Patch-Mix Transformer for Unsupervised Domain Adaptation: A Game Perspective

Jinjing Zhu[1*] Haotian Bai[1*] Lin Wang[1,2†]

[1] AI Thrust, HKUST(GZ) [2] Dept. of CSE, HKUST

zhujinjing.hkust@gmail.com, haotianwhite@outlook.com, linwang@ust.hk

Abstract

Endeavors have been recently made to leverage the vision transformer (ViT) for the challenging unsupervised domain adaptation (UDA) task. They typically adopt the cross-attention in ViT for direct domain alignment. However, as the performance of cross-attention highly relies on the quality of pseudo labels for targeted samples, it becomes less effective when the domain gap becomes large. We solve this problem from a game theory's perspective with the proposed model dubbed as ***PMTrans****, which bridges source and target domains with an intermediate domain. Specifically, we propose a novel ViT-based module called* ***PatchMix*** *that effectively builds up the intermediate domain,* i.e.*, probability distribution, by learning to sample patches from both domains based on the game-theoretical models. This way, it learns to mix the patches from the source and target domains to maximize the cross entropy (CE), while exploiting two semi-supervised mixup losses in the feature and label spaces to minimize it. As such, we interpret the process of UDA as a min-max CE game with three players, including the feature extractor, classifier, and PatchMix, to find the Nash Equilibria. Moreover, we leverage attention maps from ViT to re-weight the label of each patch by its importance, making it possible to obtain more domain-discriminative feature representations. We conduct extensive experiments on* ***four*** *benchmark datasets, and the results show that PMTrans significantly surpasses the ViT-based and CNN-based SoTA methods by* ***+3.6%*** *on Office-Home,* ***+1.4%*** *on Office-31, and* ***+17.7%*** *on DomainNet, respectively.* https://vlis2022.github.io/cvpr23/PMTrans

1. Introduction

Convolutional neural networks (CNNs) have achieved tremendous success on numerous vision tasks; however, they still suffer from the limited generalization capability to a new domain due to the domain shift problem [50]. Unsupervised domain adaptation (UDA) tackles this issue by transferring knowledge from a labeled source domain to an unlabeled target domain [30]. A significant line of solutions reduces the domain gap based on the category-level alignment which produces pseudo labels for the target samples, such as metric learning [14,53], adversarial training [12,17,34], and optimal transport [44]. Furthermore, several works [11,36] explore the potential of ViT for the non-trivial UDA task. Recently, CDTrans [45] exploits the cross-attention in ViT for direct domain alignment, buttressed by the crafted pseudo labels for target samples. However, CDTrans has a distinct limitation: as the performance of cross-attention highly depends on the quality of pseudo labels, it becomes less effective when the domain gap becomes large. As shown in Fig. 1, due to the significant gap between the domain *qdr* and the other domains, aligning distributions directly between them performs poorly.

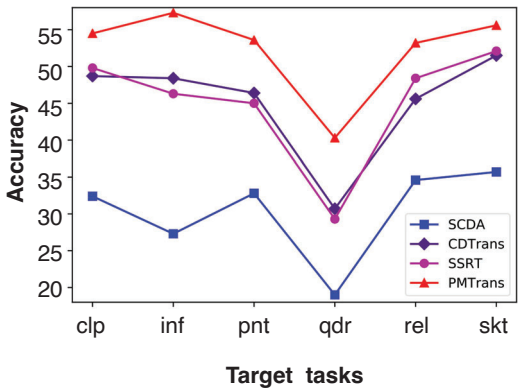

Figure 1. The classification accuracy of our PMTrans surpasses the SoTA methods by **+17.7%** on the most challenging DomainNet dataset. Note that the target tasks treat one domain of DomainNet as the target domain and the others as the source domains.

In this paper, we probe a new problem for UDA: ***how to smoothly bridge the source and target domains by constructing an intermediate domain with an effective ViT-based solution?*** The intuition behind this is that, compared to direct aligning domains, alleviating the domain gap between the intermediate and source/target domain can facili-

*These authors contributed equally to this work.
†Corresponding Author.

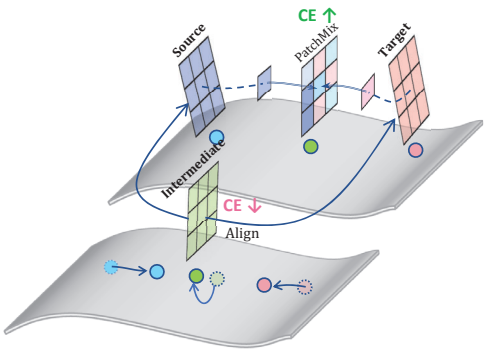

Figure 2. PMTrans builds up the intermediate domain (green patches) via a novel PatchMix module by learning to sample patches from the source (blue patches) and target (pink patches) domains. PatchMix tries to maximize the CE (↑) between the intermediate domain and source/target domain, while the feature extractor and classifier try to minimize it (↓) for aligning domains.

tate domain alignment. Accordingly, we propose a novel and effective method, called **PMTrans** (PatchMix Transformer) to construct the intermediate representations. Overall, PMTrans interprets the process of domain alignment as a min-max cross entropy (CE) game with three players, *i.e.*, the feature extractor, a classifier, and a **PatchMix** module, to find the Nash Equilibria. Importantly, the PatchMix module is proposed to effectively build up the intermediate domain, *i.e.*, probability distribution, by learning to sample patches from both domains with weights generated from a learnable Beta distribution based on the game-theoretical models [1, 3, 28], as shown in Fig. 2. That is, we aim to learn to mix patches from two domains to maximize the CE between the intermediate domain and source/target domain. Moreover, two semi-supervised mixup losses in the feature and label spaces are proposed to minimize the CE. Interestingly, **we conclude that the source and target domains are aligned if mixing the patch representations from two domains is equivalent to mixing the corresponding labels**. Therefore, the domain discrepancy can be measured based on the CE between the mixed patches and mixed labels. Eventually, the three players have no incentive to change their parameters to disturb CE, meaning the source and target domains are well aligned. Unlike existing mixup methods [38, 47, 49], our proposed PatchMix subtly learns to combine the element-wise global and local mixture by mixing patches from the source and target domains for ViT-based UDA. Moreover, we leverage the class activation mapping (CAM) from ViT to allocate the semantic information to re-weight the label of each patch, thus enabling us to obtain more domain-discriminative features.

We conduct experiments on **four** benchmark datasets, including Office-31 [33], Office-Home [40], VisDA-2017 [32], and DomainNet [31]. The results show that the performance of PMTrans significantly surpasses that of the ViT-based [36, 45, 46] and CNN-based SoTA methods [18, 29, 35] by **+3.6%** on Office-Home, **+1.4%** on Office-31, and **+17.7%** on DomainNet (See Fig. 1), respectively.

Our main contributions are four-fold: (**I**) We propose a novel ViT-based UDA framework, PMTrans, to effectively bridge source and target domains by constructing the intermediate domain. (**II**) We propose PatchMix, a novel module to build up the intermediate domain via the game-theoretical models. (**III**) We propose two semi-supervised mixup losses in the feature and label spaces to reduce CE in the min-max CE game. (**IV**) Our PMTrans surpasses the prior methods by a large margin on three benchmark datasets.

2. Related Work

Unsupervised Domain Adaptation. The prevailing UDA methods focus on domain alignment and learning discriminative domain-invariant features via metric learning, domain adversarial training, and optimal transport. Firstly, the metric learning-based methods aim to reduce the domain discrepancy by learning the domain-invariant feature representations using various metrics. For instance, some methods [14, 25, 26, 52] use the maximum mean discrepancy (MMD) loss to measure the divergence between different domains. In addition, the central moment discrepancy (CMD) loss [48] and maximum density divergence (MDD) loss [16] are also proposed to align the feature distributions. Secondly, the domain adversarial training methods learn the domain-invariant representations to encourage samples from different domains to be non-discriminative with respect to the domain labels via an adversarial loss [13, 42, 43]. The third type of approach aims to minimize the cost transported from the source to the target distribution by finding an optimal coupling cost to mitigate the domain shift [6, 7]. Unfortunately, these methods are not robust enough for the noisy pseudo target labels for accurate domain alignment. *Different from these mainstream UDA methods and [2], we interpret the process of UDA as a min-max CE game and find the Nash Equilibria for domain alignment with an intermediate domain and a pure ViT-based solution.*

Mixup. It is an effective data augmentation technique to prevent models from over-fitting by linearly interpolating two input data. Mixup types can be categorized into global mixup (e.g., Mixup [49] and Manifold-Mixup [41]) and local mixup (CutMix [47], saliency-CutMix [38], TransMix [5], and Tokenmix [22]). In CNN-based UDA tasks, several works [29, 42, 43] also use the mixup technique by linearly mixing the source and target domain data. *In comparison, we unify the global and local mixup in our PMTrans framework by learning to form a mixed patch from the source/target patch as the input to ViT. We learn the hyperparameters of the mixup ratio for each patch, which is the **first** attempt to interpolate patches based on the distribution estimation. Accordingly, we propose PatchMix which effectively builds up the intermediate domain by sampling patches from both domains based on the game-theoretical models.*

Transformer. Vision Transformer(ViT) [39] has recently been introduced to tackle the challenges in various vision tasks [4, 23]. Several works have leveraged ViT for the non-trivial UDA task. TVT [46] proposes an adaptation module to capture domain data's transferable and discriminative features. SSRT [36] proposes a framework with a transformer backbone and a safe self-refinement strategy to handle the issue in case of a large domain gap. More recently, CDTrans [45] proposes a two-step framework that utilizes the cross-attention in ViT for direct feature alignment and pre-generated pseudo labels for the target samples. *Differently, we probe to construct an intermediate domain to bridge the source and target domains for better domain alignment. Our PMTrans effectively interprets the process of domain alignment as a min-max CE game, leading to a significant UDA performance enhancement.*

3. Methodology

In UDA, denote a labeled source set $\mathcal{D}_s = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_s}$ with i-th sample $\boldsymbol{x}_i^s$ and its corresponding one-hot label $\boldsymbol{y}_i^s$ and an unlabeled target set $\mathcal{D}_t = \left\{\boldsymbol{x}_j^t\right\}_{j=1}^{n_t}$ with j-th sample $\boldsymbol{x}_j^t$, n_s and n_t as the size of samples in the source and target domains, respectively. Note that the data in two domains are sampled from two different distributions, and we assume that the two domains share the same label space. Our goal is to address the significant domain divergence issue and smoothly transfer the knowledge from the source domain to the target domain. Firstly, we define and introduce PatchMix, and interpret the process of UDA as a min-max CE game. Secondly, we describe the proposed PMTrans which smoothly aligns the source and target domains by constructing an intermediate domain via a three-player game.

3.1. PMTrans: Theoretical Analysis

3.1.1 PatchMix

Definition 1 *(PatchMix): Let $\mathcal{P}_\lambda$ be a linear interpolation operation on two pairs of randomly drawn samples $(\boldsymbol{x}^s, \boldsymbol{y}^s)$ and $(\boldsymbol{x}^t, \boldsymbol{y}^t)$. Then with $\lambda_k \sim \mathrm{Beta}(\beta, \gamma)$, it interpolates the k-th source patch $\boldsymbol{x}_k^s$ and target patch $\boldsymbol{x}_k^t$ to reconstruct a mixed representation with n patches.*

$$\begin{aligned} &\boldsymbol{x}^i = \mathcal{P}_\lambda(\boldsymbol{x}^s, \boldsymbol{x}^t),\ \boldsymbol{x}_k^i = \lambda_k \odot \boldsymbol{x}_k^s + (1-\lambda_k) \odot \boldsymbol{x}_k^t, \\ &\boldsymbol{y}^i = \mathcal{P}_\lambda(\boldsymbol{y}^s, \boldsymbol{y}^t) = \frac{(\sum_{k=1}^n \lambda_k)\boldsymbol{y}^s + (\sum_{k=1}^n (1-\lambda_k))\boldsymbol{y}^t}{n}. \end{aligned} \tag{1}$$

where $\boldsymbol{x}_k^i$ is the k-th patch of $\boldsymbol{x}^i$, and $\odot$ denotes multiplication. In Definition. 1, each image $\boldsymbol{x}^i$ of the intermediate domain composes the sampled patches $\boldsymbol{x}_k$ from the source/target domain. Here, $\lambda_k \in [0, 1]$ is the random mixing proportion that denotes the patch-level sampling weights. Furthermore, we calculate the image-level importance by aggregating patch weights $\sum_{k=1}^n (1-\lambda_k)$, which is utilized to interpolate their labels. As a result, we mix both samples $(\boldsymbol{x}^s, \boldsymbol{y}^s)$ and $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ to construct a new intermediate domain $\mathcal{D}_i = \left\{\left(\boldsymbol{x}_l^i, \boldsymbol{y}_l^i\right)\right\}_{l=1}^{n_i}$. To align the source and target domains, we need to evaluate the gap numerically. In detail, let P_S and P_T be the empirical distributions defined by $\mathcal{D}_s$ and $\mathcal{D}_t$, respectively. The domain divergence between source and target domains can be measured as

$$\begin{aligned} D(P_S, P_T) = \inf_{f\in\mathcal{F}, c\in\mathcal{C}, \mathcal{P}_\lambda\in\mathcal{P}} \mathbb{E}_{(\boldsymbol{x}^s,\boldsymbol{y}^s),(\boldsymbol{x}^t,\boldsymbol{y}^t)} \\ \ell\left(c\left(\mathcal{P}_\lambda\left(f(\boldsymbol{x}^s), f\left(\boldsymbol{x}^t\right)\right)\right), \mathcal{P}_\lambda\left(\boldsymbol{y}^s, \boldsymbol{y}^t\right)\right), \end{aligned} \tag{2}$$

where $\mathcal{F}$ denotes a set of encoding functions *i.e.*, the feature extractor and $\mathcal{C}$ denotes a set of decoding functions *i.e.* the classifier. Let $\mathcal{P}$ be the set of functions to generate the mixup ratio for building the intermediate domain. Then we can reformulate Eq. 2 as

$$\begin{aligned} D\left(P_S, P_T\right) = \inf_{\boldsymbol{h}_1^s,\ldots,\boldsymbol{h}_{n_s}^s\in\mathcal{H}^s, \boldsymbol{h}_1^t,\ldots,\boldsymbol{h}_{n_t}^t\in\mathcal{H}^t} \frac{1}{n_s \times n_t} \sum_i^{n_s}\sum_j^{n_t} \\ \left\{\inf_{c\in\mathcal{C}} \int_0^1 \ell\left(f\left(\mathcal{P}_\lambda\left(\boldsymbol{h}_i^s, \boldsymbol{h}_j^t\right)\right), \mathcal{P}_\lambda\left(\boldsymbol{y}_i^s, \boldsymbol{y}_j^t\right)\right) p(\lambda)\mathrm{d}\lambda\right\}, \end{aligned} \tag{3}$$

where ℓ is CE loss, $\boldsymbol{h}_i^s = f(\boldsymbol{s}_i^s)$ and $\boldsymbol{h}_j^t = f(\boldsymbol{x}_j^t)$. Note $\mathcal{H}^s$ and $\mathcal{H}^t$ denote the representation spaces with dimensionality $\dim(\mathcal{H})$ for source and target domains, respectively. Let $f^\star \in \mathcal{F}, c^\star \in \mathcal{C},$ and $\mathcal{P}_\lambda{}^\star \in \mathcal{P}$ be minimizers of Eq. 2.

Theorem 1 *(Domain Distribution Alignment with PatchMix): Let $d \in \mathbb{N}$ to represent the number of classes contained in three sets $\mathcal{D}_s$, $\mathcal{D}_t$, and $\mathcal{D}_i$. If $\dim(\mathcal{H}) \geq d-1$, $\mathcal{P}_\lambda{}'\ell(c^\star(f^\star(\boldsymbol{x}_i)), \boldsymbol{y}^s) + (1-\mathcal{P}_\lambda{}')\ell(c^\star(f^\star(\boldsymbol{x}_i)), \boldsymbol{y}^t) = 0$, then $D\left(P_S, P_T\right) = 0$ and the corresponding minimizer $c^\star$ is a linear function from $\mathcal{H}$ to $\mathbb{R}^d$. Denote the scaled mixup ratio sampled from a learnable Beta distribution as $\mathcal{P}_\lambda'$.*

Theorem. 1 indicates that **the source and target domains are aligned if mixing the patches from two domains is equivalent to mixing the corresponding labels**. Therefore, minimizing the CE between the mixed patches and mixed labels can effectively facilitate domain alignment. *For the proof of Theorem. 1, refer to the suppl. material.*

3.1.2 A Min-Max CE Game

We interpret UDA as a min-max CE game among three players, namely the feature extractor ($\mathcal{F}$), classifier ($\mathcal{C}$), and PatchMix module ($\mathcal{P}$), as shown in Fig. 3. To specify each player's role, we define $\boldsymbol{\omega}_\mathcal{F} \in \Omega_\mathcal{F}$, $\boldsymbol{\omega}_\mathcal{C} \in \Omega_\mathcal{C}$, and $\boldsymbol{\omega}_\mathcal{P} \in \Omega_\mathcal{P}$ as the parameters of $\mathcal{F}$, $\mathcal{C}$, and $\mathcal{P}$, respectively. The joint domain is defined as $\Omega = \Omega_\mathcal{F} \times \Omega_\mathcal{C} \times \Omega_\mathcal{P}$ and their joint parameter set is defined as $\boldsymbol{\omega} = \{\boldsymbol{\omega}_\mathcal{F}, \boldsymbol{\omega}_\mathcal{C}, \boldsymbol{\omega}_\mathcal{P}\}$. Then we use the subscript $_{-m}$ to denote all other parameters/players except m, *e.g.*, $\boldsymbol{\omega}_{-\mathcal{C}} = \{\boldsymbol{\omega}_\mathcal{F}, \boldsymbol{\omega}_\mathcal{P}\}$. In our game, m-th player is endowed with a cost function J_m and strives to reduce its cost, which contributes to the change of CE. Each player'

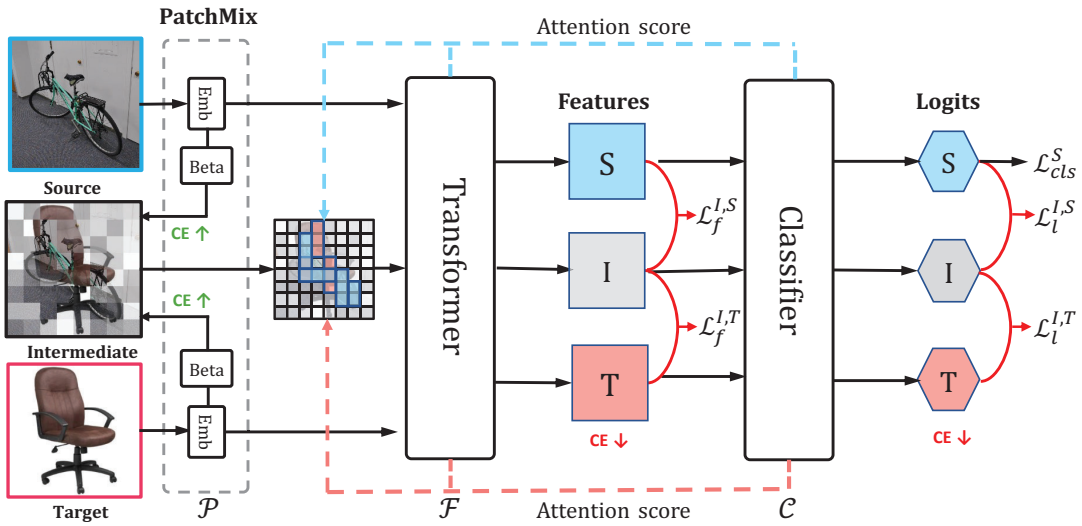

Figure 3. Overview of the proposed PMTrans framework. It consists of three players: the PatchMix module empowered by a patch embedding (**Emb**) layer and a learnable Beta distribution (**Beta**), ViT encoder, and classifier.

cost function J_m is represented as

$$\begin{aligned} J_{\mathcal{F}}\left(\boldsymbol{\omega}_{\mathcal{F}}, \boldsymbol{\omega}_{-\mathcal{F}}\right) &:= \mathcal{L}_{cls}^{S}(\boldsymbol{\omega}_{\mathcal{F}}, \boldsymbol{\omega}_{\mathcal{C}}) + \alpha \mathrm{CE}_{s,i,t}(\boldsymbol{\omega}), \\ J_{\mathcal{C}}\left(\boldsymbol{\omega}_{\mathcal{C}}, \boldsymbol{\omega}_{-\mathcal{C}}\right) &:= \mathcal{L}_{cls}^{S}(\boldsymbol{\omega}_{\mathcal{F}}, \boldsymbol{\omega}_{\mathcal{C}}) + \alpha \mathrm{CE}_{s,i,t}(\boldsymbol{\omega}), \\ J_{\mathcal{P}}\left(\boldsymbol{\omega}_{\mathcal{P}}, \boldsymbol{\omega}_{-\mathcal{P}}\right) &:= -\alpha \mathrm{CE}_{s,i,t}(\boldsymbol{\omega}), \end{aligned} \tag{4}$$

where α is the trade-off parameter, ℓ is the supervised classification loss for the source domain, and $\mathrm{CE}_{s,i,t}(\boldsymbol{\omega})$ is the discrepancy between the intermediate domain and the source/target domain. The definitions of $\mathcal{L}_{cls}^{S}(\boldsymbol{\omega}_{\mathcal{F}}, \boldsymbol{\omega}_{\mathcal{C}})$ and $\mathrm{CE}_{s,i,t}(\boldsymbol{\omega})$ are shown in Sec. 3.2. As illustrated in Eq. 4, the game is essentially a min-max process, *i.e.*, a competition for the player $\mathcal{P}$ against both players $\mathcal{F}$ and $\mathcal{C}$. Specifically, as depicted in Fig. 3, $\mathcal{P}$ strives to diverge while $\mathcal{F}$ and $\mathcal{C}$ try to align domain distributions, which is a min-max process on CE. In this min-max CE game, each player behaves selfishly to reduce its cost function, and this competition will possibly end with a situation where no one has anything to gain by changing only one's strategy. This situation is called Nash Equilibrium (NE) in game theory.

Definition 2 *(Nash Equilibrium): The equilibrium states each player's strategy is the best response to other players. And a point $\omega^* \in \Omega$ is Nash Equilibrium if*

$$\forall \boldsymbol{\omega}_m \in \Omega_i, \forall m \in \{\mathcal{F}, \mathcal{C}, \mathcal{P}\}, s.t. J_m(\boldsymbol{\omega}_m^*, \boldsymbol{\omega}_{-m}^*) \leq J_m(\boldsymbol{\omega}_m, \boldsymbol{\omega}_{-m}^*).$$

Intuitively, in our case, NE means that no player has the incentive to change its own parameters, as there is no additional pay-off.

3.2. The Proposed Framework

Overview. Fig. 3 illustrates the framework of our proposed PMTrans, which consists of a ViT encoder, a classifier, and a PatchMix module. PatchMix module is utilized to maximize the CE between the intermediate domain and source/target domain, conversely, two semi-supervised mixup losses in the feature and label spaces are proposed to minimize CE. Finally, a three-player game containing feature extractor, classifier, and PatchMix module, minimizes and maximizes the CE for aligning distributions between the source and target domains.

PatchMix. As shown in Fig. 3, the PatchMix module is proposed to construct the intermediate domain, buttressed by Definition. 1. In detail, the patch embedding layer in PatchMix transforms input images from source/target domains into patches. And the ViT encoder aims to extract features from the patch sequences. The classifier maps the outputs of ViT encoder to make predictions, each of which is exploited to select the feature map to re-weight the patch sequences. The PatchMix with a learnable Beta distribution aims to maximize the CE between the intermediate and source/target domain, and is presented as follows.

When exploiting PatchMix to construct the intermediate domain, it is worth noting that not all patches have equal contributions for the label assignment. As Chen *et al.* [5] observed, the mixed image has no valid objects due to the random process while there is still a response in the label space. To remedy this issue, we re-weight $\mathcal{P}_\lambda(\boldsymbol{y}^s, \boldsymbol{y}^t)$ in Definition. 1 with the normalized attention score a_k. For the implementation details of attention scores, refer to *the suppl. material*. The re-scaled $\mathcal{P}_\lambda(\boldsymbol{y}^s, \boldsymbol{y}^t)$ is defined as

$$\mathcal{P}_\lambda(\boldsymbol{y}^s, \boldsymbol{y}^t) = \lambda^s \boldsymbol{y}^s + \lambda^t \boldsymbol{y}^t,$$

where

$$\lambda^s = \frac{\sum_{k=1}^{n} \lambda_k a_k^s}{\sum_{k=1}^{n} \lambda_k a_k^s + \sum_{k=1}^{n} (1-\lambda_k) a_k^t},$$
$$\lambda^t = \frac{\sum_{k=1}^{n} (1-\lambda_k) a_k^t}{\sum_{k=1}^{n} \lambda_k a_k^s + \sum_{k=1}^{n} (1-\lambda_k) a_k^t}.$$

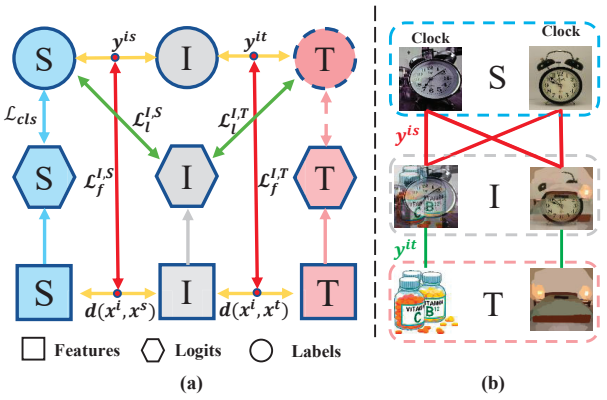

Figure 4. (a) The illustration of two proposed semi-supervised losses. (b) Label similarity $\boldsymbol{y}^{is}$ and $\boldsymbol{y}^{it}$. Better viewed in color.

Semi-supervised mixup loss. As PatchMix tries to maximize the CE between the intermediate domain and source/target domain, we now need to find a way to minimize the CE in the game. Intuitively, we propose two semi-supervised mixup losses in the feature and label spaces to minimize the discrepancy between features of mixing patches and corresponding mixing labels based on Theorem. 1. The objective of the proposed PMTrans is show in Fig. 4 (a) and consists of a classification loss of source data and two semi-supervised mixup losses.

1) Label space: As introduced in Theorem. 1, we apply a supervised mixup loss in the label space to measure the domain divergence based on the CE loss between the mixing logits and corresponding mixing labels (See green arrow in Fig. 4 (a)).

$$\mathcal{L}_l^{I,S}(\boldsymbol{\omega}) = \mathbb{E}_{(\boldsymbol{x}^i,\boldsymbol{y}^i)\sim D^i} \lambda^s \ell\left(\mathcal{C}\left(\mathcal{F}\left(\boldsymbol{x}^i\right)\right), \boldsymbol{y}^s\right),$$
$$\mathcal{L}_l^{I,T}(\boldsymbol{\omega}) = \mathbb{E}_{(\boldsymbol{x}^i,\boldsymbol{y}^i)\sim D^i} \lambda^t \ell\left(\mathcal{C}\left(\mathcal{F}\left(\boldsymbol{x}^i\right)\right), \hat{\boldsymbol{y}^t}\right),$$

where $\hat{\boldsymbol{y}^t}$ is the pseudo label for target data. For convenience, we utilize the method, commonly used in [20, 21], to generate pseudo labels $\hat{\boldsymbol{y}^t}$ for samples via k-means cluster.

2) Feature space: Nonetheless, the supervised loss alone in the label space is not sufficient to diminish the domain divergence due to the less reliable pseudo labels of the target data. Therefore, we further **propose to minimize the discrepancy between the similarity of the features and the similarity of labels in the feature space** for aligning the intermediate and source/target domain without the supervised information of the target domain. The experimental results in Tab. 5 validate its effectiveness.

Specifically, we first compute the cosine similarity between the intermediate domain and source/target domain in the feature space. The feature similarity is defined as

$$d(\boldsymbol{x}^i, \boldsymbol{x}^s) = cos(\mathcal{F}\left(\boldsymbol{x}^i\right), \mathcal{F}\left(\boldsymbol{x}^s\right)),$$

where cos denotes the cosine similarity. As shown in Fig. 4(b), for the source domain, we exploit the ground-truth to calculate the label similarity, $y^{is} = y^s(y^s)^\intercal$, as a binary matrix to represent whether samples share the same labels. For example, the intermediate image is constructed by sampling patches from the source image, *e.g.*, *clock*; therefore, the label similarity is set as 1 if it is calculated between the intermediate image and the source class *'clock'*, otherwise, it is set as 0. Then, we utilize the CE to measure the domain discrepancy based on the difference between the feature similarity and label similarity. The supervised mixup loss in the feature space (See red arrow in Fig. 4 (a)) is formulated as

$$\mathcal{L}_f^{I,S}(\boldsymbol{\omega}_\mathcal{F}, \boldsymbol{\omega}_\mathcal{P}) = \mathbb{E}_{(\boldsymbol{x}^i,\boldsymbol{y}^i)\sim D^i} \lambda^s \ell\left(d(\boldsymbol{x}^i, \boldsymbol{x}^s), \boldsymbol{y}^{is}\right).$$

Moreover, for the intermediate and target domains, due to lack of supervision, we utilize identity matrix y^{it} as the label similarity. For example, in Fig. 4 (b), as the intermediate image is built by sampling patches from the target image, *e.g.*, bottle; therefore, the label similarity between the intermediate image and the corresponding target image is set as 1 and vice versa. To measure the divergence between the intermediate and target domains in the feature space, we propose an unsupervised mixup loss as

$$\mathcal{L}_f^{I,T}(\boldsymbol{\omega}_\mathcal{F}, \boldsymbol{\omega}_\mathcal{P}) = \mathbb{E}_{(\boldsymbol{x}^i,\boldsymbol{y}^i)\sim D^i} \lambda^t \ell\left(d(\boldsymbol{x}^i, \boldsymbol{x}^t), \boldsymbol{y}^{it}\right),$$

Finally, the two semi-supervised mixup losses in the feature and label spaces are formulated as

$$\mathcal{L}_f(\boldsymbol{\omega}_\mathcal{F}, \boldsymbol{\omega}_\mathcal{P}) = \mathcal{L}_f^{I,S}(\boldsymbol{\omega}_\mathcal{F}, \boldsymbol{\omega}_\mathcal{P}) + \mathcal{L}_f^{I,T}(\boldsymbol{\omega}_\mathcal{F}, \boldsymbol{\omega}_\mathcal{P}),$$
$$\mathcal{L}_l(\boldsymbol{\omega}) = \mathcal{L}_l^{I,S}(\boldsymbol{\omega}) + \mathcal{L}_l^{I,T}(\boldsymbol{\omega}).$$

Moreover, the classification loss is applied to the labeled source domain data (See blue arrow in Fig. 4 (a)) and is formulated as

$$\mathcal{L}_{cls}^S(\boldsymbol{\omega}_\mathcal{F}, \boldsymbol{\omega}_\mathcal{C}) = \mathbb{E}_{(\boldsymbol{x}^s,\boldsymbol{y}^s)\sim D^s} \ell\left(\mathcal{C}\left(\mathcal{F}\left(\boldsymbol{x}^s\right)\right), \boldsymbol{y}^s\right).$$

A Three-Player Game. Finally, the min-max CE game aims to align distributions in the feature and label spaces. The total CE between the intermediate domain and source/target domain is

$$\mathrm{CE}_{s,i,t}(\boldsymbol{\omega}) = \mathcal{L}_f(\boldsymbol{\omega}_\mathcal{F}, \boldsymbol{\omega}_\mathcal{P}) + \mathcal{L}_l(\boldsymbol{\omega}).$$

We adopt the *random* mixup-ratio from a learnable Beta distribution in our PatchMix module to maximize the CE between the intermediate domain and source/target domain. Moreover, the feature extractor and classifier have the same objective to minimize the CE between the intermediate domain and source/target domain. Therefore, the total objective of PMTrans is achieved by reformulating Eq. 4 as

$$J\left(\boldsymbol{\omega}\right) := \mathcal{L}_{cls}^S(\boldsymbol{\omega}_\mathcal{F}, \boldsymbol{\omega}_\mathcal{C}) + \alpha \mathrm{CE}_{s,i,t}(\boldsymbol{\omega}),$$

where α is trade-off parameter. After optimizing the objective, the PatchMix module with the ideal Beta distribution will not maximize the CE anymore. Meanwhile, the feature extractor and classifier have no incentive to change their parameters to minimize the CE. Finally, the discrepancy between the intermediate domain and source/target domain is nearly zero, further indicating that the source and target domains are well aligned.

Method		A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet-50	ResNet	44.9	66.3	74.3	51.8	61.9	63.6	52.4	39.1	71.2	63.8	45.9	77.2	59.4
MCD		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
MDD		54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
BNM		56.7	77.5	81.0	67.3	76.3	77.1	65.3	55.1	82.0	73.6	57.0	84.3	71.1
FixBi		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
TVT	ViT	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
Deit-based		61.8	79.5	84.3	75.4	78.8	81.2	72.8	55.7	84.4	78.3	59.3	86.0	74.8
CDTrans-Deit		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
PMTrans-Deit		71.8	87.3	88.3	83.0	87.7	87.8	78.5	67.4	89.3	81.7	70.7	92.0	82.1
ViT-based		67.0	85.7	88.1	80.1	84.1	86.7	79.5	67.0	89.4	83.6	70.2	91.2	81.1
SSRT-ViT		75.2	89.0	91.1	85.1	88.3	89.9	85.0	74.2	91.2	85.7	78.6	91.8	85.4
PMTrans-ViT		81.2	91.6	92.4	**88.9**	91.6	93.0	**88.5**	80.0	**93.4**	**89.5**	**82.4**	94.5	88.9
Swin-based	Swin	72.7	87.1	90.6	84.3	87.3	89.3	80.6	68.6	90.3	84.8	69.4	91.3	83.6
PMTrans-Swin		**81.3**	**92.9**	**92.8**	88.4	**93.4**	**93.2**	87.9	**80.4**	93.0	89.0	80.9	**94.8**	**89.0**

Table 1. Comparison with SoTA methods on Office-Home. The best performance is marked as **bold**.

Method		A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet-50	ResNet	68.9	68.4	62.5	96.7	60.7	99.3	76.1
BNM		91.5	98.5	**100.0**	90.3	70.9	71.6	87.1
MDD		94.5	98.4	**100.0**	93.5	74.6	72.2	88.9
SCDA		94.2	98.7	99.8	95.2	75.7	76.2	90.0
FixBi		96.1	99.3	**100.0**	95.0	78.7	79.4	91.4
TVT	ViT	96.4	99.4	**100.0**	96.4	84.9	86.0	93.9
Deit-based		89.2	98.9	**100.0**	88.7	80.1	79.8	89.5
CDTrans-Deit		96.7	99.0	**100.0**	97.0	81.1	81.9	92.6
PMTrans-Deit		99.0	99.4	**100.0**	96.5	81.4	82.1	93.1
ViT-based		91.2	99.2	**100.0**	90.4	81.1	80.6	91.1
SSRT-ViT		97.7	99.2	**100.0**	98.6	83.5	82.2	93.5
PMTrans-ViT		99.1	**99.6**	**100.0**	99.4	85.7	86.3	95.0
Swin-based	Swin	97.0	99.2	**100.0**	95.8	82.4	81.8	92.7
PMTrans-Swin		**99.5**	99.4	**100.0**	**99.8**	**86.7**	**86.5**	**95.3**

Table 2. Comparison with SoTA methods on Office-31. The best performance is marked as **bold**.

4. Experiments

4.1. Datasets and implementation

Datasets. To evaluate the proposed method, we conduct experiments on four popular UDA benchmarks, including Office-Home [40], Office-31 [33], VisDA-2017 [32], and DomainNet [31]. *The details of the datasets and transfer tasks on these datasets can be found in the suppl. material.*

Implementation. In all experiments, we use the Swin-based transformer [24] pre-trained on ImageNet [9] as the backbone for our PMTrans. The base learning rate is $5e^{-6}$ with a batch size of 32, and we train models by 50 epochs. For VisDA-2017, we use a lower learning rate $1e^{-6}$. We adopt AdamW [27] with a momentum of 0.9, and a weight decay of 0.05 as the optimizer. Furthermore, for fine-tuning purposes, we set the classifier (MLP) with a higher learning rate $1e^{-5}$ for our main tasks and learn the trade-off parameter adaptively. For a fair comparison with prior works, we also conduct experiments with the same backbone **Deit-based** [37] as CDTrans [45], and **ViT-based** [11] as SSRT [36] on Office-31, Office-Home, and VisDA-2017. These two studies are trained for 60 and 100 epochs separately.

4.2. Results

We compare PMTrans with the SoTA methods, including ResNet-based and ViT-based methods. The ResNet-based methods are FixBi [29], MCD [34], SWD [15], SCDA [19], BNM [8], and MDD [51]. The ViT-based methods are SSRT [36], CDTrans [45], and TVT [46].

For the ResNet-based methods, we utilize ResNet-50 as the backbone for the Office-Home, Office-31, and DomainNet datasets, and we adopt ResNet-101 for VisDA-2017 dataset. Note that each backbone is trained with the source data only and then tested with the target data.

Results on Office-Home. Tab. 1 shows the quantitative results of methods using different backbones. As expected, our PMTrans framework achieves noticeable performance gains and surpasses TVT, SSRT, and CDTrans by a large margin. Importantly, our PMTrans achieves an improvement more than **5.4%** accuracy over the Swin backbone and yields **89.0%** accuracy. Interestingly, our proposed PMTrans can decrease the domain divergence effectively with Deit-based and ViT-based backbones. The results indicate that our method can obtain more robust transferable representations than the CNN-based and ViT-based methods.

Results on Office-31. Tab. 2 shows the quantitative comparison with the CNN-based and ViT-based methods. Overall, our PMTrans achieves the best performance on each task with **95.3%** accuracy and outperforms the SoTA methods with identical backbones. Numerically, PMTrans noticeably surpasses the SoTA methods with an increase of **+1.4%** accuracy over TVT, **+2.7%** accuracy over CDTrans, and **+1.8%** accuracy over SSRT, respectively.

Results on VisDA-2017. As shown in Tab. 3, our PMTrans achieves **88.0%** accuracy and outperforms the baseline by **11.2%**. In particular, for the 'hard' categories, such as "person", our method consistently achieves a much higher performance boost from **29.0%** to **70.3%**. These improvements indicate that our method shows an excellent generalization capability and achieves comparable performance (**88.0%**)

Method		plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg
ResNet-50	ResNet	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
BNM	ResNet	89.6	61.5	76.9	55.0	89.3	69.1	81.3	65.5	90.0	47.3	89.1	30.1	70.4
MCD	ResNet	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
SWD	ResNet	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
FixBi	ResNet	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
TVT	ViT	82.9	85.6	77.5	60.5	93.6	98.2	89.4	76.4	93.6	92.0	91.7	55.7	83.1
Deit-based	ViT	98.2	73.0	82.5	62.0	97.3	63.5	96.5	29.8	68.7	86.7	96.7	23.6	73.2
CDTrans-Deit	ViT	97.1	90.5	82.4	77.5	96.6	96.1	93.6	**88.6**	**97.9**	86.9	90.3	**62.8**	88.4
PMTrans-Deit	ViT	98.2	92.2	88.1	77.0	97.4	95.8	94.0	72.1	97.1	95.2	94.6	51.0	87.7
ViT-based	ViT	99.1	60.7	70.1	82.7	96.5	73.1	97.1	19.7	64.5	94.7	97.2	15.4	72.6
SSRT-ViT	ViT	98.9	87.6	**89.1**	**84.8**	98.3	**98.7**	96.3	81.1	94.8	97.9	94.5	43.1	**88.8**
PMTrans-ViT	ViT	98.9	**93.7**	84.5	73.3	**99.0**	98.0	96.2	67.8	94.2	**98.4**	96.6	49.0	87.5
Swin-based	Swin	99.3	63.4	85.9	68.9	95.1	79.6	**97.1**	29.0	81.4	94.2	**97.7**	29.6	76.8
PMTrans-Swin	Swin	**99.4**	88.3	88.1	78.9	98.8	98.3	95.8	70.3	94.6	98.3	96.3	48.5	88.0

Table 3. Comparison with SoTA methods on VisDA-2017. The best performance is marked as **bold**.

MCD	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	15.4	25.5	3.3	44.6	31.2	24.0
inf	24.1	-	24.0	1.6	35.2	19.7	20.9
pnt	31.1	14.8	-	1.7	48.1	22.8	23.7
qdr	8.5	2.1	4.6	-	7.9	7.1	6.0
rel	39.4	17.8	41.2	1.5	-	25.2	25.0
skt	37.3	12.6	27.2	4.1	34.5	-	23.1
Avg	28.1	12.5	24.5	2.4	34.1	21.2	20.5

SWD	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	14.7	31.9	10.1	45.3	36.5	27.7
inf	22.9	-	24.2	2.5	33.2	21.3	20.0
pnt	33.6	15.3	-	4.4	46.1	30.7	26.0
qdr	15.5	2.2	6.4	-	11.1	10.2	9.1
real	41.2	18.1	44.2	4.6	-	31.6	27.9
skt	44.2	15.2	37.3	10.3	44.7	-	30.3
Avg	31.5	13.1	28.8	6.4	36.1	26.1	23.6

BNM	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	12.1	33.1	6.2	50.8	40.2	28.5
inf	26.6	-	28.5	2.4	38.5	18.1	22.8
pnt	39.9	12.2	-	3.4	54.5	36.2	29.2
qdr	17.8	1.0	3.6	-	9.2	8.3	8.0
rel	48.6	13.2	49.7	3.6	-	33.9	29.8
skt	54.9	12.8	42.3	5.4	51.3	-	33.3
Avg	37.6	10.3	31.4	4.2	40.9	27.3	25.3

CGDM	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	16.9	35.3	10.8	53.5	36.9	30.7
inf	27.8	-	28.2	4.4	48.2	22.5	26.2
pnt	37.7	14.5	-	4.6	59.4	33.5	30.0
qdr	14.9	1.5	6.2	-	10.9	10.2	8.7
rel	49.4	20.8	47.2	4.8	-	38.2	32.0
skt	50.1	16.5	43.7	11.1	55.6	-	35.4
Avg	36.0	14.0	32.1	7.1	45.5	28.3	27.2

MDD	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	20.5	40.7	6.2	52.5	42.1	32.4
inf	33.0	-	33.8	2.6	46.2	24.5	28.0
pnt	43.7	20.4	-	2.8	51.2	41.7	32.0
qdr	18.4	3.0	8.1	-	12.9	11.8	10.8
rel	52.8	21.6	47.8	4.2	-	41.2	33.5
skt	54.3	17.5	43.1	5.7	54.2	-	35.0
Avg	40.4	16.6	34.7	4.3	43.4	32.3	28.6

SCDA	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	18.6	39.3	5.1	55.0	44.1	32.4
inf	29.6	-	34.0	1.4	46.3	25.4	27.3
pnt	44.1	19.0	-	2.6	56.2	42.0	32.8
qdr	30.0	4.9	15.0	-	25.4	19.8	19.0
rel	54.0	22.5	51.9	2.3	-	42.5	34.6
skt	55.6	18.5	44.7	6.4	53.2	-	35.7
Avg	42.6	16.7	37.0	3.6	47.2	34.8	30.3

CDTrans	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	29.4	57.2	26.0	72.6	58.1	48.7
inf	57.0	-	54.4	12.8	69.5	48.4	48.4
pnt	62.9	27.4	-	15.8	72.1	53.9	46.4
qdr	44.6	8.9	29.0	-	42.6	28.5	30.7
rel	66.2	31.0	61.5	16.2	-	52.9	45.6
skt	69.0	29.6	59.0	27.2	72.5	-	51.5
Avg	59.9	25.3	52.2	19.6	65.9	48.4	45.2

SSRT	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	33.8	60.2	19.4	75.8	59.8	49.8
inf	55.5	-	54.0	9.0	68.2	44.7	46.3
pnt	61.7	28.5	-	8.4	71.4	55.2	45.0
qdr	42.5	8.8	24.2	-	37.6	33.6	29.3
rel	69.9	37.1	66.0	10.1	-	58.9	48.4
skt	70.6	32.8	62.2	21.7	73.2	-	52.1
Avg	60.0	28.2	53.3	13.7	65.3	50.4	45.2

PMTrans	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	34.2	62.7	32.5	79.3	63.7	54.5
inf	67.4	-	61.1	22.2	78.0	57.6	57.3
pnt	69.7	33.5	-	23.9	79.8	61.2	53.6
qdr	54.6	17.4	38.9	-	49.5	41.0	40.3
rel	74.1	35.3	70.0	25.4	-	61.1	53.2
skt	73.8	33.0	62.6	30.9	77.5	-	55.6
Avg	67.9	30.7	59.1	27.0	72.8	56.9	**62.9**

Table 4. Comparison with SoTA methods on DomainNet. The best performance is marked as **bold**.

with the SoTA methods (**88.7%**). PMTrans also surpasses the SoTA methods on several sub-categories, such as "horse" and "sktbrd". In particular, it is shown that the SoTA methods, *e.g.*, CDTrans and SSRT, achieve better results on this dataset. The reason is that CDTrans and SSRT are trained with a batch size of 64 while PMTrans's batch size is 32. It indicates that when the input size is much bigger, the input can represent the data distributions better. *A detailed ablation study for this issue can be found in the suppl. material.*.

Results on DomainNet. PMTrans achieves a very high average accuracy on the most challenging DomainNet dataset, as shown in Tab. 4. Overall, our proposed PMTrans outperforms the SoTA methods by **+17.7%** accuracy. Incredibly, PMTrans surpasses the SoTA methods in all the **30 subtasks**, which demonstrates the strong ability of PMTrans to alleviate the large domain gap. Moreover, transferring knowledge is much more difficult when the domain gap becomes significant. *When taking more challenging qdr as the target domain while others as the source domain, our PMTrans achieves an average accuracy of **27.0%**, while ViT-based SSRT and CDTrans only achieve an average accuracy of **13.7%** and **19.6%**, respectively.* The comparisons on DomainNet dataset demonstrate that our PMTrans yields the best generalization ability for the challenging UDA problem.

4.3. Ablation Study

Semi-supervised mixup loss. As shown in Tab. 5, Swin with the semi-supervised mixup loss in the feature and label spaces outperforms the counterpart built on Swin with only source training by **+1.0%** and **+4.3%** on Office-Home dataset, respectively. The results indicate the effectiveness of the semi-supervised mixup loss for diminishing the domain discrepancy. Moreover, we observe that the CE loss yields better performance on the label space than that on the feature space. The reason is that the CE loss on the label space utilizing the class information performs better than on the feature space without the class information.

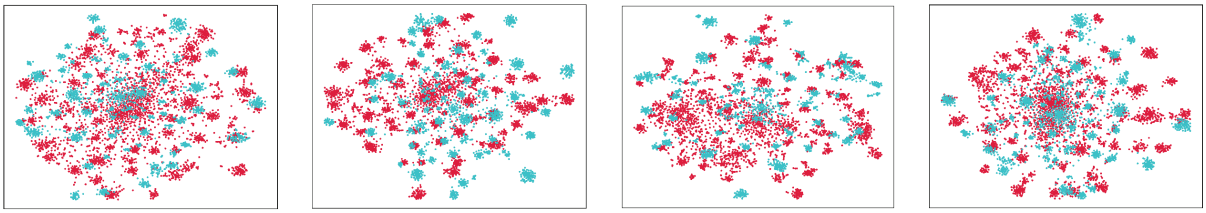

Figure 5. t-SNE visualizations for task A→C on the Office-Home dataset. Source and target instances are shown in blue and red, respectively.

$\mathcal{L}_{cls}^{S}$	$\mathcal{L}_f$	$\mathcal{L}_l$	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
✓			72.7	87.1	90.6	84.3	87.3	89.3	80.6	68.6	90.3	84.8	69.4	91.3	83.6
✓	✓		73.9	87.5	91.0	85.3	87.9	89.9	82.8	72.1	91.2	86.3	74.1	92.4	84.6
✓		✓	79.2	91.8	92.3	88.0	92.6	**93.0**	87.1	77.8	92.5	88.2	78.4	93.9	87.9
✓	✓	✓	**81.3**	**92.9**	**92.8**	**88.4**	**93.4**	**93.2**	**87.9**	**80.4**	**93.0**	**89.0**	**80.9**	**94.8**	**89.0**

Table 5. Effect of semi-supervised loss. The best performance is marked as **bold**.

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
Beta(1,1)	79.9	92.0	92.3	88.6	92.6	92.4	86.9	79.0	92.4	88.2	79.3	94.0	88.1
Beta(2,2)	79.9	92.1	92.7	88.4	92.4	92.7	86.9	79.5	92.1	88.1	79.6	94.3	88.2
Learning	**81.3**	**92.9**	**92.8**	**88.4**	**93.4**	**93.2**	**87.9**	**80.4**	**93.0**	**89.0**	**80.9**	**94.8**	**89.0**

Table 6. Effect of learning parameters. The best performance is marked as **bold**.

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
Mixup	79.4	92.4	92.6	87.5	92.8	92.4	86.8	**80.3**	92.5	88.2	79.7	**95.4**	88.3
CutMix	79.2	91.2	92.2	87.6	91.8	91.8	86.0	77.8	92.6	88.2	78.4	94.1	87.6
PatchMix	**81.3**	**92.9**	**92.8**	**88.4**	**93.4**	**93.2**	**87.9**	**80.4**	**93.0**	**89.0**	**80.9**	94.8	**89.0**

Table 7. Effect of PatchMix. The best performance is marked as **bold**.

Learning hyperparameters of mixup. Tab. 6 shows the ablation results for the effects of learning hyperparameters of the Beta distribution on the Office-Home. We compare the learning hyperparameters of mixup with fixed parameters, such as Beta(1,1) and Beta(2,2). The proposed method achieves **+0.9%** and **+0.8%** accuracy increment compared with that based on Beta(1,1) and Beta(2,2). The results demonstrate that learning to estimate the distribution to build up the intermediate domain facilitates domain alignment.

PatchMix. Comparisons of PMTrans with Mixup [49] and CutMix [47] are shown in Tab. 7. PMTrans outperforms Mixup and CutMix by **+0.7%** and **+1.4%** accuracy on the Office-Home dataset, demonstrating that PatchMix can capture the global and local mixture information better than the global mixture Mixup and local mixture CutMix methods.

Visualization. In Fig. 5, we visualize the features learned by Swin-based, PMTrans-Swin, PMTrans-ViT, and PMTrans-Deit on task A → C from the Office-Home dataset via the t-SNE [10]. Compared with Swin-based and PMTrans-Swin, our PMTrans model can better align the two domains by constructing the intermediate domain to bridge them. Moreover, comparisons between PMTrans with different transformer backbones reveal that PMTrans works successfully with different backbones on UDA tasks. *Due to the page limit, more experiments and analyses can be found in the suppl. material.*

5. Conclusion and Future Work

In this paper, we proposed a novel method, PMTrans, an optimization solution for UDA from a game perspective. Specifically, we first proposed a novel ViT-based module called PatchMix that effectively built up the intermediate domain to learn discriminative domain-invariant representations for domains. And the two semi-supervised mixup losses were proposed to assist in finding the Nash Equilibria. Moreover, we leveraged attention maps from ViT to reweight the label of each patch by its significance. PMTrans achieved the SoTA results on four benchmark UDA datasets, outperforming the SoTA methods by a large margin. In the near future, we plan to implement our PatchMix and the two semi-supervised mixup losses to solve self-supervised and semi-supervised learning problems. We will also exploit our method to tackle the challenging downstream tasks, *e.g.*, semantic segmentation and object detection.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. NSFC22FYT45.

References

[1] David Acuna, Marc T. Law, Guojun Zhang, and Sanja Fidler. Domain adversarial training: A game perspective. *CoRR*, abs/2202.05352, 2022. 2

[2] David Acuna, Marc T. Law, Guojun Zhang, and Sanja Fidler. Domain adversarial training: A game perspective. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 2

[3] Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998. 2

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE, 2021. 3

[5] Jieneng Chen, Shuyang Sun, Ju He, Philip H. S. Torr, Alan L. Yuille, and Song Bai. Transmix: Attend to mix for vision transformers. *CoRR*, abs/2111.09833, 2021. 2, 4

[6] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3730–3739, 2017. 2

[7] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(9):1853–1865, 2017. 2

[8] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3940–3949. Computer Vision Foundation / IEEE, 2020. 6

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. 6

[10] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31th International Conference on Machine Learning, ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 647–655. JMLR.org, 2014. 8

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021. 1, 6

[12] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 3937–3946. Computer Vision Foundation / IEEE, 2021. 1

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016. 2

[14] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4893–4902. Computer Vision Foundation / IEEE, 2019. 1, 2

[15] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10285–10295. Computer Vision Foundation / IEEE, 2019. 6

[16] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Heng Tao Shen. Maximum density divergence for domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(11):3918–3930, 2021. 2

[17] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2505–2514. Computer Vision Foundation / IEEE, 2021. 1

[18] Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li. Transferable semantic augmentation for domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 11516–11525. Computer Vision Foundation / IEEE, 2021. 2

[19] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. Semantic concentration for domain adaptation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9082–9091. IEEE, 2021. 6

[20] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6028–6039. PMLR, 2020. 5

[21] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 16632–16642. Computer Vision Foundation / IEEE, 2021. 5

[22] Jihao Liu, Boxiao Liu, Hang Zhou, Hongsheng Li, and Yu Liu. Tokenmix: Rethinking image mixing for data augmentation in vision transformers. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI*, volume 13686 of *Lecture Notes in Computer Science*, pages 455–471. Springer, 2022. 2

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9992–10002. IEEE, 2021. 3

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. 6

[25] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 97–105. JMLR.org, 2015. 2

[26] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2208–2217. PMLR, 2017. 2

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 6

[28] Eric Mazumdar, Lillian J. Ratliff, and S. Shankar Sastry. On gradient-based learning in continuous games. *SIAM J. Math. Data Sci.*, 2(1):103–131, 2020. 2

[29] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1094–1103. Computer Vision Foundation / IEEE, 2021. 2, 6

[30] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010. 1

[31] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1406–1415. IEEE, 2019. 2, 6

[32] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *CoRR*, abs/1710.06924, 2017. 2, 6

[33] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, volume 6314 of *Lecture Notes in Computer Science*, pages 213–226. Springer, 2010. 2, 6

[34] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3723–3732. Computer Vision Foundation / IEEE Computer Society, 2018. 1, 6

[35] Astuti Sharma, Tarun Kalluri, and Manmohan Chandraker. Instance level affinity-based transfer for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5361–5371. Computer Vision Foundation / IEEE, 2021. 2

[36] Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. Safe self-refinement for transformer-based domain adaptation. *CoRR*, abs/2204.07683, 2022. 1, 2, 3, 6

[37] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *CoRR*, abs/2012.12877, 2020. 6

[38] A. F. M. Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, and Sung-Ho Bae. Saliencymix: A saliency guided data augmentation strategy for better regularization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. 3

[40] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5385–5394. IEEE Computer Society, 2017. 2, 6

[41] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR, 2019. 2

[42] Yuan Wu, Diana Inkpen, and Ahmed El-Roby. Dual mixup regularized learning for adversarial domain adaptation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX*, volume 12374 of *Lecture Notes in Computer Science*, pages 540–555. Springer, 2020. 2

[43] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 6502–6509. AAAI Press, 2020. 2

[44] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 4393–4402. Computer Vision Foundation / IEEE, 2020. 1

[45] Tongkun Xu, Weihua Chen, Pichao Wang, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. *CoRR*, abs/2109.06165, 2021. 1, 2, 3, 6

[46] Jinyu Yang, Jingjing Liu, Ning Xu, and Junzhou Huang. TVT: transferable vision transformer for unsupervised domain adaptation. *CoRR*, abs/2108.05988, 2021. 2, 3, 6

[47] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6022–6031. IEEE, 2019. 2, 8

[48] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 2

[49] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 2, 8

[50] Yabin Zhang, Bin Deng, Hui Tang, Lei Zhang, and Kui Jia. Unsupervised multi-class domain adaptation: Theory, algorithms, and practice. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(5):2775–2792, 2022. 1

[51] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. Bridging theory and algorithm for domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7404–7413. PMLR, 2019. 6

[52] Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5989–5996. AAAI Press, 2019. 2

[53] Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. Deep subdomain adaptation network for image classification. *IEEE Trans. Neural Networks Learn. Syst.*, 32(4):1713–1722, 2021. 1