SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2024, Vol. 67, Iss. 5, 152108:1–152108:14
https://doi.org/10.1007/s11432-022-3851-1

Maximizing conditional independence for unsupervised domain adaptation

Yiming ZHAI, Chuanxian REN*, Youwei LUO & Daoqing DAI

School of Mathematics, Sun Yat-Sen University, Guangzhou 510275, *China*

Received 26 November 2022/Revised 6 April 2023/Accepted 19 May 2023/Published online 1 April 2024

Abstract Unsupervised domain adaptation (UDA) studies how to transfer a learner from a labeled source domain to an unlabeled target domain with different distributions. Existing methods mainly focus on matching marginal distributions of the source and target domains, which probably leads to a misalignment of samples from the same class but different domains. In this paper, we tackle this misalignment issue by achieving the class-conditioned transferring from a new perspective. Specifically, we propose a method named maximizing conditional independence (MCI) for UDA, which maximizes the conditional independence of feature and domain given class in the reproducing kernel Hilbert spaces. The optimization of conditional independence can be viewed as a surrogate for minimizing class-wise mutual information between feature and domain. An interpretable empirical estimation of the conditional dependence measure is deduced and connected with the unconditional case. Besides, we provide an upper bound on the target error by taking the class-conditional distribution into account, which provides a new theoretical insight for class-conditioned transferring. Extensive experiments on six benchmark datasets and various ablation studies validate the effectiveness of the proposed model in dealing with UDA.

Keywords conditional independence, kernel method, domain adaptation, class-conditioned transferring

1 Introduction

Algorithms of supervised learning have made tremendous contributions to artificial intelligence and have wide applications in real-life. Sufficient labeled data play a significant role in supervised learning. However, it is often expensive and time-consuming to collect plenty of labeled data. In contrast, it is much easier to collect considerable unlabeled data. An intuitive idea is to directly apply the learned predictive model with the labeled data to the unlabeled data. However, there may exist a large discrepancy between the training and testing datasets due to the existence of dataset shift [1]. Then, a direct application may result in a degradation of recognition performance.

Unsupervised domain adaptation (UDA) addresses the problem of transferring knowledge from a labeled dataset (source domain) to an unlabeled dataset (target domain), where the domains have similar but not identical distributions (Figure 1). To learn a discriminative predictor for the unlabeled target domain, the essence of UDA is to effectively mitigate the distribution discrepancy between domains.

Covariate shift is a common assumption in UDA, which assumes the source and target domains have identical label space but different marginal distributions of features, i.e., $P_X^s \neq P_X^t$ with $\mathcal{Y}^s = \mathcal{Y}^t$. Inspired by the learning theory [2], various UDA methods have been proposed to mitigate the marginal distribution discrepancy between domains, e.g., discrepancy minimization via maximum mean discrepancy (MMD) [3, 4] and covariance statistics [5], manifold based feature alignment [6], and adversarial learning based feature confusion [7, 8]. These studies have achieved considerable progress in UDA. However, aligning marginal distributions while ignoring the class information may lead to a misalignment across classes [9, 10] (Figure 1(left)). To be specific, samples from the same class but different domains may not be mapped nearby in the latent feature space, even with a perfect marginal distribution alignment.

To alleviate the misalignment across classes, class-conditioned transferring seeks a class-conditioned domain alignment by considering class information (including pseudo-labels) (Figure 1(right)). Zhao

* Corresponding author (email: rchuanx@mail.sysu.edu.cn)

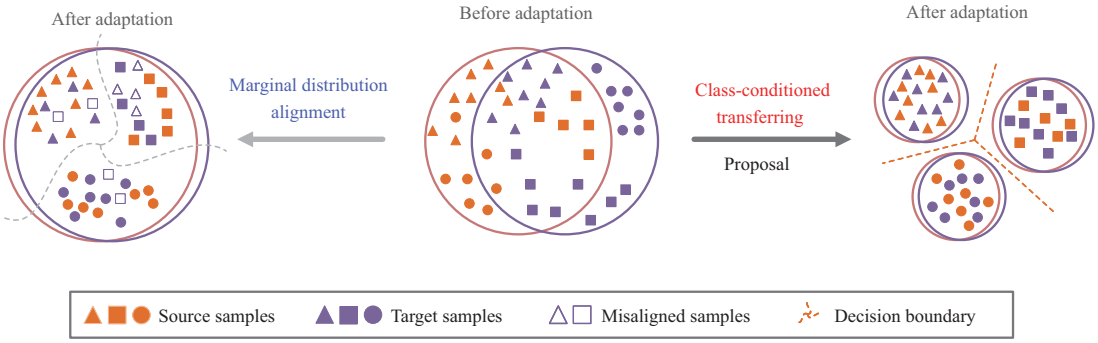

Figure 1 (Color Online) Illustration of class-conditioned transferring. Left: methods based on marginal distribution alignment may lead to a misalignment. Right: the class-conditioned transferring can deal with this misalignment.

et al. [11] theoretically showed that matching the class-conditional distributions is non-ignorable, i.e., $P^s_{X|Y} = P^t_{X|Y}$. Some metric-based methods try to measure the discrepancy between class-conditional distributions, e.g., variants of MMD [10,12,13] and maximum density divergence (MDD) [14]. Adversarial methods [15,16] incorporate multiple-level discriminators to learn both class-discriminative and domain-invariant features. There are methods simultaneously leveraging a metric and an adversarial loss [17–19]. For instance, methods [20–22] apply two classifiers and optimize the classifier discrepancy in an adversarial manner to learn task-specific decision boundaries. Class-conditioned transferring methods with different ideas have also been proposed, e.g., prediction matrix calibration [23], information maximization [24], attention mechanism [25], and transformer-based methods [26,27]. Though these methods have achieved remarkable performance, most cannot theoretically promise the class-conditional distribution alignment. Further, these class-conditioned transferring methods mainly model the variables of feature and class while without explicitly modeling the domain information. It is expected if there is a way for UDA which can not only model the relations among feature, class, and domain but also mathematically bridge the algorithm with a class-conditional distribution alignment.

Motivated by this, in this paper, we propose a novel method called maximizing conditional independence (MCI) for UDA. MCI characterizes class-conditioned transferring as conditional independence, which is a totally new statistical perspective in dealing with UDA. More precisely, MCI explicitly models a set of variables, i.e., the extracted feature X and class Y given domain Z, then maximizes the conditional independence between X and Z given Y by exploiting the conditional dependence measure. With the conditional independence, the domain-specific information can be removed from the class-conditioned feature space. From the perspective of information theory, MCI seeks a compact and informative feature space with reduced class-conditioned mutual information between feature X and domain Z. Additionally, MCI not only theoretically ensures a class-conditional distribution alignment, but also deduces that such an alignment is sufficient to minimize the target error.

To the best of our knowledge, maximizing the conditional independence for class-conditioned transferring has not been explored yet in UDA. The contributions of our work are mainly summarized as follows.

(1) We provide a class-conditional distribution based generalization error bound for UDA, which gives a new theoretical insight for class-conditioned transferring.

(2) We propose a simple yet effective method MCI for UDA, which achieves class-conditioned transferring by making feature and domain conditionally independent given class. It can also be viewed as a surrogate for minimizing class-wise mutual information.

(3) We mathematically derive that the conditional independence will lead to a class-conditional distribution alignment, which theoretically guarantees that samples from the same class but different domains are mapped nearby in the latent feature space.

(4) We derive an interpretable empirical estimation of the conditional dependence measure and connect it with the estimation in the unconditional case, which adjusts and improves the results in [28].

The rest of this paper is organized as follows. Section 2 briefly reviews related UDA work. Section 3 provides preliminaries about the dependence measure, details of MCI, and theoretical analysis. Extensive experiments along with analysis of MCI are presented in Section 4. Finally, Section 5 concludes this paper.

2 Related work

In this section, we briefly review previous UDA studies from two related aspects, including marginal distribution alignment and class-conditioned transferring.

Marginal distribution alignment. Various methods reduce the marginal distribution discrepancy between domains. Long et al. [4] employed MMD to match the distributions of deep features in reproducing kernel Hilbert spaces (RKHSs). Sun et al. [5] aligned the covariance matrices of the two domains. Gong et al. [6] embedded the two domains into Grassmann manifolds, and constructed geodesic flows between the domains to model domain discrepancy. Ganin et al. [7] tried to learn domain-invariant features to confuse a domain discriminator. Optimal transport (OT) has also been applied to UDA successively. Courty et al. [29] learned a nonlinear Wasserstein map to match the feature distributions. Zhang et al. [30] further learned a transport map in RKHSs with Gaussian priors. As for methods based on correlations, Yan et al. [31] employed the Hilbert-Schmidt independence criterion (HSIC), and Liu et al. [32] introduced an entropy regularized optimal transport independence criterion (ETIC). HSIC and ETIC match the marginal distributions by maximizing the independence between feature X and domain Z. Differently, our MCI seeks a class-conditioned distribution alignment by achieving the conditional independence between feature X and domain Z given class Y.

Class-conditioned transferring. Class-conditioned transferring methods show better performance by introducing class information. Prototype-based methods [33, 34] reduce the distance between the source and target centers with the same class. Luo et al. [35] generalized Fisher's discriminant criterion by exploiting the between-class and within-class scatters. Metric-based methods and adversarial methods have been explored to reduce the class-conditional distribution discrepancy. Long et al. [12] and Zhu et al. [13] proposed variants of MMD, which measure the discrepancy by applying MMD within or between class-wise clusters. Ren et al. [36] exploited a conditional covariance operator in RKHS to align the conditional distributions. Li et al. [14] optimized an adversarial loss and an MDD metric to maximize the inter-domain divergence and intra-class density. Based on adversarial learning, Li et al. [17] class-wisely optimized the divergence between predictions to suppress domain-variant information. Differently, Sun et al. [18] and Zhang et al. [19] minimized a sample-level prediction discrepancy. With an auxiliary classifier, Zuo et al. [21] optimized the L_1-distance between the two classifiers' predictions to learn task-specific decision boundaries. The above methods explicitly model the variables of feature X and class Y to achieve class-conditioned transferring, where the domain information is implicitly considered by aligning domains. Though Li et al. [25] explicitly captured domain-specific information by channel-aware attention, they did not model the relation between feature and class. Weighted correlation embedding learning (WCEL) [37] is built on graph learning and correlation learning. However, WCEL only explores the correlation between feature X and domain Z to find the most correlated features. Compared with existing methods, MCI provides a new perspective by characterizing class-conditioned transferring as conditional independence, which explicitly models the feature X, class Y, and domain Z simultaneously. Besides, we derive that achieving conditional independence in RKHSs ensures a class-conditional distribution alignment. In practice, we can directly measure the conditional dependence without splitting samples into class-wise clusters or pair-wise samples like prototype-based or most metric-based methods. MCI is still valid even if class Y is a continuous variable. Different from adversarial methods, the framework of MCI is simple and can be optimized in an end-to-end manner.

3 Methodology

In this section, we explain the proposed method MCI for UDA. Subsection 3.1 reviews the dependence measure. In Subsection 3.2, we employ the conditional dependence measure to characterize class-conditioned transferring. Subsection 3.3 derives the empirical estimation of the dependence measure. Finally, we provide the implementation details of MCI and theoretical analysis in Subsections 3.4 and 3.5, respectively.

3.1 Measuring conditional independence in RKHS

Let $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$ and $(\mathcal{Z}, \mathcal{B}_{\mathcal{Z}})$ be measurable spaces with Borel σ-field $\mathcal{B}_{\mathcal{X}}$ and $\mathcal{B}_{\mathcal{Z}}$, respectively. Denote $(\mathcal{F}_{\mathcal{X}}, k_{\mathcal{X}})$ and $(\mathcal{F}_{\mathcal{Z}}, k_{\mathcal{Z}})$ as corresponding RKHSs of $\mathcal{X}$ and $\mathcal{Z}$, where $k_{\mathcal{X}}$ and $k_{\mathcal{Z}}$ are measurable positive definite kernels.

Consider a random vector (X, Z) on $\mathcal{X} \times \mathcal{Z}$ with $\mathbb{E}_X[k_{\mathcal{X}}(X, X)] < \infty$ and $\mathbb{E}_Z[k_{\mathcal{Z}}(Z, Z)] < \infty$. Then, there exists a unique cross-covariance operator [38] $\Sigma_{ZX} : \mathcal{F}_{\mathcal{X}} \to \mathcal{F}_{\mathcal{Z}}$ which satisfies $\forall f_1 \in \mathcal{F}_{\mathcal{X}}$, $f_2 \in \mathcal{F}_{\mathcal{Z}}$,

$$\langle f_2, \Sigma_{ZX} f_1 \rangle_{\mathcal{F}_{\mathcal{Z}}} = \mathbb{E}[f_1(X) f_2(Z)] - \mathbb{E}[f_1(X)]\mathbb{E}[f_2(Z)],$$

where Σ_{ZX} describes higher-order correlations of X and Z via functions $f_1(X)$ and $f_2(Z)$ in RKHSs. If $Z = X$, Σ_{XX} is the known covariance operator. Besides, Σ_{ZX} can be regarded as an extension of the covariance matrix C_{ZX} on the Euclidean space. Σ_{ZX} can be expressed by the covariance of the marginals and the correlation [38], that is

$$\Sigma_{ZX} = \Sigma_{ZZ}^{\frac{1}{2}} V_{ZX} \Sigma_{XX}^{\frac{1}{2}}, \tag{1}$$

where $\mathcal{R}(V_{ZX}) \subset \overline{\mathcal{R}(\Sigma_{ZX})}$, and $\mathcal{N}(V_{ZX})^{\perp} \subset \overline{\mathcal{R}(\Sigma_{ZX})}$. $\mathcal{N}(T)$ and $\mathcal{R}(T)$ refer to the null space and the range of an operator T, respectively. V_{ZX} is also a unique bounded operator, named the normalized cross-covariance operator (NOCCO). Compared with Σ_{ZX}, V_{ZX} encodes the dependence of X and Z more directly with less influence of the marginals.

In RKHSs, conditional dependence can be derived by cross-covariance operators. Denote another variable Y on $\mathcal{Y}$ and RKHS $(\mathcal{F}_{\mathcal{Y}}, k_{\mathcal{Y}})$, where $k_{\mathcal{Y}}$ is also finite. The normalized conditional cross-covariance operator (COND) [28] can be defined by

$$V_{ZX|Y} = V_{ZX} - V_{ZY} V_{YX}, \tag{2}$$

where V_{ZY} and V_{YX} are similar defined by (1). $V_{ZX|Y}$ measures the conditional dependence of random variables X and Z given Y.

NOCCO and COND have been used to determine the independence and the conditional independence [28]. Denote $X \perp\!\!\!\perp Z$ as the independence of random variables X and Y. $X \perp\!\!\!\perp Z \mid Y$ indicates the conditional independence of X and Z given $Y = y$, $\forall y \in \mathcal{Y}$. Consider $\ddot{X} = (X, Y)$, $\ddot{Z} = (Z, Y)$ with the kernel product $k_{\ddot{\mathcal{X}}} = k_{\mathcal{X}} k_{\mathcal{Y}}$ and $k_{\ddot{\mathcal{Z}}} = k_{\mathcal{Z}} k_{\mathcal{Y}}$. Lemma 1 formulates the mentioned relation.

Lemma 1 ([28]). (i) If the product $k_{\mathcal{X}} k_{\mathcal{Z}}$ is characteristic, then

$$V_{ZX} = 0 \Longleftrightarrow X \perp\!\!\!\perp Z.$$

(ii) Assume that the product $k_{\ddot{\mathcal{X}}} k_{\ddot{\mathcal{Z}}}$ is a characteristic kernel on $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, and $\mathcal{F}_{\mathcal{Y}} + \mathbb{R}$ is dense in $L^2(P_Y)$. Then,

$$V_{\ddot{Z}\ddot{X}|Y} = 0 \Longleftrightarrow X \perp\!\!\!\perp Z \mid Y.$$

Note $\mathcal{F}_{\mathcal{Y}} + \mathbb{R}$ is dense in $L^2(P_Y)$ means that $k_{\dagger}$ is bounded and characteristic. $L^2(P_Y)$ denotes the space of the square integrable functions with the law P_Y.

To measure the distance between the zero element 0 and V_{ZX} ($V_{\ddot{Z}\ddot{X}|Y}$), the HS norm $\|\cdot\|_{\mathrm{HS}}$ of operators is employed. Denote that $V : \mathcal{F}_1 \to \mathcal{F}_2$ is a linear operator, $\{\phi_i\}$ and $\{\psi_j\}$ are complete orthonormal systems of $\mathcal{F}_1$ and $\mathcal{F}_2$. The HS norm of V is defined as $\|V\|_{\mathrm{HS}}^2 = \Sigma_{i,j} \langle \psi_j, V\phi_i \rangle_{\mathcal{F}_2}^2$. V is a HS operator if the sum $\Sigma_{i,j} \langle \psi_j, V\phi_i \rangle_{\mathcal{F}_2}^2$ is finite. Since V_{ZX} and $V_{\ddot{Z}\ddot{X}|Y}$ are HS operators, we can measure the statistical dependence as

$$I^{\mathrm{NOCCO}}(X, Z) = \|V_{ZX}\|_{\mathrm{HS}}^2,$$
$$I^{\mathrm{COND}}(X, Z|Y) = \|V_{\ddot{Z}\ddot{X}|Y}\|_{\mathrm{HS}}^2.$$

3.2 Removing domain-specific information

In UDA, we assume a labeled source domain $\mathcal{D}^S = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_s}$ and an unlabeled target domain $\mathcal{D}^T = \{\boldsymbol{x}_j^t\}_{j=1}^{n_t}$, where $\boldsymbol{x}_i^{s/t} \in \mathcal{X}$ represents the observation and $\boldsymbol{y}_i^s \in \mathcal{Y}$ is the class label. The class label space $\mathcal{Y} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_K\}$, where $\boldsymbol{e}_i$ is a K-dimensional one-hot vector. Denote the domain label space as $\mathcal{Z} = \{\boldsymbol{z}^s, \boldsymbol{z}^t\}$, where $\boldsymbol{z}^{s/t}$ is the domain label of source/target samples. The primary task of UDA is employing $\mathcal{D}^s$ and $\mathcal{D}^t$ to predict $\{\boldsymbol{y}_j^t\}_{j=1}^{n_t}$.

Generally, source and target domains are supposed to have similar but not identical distributions. We consider the random variables including feature $X \in \mathcal{X}$, class $Y \in \mathcal{Y}$, and domain $Z \in \mathcal{Z}$. As shown in Figure 2(left), some UDA methods tend to explore $Z \to X$, i.e., feature X is conditioned on domain Z.

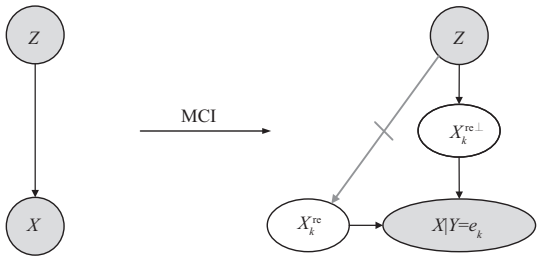

Figure 2 A directed graph of MCI. X_k^{re} denotes the class-conditioned domain-invariant features, which is the output of the feature extractor $g(\cdot)$, and $X_k^{\mathrm{re}\perp}$ denotes the remaining features related to domain label Z. MCI aims to find X_k^{re} being conditionally independent of Z given class Y.

These methods utilize various metrics to align marginal feature distributions, which aim to remove the impact of domain Z from feature X. However, these methods ignore the consideration of class Y during the transferring process, which may lead to a misalignment across classes (Figure 1(left)).

Differently, we propose MCI to explore $Z \to X|Y = \boldsymbol{e}_k$, which achieves class-conditioned transferring by MCI. It is obvious that feature X and domain Z are not conditionally independent, i.e.,

$$P(X, Z|Y) \neq P(X|Y)P(Z|Y),$$

which can be better understood by

$$P(X|Z, Y) \neq P(X|Y). \tag{3}$$

Thus, samples from the same class but different domains, i.e., $X_k^s \sim P(X|Z = \boldsymbol{z}^s, Y = \boldsymbol{e}_k)$ and $X_k^t \sim P(X|Z = \boldsymbol{z}^t, Y = \boldsymbol{e}_k)$, do not have identical conditional distributions due to the domain-specific information. To remove the domain-specific information from the class-conditioned feature space, as shown in the right of Figure 2, we decompose samples from class $\boldsymbol{e}_k$ across domains, i.e., X_k, with $X_k = X_k^{\mathrm{re}} \oplus X_k^{\mathrm{re}\perp}$, where X_k^{re} is independent of domain Z and $X_k^{\mathrm{re}\perp}$ contains all the domain-specific information relating to Z. Conditioning the whole class space of Y, in MCI, we propose to seek the conditional independence of feature X^{re} and domain Z given class Y, i.e., $X^{\mathrm{re}} \perp\!\!\!\perp Z \mid Y$.

In MCI, we use COND w.r.t. $X^{\mathrm{re}} = g(X)$, Z, and Y to explore the conditional independence in RKHSs, where $g(\cdot)$ is a feature extractor. According to Lemma 1, we seek the conditional independence $X^{\mathrm{re}} \perp\!\!\!\perp Z \mid Y$ by learning $g(\cdot)$, i.e.,

$$\min_g I^{\mathrm{COND}}(X^{\mathrm{re}}, Z|Y) = \|V_{\ddot{Z}\ddot{X}^{\mathrm{re}}|Y}\|^2_{\mathrm{HS}}, \tag{4}$$

where the extended variables $\ddot{X}^{\mathrm{re}} = (X^{\mathrm{re}}, Y)$ and $\ddot{Z} = (Z, Y)$. In the following theorem, we relate the conditional independence to class-conditional distribution alignment. The proof is provided in Appendix A.1.

Theorem 1. Assume that the product $k_{\ddot{\mathcal{X}}}k_{\mathcal{Z}}$ is a characteristic kernel on $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, and $\mathcal{F}_{\mathcal{Y}} + \mathbb{R}$ is dense in $L^2(P_{\mathcal{Y}})$. For any conditional distributions $P^S_{X|Y}$, $P^T_{X|Y} \in \Pr^S(\mathcal{X}|\mathcal{Y})$, we have

$$V_{\ddot{Z}\ddot{X}|Y} = 0 \quad \Longrightarrow \quad P^S_{X|Y} = P^T_{X|Y}.$$

Thus, if the conditional dependence measure in (4) is zero, we will have $P^S_{X^{\mathrm{re}}|Y} = P^T_{X^{\mathrm{re}}|Y}$, i.e.,

$$P\left(X^{\mathrm{re}}|Z = \boldsymbol{z}^s, Y = \boldsymbol{e}_k\right) = P(X^{\mathrm{re}}|Z = \boldsymbol{z}^t, Y = \boldsymbol{e}_k)\,.$$

In this case, the distribution of X^{re} is essentially and solely determined by class Y, and domain Z will be superfluous once Y is given.

Here we obtain a desired conclusion that our MCI, i.e., the COND-based method, not only achieves the class-conditioned transferring but also derives a class-conditional distribution alignment. By removing the domain-specific information while preserving the discriminative structure across domains, it is expected to obtain class-discriminative and domain-invariant representations.

We further analysis the COND-based objective (4) from an information theory perspective. Mutual information has been widely used to measure the information shared between two random variables. More

precisely, mutual information $I(X_k, Z) = H(X_k) - H(X_k|Z) \geqslant 0$ with equality if, and only if, X_k and Z are independent. We have the inequality $I(X_k, Z) \leqslant I^{\mathrm{NOCCO}}(X_k, Z)$ holds under the assumption of Theorem 4 in [28]. In some way, optimizing MCI via COND is equal to applying NOCCO on each class separatively. Therefore, MCI can be viewed as a surrogate for minimizing class-wise mutual information $I(X_k^{\mathrm{re}}, Z)$ theoretically. However, the direct estimation of mutual information is intractable if the joint distribution is highly complex. Comparatively, the empirical conditional dependence can be measured in the kernel space directly, without estimating any distributions.

3.3 Empirical estimation of the conditional dependence measure

In this subsection, we derive the estimation of the conditional dependence measure, i.e., $\hat{I}_n^{\mathrm{COND}}(X, Z|Y) = \|\widehat{V}_{\ddot{Z}\ddot{X}|Y}\|_{\mathrm{HS}}^2$. Denote $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i)\}_{i=1}^n$ as a set of samples, which are assumed to be drawn i.i.d from the joint distribution. Since the extended variable $\ddot{X} = (X, Y)$ and $\ddot{Z} = (Z, Y)$ are used for $\hat{I}_n^{\mathrm{COND}}(X, Z|Y)$, we concatenate samples as $\mathcal{D} = \{(\ddot{\boldsymbol{x}}_i, \boldsymbol{y}_i, \ddot{\boldsymbol{z}}_i)\}_{i=1}^n$.

We map data $\ddot{\boldsymbol{x}}_i$ to RKHS $\mathcal{F}_{\ddot{X}}$ with the implicit feature map ϕ, which satisfies the reproducing properties $\langle \phi(\ddot{\boldsymbol{x}}_i), \phi(\ddot{\boldsymbol{x}}_j)\rangle_{\mathcal{F}_{\ddot{\mathcal{X}}}} = k_{\ddot{\mathcal{X}}}(\ddot{\boldsymbol{x}}_i, \ddot{\boldsymbol{x}}_j)$ and $\langle \phi(\ddot{\boldsymbol{x}}_i), f\rangle_{\mathcal{F}_{\ddot{\mathcal{X}}}} = f(\ddot{\boldsymbol{x}}_i)$, $\forall f \in \mathcal{F}_{\ddot{X}}$. Similar properties hold for $\lambda(\ddot{\boldsymbol{z}}_i) \in \mathcal{F}_{\ddot{\mathcal{Z}}}$ and $\psi(\boldsymbol{y}_i) \in \mathcal{F}_{\mathcal{Y}}$. Let $\boldsymbol{K}_{\ddot{\boldsymbol{X}}}$, $\boldsymbol{K}_{\ddot{\boldsymbol{Z}}}$ and $\boldsymbol{K}_{\boldsymbol{Y}}$ denote kernel matrices, which can be explicitly computed as $(\boldsymbol{K}_{\ddot{\boldsymbol{X}}})_{ij} = k_{\ddot{\mathcal{X}}}(\ddot{\boldsymbol{x}}_i, \ddot{\boldsymbol{x}}_j)$, $(\boldsymbol{K}_{\ddot{\boldsymbol{Z}}})_{ij} = k_{\ddot{\mathcal{Z}}}(\ddot{\boldsymbol{z}}_i, \ddot{\boldsymbol{z}}_j)$ and $(\boldsymbol{K}_{\boldsymbol{Y}})_{ij} = k_{\mathcal{Y}}(\boldsymbol{y}_i, \boldsymbol{y}_j)$. Besides, the feature map matrices are represented as

$$\boldsymbol{\Phi} = [\phi(\ddot{\boldsymbol{x}}_1), \phi(\ddot{\boldsymbol{x}}_2), \ldots, \phi(\ddot{\boldsymbol{x}}_n)], \quad \boldsymbol{\Lambda} = [\lambda(\ddot{\boldsymbol{z}}_1), \lambda(\ddot{\boldsymbol{z}}_2), \ldots, \lambda(\ddot{\boldsymbol{z}}_n)], \quad \boldsymbol{\Psi} = [\psi(\boldsymbol{y}_1), \psi(\boldsymbol{y}_2), \ldots, \psi(\boldsymbol{y}_n)].$$

Then, the cross-covariance matrix of $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ can be written as

$$\widehat{\Sigma}_{\ddot{Z}\ddot{X}}^{(n)} = \frac{\boldsymbol{\Lambda}\boldsymbol{H}_n\boldsymbol{\Phi}^{\mathrm{T}}}{n},$$

where $\boldsymbol{H}_n = \boldsymbol{I}_n - \frac{\mathbf{1}\mathbf{1}^{\mathrm{T}}}{n}$ is the centering matrix and $\mathbf{1} \in \mathbb{R}^n$ is the all-ones vector. Similarly, we can compute $\widehat{\Sigma}_{\ddot{X}\ddot{X}}^{(n)}$ and $\widehat{\Sigma}_{\ddot{Z}\ddot{Z}}^{(n)}$ based on $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$. With regularization techniques [39], the NOCCO $V_{\ddot{Z}\ddot{X}}$ can be estimated by

$$\widehat{V}_{\ddot{Z}\ddot{X}}^{(n)} = \left(\widehat{\Sigma}_{\ddot{Z}\ddot{Z}}^{(n)} + \varepsilon_n I\right)^{-1/2} \widehat{\Sigma}_{\ddot{Z}\ddot{X}}^{(n)} \left(\widehat{\Sigma}_{\ddot{X}\ddot{X}}^{(n)} + \varepsilon_n I\right)^{-1/2},$$

where $\varepsilon_n > 0$ is a regularization constant. With similar derivations of $\widehat{V}_{\ddot{Z}Y}^{(n)}$ and $\widehat{V}_{Y\ddot{X}}^{(n)}$, we can estimate COND $V_{\ddot{Z}\ddot{X}|Y}$ by

$$\widehat{V}_{\ddot{Z}\ddot{X}|Y}^{(n)} = \widehat{V}_{\ddot{Z}\ddot{X}}^{(n)} - \widehat{V}_{\ddot{Z}Y}^{(n)}\widehat{V}_{Y\ddot{X}}^{(n)}.$$

Then, the HS norm of $\widehat{V}_{\ddot{Z}\ddot{X}|Y}^{(n)}$ can be computed by

$$\hat{I}_n^{\mathrm{COND}}(X, Z|Y) = \mathrm{Tr}(\widehat{V}_{\ddot{Z}\ddot{X}|Y}^{(n)^{\mathrm{T}}}\widehat{V}_{\ddot{Z}\ddot{X}|Y}^{(n)}).$$

We present the final estimation $\hat{I}_n^{\mathrm{COND}}(X, Z|Y)$ with kernel-based matrices in the following theorem and give proof details in Appendix A.2. Let $\boldsymbol{G}_{\ddot{\boldsymbol{X}}}$, $\boldsymbol{G}_{\ddot{\boldsymbol{Z}}}$, and $\boldsymbol{G}_{\boldsymbol{Y}}$ be the centered kernel matrices, which can be represented by corresponding kernel matrix $\boldsymbol{K}$, i.e.,

$$\boldsymbol{G} = \boldsymbol{H}_n\boldsymbol{K}\boldsymbol{H}_n^{\mathrm{T}}. \tag{5}$$

Then, we define $\boldsymbol{R}_{\ddot{\boldsymbol{X}}}$, $\boldsymbol{R}_{\ddot{\boldsymbol{Z}}}$, and $\boldsymbol{R}_{\boldsymbol{Y}}$ by

$$\boldsymbol{R} = \boldsymbol{G}(\boldsymbol{G} + n\varepsilon_n\boldsymbol{I}_n)^{-1}. \tag{6}$$

Theorem 2. Denote $\boldsymbol{S} = \boldsymbol{I}_n - \boldsymbol{R}_{\boldsymbol{Y}}$. The empirical estimation of the conditional dependence measure is

$$\hat{I}_n^{\mathrm{COND}}(X, Z|Y) = \mathrm{Tr}(\boldsymbol{R}_{\ddot{\boldsymbol{Z}}}\boldsymbol{S}\boldsymbol{R}_{\ddot{\boldsymbol{X}}}\boldsymbol{S}). \tag{7}$$

Theorem 2 deduces an interpretable empirical estimation of the conditional dependence measure, which improves the result in [28]. From (7), it is intuitive that $\hat{I}_n^{\mathrm{COND}}(X, Z|Y)$ incorporates all the conditions of $\boldsymbol{Y}$ by adjusting $\boldsymbol{R}_{\ddot{\mathcal{Z}}}$ and $\boldsymbol{R}_{\ddot{\mathcal{X}}}$ with $\boldsymbol{S}$, i.e., $\boldsymbol{I}_n - \boldsymbol{R}_{\boldsymbol{Y}}$. With Theorem 2, we can make an empirical estimation for the COND-based objective in (4), i.e., $\hat{I}_n^{\mathrm{COND}}(X^{\mathrm{re}}, Z|Y)$.

With the assistance of the incomplete Cholesky decomposition [40] of rank-r, the computational complexity of $\hat{I}_n^{\mathrm{COND}}$ in (7) is $\mathcal{O}(r^2 n)$. To be specific, the centered kernel matrix $\boldsymbol{G}$ can be decomposed as $\boldsymbol{G} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$, where $\boldsymbol{L}^{\mathrm{T}} \in \mathbb{R}^{n\times r}$. Such procedure requires $\mathcal{O}(r^2 n)$ operations. Then, $\boldsymbol{R}$ can be rewritten as

$$\boldsymbol{R} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}(\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} + n\varepsilon_n \boldsymbol{I}_n)^{-1}. \tag{8}$$

Applying the Sherman-Morrison-Woodbury formula on the matrix inverse in (8), the complexity of $\boldsymbol{R}$ is $\mathcal{O}(r^3)$. Thus, the overall computational complexity of $\hat{I}_n^{\mathrm{COND}}$ is $\mathcal{O}(r^2 n)$.

The convergence of COND $\widehat{V}_{ZX|Y}^{(n)}$ in HS norm is provided in [28]. Since V_{ZX}, V_{ZY}, and V_{YX} are HS operators, and that the regularization constant ε_n satisfies $\varepsilon_n \to 0$ and $\varepsilon_n^3 n \to \infty$, we have the convergence in probability

$$\|\widehat{V}_{ZX|Y}^{(n)} - V_{ZX|Y}\|_{\mathrm{HS}} \to 0 \ \ (n \to \infty).$$

Thus, the empirical conditional dependence measure, i.e., $\hat{I}_n^{\mathrm{COND}}(X, Z|Y) = \|\widehat{V}_{\ddot{Z}\ddot{X}|Y}\|_{\mathrm{HS}}^2$, also converges to $I^{\mathrm{COND}}(X, Z|Y)$ in probability rate $\varepsilon_n^{-\frac{3}{2}} n^{-\frac{1}{2}}$.

To explore how condition Y works in the empirical conditional dependence measure $\hat{I}_n^{\mathrm{COND}}(X, Z|Y)$, the empirical estimation of the dependence measure, i.e., $\hat{I}_n^{\mathrm{NOCCO}}(X, Z)$, is provided for comparison,

$$\hat{I}_n^{\mathrm{NOCCO}}(X, Z) = \mathrm{Tr}(\boldsymbol{R}_{\boldsymbol{Z}} \boldsymbol{I}_n \boldsymbol{R}_{\boldsymbol{X}} \boldsymbol{I}_n). \tag{9}$$

Compared with $I^{\mathrm{NOCCO}}(X, Z)$, $I^{\mathrm{COND}}(X, Z|Y)$ further defines a random variable Y as the condition and characterizes the conditional dependence relation. By observing the estimations in (7) and (9), we notice that the condition information w.r.t $\boldsymbol{Y}$ is reflected in $\boldsymbol{R}_{\boldsymbol{Y}}$, which is used to adjust the identity weights for the kernel-based matrix $\boldsymbol{R}$. Thus, in MCI, the intrinsic structure between feature X and domain Z is explored by considering the class condition Y. Actually, $I^{\mathrm{COND}}(X, Z|Y)$ is still valid in dealing with the dependence case like $I^{\mathrm{NOCCO}}(X, Z)$ even if the condition Y serves as a constant. This property is explained as follows.

Proposition 1. Assuming that $k_{\ddot{\mathcal{X}}}$ and $k_{\ddot{\mathcal{Z}}}$ are radial kernels. If Y is a constant random variable, then the empirical estimations of the conditional dependence and dependence are equal. Then,

$$\hat{I}_n^{\mathrm{NOCCO}}(X, Z) = \hat{I}_n^{\mathrm{COND}}(X, Z|Y),$$

where the radial kernel is $k(x, y) = k(\|x - y\|)$.

The constant random variable Y means that it takes a constant value, regardless of any event that occurs. Thus, Y is independent of both X and Z. In this case, the empirical conditional dependence measure $\hat{I}_n^{\mathrm{COND}}(X, Z|Y)$ boils down to the empirical dependence measure $\hat{I}_n^{\mathrm{NOCCO}}(X, Z)$. For UDA, Yan et al. [31] estimated $\hat{I}_n^{\mathrm{NOCCO}}(X, Z)$ between feature X and domain Z, which aims to align marginal feature distributions across domains. Differently, our MCI pursues the class-conditional distribution alignment by employing the empirical conditional dependence measure $\hat{I}_n^{\mathrm{COND}}(X, Z|Y)$. Based on the relation between $\hat{I}_n^{\mathrm{COND}}(X, Z|Y)$ and $\hat{I}_n^{\mathrm{NOCCO}}(X, Z)$ in Proposition 1, MCI is also applicable to achieve a marginal distribution alignment if the class variable Y is ignored. Thus, MCI not only seeks a class-level distribution alignment but also can deal with more general cases.

3.4 MCI for UDA

As described in Subsection 3.2, we propose MCI for UDA, which can achieve a class-conditional distribution alignment by seeking feature X^{re} that satisfies the conditional independence $X^{\mathrm{re}} \perp\!\!\!\perp Z \mid Y$. In this subsection, we discuss the implementation details of MCI for UDA.

In MCI, the entire network structure consists of a feature extractor $g : \boldsymbol{x} \mapsto \boldsymbol{x}^{\mathrm{re}}$ and a classifier $C : \boldsymbol{x}^{\mathrm{re}} \mapsto \hat{\boldsymbol{y}}$. With the feature extractor $g(\cdot)$, the feature matrix is derived by $\boldsymbol{X}^{\mathrm{re}} = g(\boldsymbol{X}) \in \mathbb{R}^{d\times n}$, where $n = n_s + n_t$. Domain matrix $\boldsymbol{Z} \in \mathbb{R}^{2\times n}$ can be defined by the binary domain labels with a one-hot coding scheme. Denote $\boldsymbol{Y} \in \mathbb{R}^{K\times n}$ as the class matrix. It is worth noting that target labels are unavailable in UDA. To construct $\boldsymbol{Y}$, we use probability (soft) predictions $\{\hat{\boldsymbol{y}}_j^t\}_{j=1}^{n_t}$ as pseudo-labels. In

Algorithm 1 MCI

Require: Source dataset $\mathcal{D}^s$, target dataset $\mathcal{D}^t$, batch size $b_{s/t}$, ε_n, entropy weight β_{Ent}, and COND weight β_{COND}.
Ensure: Network parameters $\boldsymbol{W}_g$, $\boldsymbol{W}_C$, predictions $\{\hat{\boldsymbol{y}}_j^t\}_{j=1}^{n_t}$.
while not converged **do**
  Sample data $\mathcal{B}^s = \{\boldsymbol{x}_i^s, \boldsymbol{y}_i^s\}_{i=1}^{b_s}$ and $\mathcal{B}^t = \{\boldsymbol{x}_j^t\}_{j=1}^{b_t}$ from $\mathcal{D}^s$ and $\mathcal{D}^t$, respectively;
  Forward propagate data $\boldsymbol{X}_s^{\text{re}} = g(\boldsymbol{X}^s)$, $\boldsymbol{X}_t^{\text{re}} = g(\boldsymbol{X}^t)$, $\hat{\boldsymbol{Y}}^s = C(\boldsymbol{X}_s^{\text{re}})$, $\hat{\boldsymbol{Y}}^t = C(\boldsymbol{X}_t^{\text{re}})$;
  Estimate the cross-entropy loss $\mathcal{L}_{\text{CE}}$ by (10) and the target entropy loss $\mathcal{L}_{\text{Ent}}$ by (11);
  % Estimation of $\mathcal{L}_{\text{COND}}$
  Construct domain matrices $\boldsymbol{Z}^s \in \mathbb{R}^{2\times b_s}$ for $\boldsymbol{X}^s$ and $\boldsymbol{Z}^t \in \mathbb{R}^{2\times b_t}$ for $\boldsymbol{X}^t$;
  Construct matrices $\boldsymbol{X}^{\text{re}} = (\boldsymbol{X}_s^{\text{re}}, \boldsymbol{X}_t^{\text{re}}) \in \mathbb{R}^{d\times(b_s+b_t)}$, $\boldsymbol{Y} = (\boldsymbol{Y}^s, \hat{\boldsymbol{Y}}^t) \in \mathbb{R}^{K\times(b_s+b_t)}$, $\boldsymbol{Z} = (\boldsymbol{Z}^s, \boldsymbol{Z}^t) \in \mathbb{R}^{2\times(b_s+b_t)}$;
  Concatenate matrices by $\ddot{\boldsymbol{X}}^{\text{re}} = (\boldsymbol{X}^{\text{re}}; \boldsymbol{Y}) \in \mathbb{R}^{(d+K)\times n}$ and $\ddot{\boldsymbol{Z}} = (\boldsymbol{Z}; \boldsymbol{Y}) \in \mathbb{R}^{(2+K)\times n}$;
  Map $\ddot{\boldsymbol{X}}^{\text{re}}$, $\ddot{\boldsymbol{Z}}$, and $\boldsymbol{Y}$ into RKHSs by computing kernel matrices $\boldsymbol{K}_{\ddot{\boldsymbol{X}}^{\text{re}}}$, $\boldsymbol{K}_{\ddot{\boldsymbol{Z}}}$, and $\boldsymbol{K}_{\boldsymbol{Y}}$ as (16);
  Compute $\boldsymbol{R}_{\ddot{\boldsymbol{X}}^{\text{re}}}$, $\boldsymbol{R}_{\ddot{\boldsymbol{Z}}}$, and $\boldsymbol{R}_{\boldsymbol{Y}}$ via (5) and (6);
  Estimate the conditional dependence loss $\mathcal{L}_{\text{COND}}$ by (12);
  % Model update
  Update $\boldsymbol{W}_g$, $\boldsymbol{W}_C$ by minimizing the overall objective function in (13);
end while

experiments, we will discuss the accuracy of pseudo-labels. With $\boldsymbol{X}^{\text{re}}$, $\boldsymbol{Z}$, and $\boldsymbol{Y}$, MCI can calculate and minimize the conditional dependence measure $\hat{I}_n^{\text{COND}}(X^{\text{re}}, Z|Y)$.

To build a basic classification network, a supervised learning task performed on the source domain is considered. Let $\boldsymbol{W}_g$ and $\boldsymbol{W}_C$ represent the parameters of the feature extractor $g(\cdot)$ and the classifier $C(\cdot)$, respectively. We apply the cross-entropy function on the labeled source samples, i.e.,

$$\mathcal{L}_{\text{CE}}(\boldsymbol{W}_g, \boldsymbol{W}_C) = \sum_{i=1}^{K}\sum_{j=1}^{n_s} -y_{ij}^s \log \hat{y}_{ij}^s, \tag{10}$$

where $\hat{y}_{ij}^s = C(g(\boldsymbol{x}_j^s))$ and $\sum_{i=1}^K \hat{y}_{ij}^s = 1$. $\hat{y}_{ij}^s$ is the prediction probability of $\boldsymbol{x}_j^s$ belonging to the i-th class. $\boldsymbol{y}_j^s$ is the ground-truth label of $\boldsymbol{x}_j^s$.

To explore the intrinsic structure of the target domain, we employ the entropy function, i.e.,

$$\mathcal{L}_{\text{Ent}}(\boldsymbol{W}_g, \boldsymbol{W}_C) = \sum_{i=1}^{K}\sum_{j=1}^{n_t} -\hat{y}_{ij}^t \log \hat{y}_{ij}^t, \tag{11}$$

where $\hat{y}_{ij}^t = C(g(\boldsymbol{x}_j^t))$ and $\sum_{i=1}^K \hat{y}_{ij}^t = 1$. $\hat{y}_{ij}^t$ is the probability prediction of $\boldsymbol{x}_j^t$ belonging to the i-th class. The entropy loss $\mathcal{L}_{\text{Ent}}$ improves the quality of target pseudo-labels by reducing the uncertainty of target predictions, which benefits the estimation of the conditional dependence measure.

To perform the class-conditioned transferring, the conditional dependence $I^{\text{COND}}(X^{\text{re}}, Z|Y)$ shown in (4) is estimated by the extracted feature $\boldsymbol{X}^{\text{re}}$, domain $\boldsymbol{Z}$ and class $\boldsymbol{Y}$. We firstly extend variables as $\ddot{X}^{\text{re}} = (X^{\text{re}}, Y)$ and $\ddot{Z} = (Z, Y)$. Corresponding samples are concatenated as $\ddot{\boldsymbol{X}}^{\text{re}} \in \mathbb{R}^{(d+K)\times n}$ and $\ddot{\boldsymbol{Z}} \in \mathbb{R}^{(2+K)\times n}$, respectively. Then, the kernel matrix $\boldsymbol{K}_{\ddot{\boldsymbol{X}}^{\text{re}}}$ can be explicitly computed by $k_{\ddot{\mathcal{X}}}(\ddot{\boldsymbol{x}}_i^{\text{re}}, \ddot{\boldsymbol{x}}_j^{\text{re}})$. Kernel matrices $\boldsymbol{K}_{\ddot{\boldsymbol{Z}}}$ and $\boldsymbol{K}_{\boldsymbol{Y}}$ can be derived in a similar way. According to the estimation of conditional dependence in Theorem 2, $\hat{I}_n^{\text{COND}}(X^{\text{re}}, Z|Y)$ is estimated as

$$\mathcal{L}_{\text{COND}}(\boldsymbol{W}_g) = \boldsymbol{R}_{\ddot{\boldsymbol{Z}}} \boldsymbol{S} \boldsymbol{R}_{\ddot{\boldsymbol{X}}^{\text{re}}} \boldsymbol{S}. \tag{12}$$

With positive hyper-parameters β_{COND} and β_{Ent}, the objective function of MCI can be written as

$$\mathcal{L}_{\text{MCI}}(\boldsymbol{W}_g, \boldsymbol{W}_C) = \mathcal{L}_{\text{CE}} + \beta_{\text{COND}}\mathcal{L}_{\text{COND}} + \beta_{\text{Ent}}\mathcal{L}_{\text{Ent}}. \tag{13}$$

The training pipeline of MCI is provided in Algorithm 1. According to Lemma 1, minimizing $\mathcal{L}_{\text{COND}}$ ensures a conditional independence, i.e., $X^{\text{re}} \perp\!\!\!\perp Z \mid Y$. Then, Theorem 1 indicates that it is expected to align the class-conditional distributions, i.e., $P^S_{X^{\text{re}}|Y} = P^T_{X^{\text{re}}|Y}$. Thus, the classifier trained in the aligned feature space tends to give more accurate target pseudo-labels. The estimation of $\mathcal{L}_{\text{COND}}$ will be more precise and reliable, which leads to a better class-conditional distribution alignment. Therefore, feature learning and classifier learning can benefit from each other and promote the training of the adaptation model.

3.5 Theoretical analysis

Following we provide a new theoretical insight based on the divergence between class-conditional distributions $P^S_{X|Y}$ and $P^T_{X|Y}$. Based on [2], domain $\mathcal{D} = (\mu, f)$ is defined by a distribution μ on inputs $\mathcal{X}$ and a labeling function f. The probability according to the distribution μ that a hypothesis h disagrees with a labeling function f (which can also be a hypothesis) is defined as

$$\epsilon_{\mathcal{D}}(h) = \epsilon_{\mathcal{D}}(h, f) = \mathbb{E}_{\boldsymbol{x}\sim\mu}[\mathbb{I}(h(\boldsymbol{x}), f(\boldsymbol{x}))],$$

where $\mathbb{I}(\cdot,\cdot)$ is an indicator function. For the source and target domains, we denote the source error and target error of a hypothesis h as $\epsilon_S(h)$ and $\epsilon_T(h)$, respectively.

Theorem 3. Let $\mathcal{H}$ be a hypothesis space of VC dimension d, m be the sample size of the source domain and f_s be the ground truth labeling function for the source domain. If $\hat{h} \in \mathcal{H}$ is the empirical minimizer of $\hat{\epsilon}_S(h)$ and $h^*_T = \text{argmin}_{h\in\mathcal{H}}\, \epsilon_T(h)$ is the target error minimizer, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\epsilon_T(\hat{h}) \leqslant \epsilon_T(h^*) + 2\left(\lambda + \frac{1}{2}\mathbb{E}_Y[d_{\mathcal{H}\Delta\mathcal{H}}(P^S_{X|Y}, P^T_{X|Y})] + \|P^S_Y - P^T_Y\|_1\right) + 2\sqrt{\frac{1}{2m}\left(\log\frac{d}{\delta}\right)}, \tag{14}$$

where $\lambda = \min_{h\in\mathcal{H}}\{\epsilon_S(h) + \epsilon_T(h)\}$.

Theorem 3 shows the upper bound on the target error of the learned hypothesis. Here we focus on the expectation of divergence between the class-conditional distributions, i.e., $\mathbb{E}_Y[d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{P}^S_{X|Y}, \mathcal{P}^T_{X|Y})]$, and the joint prediction error λ. The former item evaluates the discrepancy between class-conditional distributions, which motivates class-conditioned transferring based UDA methods. MCI aims to remove the domain-specific information by achieving the conditional independence with $V_{\ddot{X}^{\text{re}}\ddot{Z}|Y} = 0$. Theorem 1 ensures that MCI derives a class-conditional distribution alignment, i.e., $P^S_{X^{\text{re}}|Y} = P^T_{X^{\text{re}}|Y}$. This indicates that optimizing MCI is equal to minimizing the expectation of the class-conditioned $\mathcal{H}\Delta\mathcal{H}$-divergence.

If the joint prediction error λ in (14) is large, it is impossible to learn a classifier that performs well on both domains. Thus, it is also important to bound λ. We mathematically show that MCI optimizes the upper bound of λ by using the pseudo-labels. Based on the triangle inequality for classification error [41], i.e., $\epsilon(f_1, f_2) \leqslant \epsilon(f_1, f_3) + \epsilon(f_3, f_2)$, for any labeling functions f_1, f_2 and f_3, we have

$$\lambda = \min_{h\in\mathcal{H}} \epsilon_S(h, f_s) + \epsilon_T(h, f_t) \leqslant \min_{h\in\mathcal{H}} \epsilon_S(h, f_s) + \epsilon_T(h, f_s) + \epsilon_T(f_s, f_t). \tag{15}$$

To present a more clear illustration, we decompose the hypothesis into the feature extractor $g(\cdot)$ and classifier $C(\cdot)$. Thus, Eq. (15) can be rewritten as

$$\min_{g,C} \epsilon_S(C\circ g, C_s\circ g) + \epsilon_T(C\circ g, C_s\circ g) + \epsilon_T(C_s\circ g, C_t\circ g),$$

where $f_s = C_s \circ g$ and $f_t = C_t \circ g$. The first and second items denote the disagreements between the classifier $C(\cdot)$ and the source classifier $C_s(\cdot)$ on source and target domains, respectively. With the supervised training on the labeled source domain, the disagreements can be decreased by approximating $C_s(\cdot)$. The last item originally denotes the disagreement between the source labeling function f_s and the target labeling function f_t on the target domain, which is nonnegative. If $g(\cdot)$ maps samples from the same class but different domains nearby in the latent feature space, $C_s(\cdot)$ and $C_t(\cdot)$ will have similar decision boundaries on the target domain. Then, the last item can be decreased by learning class-discriminative and domain-invariant features, which can be sufficiently guaranteed by the class-conditional distribution alignment. Thus, the three items in (15) are expected to be small. The joint prediction error λ can be optimized by the training of MCI.

4 Experiments

4.1 Datasets and implementation details

Image-CLEF [42]**.** This dataset has 3 domains with 12 classes, i.e., Caltech (C), ImageNet (I), and Pascal (P). Especially, it is a balanced dataset as each domain contains 600 images.

Office-10 [6]**.** This dataset contains 2533 images from 4 domains with 10 classes, i.e., Amazon (A), Caltech (C), DSLR (D), and Webcam (W). There are 8 to 151 samples per class per domain.

Office-31 [43]**.** This dataset contains 4110 images from 3 domains with 31 classes, i.e., Amazon (A), Webcam (W), DSLR (D).

Office-Home [44]**.** This dataset consists of 15500 images from 4 domains with 65 classes. The domains include Artistic (Ar), Clipart (Cl), Product (Pr), and Real-World (Rw). Each class has around 70 images and 99 images maximally.

VisDA-2017 [45]**.** This is a challenging large-scale dataset, which consists of 280k images in 12 classes. Domain synthetic (S) includes 152397 synthetic images generated by 3D models. Domain real-image (R) collects 55388 object images. A synthetic-to-real domain gap will be explored.

DomainNet [46]**.** This is the largest domain adaptation dataset so far. It contains about 0.6 million images from 6 domains with of 345 classes: Clipart (clp), Infograph (inf), Painting (pnt), Quickdraw (qdr), Real (rel), and Sketch (skt).

MCI is trained with back-propagation in a mini-batch manner. The feature extractor $g(\cdot)$ is based on a backbone followed by two/three fully-connected layers with 512/256 output units, where the backbones include convolutional neural networks (CNNs) and vision transformer (ViT) [47]. ResNet-50/101 [48] and AlexNet [49] are employed as the CNNs backbones. ViT-base with 16×16 patch size [47] is employed as the ViT backbones. All backbones are pre-trained on ImageNet [50]. The classifier $C(\cdot)$ is a single fully-connected layer with K output units and a softmax activate function. The Gaussian kernel is used due to its characteristic property. Thus, kernel matrix $\boldsymbol{K}$ can be computed by

$$\boldsymbol{K}_{ij} = \exp\left(-\boldsymbol{D}_{ij}/\sigma^2\right), \quad \boldsymbol{D}_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2, \quad \sigma^2 = \mathbf{1}^{\mathrm{T}}\boldsymbol{D}\mathbf{1}/n^2, \tag{16}$$

where $\mathbf{1} \in \mathbb{R}^n$ is the all-ones vector and σ^2 equals to the mean of all the square Euclidean distances $\boldsymbol{D}_{ij}$. The ε_n in (6) is set as 1e−5 for Image-CLEF and Office-10, and 1e−4 for other datasets.

4.2 Main results

In this subsection, we evaluate MCI with two kinds of backbones, i.e., CNNs (AlexNet, ResNet-50/101) and ViT (ViT-Base), and present comparisons against state-of-the-art UDA methods.

CNNs-based results. The results on Image-CLEF, Office-10, and Office-31 are reported in Table 1. MCI substantially achieves the highest mean accuracy on these datasets. The encouraging results indicate the importance of class-conditioned transferring and the effectiveness of MCI. The Source-Only model basically trains a cross-entropy loss on the source domain. The average accuracies of marginal adaptation methods (e.g., CORAL [5], GFK [6], DAN [4], DANN [7], OT-IT [29], KGOT [30], and ETIC [32]) are higher than the Source-Only model, which shows that matching marginal distributions can alleviates domain discrepancy. Specifically, ETIC outperforms most marginal methods, which indicates the advantage of the independence criterion. We also find that class-conditioned transferring methods (e.g., CTSN [21], DSAN [13], DMP [35], ATM [14], BuresNet [36], SymmNets-V2 [22], WCEL [37], DCAN+SCDA [17], and MCI) improve significantly over these marginal adaptation methods. CDAN+E [9] formulates the joint adaptation by incorporating the class variable into DANN, and its average accuracy lies between most marginal adaptation methods and class-conditioned transferring methods. These results indicate the importance of class information in promoting the transferability and discriminability of adaptation models. Specifically, MCI outperforms other class-conditioned transferring methods, which indicates the superiority of MCI in explicitly exploring the relations among feature, domain, and class.

The results on Office-Home, VisDA-2017, and DomainNet are presented in Tables B2, B4, and B5, respectively. Class-conditioned transferring methods and marginal methods have similar conclusions as mentioned above. The performance of SHOT [24] is slightly better than MCI on Office-Home and VisDA-2017, probably due to its efforts to obtain reliable target predictions. DCAN+SCDA employs class-wise MMD, pair-wise Jensen-Shannon divergence, and domain-wise attention, while MCI only relies on a conditional dependence measure. Thus, it is reasonable that DCAN+SCDA performs better than MCI on Office-Home. Nevertheless, the mean accuracy of MCI is higher than most methods, e.g., DSAN, DMP, ATM, BuresNet, SymmNets-V2, and WCEL, on Office-Home and VisDA-2017. Though WCEL is built upon correlation learning, MCI improves the mean accuracy over WCEL by 2.1% and 1.5% on Office-Home and VisDA-2017, respectively. This result further confirms that MCI is effective in exploring the relations among feature, class, and domain. MCI obtains the highest average accuracy with 36.9% on DomainNet. The average accuracy of MCI makes an improvement over MDD+SCDA [17] by 3.6%,

Table 1 Accuracies (%) on Office-31, Image-CLEF (ResNet-50) and Office-10 (AlexNet)[a)]

Method	Office-31							Image-CLEF						
	A→W	D→W	W→D	A→D	D→A	W→A	Mean	I→P	P→I	I→C	C→I	C→P	P→C	Mean
Source-Only [48]	68.4	96.7	99.3	68.9	62.5	60.7	76.1	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN [4]	80.5	97.1	99.6	78.6	63.6	62.8	80.4	74.5	82.2	92.8	86.3	69.2	89.8	82.5
DANN [7]	82.0	96.9	99.1	79.7	68.2	67.4	82.2	75.0	86.0	96.2	87.0	74.3	91.5	85.0
KGOT [30]	75.3	96.2	98.4	80.3	65.2	63.5	79.8	76.3	83.3	93.5	87.5	74.8	89.0	84.1
ETIC [32]	88.0	**100.0**	98.0	85.9	68.2	69.0	84.8	80.4	91.3	95.1	90.9	78.4	94.2	88.4
CDAN+E [9]	94.1	98.6	**100.0**	92.9	71.0	69.3	87.7	77.7	90.7	97.7	91.3	74.2	94.3	87.7
DSAN [13]	93.6	98.3	**100.0**	90.2	73.5	74.8	88.4	80.2	93.3	97.2	93.8	80.8	95.9	90.2
DMP [35]	93.0	98.7	**100.0**	92.4	75.4	74.2	88.9	80.7	92.5	97.2	90.5	77.7	96.2	89.1
ATM [14]	**95.7**	99.3	**100.0**	96.4	74.1	73.5	89.8	80.3	92.9	**98.6**	93.5	77.8	96.7	90.0
SHOT [24]	90.1	98.4	99.9	94.0	74.7	74.3	88.6	–	–	–	–	–	–	–
DCAN+SCDA [17]	94.8	98.2	**100.0**	94.6	77.5	76.4	90.3	–	–	–	–	–	–	–
BuresNet [36]	–	–	–	–	–	–	–	80.7	**93.7**	97.0	93.5	79.2	**97.0**	90.2
SymmNets-V2 [22]	94.2	98.8	**100.0**	93.5	74.4	73.4	89.1	79.0	93.5	96.9	93.4	79.2	96.2	89.7
MCI	93.8	99.0	**100.0**	**96.8**	**86.2**	**83.4**	**93.2**	**82.0**	92.8	97.0	**95.8**	**82.2**	96.0	**90.9**

	Office-10												
	A→C	A→D	A→W	C→A	C→D	C→W	D→A	D→C	D→W	W→A	W→C	W→D	Mean
Source-Only [49]	82.7	85.4	78.3	91.5	88.5	83.1	80.6	74.6	99.0	77.0	69.6	**100.0**	84.2
GFK [6]	78.1	84.7	76.3	89.1	88.5	80.3	89.0	78.4	99.3	83.9	76.2	**100.0**	85.3
CORAL [5]	85.3	80.8	76.3	91.1	86.6	81.1	88.7	80.4	99.3	82.1	78.7	**100.0**	85.9
OT-IT [29]	83.3	84.1	77.3	88.7	90.5	88.5	83.3	84.0	98.3	88.9	79.1	99.4	87.1
KGOT [30]	85.7	86.6	82.4	91.4	92.4	87.1	91.8	85.6	99.3	89.7	85.0	**100.0**	89.7
ETIC [32]	84.5	91.1	93.2	93.2	**96.0**	91.4	79.0	72.3	97.3	77.0	68.8	**100.0**	87.0
DMP [35]	86.6	90.4	91.3	92.8	93.0	88.5	91.4	85.3	97.7	91.9	85.6	**100.0**	91.2
BuresNet [36]	87.0	**93.6**	90.2	93.4	93.6	90.8	92.7	83.5	**100.0**	92.4	84.3	**100.0**	91.8
MCI	**87.9**	92.7	**96.0**	**93.7**	94.5	**94.1**	**93.1**	**87.0**	99.6	**93.8**	**86.6**	100.0	**93.2**

a) Bold font indicates the highest accuracy.

which shows the superiority of MCI. Overall, we conclude that MCI is helpful in reducing the domain discrepancy on challenging datasets.

ViT-based results and backbone analysis. With the surge of a transformer, we exploit ViT-B as the feature extractor to evaluate MCI. The comparison methods include CDtrans [26], TVT [27], SSRT-B [18], and SHOT [24]. The transformer-related results on Office-31 are reported in the second row of Table B1. We can observe that MCI achieves the highest accuracy with 94.0%. Considering the results based on AlexNet and ResNet-50, we conclude that MCI can promote performance under different backbones. The results on Office-Home are presented in Table B3. The mean accuracy of MCI is 5.3% higher than SHOT, which indicates that MCI can show more advantages under a stronger backbone. Since Office-Home has 65 categories, the self-training strategy in SSRT-B and the information maximization loss in TVT play a vital role in boosting performance. It might be why SSRT-B and TVT perform slightly better than MCI. Though CDtrans designs a cross-attention module and labeling strategy, MCI achieves a better performance than it. Overall, MCI is effective in aligning domains with different backbones.

4.3 Ablation studies and discussion

Parameter sensitivity. We evaluate the parameter sensitivity on Image-CLEF. Figures 3(a) and (b) show grid search results by selecting β_{COND} from {1e−2, 1e−1, 1e0, 1e1} and β_{Ent} from {5e−3, 5e−2, 5e−1, 5e−0, 5e1}, where $(\beta_{\mathrm{COND}}, \beta_{\mathrm{Ent}})$ = (1e−1, 5e−2) is the optimal setting for the two tasks. In general, MCI is stable under different parameter settings. We can find that the accuracy decreases with smaller values of β_{COND}. Such results indicate that maximizing the conditional independence is vital for achieving better performance.

Ablation study. To explore the impact of $\mathcal{L}_{\mathrm{COND}}$ and $\mathcal{L}_{\mathrm{Ent}}$, we design ablation experiments from three aspects: MCI without $\mathcal{L}_{\mathrm{COND}}$, MCI without $\mathcal{L}_{\mathrm{Ent}}$, and HSIC [31] based on (9), where HSIC matches marginal distributions by maximizing independence of the feature and domain. In Table 2, we observe

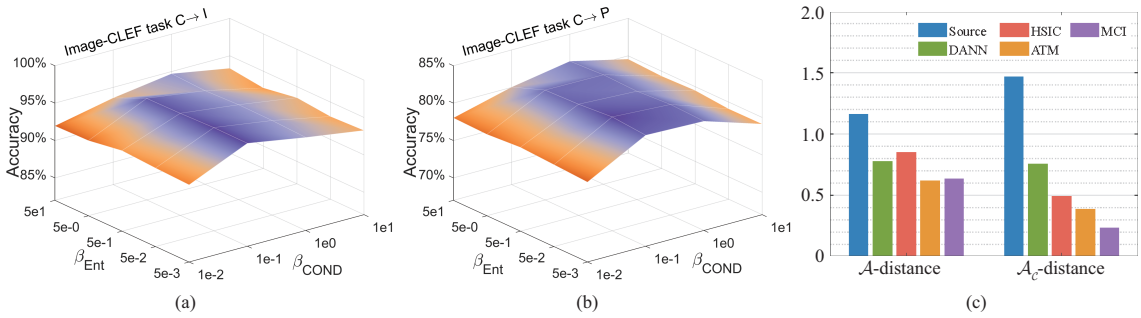

Figure 3 (Color online) Parameter sensitivity of β_{COND} and β_{Ent} on Image-CLEF tasks (a) C → I and (b) C → P; (c) $\mathcal{A}$-distance and $\mathcal{A}_C$-distance on Image-CLEF task C → I. The lower, the better.

Table 2 Accuracies (%) of ablation studies on Image-CLEF and Office-Home datasets

Method	Image-CLEF				Office-Home			
	C→P	I→P	C→I	P→C	Cl→ Rw	Pr → Ar	Rw → Ar	Rw → Pr
MCI (w/o $\mathcal{L}_{\text{COND}}$)	77.6	79.5	91.5	93.9	70.3	61.1	67.6	80.8
MCI (w/o $\mathcal{L}_{\text{Ent}}$)	81.8	81.7	95.5	95.2	75.5	64.4	69.1	83.0
HSIC [31]	80.7	81.6	95.1	93.8	71.3	61.6	68.0	82.2
MCI	**82.2**	**82.0**	**95.8**	**96.0**	**76.3**	**64.8**	**69.3**	**83.3**

Table 3 Dependence test on Image-CLEF task C → I[a)]

	Source-Only [48]	DANN [7]	HSIC [31]	ATM [14]	MCI (w/o $\mathcal{L}_{\text{Ent}}$)	MCI
$\hat{I}_n^{\text{NOCCO}}$	0.3444	0.2402	0.0572	0.1349	0.0672	0.0485
$\hat{I}_{n_C}^{\text{NOCCO}}$	0.7370	0.7205	0.5354	0.6337	0.4959	0.4293
Accuracy (%)	78.0	87.0	95.1	93.5	95.5	95.8

a) Lower values indicate lower dependence.

that MCI consistently achieves the best, which suggests the superiority of MCI. MCI (w/o $\mathcal{L}_{\text{Ent}}$) surpasses MCI (w/o $\mathcal{L}_{\text{COND}}$) with at least 1.6% in accuracy, which indicates that loss $\mathcal{L}_{\text{COND}}$ plays a key role in the class-conditioned transferring. The accuracies of MCI are higher than HSIC, which validates that the class-discriminative and domain-invariant features are helpful in training a discriminative classifier.

Dependence test. We estimate $\hat{I}_n^{\text{NOCCO}}$ to explore the dependence of feature and domain. $\hat{I}_{n_C}^{\text{NOCCO}} = \mathbb{E}[\hat{I}_{n_c}^{\text{NOCCO}}]$ is defined to estimate a class-level dependence. In Table 3, the Source-Only has the highest domain-level and class-level dependence values and lowest classification accuracy. DANN reduces the domain-level dependence and improves the accuracy of Source-Only by learning domain-invariant features. Class-conditioned transferring methods ATM and MCI perform better with much lower dependence values than DANN. HSIC achieves lower dependence values and gives better accuracy than ATM, which shows the superiority of the dependence measure in dealing with UDA. MCI achieves the best result along with the lowest dependence at both levels, which validates the superiority of MCI.

Distribution discrepancy. To measure the domain discrepancy, we employ $\mathcal{A}$-distance $d_{\mathcal{A}}=2(1-2\epsilon)$, where ϵ is the test error of a classifier which is trained to discriminate domains [2]. We also estimate the class-level distribution discrepancy by $d_{\mathcal{A}_C}=\mathbb{E}[d_{\mathcal{A}_c}]$, where $d_{\mathcal{A}_c}$ is the $\mathcal{A}$-distance of the class-conditional distributions based on class $\boldsymbol{y}_c$. In Figure 3(c), MCI (w/o $\mathcal{L}_{\text{COND}}$) has smaller $\mathcal{A}$-distance and $\mathcal{A}_C$-distance than MCI (w/o $\mathcal{L}_{\text{Ent}}$), which validates that $\mathcal{L}_{\text{COND}}$ is the key of MCI. Though ATM and MCI have similar $\mathcal{A}$-distance, MCI has a smaller $\mathcal{A}_C$-distance. Thus, MCI is helpful to learn more separable features.

Feature visualization. To evaluate the aligned features intuitively, we use t-SNE [51] to visualize the features of Source-Only, DANN, and MCI on Image-CLEF task C→I, as shown in Figures 4(a)–(c). In Figure 4(a), the source and target domains have different spatial distributions before adaptation. Compared with DANN, MCI has intra-class compactness and inter-class separability, which validates that MCI leads to a class-conditional distribution alignment by MCI.

Pseudo labeling. To compute $\mathcal{L}_{\text{COND}}$, we exploit pseudo-labels of target samples. The pseudo-labels are dynamically updated in each iteration. From Figure B1, we can see that the accuracy of pseudo-labels is steadily increasing with the iteration until convergence. Compared with MCI w/o $\mathcal{L}_{\text{Ent}}$ and MCI w/

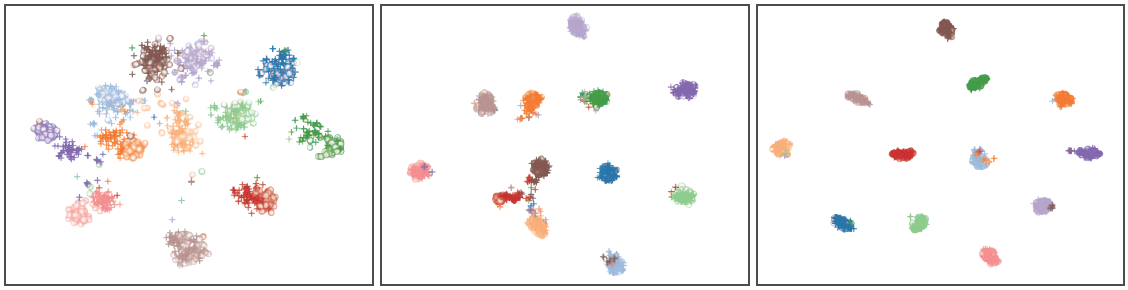

Figure 4 (Color online) The t-SNE visualization of features generated by (a) Source-Only, (b) DANN, and (c) MCI on ImageCLEF task C → I. Here, "o" means source domain and "+" means target domain. Each color denotes one class.

hard predictions, the pseudo-label accuracy of MCI converges at a higher level, which shows that $\mathcal{L}_{\text{Ent}}$ and soft predictions can boost the reliance of pseudo-labels.

Time comparison. We compare the time of metric estimation in Table B6. For MCI, the time is the cost of computing $\mathcal{L}_{\text{COND}}$ with extracted features, similarly for HSIC and ETIC. All experiments are run on a device with an NVIDIA GTX1080Ti GPU. From Table B6, we obtain the following observations: (1) MCI and HSIC take longer time than ETIC since they have to calculate the inverse of the kernel matrix. MCI and HSIC are built upon covariance operators in RKHSs. The time of MCI is slightly longer than HSIC due to MCI models one more variable. (2) MCI achieves higher accuracies than other methods and significantly improves the harder task W→A. Though MCI takes a longer time, it has superiority in classification performance. Therefore, MCI is generally practical, which ensures significant accuracy improvement with slightly increased computation cost.

5 Conclusion

In this paper, we deal with UDA by removing the domain-specific information while preserving the discriminative structure simultaneously. Specifically, we explore the class-conditioned transferring from a new statistical perspective, which maximizes the conditional independence of the extracted features and domain-specific information. Meanwhile, this transferring derives a class-conditional distribution alignment mathematically. By providing an interpretable empirical estimation of the conditional dependence, it is clear that the class-conditional information is sufficiently considered to learn the class-conditioned domain-invariant features. We also derive an informative upper bound of the target error based on the class-conditional distributions, which provide a theoretical insight into our proposal. Extensive experiments validate the effectiveness of our MCI in dealing with UDA.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61976229, 62376291), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023B1515020004), Open Research Projects of Zhejiang Lab (Grant No. 2021KHOABO8), Guangdong Province Key Laboratory of Computational Science at the Sun Yat-Sen University (Grant No. 2020B1212060032), and Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education.

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Pan S J, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng, 2010, 22: 1345–1359
2 Ben-David S, Blitzer J, Crammer K, et al. A theory of learning from different domains. Mach Learn, 2010, 79: 151–175
3 Gretton A, Borgwardt K M, Rasch M J, et al. A kernel two-sample test. J Mach Learn Res, 2012, 13: 723–773
4 Long M S, Cao Y, Cao Z J, et al. Transferable representation learning with deep adaptation networks. IEEE Trans Pattern Anal Mach Intell, 2019, 41: 3071–3085
5 Sun B C, Feng J S, Saeko K. Return of frustratingly easy domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2016. 2058–2065
6 Gong B Q, Shi Y, Sha F, et al. Geodesic flow kernel for unsupervised domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision, 2012. 2066–2073
7 Ganin Y, Ustinova E, Ajakan H, et al. Domain-adversarial training of neural networks. J Mach Learn Res, 2016, 17: 2096–2030
8 Tzeng E, Hoffman J, Saenko K, et al. Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 7167–7176
9 Long M S, Cao Z J, Wang J M, et al. Conditional adversarial domain adaptation. In: Proceedings of the International Conference on Neural Information Processing Systems, 2018. 1640–1650

10 Kang G L, Jiang L, Wei Y C, et al. Contrastive adaptation network for single-and multi-source domain adaptation. IEEE Trans Pattern Anal Mach Intell, 2020, 44: 1793–1804
11 Zhao H, Des Combes R T, Zhang K, et al. On learning invariant representations for domain adaptation. In: Proceedings of the International Conference on Machine Learning, 2019. 7523–7532
12 Long M S, Wang J M, Ding G G, et al. Transfer feature learning with joint distribution adaptation. In: Proceedings of the IEEE International Conference on Computer Vision, 2013. 2200–2207
13 Zhu Y C, Zhuang F Z, Wang J D, et al. Deep subdomain adaptation network for image classification. IEEE Trans Neural Netw Learn Syst, 2021, 32: 1713–1722
14 Li J J, Chen E P, Ding Z M, et al. Maximum density divergence for domain adaptation. IEEE Trans Pattern Anal Mach Intell, 2021, 43: 3918–3930
15 Hu L Q, Kan M N, Shan S G, et al. Unsupervised domain adaptation with hierarchical gradient synchronization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 4043–4052
16 Le T, Nguyen T, Ho N, et al. LAMDA: label matching deep domain adaptation. In: Proceedings of the International Conference on Machine Learning, 2021. 6043–6054
17 Li S, Xie M X, Lv F R, et al. Semantic concentration for domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision, 2021. 9102–9111
18 Sun T, Lu C, Zhang T S, et al. Safe self-refinement for transformer-based domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022. 7191–7200
19 Zhang J Y, Huang J X, Tian Z C, et al. Spectral unsupervised domain adaptation for visual recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022. 9829–9840
20 Saito K, Watanabe K, Ushiku Y, et al. Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 3723–3732
21 Zuo L, Jing M M, Li J J, et al. Challenging tough samples in unsupervised domain adaptation. Pattern Recogn, 2021, 110: 107540
22 Zhang Y B, Deng B, Tang H, et al. Unsupervised multi-class domain adaptation: theory, algorithms, and practice. IEEE Trans Pattern Anal Mach Intell, 2020, 2066–2073
23 Zhang Y L, Jing C X, Lin H X, et al. Hard class rectification for domain adaptation. Knowl-Based Syst, 2021, 222: 107011
24 Liang J, Hu D P, Feng J S. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In: Proceedings of the International Conference on Machine Learning, 2020. 6028–6039
25 Li S, Xie B H, Lin Q X, et al. Generalized domain conditioned adaptation network. IEEE Trans Pattern Anal Mach Intell, 2021, 4093–4109
26 Xu T K, Chen W H, Wang P C, et al. CDTrans: cross-domain transformer for unsupervised domain adaptation. In: Proceedings of the International Conference on Learning Representations, 2022
27 Yang J Y, Liu J J, Xu N, et al. TVT: transferable vision transformer for unsupervised domain adaptation. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2023. 520–530
28 Fukumizu K, Gretton A, Sun X, et al. Kernel measures of conditional dependence. In: Proceedings of the International Conference on Neural Information Processing Systems, 2008. 489–496
29 Courty N, Flamary R, Tuia D, et al. Optimal transport for domain adaptation. IEEE Trans Pattern Anal Mach Intell, 2017, 39: 1853–1865
30 Zhang Z, Wang M Z, Nehorai A. Optimal transport in reproducing kernel Hilbert spaces: theory and applications. IEEE Trans Pattern Anal Mach Intell, 2020, 42: 1741–1754
31 Yan K, Kou L, Zhang D. Learning domain-invariant subspace using domain features and independence maximization. IEEE Trans Cybern, 2018, 48: 288–299
32 Liu L, Pal S, Harchaoui Z. Entropy regularized optimal transport independence criterion. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2022. 11247–11279
33 Deng Z J, Luo Y C, Zhu J. Cluster alignment with a teacher for unsupervised domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 9944–9953
34 Chen C Q, Xie W P, Huang W B, et al. Progressive feature alignment for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 627–636
35 Luo Y W, Ren C X, Dai D Q, et al. Unsupervised domain adaptation via discriminative manifold propagation. IEEE Trans Pattern Anal Mach Intell, 2022, 44: 1653–1669
36 Ren C X, Luo Y W, Dai D Q. BuresNet: conditional Bures metric for transferable representation learning. IEEE Trans Pattern Anal Mach Intell, 2022, 1–16
37 Lu Y W, Zhu Q, Zhang B, et al. Weighted correlation embedding learning for domain adaptation. IEEE Trans Image Process, 2022, 31: 5303–5316
38 Baker C R. Joint measures and cross-covariance operators. Trans Amer Math Soc, 1973, 186: 273–289
39 Bach F R, Jordan M I. Kernel independent component analysis. J Mach Learn Res, 2002, 3: 1–48
40 Fine S, Scheinberg K. Efficient SVM training using low-rank kernel representations. J Mach Learn Res, 2001, 2: 243–264
41 Koby C, Michael K, Jennifer W. Learning from multiple sources. J Mach Learn Res, 2008, 9: 1757–1774
42 Caputo B, Müller H, Martinez-Gomez J, et al. ImageCLEF 2014: overview and analysis of the results. In: Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages, 2014. 192–211
43 Saenko K, Kulis B, Fritz M, et al. Adapting visual category models to new domains. In: Proceedings of the European Conference on Computer Vision, 2010. 213–226
44 Venkateswara H, Eusebio J, Chakraborty S, et al. Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 5018–5027
45 Peng X, Usman B, Kaushik N, et al. VisDA: the visual domain adaptation challenge. 2017. ArXiv:1710.06924
46 Peng X C, Bai Q X, Xia X D, et al. Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 1406–1415
47 Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16×16 words: transformers for image recognition at scale. 2020. ArXiv:2010.11929
48 He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778
49 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Proceedings of the International Conference on Neural Information Processing Systems, 2012. 1097–1105
50 Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–255
51 van der Maaten L, Hinton G. Visualizing data using t-SNE. J Mach Learn Res, 2008, 9: 2579–2605