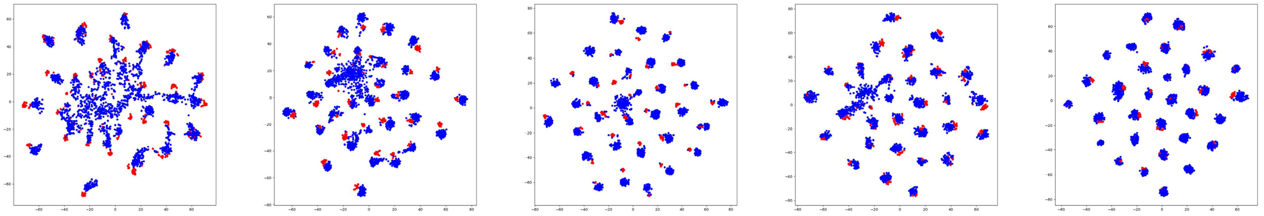
Results on Office-Home dataset are shown in Table III. Our TransVQA obtained the highest average accuracy rates, 84.6% (ViT-B, IN-1K), 86.4% (ViT-B, IN-21K), and 87.6% (Swin-B, IN-1K). And the highest accuracy is obtained on eight UDA tasks in Ar → Cl, Ar → Pr, Cl → Pr, Pr → Cl, Pr → Rw, Rw → Ar, Rw → Cl, and Rw → Pr.

Results on Office-31 dataset are shown in Table IV. Our TransVQA outperforms other baselines and obtains the highest average accuracy, 95.2% (ViT-B, IN-21K) and 95.6% (Swin-B, IN-1K).

These outstanding results demonstrate the effectiveness of our proposed TransVQA method in UDA tasks.

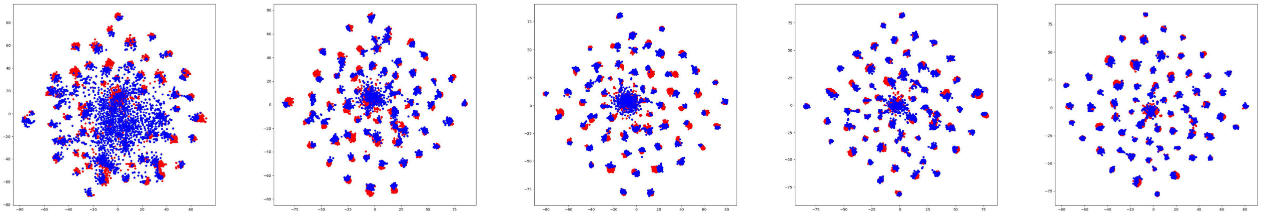
C. Indepth Analysis

Ablation Study: To study the capabilities of the different parts of TransVQA, we performed an ablation analysis based on ViT-B and Swin-B (IN-1K) of VisDA-2017, Office-Home, and Office-31. (1) Source only means to use $\mathcal{L}_{CE}$ only on the source domain. (2) $+\mathcal{L}_{VQA1}$ denotes the addition of the global alignment via vector quantization loss term to Source only. (3) $+\mathcal{L}_{VQA2}$ denotes the addition of the local alignment via pseudo label loss term to Source only. (4) $+\mathcal{L}_{VQA}$ denotes the addition of the Two-step vector quantization alignment



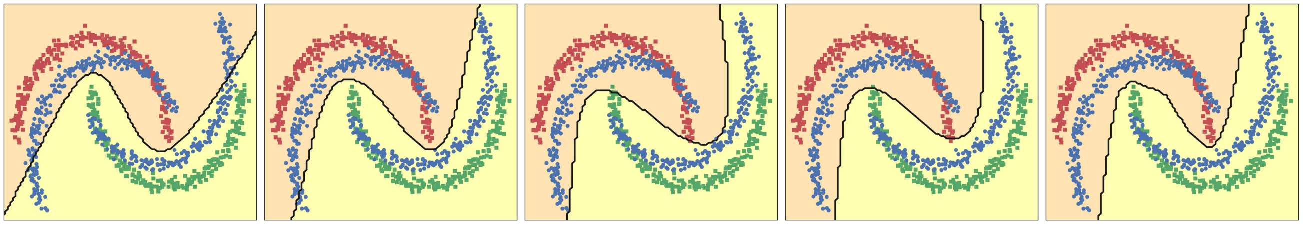
(a) Source only (ViT-B) (b) $\mathcal{L}_{MIMC}$ loss (c) $\mathcal{L}_{VQA1}$ loss (d) $\mathcal{L}_{VQA2}$ loss (e) Our TransVQA

Fig. 4. Visualization of the feature representation learned by different partial losses on task W → A of Office-31. The red and blue points mean source and target domain features, respectively. Note the improved alignment between red and blue over other competing methods by the proposed TransVQA.



(a) Source only (ViT-B) (b) $\mathcal{L}_{MIMC}$ loss (c) $\mathcal{L}_{VQA1}$ loss (d) $\mathcal{L}_{VQA2}$ loss (e) Our TransVQA

Fig. 5. Visualization of the feature representation learned by different partial losses on task Ar → Cl of Office-Home. The red and blue points mean source and target domain features, respectively. Note the improved alignment between red and blue over other competing methods by the proposed TransVQA.



(a) Source only (b) $\mathcal{L}_{MIMC}$ loss (c) $\mathcal{L}_{VQA1}$ loss (d) $\mathcal{L}_{VQA2}$ loss (e) Our TransVQA

Fig. 6. Decision boundaries on twin Moons datasets. Green and red points denote different classes in the source domain. The blue points represent the target domain, generated by rotating the source domain 30 degrees. The solid black line is the decision boundary.

TABLE II

ACCURACY (%) ON VISDA-2017 FOR UDA. (IN-1K/21K DENOTES THE PRETRAINNED MODEL ON IMAGENET-1K/21K)

Method	Pretrain		plane	bcycl	bus	car	horse	knife	mcycle	person	plant	sktbrd	train	truck	Avg
Source Only		ResNet-101	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
DANN [11]	IN-1K		81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN [16]	IN-1K		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 [13]	IN-1K		87.0	60.9	83.7	63.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
SHOT [31]	IN-1K		94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
FixBi [32]	IN-1K		96.1	87.8	90.5	90.3	96.8	95.3	92.8	88.7	97.2	94.2	90.9	25.7	87.2
Source Only		ViT-B	98.2	73.0	82.5	62.0	97.3	63.5	96.5	29.8	68.7	86.7	**96.7**	23.7	73.2
TVT [26]	IN-1K		97.1	88.8	86.4	64.4	96.4	97.4	90.6	64.1	92.0	90.3	93.7	59.6	85.1
TVT [26]	IN-21K		97.1	92.9	85.4	66.4	97.1	97.1	89.3	75.5	95.0	94.7	94.5	55.1	86.7
CDTrans [27]	IN-1K		97.1	90.5	82.4	77.5	96.6	96.1	93.6	**88.5**	**97.9**	86.9	90.3	62.8	88.4
BCAT [28]			98.9	91.3	87.4	73.8	97.9	98.1	**94.2**	64.0	88.9	97.4	96.0	60.7	87.4
DOT [35]	IN-1K		**99.3**	92.7	**89.0**	78.8	98.2	96.1	93.2	80.2	97.6	95.8	94.4	69.0	90.3
TransVQA	IN-1K		96.7	93.1	88.8	84.9	98.6	96.2	**94.2**	82.6	97.0	98.0	95.1	70.0	91.3
TransVQA	IN-21K		98.9	**94.4**	**89.0**	**85.4**	**98.6**	**98.8**	93.9	84.6	96.5	**97.9**	96.2	**71.8**	**92.2**
Source Only		Swin-B	98.7	63.0	86.7	68.5	84.6	59.4	98.0	22.0	81.9	91.4	**96.7**	25.7	73.9
BCAT [28]			**99.1**	91.5	86.8	72.4	**98.6**	**98.1**	**96.5**	**82.1**	94.4	96.0	93.9	61.1	89.2
TransVQA	IN-1K		99.0	**95.2**	**91.4**	**85.0**	98.4	97.2	94.4	80.5	**96.6**	**99.0**	**97.0**	**76.4**	**92.5**

loss term to Source only. (5) $+\mathcal{L}_{MIMC}$ means the addition of the mutual information weighted maximization confusion matrix loss term to Source only. (6) TransVQA denotes the complete method we proposed. The results are shown in Table V. We can find that the full TransVQA outperforms other components. The average accuracy with $\mathcal{L}_{VQA}$ loss is 7.8% and 8.1% higher than the Source Only. It shows that the two-step vector quantization alignment module based on the codebook is significantly effective in cross-domain alignment. Based on the VIT-B backbone network, the average accuracy with $\mathcal{L}_{VQA1}$ loss and $\mathcal{L}_{VQA2}$ loss are 6.4% and 6.6% higher than the Source Only. It shows that each part of the two-step vector quantization alignment module is effective in cross-domain alignment. The average accuracy with $\mathcal{L}_{MIMC}$ loss is 6.3% and 6.2% higher than the Source Only. It shows that the MIMC module can effectively improve pseudo-label accuracy.

Training Efficient Study: We computed average Accuracy (%), occupied GPU memory, and FPS (Frames Per Second)

TABLE III
ACCURACY (%) ON OFFICE-HOME FOR UDA

Method	Pretrain		Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl →Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
Source Only		ResNet-50	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
DANN [11]	IN-1k		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
MCD [13]	IN-1k		48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
DCAN [24]	IN-1k		54.5	75.7	81.2	67.4	74.0	76.3	67.4	52.7	80.6	74.1	59.1	83.5	70.5
SHOT [31]	IN-1k		57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
FixBi [32]	IN-1k		58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
Source Only		ViT-B	66.2	84.3	86.6	77.9	83.3	84.3	76.0	62.7	88.7	80.1	66.2	88.6	78.8
TVT [26]	IN-1K		67.1	83.5	87.3	77.4	85.0	85.6	75.6	64.9	86.6	79.1	67.2	88.0	78.9
TVT [26]	IN-21K		74.9	86.8	89.5	82.8	88.0	88.3	79.8	71.9	90.1	85.5	74.6	90.6	83.6
CDTrans [27]	IN-1K		68.8	85.0	86.9	81.5	87.1	87.3	79.6	63.3	88.2	82.0	66.0	90.6	80.5
BCAT [28]			74.2	90.6	90.9	84.2	90.9	89.9	**84.1**	74.5	90.8	**85.7**	**74.8**	92.2	85.2
DOT [35]	IN-1K		69.0	85.6	87.0	80.0	85.2	86.4	78.2	65.4	87.9	79.7	67.3	89.3	80.1
DOT [35]	IN-21K		73.1	89.1	90.1	85.5	89.4	89.6	83.2	72.1	90.4	84.4	72.9	91.5	84.3
TransVQA	IN-1K		74.0	89.1	92.0	83.5	90.2	90.7	81.8	72.4	91.0	84.0	73.6	92.5	84.6
TransVQA	IN-21K		**78.2**	**91.1**	**92.8**	**85.6**	**91.9**	**91.8**	83.9	**75.9**	**91.7**	85.6	74.7	**93.6**	**86.4**
Source Only		Swin-B	64.1	84.8	87.6	82.2	84.6	86.7	78.8	60.3	88.9	82.8	65.3	89.6	79.7
BCAT [28]			75.3	90.0	**92.9**	**88.6**	90.3	**92.7**	**87.4**	73.7	92.5	86.7	75.4	93.5	86.6
TransVQA	IN-1K		**79.5**	**91.5**	92.8	87.0	**91.3**	92.1	86.8	**76.1**	**92.8**	**88.1**	**78.5**	**94.6**	**87.6**

TABLE IV
ACCURACY (%) ON OFFICE-31 FOR UDA

Method	Pretrain		A→W	D→W	W→D	A →D	D→A	W→A	Avg
Source Only	IN-1K	ResNet-50	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DANN [11]	IN-1K		82.0	96.9	99.1	79.7	68.2	67.4	82.2
MCD [13]	IN-1K		88.6	98.5	100	92.2	69.5	69.7	86.5
SHOT [31]	IN-1K		90.1	98.4	99.9	94.0	74.7	74.3	88.6
SCDA [14]	IN-1K		94.2	98.7	99.8	95.2	75.5	76.2	90.0
FixBi [32]	IN-1K		96.1	99.3	100	95.0	78.7	79.4	91.4
Source Only	IN-1K	ViT-B	89.2	98.9	100	88.8	80.1	79.8	89.5
CDTrans [27]	IN-1K		97.6	99.0	100	97.0	81.1	81.9	92.8
TVT [26]	IN-1K		95.7	98.7	100	95.4	80.6	80.3	91.8
TVT [26]	IN-21K		96.4	99.4	100	96.4	84.9	86.1	93.8
BCAT [28]			96.9	98.7	100	97.5	85.5	86.0	94.1
TransVQA	IN-1K		98.2	99.2	100	97.9	86.2	86.0	94.6
TransVQA	IN-21K		**98.6**	**99.6**	100	**98.7**	**86.8**	**87.3**	**95.2**
Source Only	IN-1K	Swin-B	89.2	94.1	100	93.1	80.9	81.3	89.8
BCAT [28]			99.4	99.1	100	**99.8**	85.7	86.1	95.1
TransVQA	IN-1K		**99.6**	**99.5**	100	**99.8**	**87.3**	**87.5**	**95.6**

TABLE V
ABLATION STUDY OF TRANSVQA BASED ON VIT-B AND SWIN-B (PRETRAINED ON IMAGENET-1K)

Method		VisDA-2017	Office-Home	Office-31	Avg	△
Source only	ViT-B	73.2	78.8	89.5	80.5	
+$\mathcal{L}_{VQA1}$		87.5	81.4	91.9	86.9	↑ 6.4
+$\mathcal{L}_{VQA2}$		88.0	81.2	92.1	87.1	↑ 6.6
+$\mathcal{L}_{VQA}$		89.1	82.9	92.9	88.3	↑ 7.8
+$\mathcal{L}_{MIMC}$		87.7	81.3	91.5	86.8	↑ 6.3
TransVQA		91.3	84.6	94.6	90.2	↑ 9.7
Source only	Swin-B	73.9	79.7	89.8	81.1	
+$\mathcal{L}_{VQA}$		89.8	84.0	93.7	89.2	↑ 8.1
+$\mathcal{L}_{MIMC}$		87.2	82.7	92.1	87.3	↑ 6.2
TransVQA		92.5	87.6	95.6	91.9	↑ 10.8

TABLE VI
AVERAGE ACCURACY (%), OCCUPIED GPU MEMORY, AND FPS ON THE OFFICE-HOME

Method		Accuracy	GPU memory	FPS
BCAT [28]	ViT-B	85.5	2241MB	78.6
TransVQA		86.4	1616MB	283.5
BCAT [28]	Swin-B	86.0	4087MB	74.2
TransVQA		87.6	1848MB	270.1

based on ViT-B and Swin-B (IN-1K) of Office-Home to study the training efficiency compared with the multi-branch solution (BCAT method). The results are shown in Table VI. We can find that: the proposed TransVQA method occupies less memory and processes images faster than BCAT with better performance. Those results show that the proposed model is both effective and efficient.

t-SNE Visualization: In this section, we study the effectiveness of each module in the proposed TrandVQA method for eliminating the domain shift issue. We take the W → A task for Office-31 and Ar → Cl task for Office-Home as examples. The t-SNE [49] was used to visualize the distribution of features obtained by each module loss. The distribution of feature learned by Source only (ViT-B), $\mathcal{L}_{MIMC}$ loss (Source only + Mutual Information weighted Maximization Confusion matrix loss), $\mathcal{L}_{VQA1}$ loss (Source only + Global Alignment via Vector Quantization loss), $\mathcal{L}_{VQA2}$ loss (Source only + Local Alignment via pseudo label loss), and our complete TransVQA results are shown in Fig. 4 and Fig. 5. From the results, we can find that: (1) Mutual Information weighted Maximization Confusion matrix module can enhance the classification performance of the target domain; (2) Global Alignment via Vector Quantization module can reduce across-domain global shift; (3) Local Alignment via Vector Quantization module can reduce the cross-domain intra-class shift; (4) Our TransVQA method can effectively solve the domain shift issue.

Decision Boundaries on Twin Moons Datasets: To better demonstrate the effectiveness of each part of our TransVQA method for UDA, we conducted comparison experiments with source-only, $\mathcal{L}_{MIMC}$ loss, $\mathcal{L}_{VQA1}$ loss, $\mathcal{L}_{VQA2}$ loss, and our TransVQA on 2D twin moon datasets [50]. The backbone network uses a shallow MLP network and visualizes the decision boundaries for the different parts of the loss decisions. The results are shown in Figure 6. From the results, we can find: (1) Due to the domain shift issue, the Source-only method cannot completely and correctly classify the target domain. (2) The $\mathcal{L}_{MIMC}$ loss post-processes the target domain output, which can improve the recognition accuracy of the target domain. (3) $\mathcal{L}_{VQA1}$ loss and $\mathcal{L}_{VQA2}$ loss can enhance the recognition accuracy of the target domain by cross-domain alignment. (4) Our TransVQA decision boundary correctly classifies all samples in the source and target domains.

Visualization of Attention Maps: In this study, attention maps will be visualized with the W → A task of the Office-31 dataset and the Cl → Rw task of the Office-Home dataset as examples. The results are shown in Fig. 7. The results show that: the proposed TransVQA method pays more attention to important regions more accurately than the source-only (ViT-B) method. For example, in the "Mouse" image on Office-31, the Source-only method focuses more on the background,

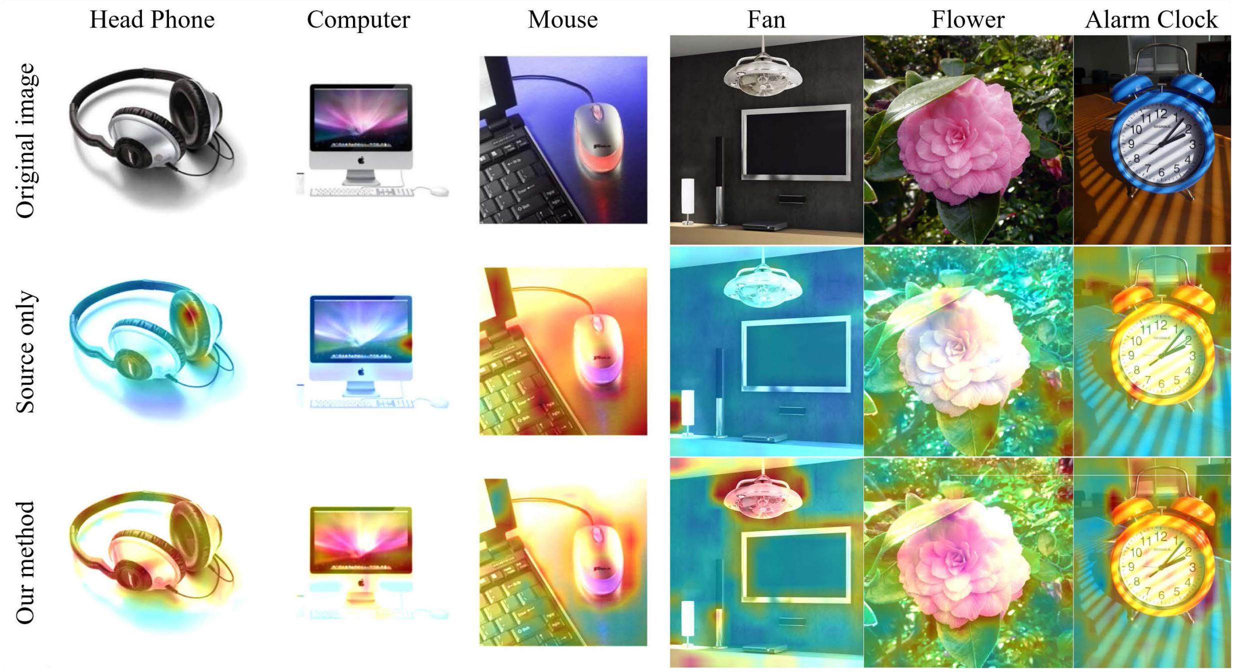


Fig. 7. Attention maps of images. “Head Phone”, “Computer”, and “Mouse” in the Office-31 dataset; “Fan”, “Flower”, and “Alarm Clock” in the Office-Home dataset.

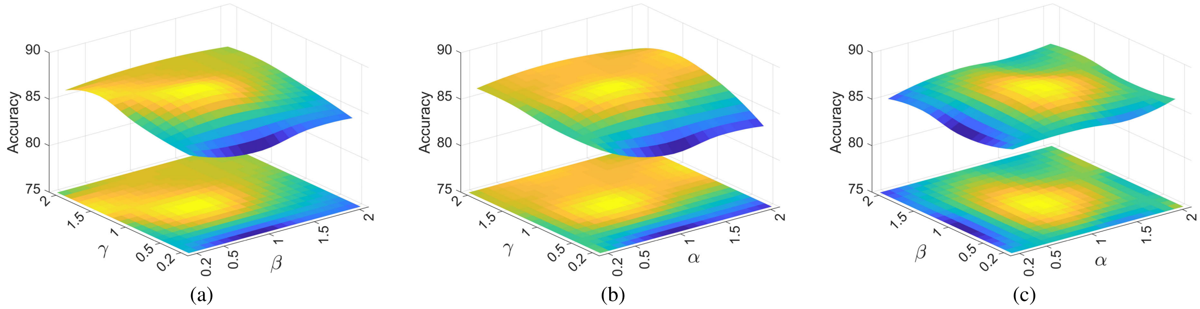


Fig. 8. Parameter sensitivity of TransVQA on task W → A of Office-31, where (a) the effects of tuning β and λ on the performance by fixing $\alpha = 1$; (b) the effects of tuning α and λ on the performance by fixing $\beta = 1$; (c) the effects of tuning α and β on the performance by fixing $\lambda = 1$.

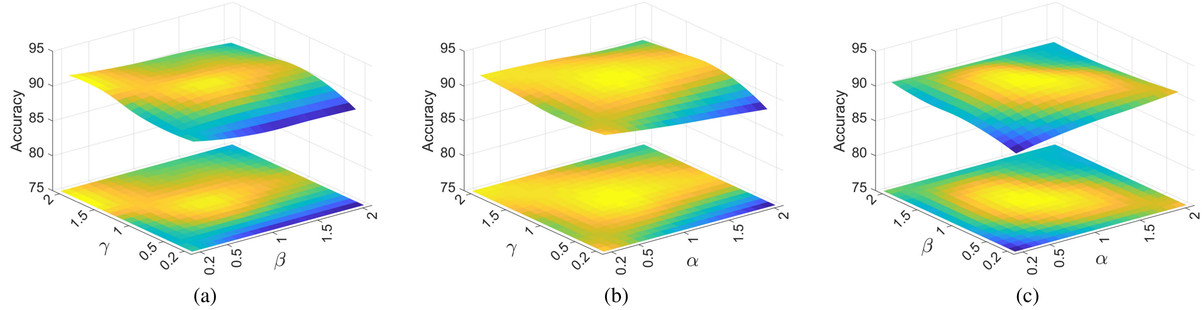
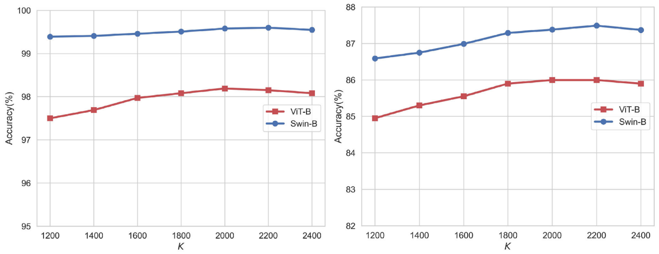


Fig. 9. Parameter sensitivity of TransVQA on task Pr → Rw of Office-Home, where (a) the effects of tuning β and λ on the performance by fixing $\alpha = 1$; (b) the effects of tuning α and λ on the performance by fixing $\beta = 1$; (c) the effects of tuning α and β on the performance by fixing $\lambda = 1$.

such as the computer and desktop, while our TransVQA main attention is focused on the mouse. In the “Fan” image on Office-Home, the Source-only method pays more attention to the “table lamp”, while our TransVQA method pays more attention to the “Fan”.

Parameter Sensitivity: In this study, we check the sensitivity of our TransVQA to hyper-parameters with several experiments. Note that our TransVQA has three parameters α, β, and γ in the loss function. We take the W → A task of Office-31 and Pr → Rw task of Office-Home as examples. We will fix one of the parameters and analyze the other two parameters by grid search. The results are shown in Fig. 8 and Fig. 9. The results show that: (1) For the parameter α and β, the higher accuracy is between [0.5, 1.5]. For the parameter γ, the higher accuracy is between [0.8, 2]. (2) Our TransVQA method maintains robust performance over a wide range of parameter choices.

Codebook Size Sensitivity: In this study, we only change the codebook size parameter K to analyze its recognition accuracy in tasks A → W and W → A of Office-31. We use



(a) (b)

Fig. 10. Codebook size sensitivity of TransVQA on Office31, where (a) the task A → W; (b) the task W → A.

the backbone network (ViT-B and Swin-B) pre-trained on ImageNet (ImageNet 1K). The codebook contains 1200, 1400, 1600, 1800, 2000, 2200, and 2400 items. The results are shown in Fig. 10. The results show that: For the codebook size K, the higher accuracy is between [1800, 2400].

V. Conclusion

In this paper, we proposed a novel UDA method named Transferable Vector Quantization Alignment for Unsupervised Domain Adaptation (**TransVQA**), which integrates the Transformer-Net feature extract (Trans), The two-step vector quantization domain alignment (VQA) module, and mutual information weighted maximization confusion matrix of intra-class discrimination (MIMC) into a unified domain adaptation framework. The two-step alignment module solves the domain shift issue by vector quantization for global alignment and pseudo-labels for intra-class local alignment. Experiments on the DomainNet, Office-31, Office-Home, and VisDA-2017 datasets of the UDA task show that TransVQA outperforms state-of-the-art methods.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[4] A. Dosovitskiy, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. ICLR*, 2021.

[5] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[6] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.

[7] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," 2017, *arXiv:1702.05374*.

[8] Y.-W. Luo, C.-X. Ren, P. Ge, K.-K. Huang, and Y.-F. Yu, "Unsupervised domain adaptation via discriminative manifold embedding and alignment," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 5029–5036.

[9] S. Zhao et al., "A review of single-source deep unsupervised visual domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 473–493, Feb. 2022.

[10] F. Zhuang et al., "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

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

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

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

[14] S. Li et al., "Semantic concentration for domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9082–9091.

[15] Y. Jin, X. Wang, M. Long, and J. Wang, "Minimum class confusion for versatile domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 464–480.

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

[17] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19, 2006, pp. 513–520.

[18] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7404–7413.

[19] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1647–1657.

[20] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2962–2971.

[21] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Computer Vision—ECCV 2016*. Cham, Switzerland: Springer, 2016, pp. 443–450.

[22] Y. Zhu et al., "Deep subdomain adaptation network for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1713–1722, Apr. 2021.

[23] S. Li, F. Lv, B. Xie, C. H. Liu, J. Liang, and C. Qin, "Bi-classifier determinacy maximization for unsupervised domain adaptation," in *Proc. AAAI*, vol. 2, 2021, p. 5.

[24] S. Li et al., "Domain conditioned adaptation network," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11386–11393.

[25] 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.

[26] J. Yang, J. Liu, N. Xu, and J. Huang, "TVT: Transferable vision transformer for unsupervised domain adaptation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 520–530.

[27] T. Xu, W. Chen, P. Wang, F. Wang, H. Li, and R. Jin, "CDTrans: Cross-domain transformer for unsupervised domain adaptation," in *Proc. ICLR*, 2022.

[28] X. Wang, P. Guo, and Y. Zhang, "Domain adaptation via bidirectional cross-attention transformer," 2022, *arXiv:2201.05887*.

[29] S. Li, C. H. Liu, B. Xie, L. Su, Z. Ding, and G. Huang, "Joint adversarial domain adaptation," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 729–737.

[30] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3801–3809.

[31] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng, "Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8602–8617, Nov. 2022.

[32] J. Na, H. Jung, H. J. Chang, and W. Hwang, "FixBi: Bridging domain spaces for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1094–1103.

[33] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[34] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12159–12168.

[35] W. Ma, J. Zhang, S. Li, C. H. Liu, Y. Wang, and W. Li, "Making the best of both worlds: A domain-oriented transformer for unsupervised domain adaptation," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 5620–5629.

[36] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[37] A. Van Den Oord et al., "Neural discrete representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6306–6315.

[38] A. Razavi, A. Van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 14837–14847.

[39] Z. Li, R. Togo, T. Ogawa, and M. Haseyama, "Variational autoencoder based unsupervised domain adaptation for semantic segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 2426–2430.

[40] Y. Li, Y. Zhang, and C. Yang, "Unsupervised domain adaptation with joint adversarial variational AutoEncoder," *Knowl.-Based Syst.*, vol. 250, Aug. 2022, Art. no. 109065.

[41] S. Li, M. Xie, K. Gong, C. H. Liu, Y. Wang, and W. Li, "Transferable semantic augmentation for domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11511–11520.

[42] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1406–1415.

[43] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "VisDA: The visual domain adaptation challenge," 2017, *arXiv:1710.06924*.

[44] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5385–5394.

[45] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2010, pp. 213–226.

[46] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[47] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 10, pp. 400–407, Sep. 1951.

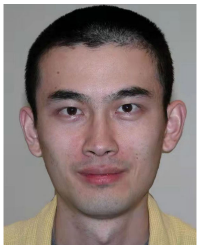
[48] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

[49] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[50] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12 no. 10, pp. 2825–2830, 2012.

Yulin Sun (Member, IEEE) received the M.S. degree in computer science and technology from Soochow University, Suzhou, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Artificial Intelligence, Xidian University, Xi'an, China. His current research interests include pattern recognition, deep learning, and their applications.

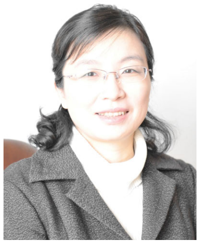
Weisheng Dong (Member, IEEE) received the B.S. degree in electronic engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2004, and the Ph.D. degree in circuits and systems from Xidian University, Xi'an, China, in 2010. He was a Visiting Student with Microsoft Research Asia, Beijing, China, in 2006. From 2009 to 2010, he was a Research Assistant with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. In 2010, he joined the School of Electronic Engineering, Xidian University, as a Lecturer, where he has been a Professor since 2016. His research interests include inverse problems in image processing, deep learning, computer vision, and computational imaging. He was a recipient of the Best Paper Award from the SPIE Visual Communication and Image Processing (VCIP) in 2010. He has served as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING. He is currently serving as an Associate Editor for *SIAM Journal on Imaging Sciences*.



Xin Li (Fellow, IEEE) received the B.S. degree (Hons.) in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 1996, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 2000. He was a Technical Staff Member with Sharp Laboratories of America, Camas, WA, USA, from 2000 to 2002. Since 2003, he has been a Faculty Member with the Lane Department of Computer Science and Electrical Engineering. His research interests include image/video coding and processing. He was a recipient of the Best Student Paper Award from the Conference of Visual Communications and Image Processing in 2001, the Best Student Paper Award from the IEEE Asilomar Conference on Signals, Systems and Computers in 2006, and the Best Paper Award from the Conference of Visual Communications and Image Processing in 2010. He is currently serving as a member for the Image, Video, and Multidimensional Signal Processing Technical Committee and an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Le Dong (Member, IEEE) received the B.S. degree from Wuhan University, Wuhan, Hubei, in 2016, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, in 2021. She is currently a Lecturer with Xidian University, Xi'an, China. Her main research interests include hyperspectral image analysis, pattern recognition, and machine learning.

Guangming Shi (Fellow, IEEE) received the B.S. degree in automatic control, the M.S. degree in computer control, and the Ph.D. degree in electronic information technology from Xidian University in 1985, 1988, and 2002, respectively. From 1994 to 1996, he was a Research Assistant cooperated with the Department of Electronic Engineering, The University of Hong Kong. He joined the School of Electronic Engineering, Xidian University, in 1988. Since 2003, he has been a Professor with the School of Electronic Engineering, Xidian University. In 2004, he was the Head of the National Instruction Base of Electrician and Electronic (NIBEE). From June 2004 to December 2004, he studied with the Department of Electronic Engineering, University of Illinois at Urbana–Champaign (UIUC). He is currently the Deputy Director of the School of Electronic Engineering, Xidian University, and the Academic Leader in the subject of circuits and systems. He has authored or coauthored more than 100 research articles. His research interests include compressed sensing, the theory and design of multirate filter banks, image denoising, low-bit-rate image/video coding, and the implementation of algorithms for intelligent signal processing (using DSP and FPGA).



Xuemei Xie (Senior Member, IEEE) received the M.S. degree in electronic engineering from Xidian University, Xi'an, China, in 1994, and the Ph.D. degree in electrical and electronic engineering from The University of Hong Kong, Hong Kong, in 2004. She is currently a Professor with the School of Artificial Intelligence, Xidian University. She has authored more than 50 academic papers in international and national journals, and international conferences. Her research interests include artificial intelligence, compressive sensing, deep learning, image and video processing, and filter banks.