Open Set Domain Adaptation via Known Joint Distribution Matching and Unknown Classification Risk Reformulation

Sentao Chen, Ping Xuan, and Lifang He, *Member, IEEE*

***Abstract*—Open Set Domain Adaptation (OSDA) is an important problem in machine learning and computer vision. In OSDA, one is given a labeled dataset from a source domain (source joint distribution) and an unlabeled dataset from a target domain (target joint distribution), where the target domain contains not only the known classes presented in the source domain but also the unknown class. The goal of OSDA is to train a neural network with minimal target classification risk. From the statistical learning perspective, there are two fundamental challenges in this problem: (1) the source-target joint distribution difference regarding the known classes and (2) the target classification risk estimation regarding the unknown class. Although prior works have proposed various sophisticated solutions to the problem and achieved inspiring experimental results, they do not fully resolve these two challenges. In this paper, we introduce a principled approach named Known Joint Distribution Matching and Unknown Classification Risk Reformulation (KMUR). KMUR tackles the first challenge by matching the source joint distribution to the target known joint distribution such that the distribution difference can be reduced, and addresses the second challenge by reformulating the target unknown classification risk such that the reformulated risk can be estimated on the unlabeled target and source data. To be specific, we exploit cross entropy as the classification loss and Triangular Discrimination (TD) distance as the joint distribution matching loss. Since the TD distance needs to be estimated from data, we develop an innovative technique named Least Squares Triangular Discrimination Estimation (LSTDE), which casts the estimation into least squares classification. To achieve the OSDA goal, we train the network to minimize the estimations of target classification risk and TD distance. Experiments on benchmark and real world datasets confirm the effectiveness of our approach. The introductory video and PyTorch code are available on GitHub (https://github.com/sentaochen/Known-Joint-Distribution-Matching-and-Unknown-Classification-Risk-Reformulation). Interested readers are also welcome to visit https://github.com/sentaochen for more source code on domain adaptation, multi-source domain adaptation, partial domain adaptation, and domain generalization approaches.**

***Index Terms*—Distribution matching, open set classification, risk minimization, statistical estimation.**

This work was supported in part by the National Natural Science Foundation of China under Grants 62106137, and 62372282, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515012954, and in part by Shantou University under Grant NTF21035. *(Corresponding author: Sentao Chen.)*

Sentao Chen is with the Department of Computer Science and Technology, School of Mathematics and Computer Science, Shantou University, Shantou 515063, China (e-mail: sentaochenmail@gmail.com).

Ping Xuan is with the School of Cyberspace Security, Hainan University, Haikou 570228, China.

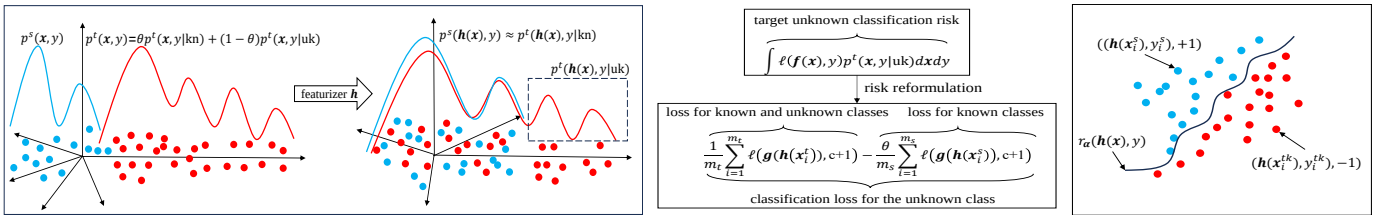
Lifang He is with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, Pennsylvania 18015, USA.

I. Introduction

In supervised classification, it is generally assumed that the training joint distribution $p^{tr}(\boldsymbol{x}, y)$ and testing joint distribution $p^{te}(\boldsymbol{x}, y)$ are identical and share the same class label space, where $\boldsymbol{x}$ are the input features and y is the class label [1]–[3]. Given a labeled training dataset $\{(\boldsymbol{x}_i^{tr}, y_i^{tr})\}_{i=1}^{m_{tr}}$ from the training joint distribution $p^{tr}(\boldsymbol{x}, y)$, the goal of supervised classification is to train a neural network $\boldsymbol{f}(\boldsymbol{x})$ with minimal testing classification risk $\int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^{te}(\boldsymbol{x}, y) d\boldsymbol{x} dy$, where ℓ is the classification loss [1]–[3]. Since the training and testing joint distributions are identical and share the same label space, the testing classification risk can therefore be directly estimated as the average training classification loss $\frac{1}{m_{tr}} \sum_{i=1}^{m_{tr}} \ell(\boldsymbol{f}(\boldsymbol{x}_i^{tr}), y_i^{tr})$, and the network can be practically trained by minimizing this loss. This is known as the Empirical Risk Minimization (ERM) principle [1] in statistical learning.

Open Set Domain Adaptation (OSDA) [4] extends supervised classification by relaxing the assumptions of identical joint distribution and identical class label space. In OSDA, the source domain (source joint distribution) $p^s(\boldsymbol{x}, y)$ and the target domain (target joint distribution) $p^t(\boldsymbol{x}, y)$ are different, and the target label space is also different from the source one and contains it as a subset. This means that besides the known classes presented in the source domain, the target domain also contains additional unknown classes. Since there is no prior knowledge on the unknown classes, following mainstream OSDA works [4]–[7], this paper combines all the unknown classes into a single class. Given a labeled source dataset and an unlabeled target dataset containing the unknown class, the goal of OSDA is to train a neural network $\boldsymbol{f}(\boldsymbol{x})$ with minimal target classification risk $\int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^t(\boldsymbol{x}, y) d\boldsymbol{x} dy$. Namely, the trained network should well classify the unlabeled target data into one of the known classes or the unknown class.

From the statistical learning perspective, there are two fundamental challenges in OSDA. One is the difference between source joint distribution $p^s(\boldsymbol{x}, y)$ and target known joint distribution $p^t(\boldsymbol{x}, y|\mathrm{kn})$, *i.e.*, $p^s(\boldsymbol{x}, y) \neq p^t(\boldsymbol{x}, y|\mathrm{kn})$. This challenge hinders the neural network from classifying the unlabeled target data into the correct known classes, since the network is trained and tested on different distributions [8]–[10]. The other one is the estimation of target unknown classification risk, whose result is a natural optimization objective to train the network for classifying the unknown. In general, estimating the classification risk requires the input features and class labels to compute the average classification loss [1]–[3]. However, in



(a) Known joint distribution matching (b) Unknown classification risk reformulation (c) TD distance estimation

Fig. 1: Illustration of our KMUR approach and LSTDE technique. (a) KMUR matches source joint distribution $p^s(\boldsymbol{x}, y)$ to target known joint distribution $p^t(\boldsymbol{x}, y|\mathrm{kn})$ via featurizer $\boldsymbol{h}$. (b) KMUR reformulates target unknown classification risk, and estimates the reformulated risk as the difference between the average loss on unlabeled target data (of known and unknown classes) and the average loss on unlabeled source data (of known classes). (c) LSTDE estimates the TD distance between $p^s(\boldsymbol{h}(\boldsymbol{x}), y)$ and $p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})$ via kernel-based least squares classification, where the kernel-based model $r_{\boldsymbol{\alpha}}(\boldsymbol{h}(\boldsymbol{x}), y)$ classifies the source sample $(\boldsymbol{h}(\boldsymbol{x}_i^s), y_i^s)$ into positive class $+1$ and the target known sample $(\boldsymbol{h}(\boldsymbol{x}_i^{tk}), y_i^{tk})$ into negative class -1.

OSDA, the unknown class labels are not given, which poses a challenge in estimating the target unknown classification risk.

Mainstream OSDA works match marginal distributions [4], [5], [7] or class-conditional distributions [11]–[13] regarding the known classes between domains, and recognize the target unknown class via setting threshold value [4] or designing weighting mechanisms [5], [7], [14]. For example, Saito *et al*. [4] introduced an adversarial learning method that matches the source marginal distribution $p^s(\boldsymbol{x})$ to the target known marginal distribution $p^t(\boldsymbol{x}|\mathrm{kn})$ in the feature space, and recognizes the target unknown class by a predefined threshold value. Liu *et al*. [11] matched the source class-conditional distribution $p^s(\boldsymbol{x}|y)$ to the target known class-conditional distribution $p^t(\boldsymbol{x}|y, \mathrm{kn})$ in the subspace, and trained the classifier by minimizing the average source classification loss and the average classification loss computed on the pseudo labeled target unknown data. Jang *et al*. [7] optimized an adversarial learning objective to align the source and target known marginal distributions and segregate the target unknown marginal distribution $p^t(\boldsymbol{x}|\mathrm{uk})$, and trained the neural network by minimizing the average source classification loss and the weighted average target classification loss. Besides, there also exist works that propose to address OSDA via self-supervised learning [15], causal alignment [16], or other strategies [17], [18]. While these works have achieved inspiring experimental results, in essence, they do not fully resolve the two fundamental challenges in OSDA. As a result, it is not clear whether their solutions can achieve the OSDA goal and train a neural network with minimal target classification risk.

Our contributions. Our research in this paper aims to address the two fundamental challenges and achieve the OSDA goal. We use a basic neural network that contains a featurizer and a classifier. Starting from the OSDA goal, we express the target classification risk as the mixture of two risks: the target known classification risk and the target unknown classification risk, which are associated with the target known joint distribution $p^t(\boldsymbol{x}, y|\mathrm{kn})$ and the target unknown joint distribution $p^t(\boldsymbol{x}, y|\mathrm{uk})$, respectively. To address the first challenge, we exploit the featurizer to match the source joint distribution $p^s(\boldsymbol{x}, y)$ to the target known joint distribution $p^t(\boldsymbol{x}, y|\mathrm{kn})$ in the feature space to reduce their difference. Since the joint distributions are matched, the target known classification risk can therefore be estimated as the average source classification loss computed on the labeled source data. To address the second challenge, we reformulate the target unknown classification risk in terms of the target marginal distribution $p^t(\boldsymbol{x})$ and target known marginal distribution $p^t(\boldsymbol{x}|\mathrm{kn})$. Since the previous joint distribution matching also leads to the matching of target known marginal distribution $p^t(\boldsymbol{x}|\mathrm{kn})$ and source marginal distribution $p^s(\boldsymbol{x})$, the reformulated risk can therefore be estimated as the difference of two average classification losses computed on the unlabeled target and source data, without requiring the target unknown class labels. By combining the estimation results of target known and unknown risks, we derive the estimated target classification risk. In the reminder, we name this OSDA approach **K**nown Joint Distribution **M**atching and **U**nknown Classification Risk **R**eformulation (KMUR). See Figs. 1(a) and 1(b) for the illustration of KMUR. To be specific, we exploit cross entropy as the classification loss and Triangular Discrimination (TD) distance [19] as the joint distribution matching loss. Since the source joint distribution and target known joint distribution are unknown in practice, the TD distance between them is also unknown and needs to be estimated from data. We show that the estimation of TD distance can be elegantly cast into the kernel-based least squares classification problem that has an analytic solution. With this solution, we manage to express the estimated TD distance as an explicit function of the featurizer. Hence, we can directly optimize the featurizer to minimize the estimated TD distance and match joint distributions in the feature space. In the reminder, we name this estimation technique **L**east **S**quares **T**riangular **D**iscrimination **E**stimation (LSTDE). See Fig. 1(c) for the illustration of LSTDE. Of course, it is also feasible to use other statistical distances as the joint distribution matching loss, *e.g*., the related Kullback-Leibler (KL) divergence and Jensen-Shannon (JS) divergence. However, as practiced in [5], [7], [20], [21], optimizing the featurizer to minimize these distances usually results in adversarial training, which is time-consuming and unstable [22]. Considering such drawback, we use TD distance to free us from the challenging adversarial training. To achieve the OSDA goal, we train the neural network to minimize both the estimated target classification

risk and the estimated TD distance. In summary, we list the main contributions of this research as follows:

1) We introduce the KMUR approach to solve the OSDA challenges. Our approach matches the source and target joint distributions regarding the known classes in the feature space, and reformulates the target classification risk regarding the unknown class such that the reformulated risk can be estimated on unlabeled target and source data.
2) We develop the LSTDE technique to estimate the TD distance (joint distribution matching loss) from data. Our technique casts the estimation into least squares classification, and expresses the estimated TD distance as an explicit function of the featurizer, which allows us to directly optimize the featurizer to minimize the distance and match joint distributions in the feature space.
3) We conduct experiments on several benchmark image datasets and a real world skin disease dataset. Our experimental results and analysis demonstrate the advantage of the proposed approach compared to other methods.

II. Methodology

A. Problem Formulation

In OSDA, there exists a source joint distribution $p^s(\boldsymbol{x}, y)$ defined on the source label space with c classes, and a target joint distribution $p^t(\boldsymbol{x}, y)$ defined on the target label space with $c+1$ classes. For the target label space, its first c classes are the known classes shared with the source label space, and its last $c+1$ class is the unknown class not presented in the source label space. This research combines all the unknown classes into a single class, since there is no prior knowledge on the unknown classes [4]–[7]. Given a labeled source dataset $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{m_s}$ from source joint distribution $p^s(\boldsymbol{x}, y)$ and an unlabeled target dataset $\mathcal{D}^u = \{\boldsymbol{x}_i^t\}_{i=1}^{m_t}$ from target joint distribution $p^t(\boldsymbol{x}, y)$, the goal of OSDA is to train a neural network $\boldsymbol{f}(\boldsymbol{x})$ with minimal target classification risk $\int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^t(\boldsymbol{x}, y) d\boldsymbol{x} dy$, where $\boldsymbol{f}(\boldsymbol{x})$ has $c+1$ outputs and ℓ is the classification loss. In other words, the trained network should well classify the unlabeled target data into one of the known classes or the unknown class. The classification rule is given by $\widehat{y} = \operatorname{argmax}_{k \in \{1, \cdots, c+1\}} f_k(\boldsymbol{x})$, where $f_k(\boldsymbol{x})$ is the k-th output of $\boldsymbol{f}(\boldsymbol{x})$. In this OSDA problem, there are two challenges: (1) the source-target joint distribution difference regarding the known classes and (2) the target classification risk estimation regarding the unknown class.

B. KMUR Approach

Target classification risk. Unlike prior works [14], [16], [17] that focus on designing complex neural networks, this research uses a basic neural network $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}))$ that contains the featurizer $\boldsymbol{h}$ and classifier $\boldsymbol{g}$. The featurizer generates the feature space and the classifier yields the $c+1$ outputs. Starting from the OSDA goal, we express the target classification risk as

$$\begin{aligned} &\int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^t(\boldsymbol{x}, y) d\boldsymbol{x} dy \\ &= p^t(\mathrm{kn}) \int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^t(\boldsymbol{x}, y|\mathrm{kn}) d\boldsymbol{x} dy \\ &\quad + p^t(\mathrm{uk}) \int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^t(\boldsymbol{x}, y|\mathrm{uk}) d\boldsymbol{x} dy \qquad (1) \\ &= \theta \int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^t(\boldsymbol{x}, y|\mathrm{kn}) d\boldsymbol{x} dy \\ &\quad + (1-\theta) \int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^t(\boldsymbol{x}, y|\mathrm{uk}) d\boldsymbol{x} dy. \qquad (2) \end{aligned}$$

Eq. (1) uses the law of total probability [23] and decomposes the target joint distribution as $p^t(\boldsymbol{x}, y) = p^t(\mathrm{kn}) p^t(\boldsymbol{x}, y|\mathrm{kn}) + p^t(\mathrm{uk}) p^t(\boldsymbol{x}, y|\mathrm{uk})$, where $p^t(\mathrm{kn})$ is the prior probability for the known classes, $p^t(\mathrm{uk})$ is the prior probability for the unknown class, and $p^t(\mathrm{kn}) + p^t(\mathrm{uk}) = 1$. Eq. (2) introduces the mixture parameter $\theta = p^t(\mathrm{kn})$ and writes the target joint distribution as a mixture distribution

$$p^t(\boldsymbol{x}, y) = \theta p^t(\boldsymbol{x}, y|\mathrm{kn}) + (1-\theta) p^t(\boldsymbol{x}, y|\mathrm{uk}). \qquad (3)$$

As a result, the target classification risk is expressed as the mixture of two risks: the target known classification risk $\int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^t(\boldsymbol{x}, y|\mathrm{kn}) d\boldsymbol{x} dy$ and the target unknown classification risk $\int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^t(\boldsymbol{x}, y|\mathrm{uk}) d\boldsymbol{x} dy$.

Known joint distribution matching. To address the first OSDA challenge, we exploit featurizer $\boldsymbol{h}$ to match the source joint distribution $p^s(\boldsymbol{x}, y)$ to the target known joint distribution $p^t(\boldsymbol{x}, y|\mathrm{kn})$ in the feature space. See Fig. 1(a) for the illustration. Accordingly, the target known classification risk can be estimated as

$$\begin{aligned} &\int \ell(\boldsymbol{f}(\boldsymbol{x}), y) p^t(\boldsymbol{x}, y|\mathrm{kn}) d\boldsymbol{x} dy \\ &= \int \ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x})), y) p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn}) d\boldsymbol{h}(\boldsymbol{x}) dy \qquad (4) \\ &\approx \int \ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x})), y) p^s(\boldsymbol{h}(\boldsymbol{x}), y) d\boldsymbol{h}(\boldsymbol{x}) dy \qquad (5) \\ &\approx \frac{1}{m_s} \sum_{i=1}^{m_s} \ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^s)), y_i^s). \qquad (6) \end{aligned}$$

Eq. (5) approximates Eq. (4) via the known joint distribution matching

$$p^s(\boldsymbol{h}(\boldsymbol{x}), y) \approx p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn}). \qquad (7)$$

Eq. (6) estimates the expectation in Eq. (5) via data average, and computes the average source classification loss on the labeled source data. Obviously, Eq. (6) is a natural optimization objective to train the neural network $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}))$ for classifying the known classes.

Unknown classification risk reformulation. To address the second OSDA challenge, we reformulate and estimate the

target unknown classification risk as

$$\int \ell(\boldsymbol{f}(\boldsymbol{x}), y)p^t(\boldsymbol{x}, y|\mathrm{uk})d\boldsymbol{x}dy$$
$$= \int \ell(\boldsymbol{f}(\boldsymbol{x}), \mathrm{c{+}1})p^t(\boldsymbol{x}|\mathrm{uk})d\boldsymbol{x} \tag{8}$$
$$= \frac{1}{1-\theta}\int \ell(\boldsymbol{f}(\boldsymbol{x}), \mathrm{c{+}1})p^t(\boldsymbol{x})d\boldsymbol{x} - \frac{\theta}{1-\theta}\int \ell(\boldsymbol{f}(\boldsymbol{x}), \mathrm{c{+}1})p^t(\boldsymbol{x}|\mathrm{kn})d\boldsymbol{x} \tag{9}$$
$$\approx \frac{1}{1-\theta}\int \ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x})), \mathrm{c{+}1})p^t(\boldsymbol{h}(\boldsymbol{x}))d\boldsymbol{h}(\boldsymbol{x}) - \frac{\theta}{1-\theta}\int \ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x})), \mathrm{c{+}1})p^s(\boldsymbol{h}(\boldsymbol{x}))d\boldsymbol{h}(\boldsymbol{x}) \tag{10}$$
$$\approx \frac{1}{1-\theta}\frac{1}{m_t}\sum_{i=1}^{m_t}\ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^t)), \mathrm{c{+}1}) - \frac{\theta}{1-\theta}\frac{1}{m_s}\sum_{i=1}^{m_s}\ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^s)), \mathrm{c{+}1}). \tag{11}$$

Eq. (8) simplifies the target unknown classification risk by the factorization $p^t(\boldsymbol{x}, y|\mathrm{uk}) = p^t(\boldsymbol{x}|\mathrm{uk})p^t(y|\boldsymbol{x}, \mathrm{uk})$, where $p^t(y|\boldsymbol{x}, \mathrm{uk})$ equals 1 if y is the unknown class c+1 or 0 if y is one of the c known classes. Eq. (9) is a crucial step in the unknown classification risk reformulation. It reformulates the simplified target unknown classification risk over $p^t(\boldsymbol{x}|\mathrm{uk})$ in terms of the target marginal distribution $p^t(\boldsymbol{x})$ and the target known marginal distribution $p^t(\boldsymbol{x}|\mathrm{kn})$, since $p^t(\boldsymbol{x}|\mathrm{uk})$, $p^t(\boldsymbol{x})$, and $p^t(\boldsymbol{x}|\mathrm{kn})$ are related by the equation $p^t(\boldsymbol{x}|\mathrm{uk}) = \frac{1}{1-\theta}p^t(\boldsymbol{x}) - \frac{\theta}{1-\theta}p^t(\boldsymbol{x}|\mathrm{kn})$. This equation is derived from Eq. (3), which implies that $p^t(\boldsymbol{x}) = \theta p^t(\boldsymbol{x}|\mathrm{kn}) + (1-\theta)p^t(\boldsymbol{x}|\mathrm{uk})$. Eq. (10) approximates Eq. (9) via the known marginal distribution matching $p^t(\boldsymbol{h}(\boldsymbol{x})|\mathrm{kn}) \approx p^s(\boldsymbol{h}(\boldsymbol{x}))$, which is a result from Eq. (7). To be specific, by marginalizing both sides of Eq. (7) over y: $\int p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})dy \approx \int p^s(\boldsymbol{h}(\boldsymbol{x}), y)dy$, we have $p^t(\boldsymbol{h}(\boldsymbol{x})|\mathrm{kn}) \approx p^s(\boldsymbol{h}(\boldsymbol{x}))$. Eq. (11) estimates the two expectations in Eq. (10) via data averages, and derives the estimated target unknown classification risk as the difference of two average classification losses computed on the unlabeled target data and unlabeled source data. See Fig. 1(b) for the illustration. Obviously, without requiring the target unknown class labels, we successfully derive the estimated target unknown classification risk in Eq. (11). It is a natural optimization objective to train the network $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}))$ for classifying the unknown class.

Estimated target classification risk. We combine the results from Eqs. (6) and (11) to estimate the target classification risk as

$$\int \ell(\boldsymbol{f}(\boldsymbol{x}), y)p^t(\boldsymbol{x}, y)d\boldsymbol{x}dy \approx \frac{\theta}{m_s}\sum_{i=1}^{m_s}\ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^s)), y_i^s) + \Big(\frac{1}{m_t}\sum_{i=1}^{m_t}\ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^t)), \mathrm{c{+}1}) - \frac{\theta}{m_s}\sum_{i=1}^{m_s}\ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^s)), \mathrm{c{+}1})\Big), \tag{12}$$

where $\boldsymbol{h}$ is the featurizer that matches joint distributions in the feature space such that $p^s(\boldsymbol{h}(\boldsymbol{x}), y) \approx p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})$.

C. LSTDE Technique

Triangular Discrimination (TD) distance. We exploit the TD distance [19] as the joint distribution matching loss. Mathematically, the TD distance between source joint distribution $p^s(\boldsymbol{h}(\boldsymbol{x}), y)$ and target known joint distribution $p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})$ is defined as

$$\mathrm{TD}\big(p^s(\boldsymbol{h}(\boldsymbol{x}), y), p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})\big) = \int \frac{\big(p^s(\boldsymbol{h}(\boldsymbol{x}), y) - p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})\big)^2}{p^s(\boldsymbol{h}(\boldsymbol{x}), y) + p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})}d\boldsymbol{h}(\boldsymbol{x})dy. \tag{13}$$

It is symmetric, lower bounded by 0, and upper bounded by 2. Importantly, when $p^s(\boldsymbol{h}(\boldsymbol{x}), y) = p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})$, the TD distance reduces to zero. Therefore, the TD distance is a suitable joint distribution matching loss and minimizing it with respect to the featurizer $\boldsymbol{h}$ matches joint distributions $p^s(\boldsymbol{h}(\boldsymbol{x}), y)$ and $p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})$. Besides, the TD distance is related to the Kullback-Leibler (KL) divergence and Jensen-Shannon (JS) divergence, since they all belong to the f-divergence family, which is a popular class of statistical distance in machine learning [21], [22], [24], [25]. In our research, it is also feasible to use the KL or JS divergence. However, minimizing them with respect to the featurizer usually results in adversarial training [5], [7], [20], [21], which is time-consuming and unstable [22]. By contrast, we find that the TD distance can free us from adversarial training, since it can be estimated as an explicit function of the featurizer. See our result in Eq. (21). This charming characteristic motivates us to select the TD distance, instead of the KL or JS divergence. We know that the Maximum Mean Discrepancy (MMD) [26] can also avoid adversarial training. However, to our best knowledge, MMD has not yet been well defined to measure the distance between two joint (features and class label) distributions in theory.

TD distance estimation via least squares classification. Since source joint distribution $p^s(\boldsymbol{h}(\boldsymbol{x}), y)$ and target known joint distribution $p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})$ are both unknown, the TD distance between them is also unknown and needs to be estimated from data. Below, we show the TD distance estimation can be elegantly cast into least squares classification.

$$\mathrm{TD}\big(p^s(\boldsymbol{h}(\boldsymbol{x}), y), p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})\big) = \int \frac{\big(p^s(\boldsymbol{h}(\boldsymbol{x}), y) + p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})\big)^2}{p^s(\boldsymbol{h}(\boldsymbol{x}), y) + p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})}d\boldsymbol{h}(\boldsymbol{x})dy - \int \frac{4p^s(\boldsymbol{h}(\boldsymbol{x}), y)p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})}{p^s(\boldsymbol{h}(\boldsymbol{x}), y) + p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})}d\boldsymbol{h}(\boldsymbol{x})dy \tag{14}$$
$$= 2 - \int \frac{4p^s(\boldsymbol{h}(\boldsymbol{x}), y)p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})}{p^s(\boldsymbol{h}(\boldsymbol{x}), y) + p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})}d\boldsymbol{h}(\boldsymbol{x})dy \tag{15}$$
$$= 2 - \min_r \Big(\int \big(r(\boldsymbol{h}(\boldsymbol{x}), y) - 1\big)^2 p^s(\boldsymbol{h}(\boldsymbol{x}), y)d\boldsymbol{h}(\boldsymbol{x})dy + \int \big(r(\boldsymbol{h}(\boldsymbol{x}), y) + 1\big)^2 p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})d\boldsymbol{h}(\boldsymbol{x})dy\Big) \tag{16}$$
$$\approx 2 - \min_{\boldsymbol{\alpha}} \Big(\frac{1}{m_s}\sum_{i=1}^{m_s}\big(r_{\boldsymbol{\alpha}}(\boldsymbol{h}(\boldsymbol{x}_i^s), y_i^s) - 1\big)^2 + \frac{1}{m_{tk}}\sum_{i=1}^{m_{tk}}\big(r_{\boldsymbol{\alpha}}(\boldsymbol{h}(\boldsymbol{x}_i^{tk}), y_i^{tk}) + 1\big)^2\Big). \tag{17}$$

Eq. (14) rewrites the original definition of TD distance in Eq. (13) by using the equation $(a-b)^2 = (a+b)^2 - 4ab$, where $a = p^s(\boldsymbol{h}(\boldsymbol{x}), y)$ and $b = p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})$. Eq. (15) is due to the fact that $p^s(\boldsymbol{h}(\boldsymbol{x}), y)$ and $p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})$ are probability distributions and as a result $\int \big(p^s(\boldsymbol{h}(\boldsymbol{x}), y) + p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})\big)d\boldsymbol{h}(\boldsymbol{x})dy = 2$. Note that, Eq. (15) also suggests that the TD distance is upper bounded by 2, since the

second term is non-negative. Eq. (16) leverages the correspondence between least squares loss and TD distance [27], and expresses the TD distance as a minimization problem with respect to the function $r(\boldsymbol{h}(\boldsymbol{x}), y)$. The minimal value is attained when $r(\boldsymbol{h}(\boldsymbol{x}), y) = \frac{p^s(\boldsymbol{h}(\boldsymbol{x}),y) - p^t(\boldsymbol{h}(\boldsymbol{x}),y|\mathrm{kn})}{p^s(\boldsymbol{h}(\boldsymbol{x}),y) + p^t(\boldsymbol{h}(\boldsymbol{x}),y|\mathrm{kn})}$. Eq. (17) estimates the two expectations in Eq. (16) by the averages of the labeled source dataset $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{m_s}$ from $p^s(\boldsymbol{x}, y)$ and the labeled target known dataset $\mathcal{D}^{tk} = \{(\boldsymbol{x}_i^{tk}, y_i^{tk})\}_{i=1}^{m_{tk}}$ from $p^t(\boldsymbol{x}, y|\mathrm{kn})$. Here, we assume that the labeled target known dataset is available. The process of obtaining this dataset is detailed in the next subsection. Besides, inspired by the RBF network [28], Eq. (17) utilizes a kernel-based model

$$r_{\boldsymbol{\alpha}}(\boldsymbol{h}(\boldsymbol{x}), y) = \sum_{i=1}^{m_s} \alpha_i k(\boldsymbol{h}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x}_i^s)) \delta(y, y_i^s) \tag{18}$$

for $r(\boldsymbol{h}(\boldsymbol{x}), y)$, where $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_{m_s})^\top$ are the model parameters to be learned, $k(\boldsymbol{h}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x}_i^s)) = \exp(-\|\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{h}(\boldsymbol{x}_i^s)\|_2^2/\sigma)$ is the RBF kernel with kernel width σ (> 0), and $\delta(y, y_i^s)$ is the delta kernel that evaluates 1 if $y = y_i^s$ and 0 otherwise. In particular, the kernel width σ is set to the median pairwise squared distances on the unlabeled source data in the experiments. Similar to the RBF network [28], on the one hand, the model in Eq. (18) is non-linear in its input $(\boldsymbol{h}(\boldsymbol{x}), y)$, which makes the model flexible and expressive. On the other hand, the model is linear in its parameters $\boldsymbol{\alpha}$, which, together with the least squares loss from Eq. (16), makes it possible to learn $\boldsymbol{\alpha}$ in an analytic manner. Clearly, the minimization problem in Eq. (17) is a special kernel-based least squares classification problem, where the numbers of positive and negative samples are the same. In this problem, the kernel-based classification model $r_{\boldsymbol{\alpha}}(\boldsymbol{h}(\boldsymbol{x}), y)$ classifies the source sample $(\boldsymbol{h}(\boldsymbol{x}_i^s), y_i^s)$ into positive class $+1$ and the target known sample $(\boldsymbol{h}(\boldsymbol{x}_i^{tk}), y_i^{tk})$ into negative class -1. See Fig. 1(c) for the illustration. From Eq. (17), we learn that the harder the source samples and target known samples can be classified, the larger the average least squares classification loss will become, and the smaller the TD distance between source joint distribution $p^s(\boldsymbol{h}(\boldsymbol{x}), y)$ and target known joint distribution $p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})$ will be.

Estimated TD distance. We solve the least squares classification problem in Eq. (17), and derive the estimated TD distance as

$$\begin{aligned}
&\widehat{\mathrm{TD}}\big(p^s(\boldsymbol{h}(\boldsymbol{x}), y), p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})\big) \\
&= \max_{\boldsymbol{\alpha}} \Big(\frac{2}{m_s} \sum_{i=1}^{m_s} r_{\boldsymbol{\alpha}}(\boldsymbol{h}(\boldsymbol{x}_i^s), y_i^s) - \frac{2}{m_{tk}} \sum_{i=1}^{m_{tk}} r_{\boldsymbol{\alpha}}(\boldsymbol{h}(\boldsymbol{x}_i^{tk}), y_i^{tk}) \\
&\quad - \frac{1}{m_s} \sum_{i=1}^{m_s} r_{\boldsymbol{\alpha}}(\boldsymbol{h}(\boldsymbol{x}_i^s), y_i^s)^2 - \frac{1}{m_{tk}} \sum_{i=1}^{m_{tk}} r_{\boldsymbol{\alpha}}(\boldsymbol{h}(\boldsymbol{x}_i^{tk}), y_i^{tk})^2 \Big) \qquad (19) \\
&= \max_{\boldsymbol{\alpha}} \Big(2\big[\frac{1}{m_s} \mathbf{1}_{m_s}^\top \boldsymbol{A} - \frac{1}{m_t} \mathbf{1}_{m_{tk}}^\top \boldsymbol{B}\big] \boldsymbol{\alpha} \\
&\qquad - \boldsymbol{\alpha}^\top \big[\frac{1}{m_s} \boldsymbol{A}^\top \boldsymbol{A} + \frac{1}{m_t} \boldsymbol{B}^\top \boldsymbol{B}\big] \boldsymbol{\alpha} \Big) \qquad (20) \\
&= 2\big[\frac{1}{m_s} \mathbf{1}_{m_s}^\top \boldsymbol{A} - \frac{1}{m_t} \mathbf{1}_{m_{tk}}^\top \boldsymbol{B}\big] \widehat{\boldsymbol{\alpha}} \\
&\qquad - \widehat{\boldsymbol{\alpha}}^\top \big[\frac{1}{m_s} \boldsymbol{A}^\top \boldsymbol{A} + \frac{1}{m_{tk}} \boldsymbol{B}^\top \boldsymbol{B}\big] \widehat{\boldsymbol{\alpha}}. \qquad (21)
\end{aligned}$$

Eq. (19) expands the quadratic terms in Eq. (17) and rewrites the minimization problem as a maximization problem. Eq. (20) introduces new notations to explicitly present the quadratic maximization problem, where $\mathbf{1}_{m_s}$, $\mathbf{1}_{m_{tk}}$ are two column vectors of ones with m_s, m_{tk} dimensions, and $\boldsymbol{A} \in \mathbb{R}^{m_s \times m_s}$, $\boldsymbol{B} \in \mathbb{R}^{m_{tk} \times m_s}$ are two kernel matrices defined as

$$(\boldsymbol{A})_{ij} = k(\boldsymbol{h}(\boldsymbol{x}_i^s), \boldsymbol{h}(\boldsymbol{x}_j^s)) \delta(y_i^s, y_j^s), \tag{22}$$

$$(\boldsymbol{B})_{ij} = k(\boldsymbol{h}(\boldsymbol{x}_i^{tk}), \boldsymbol{h}(\boldsymbol{x}_j^s)) \delta(y_i^{tk}, y_j^s). \tag{23}$$

Finally, Eq. (21) solves the max problem with analytic solution

$$\begin{aligned}
\widehat{\boldsymbol{\alpha}} = &\Big(\frac{1}{m_s} \boldsymbol{A}^\top \boldsymbol{A} + \frac{1}{m_t} \boldsymbol{B}^\top \boldsymbol{B} + \epsilon \mathbf{I}\Big)^{-1} \\
&\times \Big[\frac{1}{m_s} \boldsymbol{A}^\top \mathbf{1}_{m_s} - \frac{1}{m_t} \boldsymbol{B}^\top \mathbf{1}_{m_{tk}}\Big]. \qquad (24)
\end{aligned}$$

Here, a diagonal matrix $\epsilon \mathbf{I}$ is included in the solution to ensure numerical stability, where ϵ is a small positive value set to 0.001 in our experiments and $\mathbf{I}$ is the identity matrix. For clarity, we summarize our TD distance estimation technique in Algorithm 1. In Subsection IV-C "**TD distance estimation**", we demonstrate that Algorithm 1 yields reliable estimation results. With the analytic solution $\widehat{\boldsymbol{\alpha}}$, the estimated TD distance in Eq. (21) is expressed as an explicit function of the featurizer $\boldsymbol{h}$, since $\boldsymbol{A}$, $\boldsymbol{B}$, and $\widehat{\boldsymbol{\alpha}}$ are all dependent on $\boldsymbol{h}$. Hence, we can directly optimize the featurizer to minimize the estimated TD distance and match joint distributions in the feature space.

Algorithm 1 TD Distance Estimation via LSTDE Technique

Input: Labeled datasets $\mathcal{D}^s, \mathcal{D}^{tk}$.
Output: Estimated value $\widehat{\mathrm{TD}}\big(p^s(\boldsymbol{h}(\boldsymbol{x}), y), p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})\big)$.
1: Build matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and vector $\widehat{\boldsymbol{\alpha}}$ by Eqs. (22)-(24).
2: Obtain $\widehat{\mathrm{TD}}\big(p^s(\boldsymbol{h}(\boldsymbol{x}), y), p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})\big)$ via Eq. (21).

D. Network Training

Network training objective. To achieve the OSDA goal, we use cross entropy loss ℓ_{ce} as the classification loss, and train the network $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}))$ to minimize the estimated target classification risk in Eq. (12) and the estimated TD distance in Eq. (21). The optimization problem is formulated as

$$\begin{aligned}
\min_{\boldsymbol{g},\boldsymbol{h}} \; &\underbrace{\frac{\theta}{m_s} \sum_{i=1}^{m_s} \ell_{\mathrm{ce}}(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^s)), y_i^s)}_{\text{classification loss for known classes}} \\
&+ \underbrace{\Big| \frac{1}{m_t} \sum_{i=1}^{m_t} \ell_{\mathrm{ce}}(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^t)), \mathrm{c{+}1}) - \frac{\theta}{m_s} \sum_{i=1}^{m_s} \ell_{\mathrm{ce}}(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^s)), \mathrm{c{+}1}) \Big|}_{\text{classification loss for unknown class}} \\
&+ \underbrace{\lambda_{\mathrm{TD}} \widehat{\mathrm{TD}}\big(p^s(\boldsymbol{h}(\boldsymbol{x}), y), p^t(\boldsymbol{h}(\boldsymbol{x}), y|\mathrm{kn})\big)}_{\text{joint distribution matching loss for known classes}}, \qquad (25)
\end{aligned}$$

where $\theta \in (0, 1)$ and $\lambda_{\mathrm{TD}} (> 0)$ are the parameters. Theoretically, the classification loss for the unknown class should be non-negative. However, due to the data average approximation and the flexibility of neural networks, the loss can become negative during training and adversely affect performance. Similar issue was also encountered and addressed in the previous study of Lu *et al.* [29]. Following [29], we wrap the loss by the absolute value function to ensure its non-negativity. As shown in (25), our objective function consists of three losses: the classification loss for known classes, the classification loss for unknown class, and the joint distribution matching loss for known classes. In Subsection IV-C "**Ablation study**", we

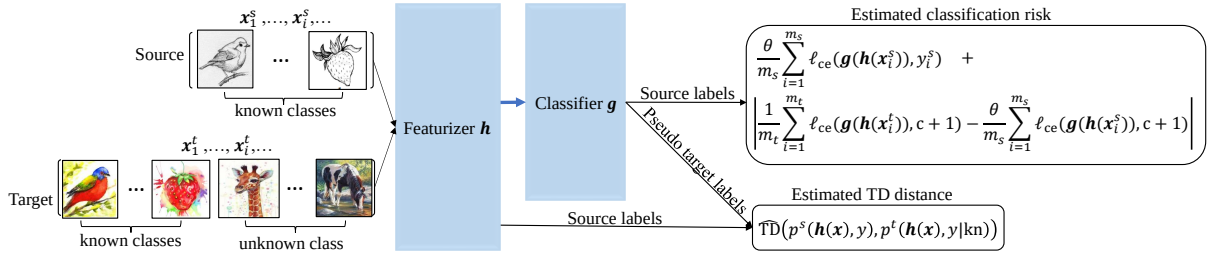


Fig. 2: Illustration of the input data $\boldsymbol{x}$, the neural network architecture $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}))$, and the losses in our KMUR approach.

show that all the three losses are indispensable. For clarity, we illustrate our approach with objective function (25) in Fig. 2.

Network training algorithm. Recall from Subsection II-C that the estimated TD distance depends on the labeled target known dataset $\mathcal{D}^{tk} = \{(\boldsymbol{x}_i^{tk}, y_i^{tk})\}_{i=1}^{m_{tk}}$, which is not available under the OSDA problem setting. To address this issue, we follow the popular pseudo labeling idea [6], [11]–[13] to generate the labeled target known dataset from the unlabeled target dataset $\mathcal{D}^u = \{\boldsymbol{x}_i^t\}_{i=1}^{m_t}$. As a form of self-training, pseudo labeling has been theoretically proven to be effective for improving the target classification accuracy even when the domain shift is large [30]. Furthermore, its effectiveness has also been empirically verified by a series of works [6], [11]–[13], [31]–[34]. In our research, the pseudo labeling idea is practiced via the following two-step procedure. In the first step, we use the Adam optimizer [35] to minimize objective function (25) without the estimated TD distance term to train the neural network. In the second step, we utilize the trained network to predict pseudo labels for $\mathcal{D}^u$ and generate the pseudo labeled target dataset $\mathcal{D}^t = \{(\boldsymbol{x}_i^t, y_i^t)\}_{i=1}^{m_t}$. The labeled target known dataset $\mathcal{D}^{tk}$ is a subset selected from $\mathcal{D}^t$ with the known class labels. We call this procedure the pre-training phase for generating the pseudo labels. In Subsection IV-C "**Pseudo labels under increasing domain shift**", we study the performance of our approach with respect to the initial pseudo labels generated under increasing domain shift, and show that our approach still works when the quality of pseudo labels decreases. With the pseudo labeled target known dataset and the resulting estimated TD distance, objective function (25) is complete and can be optimized. In the formal training phase, we again use Adam optimizer to minimize objective function (25) and train the neural network. Since the pseudo labels may not be accurate, during the formal training phase, we utilize the network to update the pseudo labels and improve their accuracy. For clarity, we summarize our network training procedure in Algorithm 2, where B is the number of iterations per training epoch. In Subsection IV-C "**Convergence analysis**", we demonstrate that this algorithm is convergent.

Algorithm 2 Neural Network Training via KMUR Approach

Input: Labeled source and unlabeled target datasets $\mathcal{D}^s$, $\mathcal{D}^u$.
Output: Trained neural network $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}))$.

% Pre-training for generating pseudo labels
while training does not end **do**
  for b in $1 : B$ **do**
    Sample minibatches $\mathcal{D}_b^s$, $\mathcal{D}_b^u$ from datasets $\mathcal{D}^s$, $\mathcal{D}^u$.
    Compute objective function (25) without the estimated TD distance on $\mathcal{D}_b^s$, $\mathcal{D}_b^u$.
    Update network parameters by the objective function.
  end for
end while
Generate pseudo labeled target dataset $\mathcal{D}^t$ by utilizing the network to label $\mathcal{D}^u$.
% Formal training
while training does not end **do**
  for b in $1 : B$ **do**
    Sample minibatches $\mathcal{D}_b^s$, $\mathcal{D}_b^t$ from datasets $\mathcal{D}^s$, $\mathcal{D}^t$.
    Compute objective function (25) on $\mathcal{D}_b^s$, $\mathcal{D}_b^t$, where the estimated TD distance is computed on $\mathcal{D}_b^s$ and $\mathcal{D}_b^{tk}$ via Algorithm 1 and $\mathcal{D}_b^{tk}$ is a subset selected from $\mathcal{D}_b^t$ with the known class labels.
    Update network parameters by the objective function.
  end for
  Update $\mathcal{D}^t$ by utilizing the network to label $\mathcal{D}^u$.
end while

III. Related Work and Discussion

A. Domain Adaptation

Domain Adaptation (DA) assumes the source and target domains share the same class label space, and aims to transfer a classification model from the source to target by matching distributions between domains [8]. This DA problem can be regarded as a simplified OSDA problem without the open set recognition challenge. Mainstream DA works propose to match the marginal distributions $p^s(\boldsymbol{x})$ and $p^t(\boldsymbol{x})$ [20]–[22], [36], the class-conditional distributions $p^s(\boldsymbol{x}|y)$ and $p^t(\boldsymbol{x}|y)$ [31], [32], [37], or more directly, the joint distributions $p^s(\boldsymbol{x}, y)$ and $p^t(\boldsymbol{x}, y)$ [33], [38], [39] between the two domains. The distributions are matched in the feature space by adversarial training [20], [21] or minimizing statistical distances between distributions [31]–[33], [36], [40], [41]. On addressing the common distribution difference challenge in DA and OSDA, our OSDA work is related to the DA works [33], [38], [39] in matching the joint distributions, since the joint distribution difference is a fundamental challenge that affects the transfer performance [8]–[10]. However, our OSDA work is different from [33], [38], [39] in that our work proposes to match the source joint distribution $p^s(\boldsymbol{x}, y)$ to the target known joint distribution $p^t(\boldsymbol{x}, y|\mathrm{kn})$, rather than the whole target joint distribution $p^t(\boldsymbol{x}, y)$. In

fact, for the OSDA problem where the target domain contains the unknown class not presented in the source domain, it can be problematic to match the source joint distribution $p^s(\boldsymbol{x}, y)$ to the whole target joint distribution $p^t(\boldsymbol{x}, y)$ in the feature space such that $p^s(\boldsymbol{h}(\boldsymbol{x}), y) \approx p^t(\boldsymbol{h}(\boldsymbol{x}), y)$. Because by marginalizing both sides of the matching equation over $\boldsymbol{h}(\boldsymbol{x})$, *i.e.*, $\int p^s(\boldsymbol{h}(\boldsymbol{x}), y) d\boldsymbol{h}(\boldsymbol{x}) \approx \int p^t(\boldsymbol{h}(\boldsymbol{x}), y) d\boldsymbol{h}(\boldsymbol{x})$, one will get $p^s(y) \approx p^t(y)$, which results in the paradox $0 = p^s(\text{c+1}) \approx p^t(\text{c+1}) > 0$ for the unknown class c+1. Therefore, our OSDA work matches the source joint distribution to the target known joint distribution to improve the cross domain transfer performance the on known classes.

B. Open Set Recognition

Open Set Recognition (OSR) considers the problem where new classes unseen in the training phase can appear in the testing phase, and aims to train a classification model that accurately recognizes not only the known classes but also the unknown classes [42]. This OSR problem can be regarded as a simplified OSDA problem without the distribution difference challenge. Most OSR works propose to explore the relationship between known and unknown classes by employing specific techniques or concepts, including open space risk [43], nearest neighbor approach [44], extreme value theory [45], [46], and reconstruction approach [47]. On addressing the common open set recognition challenge in OSR and OSDA, our OSDA work is different from the OSR works. To be specific, our work is founded on the general Empirical Risk Minimization (ERM) principle [1] in statistical learning and minimizes the empirical unknown classification risk for classifying the unknown class, while most OSR works implicitly utilize the specific domain knowledge like the feature semantic information for recognizing the unknown class.

C. Open Set Domain Adaptation

OSDA integrates the challenges from both DA and OSR. Therefore, most OSDA works propose to match distributions between domains to improve the classification performance on the known classes and design mechanisms to recognize the unknown class [4], [5], [7], [11]–[13], [48], [49]. Among these works, we are especially interested in the works of Liu *et al*. [11] and Jang *et al*. [7], since they are most related to our research. Below, we elaborate on the discussion of these two works. Liu *et al*. [11] exploited a projection matrix $\boldsymbol{\phi}$ to match the source marginal $p^s(\boldsymbol{x})$ to the target known marginal $p^t(\boldsymbol{x}|\text{kn})$, and align the source class conditional $p^s(\boldsymbol{x}|y)$ to the target known class conditional $p^t(\boldsymbol{x}|y, \text{kn})$ under the MMD distance in the projection space. Furthermore, they also segregated the target unknown marginal $p^t(\boldsymbol{x}|\text{uk})$ from the source marginal $p^s(\boldsymbol{x})$, and separated the source and target class-conditionals $p^s(\boldsymbol{x}|y)$, $p^t(\boldsymbol{x}|\tilde{y}, \text{kn})$ conditioned on different known classes, *i.e.*, $y \neq \tilde{y}$. The optimization problem for learning the projection $\boldsymbol{\phi}$ is formulated as

$$\begin{aligned}\min_{\boldsymbol{\phi}} \Big(& \alpha \widehat{\text{MMD}}\big(p^s(\boldsymbol{\phi}(\boldsymbol{x})), p^t(\boldsymbol{\phi}(\boldsymbol{x})|\text{kn})\big) \\ & + \alpha \sum_{y=1}^{c} \widehat{\text{MMD}}\big(p^s(\boldsymbol{\phi}(\boldsymbol{x})|y), p^t(\boldsymbol{\phi}(\boldsymbol{x})|y, \text{kn})\big) \\ & - \lambda_1 \widehat{\text{MMD}}\big(p^s(\boldsymbol{\phi}(\boldsymbol{x})), p^t(\boldsymbol{\phi}(\boldsymbol{x})|\text{uk})\big) \\ & - \lambda_2 \sum_{y \neq \tilde{y}}^{c} \widehat{\text{MMD}}\big(p^s(\boldsymbol{\phi}(\boldsymbol{x})|y), p^t(\boldsymbol{\phi}(\boldsymbol{x})|\tilde{y}, \text{kn})\big)\Big), \end{aligned} \tag{26}$$

where α, λ_1, $\lambda_2 (> 0)$ are three tradeoff parameters for balancing the four estimated MMD distances. In objective function (26), the first and second terms are the losses for distribution matching, and the third and fourth terms are the losses for distribution segregation and separation. It is worth mentioning that since estimating the four MMDs in (26) requires the labels for the target data, the authors therefore utilized the Open Set Nearest Neighbor (OSNN) classifier [44] to generate the pseudo target labels of both the known classes and the unknown class. With the pseudo target labels, optimization problem (26) is solved to learn the optimal projection $\widehat{\boldsymbol{\phi}}$. Then, they used square loss as the classification loss, and trained the classifier $\boldsymbol{g}$ to minimize the average source classification loss and the average classification loss computed on the pseudo labeled target unknown data $\{(\boldsymbol{x}_i^{tu}, \text{c+1})\}_{i=1}^{m_{tu}}$. The optimization problem for training the classifier $\boldsymbol{g}$ is formulated as

$$\min_{\boldsymbol{g}} \frac{1}{m_s} \sum_{i=1}^{m_s} \ell(\boldsymbol{g}(\boldsymbol{z}_i^s), y_i^s) + \frac{\gamma}{m_{tu}} \sum_{i=1}^{m_{tu}} \ell(\boldsymbol{g}(\boldsymbol{z}_i^{tu}), \text{c+1}), \tag{27}$$

where $\boldsymbol{z} = \widehat{\boldsymbol{\phi}}(\boldsymbol{x})$ and $\gamma (> 0)$ is a tradeoff parameter for balancing the two loss terms. In objective function (27), the first term is the loss for the known classes, and the second term is the loss for the unknown class. Specifically, computing the second term also requires the pseudo target labels generated by the OSNN classifier. With the pseudo labels, optimization problem (27) is solved to learn the optimal classifier $\widehat{\boldsymbol{g}}$. Finally, the authors updated the pseudo labels by using classifier $\widehat{\boldsymbol{g}}$, and conducted the update procedure until convergence or maximum iteration. Jang *et al*. [7] exploited the featurizer $\boldsymbol{h}$ to match the source marginal $p^s(\boldsymbol{x})$ to the average marginal $p^a(\boldsymbol{x}) = (p^s(\boldsymbol{x}) + \theta p^t(\boldsymbol{x}|\text{kn}) + (1-\theta)p^t(\boldsymbol{x}|\text{uk}))/2$, and also align the target known marginal $p^t(\boldsymbol{x}|\text{kn})$ to the average marginal $p^a(\boldsymbol{x})$ under the KL divergence in the feature space, where $\theta = p^t(\text{kn})$ is the prior probability for the known classes. Besides, they segregated the target unknown marginal $p^t(\boldsymbol{x}|\text{uk})$ from the average marginal $p^a(\boldsymbol{x})$. Concurrently, the authors used cross entropy as classification loss, and trained the classifier $\boldsymbol{g}$ to minimize the average source classification loss and the average classification loss computed on the weighted unlabeled target data. The optimization problem for training the neural network $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}))$ is formulated as

$$\begin{aligned}\min_{\boldsymbol{g},\boldsymbol{h}} \Big(& \frac{1}{m_s} \sum_{i=1}^{m_s} \ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^s)), y_i^s) + \frac{1}{m_t} \sum_{i=1}^{m_t} w_i^t \ell(\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}_i^t)), \text{c+1}) \\ & + \widehat{\text{KL}}\big(p^s(\boldsymbol{h}(\boldsymbol{x})), p^a(\boldsymbol{h}(\boldsymbol{x}))\big) + \theta \widehat{\text{KL}}\big(p^t(\boldsymbol{h}(\boldsymbol{x})|\text{kn}), p^a(\boldsymbol{h}(\boldsymbol{x}))\big) \\ & - (1-\theta) \widehat{\text{KL}}\big(p^t(\boldsymbol{h}(\boldsymbol{x})|\text{uk}), p^a(\boldsymbol{h}(\boldsymbol{x}))\big)\Big), \end{aligned} \tag{28}$$

where $w_i^t \in (0, 1)$ is the weight learned via posterior inference. In objective function (28), the first term is the loss for the known classes, the second term is the loss for the unknown class, the third and fourth terms are the losses for distribution matching, and the last term is the loss for distribution segregation. Similar to the adversarial domain

adaptation works [5], [20], here the distribution matching and distribution segregation under the KL divergence are also performed via adversarial training.

Our work is different from the works of Liu *et al*. [11] and Jang *et al*. [7] from the following two perspectives. **(1)** From the perspective of distribution matching regarding the known classes, we match joint distributions $p^s(\boldsymbol{x}, y)$, $p^t(\boldsymbol{x}, y|\mathrm{kn})$ in (25), rather than marginal distributions $p^s(\boldsymbol{x})$, $p^t(\boldsymbol{x}|\mathrm{kn})$ in (26) and (28), or class-conditional distributions $p^s(\boldsymbol{x}|y)$, $p^t(\boldsymbol{x}|y, \mathrm{kn})$ in (26). In particular, we match distributions under the TD distance in (25), rather than the MMD distance in (26), or the KL divergence in (28). In Subsection IV-C "**Known joint distribution matching**", we demonstrate that our known joint distribution matching under TD distance is superior to the distribution matching and segregation under MMD or KL in [11] or [7]. Again, as aforementioned in Subsection II-C, we select the TD distance because it is better suited for measuring the distance between two joint distributions and can free us from the unstable adversarial training. Like the MMD distance or KL divergence, the TD distance also needs to be estimated from data for practical application. In our research, we show that the estimation of TD distance can be elegantly cast into the familiar least squares classification problem that has an analytic solution. **(2)** From the perspective of unknown class recognition, we derive the classification loss regarding the unknown class in a theoretically justified manner, rather than the heuristic manners in [7], [11]. In particular, thanks to the joint distribution matching and risk reformulation, our unknown loss can be directly computed on the unlabeled target and source data in (25). Unfortunately, the unknown loss in (27) should rely on the additional pseudo labels generated by OSNN classifier, and the unknown loss in (28) should rely on the additional weights learned by posterior inference. In Subsection IV-C "**Unknown classification risk reformulation**", we demonstrate that our theoretically justified unknown class recognition is superior to the heuristic unknown class recognition in [7], [11].

IV. Experiments

A. Experimental Setup

Datasets. We conduct experiments on 5 benchmark image datasets commonly used in OSDA. **OfficeHome** [50] consists of 65 classes and 4 domains: RealWorld (**R**), Product (**P**), Clipart (**C**), and Art (**A**). For this dataset, following [7], [15], [16], [51], the first 25 categories in alphabetic order are selected as the known classes and the remaining 40 classes are treated as the unknown. **ImageCLEF**[1] has 12 classes and 4 domains: Pascal (**P**), ImageNet (**I**), Caltech (**C**), and Bing (**B**). For this dataset, following [16], [51], the first 6 categories are used as the known classes and the remaining 6 classes are combined as the unknown. **Office** [52] contains 31 classes and 3 domains: Amazon (**A**), DSLR (**D**), and Webcam (**W**). For this dataset, following [7], [15], [16], [51], the first 10 classes are defined as the known classes and the last 11 classes are treated as the unknown. **DomainNet** [53] includes 345 classes and 6 domains. For this dataset, following [54], [55], we use 3 domains: Painting (**P**), Real (**R**), and Sketch (**S**) in the experiments. Then, the first 150 categories in alphabetic order are selected as the known classes and the remaining classes are grouped as the unknown. **VisDA** [56] contains 12 classes and 2 domains: Real (**R**) and Synthetic (**S**). For this dataset, following [4], [51], [57], the 6 classes: bicycle, bus, car, motorcycle, train, and truck are denoted as the known classes, and the remaining classes are denoted as the unknown. Apart from the benchmarks, we also conduct experiments on a real world image dataset. **SkinDisease** contains 2 domains: PAD-UFES-20[2] (**P**) and ISIC-SkinCancer[3] (**I**). PAD-UFES-20 includes clinical skin lesion images collected via smartphone devices and has 6 disease classes: actinic keratosis, $\cdots$, squamous cell carcinoma. ISIC-SkinCancer is the other skin disease dataset collected via professional equipment and has 9 disease classes. These 9 classes include the aforementioned 6 classes, and 3 additional classes: dermatofibroma, pigmented benign keratosis, and vascular lesion. For this dataset, PAD-UFES-20 is naturally regarded as the source domain, and ISIC-SkinCancer is the target domain, where the 3 additional classes in the target domain are grouped as unknown.

Evaluation protocol. We select two different domains D1 and D2 from each dataset, and build the OSDA task as "D1 $\rightarrow$ D2". On each task, we follow [7], [15], [16] and use 3 metrics, OS*, UNK, and HOS, to measure a method's performance on the target domain. OS* is the class-wise averaged classification accuracy (%) on known classes, UNK is the classification accuracy on unknown class, and HOS $= 2\frac{\mathrm{OS}^* \times \mathrm{UNK}}{\mathrm{OS}^* + \mathrm{UNK}}$ is the harmonic mean of OS* and UNK. Among these 3 metrics, HOS is the core metric in mainstream OSDA literature [7], [15], [16], because it requires a method to perform well on both the known and unknown classes in an unbiased manner.

Comparison methods. We compare our KMUR approach against other OSDA methods, including OSBP [4], STA [5], ROS [15], JACS [11], UADAL [7], ANNA [16], PSDC [58], WDAN [59], MTS [57], LIWUDA [60], and RTA [51]. Since our experimental setup aligns with the comparison methods, we directly cite their experimental results reported in the original papers. For the comparison methods that do not provide results on a certain dataset, we conduct experiments by using their released source codes, and report their best results.

Implementation details. We run the experiments on Pytorch platform, and implement our neural network as the ResNet50 model for all datasets except VisDA, for which we follow [51] and use VGGNet. In the network, the featurizer is pretrained on ImageNet and the classifier $\boldsymbol{g}$ is trained from scratch. To train the neural network, we use the Adam optimizer [35] and set the learning rate to 10^{-5}. Again, see Algorithm 2 for the training procedure. As defined in objective function (25), our KMUR approach has 2 parameters: θ and λ_{TD}. For parameter θ associated with the classification loss, we utilize the class prior estimation technique from [61] to estimate it from the source and target data. As a result, θ is respectively set to 0.36, 0.47, 0.50, 0.45, 0.52, and 0.68

[1]https://www.imageclef.org/2014/adaptation

[2]https://data.mendeley.com/datasets/zr7vgbcyr2/1

[3]https://www.kaggle.com/datasets/rajivaiml/isic-skin-cancer-dataset

TABLE I: OS* (%), UNK (%), and HOS (%) on benchmark dataset OfficeHome (ResNet50).

Method	A→C			A→P			A→R			C→A			C→P			C→R		
	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**
OSBP [4]	50.20	61.10	55.10	71.80	59.80	65.20	79.30	67.50	72.90	59.40	70.30	64.30	67.00	62.70	64.70	**72.00**	69.20	70.60
STA [5]	46.00	72.30	55.80	68.00	48.40	54.00	78.60	60.40	68.30	51.40	65.00	57.40	61.80	59.10	60.40	67.00	66.70	66.80
ROS [15]	50.60	74.10	60.10	68.40	70.30	69.30	75.80	77.20	76.50	53.60	65.50	58.90	59.80	71.60	65.20	65.30	72.20	68.60
JACS [11]	57.65	75.32	65.31	66.73	74.50	70.40	71.39	69.09	70.22	56.24	69.66	62.23	60.15	70.43	64.89	64.11	74.53	68.93
UADAL [7]	54.90	74.70	63.20	69.10	72.50	70.80	81.30	73.70	77.40	53.50	80.50	64.20	62.10	**78.80**	**69.50**	69.10	78.30	73.40
ANNA [16]	61.40	78.70	69.00	68.30	**79.90**	73.70	74.10	79.70	76.80	58.00	73.10	64.70	64.20	73.60	68.60	66.90	80.20	73.00
PSDC [58]	**65.50**	71.40	68.30	78.60	68.20	73.60	76.20	67.50	71.60	52.80	68.60	59.70	**67.80**	69.00	68.40	69.00	68.60	68.80
WDAN [59]	50.30	66.50	57.30	71.80	59.80	65.20	78.70	75.20	76.90	59.40	70.30	64.30	67.00	62.70	64.70	**72.00**	69.20	70.60
LIWUDA [60]	52.40	69.30	59.70	65.20	65.00	65.00	68.20	66.90	67.50	56.20	**91.90**	**69.80**	65.30	70.00	67.60	61.10	73.10	66.60
RTA [51]	55.50	74.20	63.50	67.20	77.60	77.00	**85.30**	74.50	79.60	59.70	72.50	65.50	59.90	75.80	66.90	70.00	72.80	71.40
KMUR (ours)	59.90	**86.46**	**70.77**	**84.59**	74.45	**79.20**	73.19	**92.45**	**81.70**	**60.43**	79.59	68.70	66.01	70.36	68.12	67.80	**81.30**	**73.94**

Method	P→A			P→C			P→R			R→A			R→C			R→P			Avg	
	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**
OSBP	59.10	68.10	63.20	44.50	66.30	53.20	76.20	71.70	73.90	66.10	67.30	66.70	48.00	63.00	54.50	76.30	68.60	72.30	64.16	66.30	64.72
STA	54.20	72.40	61.90	44.20	67.10	53.20	76.20	64.30	69.50	67.50	66.70	67.10	49.90	61.10	54.50	77.10	55.40	64.50	61.83	63.24	61.12
ROS	57.30	64.30	60.60	46.50	71.20	56.30	70.80	78.40	74.40	67.00	70.80	68.80	51.50	73.00	60.40	72.00	80.00	75.70	61.55	72.38	66.23
JACS	61.55	69.10	65.11	52.22	70.66	60.06	67.25	75.74	71.24	62.88	72.15	67.20	61.80	70.25	65.75	71.00	78.25	74.45	62.75	72.47	67.15
UADAL	50.50	**83.70**	63.00	43.40	81.50	56.60	71.60	83.10	76.90	66.70	78.60	72.10	51.10	74.50	60.60	77.40	76.20	76.80	62.56	78.01	68.71
ANNA	**63.00**	70.30	66.50	54.60	74.80	63.10	74.30	78.90	76.60	66.10	77.30	71.30	59.70	73.10	65.70	76.40	81.00	**78.70**	65.58	76.72	70.64
PSDC	61.80	65.30	63.50	**64.70**	64.10	64.40	77.50	63.20	69.60	60.30	65.00	62.50	**66.60**	69.10	**67.80**	82.20	74.20	78.00	**68.60**	68.00	68.00
WDAN	62.20	64.60	63.40	44.70	63.90	52.60	78.80	71.50	75.00	66.30	78.50	71.90	48.40	66.60	56.10	74.40	79.10	76.70	64.50	68.99	66.23
LIWUDA	57.70	71.80	63.90	52.80	70.90	60.50	64.60	76.80	70.20	64.50	70.90	67.50	61.50	72.20	66.40	70.60	71.40	71.00	61.70	72.60	66.30
RTA	58.60	80.90	68.00	51.60	78.60	62.30	**82.10**	77.20	79.50	**72.50**	77.80	75.00	55.00	70.70	61.70	**84.40**	72.40	77.90	66.70	75.60	70.90
KMUR	60.12	81.61	**69.23**	53.11	**83.49**	**64.92**	78.35	**85.74**	**81.88**	70.98	**81.98**	**76.09**	50.32	**87.17**	63.81	70.61	**85.53**	77.35	66.28	**82.51**	**72.98**

TABLE II: OS* (%), UNK (%), and HOS (%) on benchmark dataset ImageCLEF (ResNet50).

Method	B→C			B→I			B→P			C→B			C→I			C→P		
	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**
OSBP [4]	87.00	81.00	83.90	85.30	65.70	74.30	66.30	66.70	66.50	**62.00**	58.00	59.90	89.00	80.00	84.30	**87.70**	53.70	66.70
STA [5]	93.30	51.70	66.50	86.00	60.70	71.20	77.70	48.70	59.80	61.30	69.70	65.20	91.70	66.70	77.20	84.00	54.00	65.70
ROS [15]	78.30	90.00	83.80	73.00	76.30	74.60	59.00	67.30	62.90	59.00	68.30	63.30	78.30	83.00	80.60	68.70	78.70	73.30
JACS [11]	96.30	82.30	88.80	**88.30**	84.70	86.50	75.00	**75.30**	75.10	54.00	**95.30**	**68.90**	90.00	**97.00**	93.40	65.70	**89.30**	75.70
UADAL [7]	85.34	92.44	88.75	76.40	83.39	79.74	67.10	73.80	70.29	55.20	77.00	64.30	78.00	88.60	82.96	72.40	75.00	73.68
ANNA [16]	95.30	**98.30**	96.80	81.30	84.70	83.00	74.00	75.00	74.50	58.00	83.00	68.30	87.00	93.00	89.90	78.70	84.00	81.20
RTA [51]	95.00	96.30	95.70	79.00	92.70	85.30	73.30	72.00	72.60	57.30	85.70	68.70	86.00	94.00	89.80	75.30	88.30	81.30
KMUR (ours)	**97.66**	97.00	**97.33**	83.66	**95.00**	**88.97**	**84.66**	72.00	**77.82**	**62.00**	67.33	64.55	**92.66**	**97.00**	**94.78**	78.33	87.66	**82.73**

Method	I→B			I→C			I→P			P→B			P→C			P→I			Avg	
	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**
OSBP	55.70	60.70	58.10	80.70	92.70	86.30	66.30	74.30	70.10	52.30	61.00	56.30	94.00	68.00	78.90	66.00	80.70	72.60	74.40	70.20	71.50
STA	**62.30**	54.00	57.90	94.00	53.70	68.40	80.70	59.00	68.20	61.30	43.70	51.00	93.70	47.70	63.20	90.00	51.00	65.10	**81.30**	55.10	65.00
ROS	58.00	59.70	58.80	88.70	92.70	90.60	78.00	76.00	77.00	47.30	59.30	52.70	71.30	90.30	79.70	79.70	81.30	80.50	69.90	76.90	73.10
JACS	56.30	75.30	64.40	96.70	96.70	96.70	82.70	78.30	80.40	53.00	82.70	64.60	95.30	74.30	83.50	**92.00**	**92.00**	**92.00**	78.80	85.20	80.80
UADAL	55.02	69.30	61.34	87.73	95.00	91.22	74.40	85.70	79.65	51.50	64.87	57.42	86.53	78.88	82.53	84.66	80.12	82.33	72.86	80.34	76.18
ANNA	56.00	78.00	65.20	94.30	**97.70**	96.00	80.70	82.70	81.70	**54.00**	73.70	62.30	94.00	**93.70**	93.80	85.00	83.30	84.20	78.20	85.60	81.40
RTA	58.00	79.30	67.00	95.40	98.00	96.70	79.00	**86.00**	82.40	52.00	78.30	62.50	95.30	89.70	92.40	86.70	84.70	85.70	77.70	87.10	82.10
KMUR	52.66	**94.33**	**67.59**	**97.00**	97.00	**97.00**	**85.00**	81.33	**83.12**	49.33	**94.00**	**64.70**	**95.33**	93.00	**94.15**	90.66	85.33	87.91	80.75	**88.42**	**83.39**

TABLE III: OS* (%), UNK (%), and HOS (%) on benchmark dataset Office (ResNet50).

Method	A→D			A→W			D→A			D→W			W→A			W→D			Avg		
	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**
OSBP [4]	90.50	75.50	82.40	86.80	79.20	82.70	76.10	72.30	75.10	97.70	96.70	97.20	73.00	74.40	73.70	99.10	84.20	91.10	87.20	80.40	83.70
STA [5]	91.00	63.90	75.00	86.70	67.60	75.90	83.10	65.90	73.20	94.10	55.50	69.80	66.20	68.00	66.10	84.90	67.80	75.20	84.30	64.80	72.60
ROS [15]	87.50	77.80	82.40	88.40	76.70	82.10	74.80	81.20	77.90	99.30	93.00	96.00	69.70	86.60	77.20	**100.00**	**99.40**	**99.70**	86.60	85.80	85.90
JACS [11]	90.20	88.60	89.40	88.10	90.30	89.20	88.20	**96.90**	**92.30**	97.20	99.60	98.40	90.60	**95.00**	**92.70**	**100.00**	92.60	96.20	92.40	93.80	93.00
UADAL [7]	85.60	90.40	87.90	85.50	**95.10**	90.10	74.20	87.80	80.50	98.70	97.70	98.20	65.60	87.80	75.10	99.30	**99.40**	99.40	84.82	93.03	88.53
ANNA [16]	93.20	76.10	83.80	82.80	88.40	85.50	75.40	91.10	82.50	99.40	99.60	99.50	76.00	87.90	81.60	**100.00**	96.80	98.40	87.80	89.98	88.55
PSDC [58]	94.60	81.70	87.70	97.80	85.90	91.50	**94.20**	88.80	91.40	**100.00**	99.90	**99.90**	**94.90**	86.00	90.20	99.00	98.90	99.00	**96.80**	90.20	93.30
LIWUDA [60]	99.20	83.10	90.40	89.80	86.70	88.20	70.30	85.20	77.00	94.40	91.60	93.00	84.10	71.60	77.30	94.90	86.20	90.30	88.80	84.10	86.10
RTA [51]	94.70	91.20	92.90	92.20	93.80	93.00	87.90	94.10	90.90	98.00	**100.00**	99.00	89.00	92.50	90.70	98.50	99.10	98.80	93.40	**95.10**	**94.20**
KMUR (ours)	**100.00**	**91.48**	**95.55**	**99.52**	90.26	**94.66**	87.40	90.31	88.83	**100.00**	94.38	97.11	85.00	93.77	89.17	99.33	93.08	96.11	95.21	92.21	93.57

for OfficeHome, ImageCLEF, Office, DomainNet, VisDA, and SkinDisease. In Subsection IV-C "**Parameter sensitivity**", we study the sensitivity of our approach to this parameter. For parameter λ_{TD} associated with the joint distribution matching loss, following the common strategy in [20], [31], [33], [34], we change it from 0 to 1 during the training procedure by using the formula $\lambda_{\text{TD}} = \frac{2}{1+\exp(-10i/I)} - 1$, where $i \in [0, I]$ is the current iteration and I is the total iteration. In [20], [31], [33], [34], this strategy has proven effective for balancing the distribution matching loss.

B. Experimental Results

Results. We report the performance metrics (OS*, UNK, and HOS) of the OSDA methods in Tables I-V for the 5 benchmark datasets OfficeHome, ⋯, VisDA, and in Table VI for the real world dataset SkinDisease, where the best results are highlighted in **bold** and the second best are underlined. It is important to emphasize that HOS is the most critical metric, as it requires a method to achieve balanced performance on both the known and unknown classes without bias. In Table I for the benchmark dataset OfficeHome, our KMUR significantly outperforms the comparison methods on majority of the 12 tasks, achieving the highest average UNK of 82.51%, and the highest average HOS of 72.98%. For the comparison with JACS and UADAL, which perform marginal distribution matching and segregation under MMD or KL and design heuristic mechanisms for unknown recognition, the outperformance

TABLE IV: OS* (%), UNK (%), and HOS (%) on benchmark dataset DomainNet (ResNet50).

Method	P→R			P→S			R→P			R→S			S→P			S→R			Avg		
	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**
OSBP [4]	55.07	65.80	59.96	29.47	63.44	40.24	40.41	67.44	50.54	29.60	68.25	41.29	32.71	64.56	43.42	49.67	66.01	56.69	39.49	65.92	48.69
STA [5]	35.33	69.96	46.95	21.16	69.72	32.47	35.62	63.14	45.54	28.39	61.56	38.86	23.08	55.24	32.55	33.39	77.70	46.71	29.49	66.22	40.51
UADAL [7]	32.40	83.20	46.64	28.60	69.70	40.56	36.00	85.50	50.67	24.30	**94.90**	38.69	28.40	55.40	37.55	50.44	88.50	64.26	33.36	79.53	46.39
ANNA [16]	55.17	85.75	67.14	34.98	72.56	47.21	38.91	73.69	50.93	37.11	68.79	48.21	**40.02**	60.96	48.32	50.76	88.63	64.55	42.82	75.06	54.39
RTA [51]	51.30	88.80	65.03	30.80	**90.90**	46.01	34.00	85.40	48.64	**37.70**	93.40	**53.72**	33.20	84.50	47.67	48.10	**92.40**	63.27	39.18	89.23	54.06
KMUR (ours)	**57.90**	**90.75**	**70.70**	**36.75**	88.85	**51.99**	**41.26**	**90.50**	**56.68**	35.66	85.43	50.32	38.76	**89.90**	**54.17**	**56.80**	91.01	**69.95**	**44.52**	**89.41**	**58.97**

TABLE V: OS* (%), UNK (%), and HOS (%) on benchmark dataset VisDA with OSDA task S→R (VGGNet).

Method	Bicycle	Bus	Car	Motorcycle	Train	Truck	OS*	UNK	**HOS**
OSBP [4]	51.10	67.10	42.80	84.20	81.80	28.00	59.20	85.10	69.80
STA [5]	52.40	69.60	59.90	87.80	86.50	27.20	63.90	84.10	72.60
PSDC [58]	75.50	84.40	60.50	88.20	73.20	28.60	68.40	85.40	76.00
MTS [57]	55.60	68.40	65.50	88.20	85.30	31.50	65.60	**88.00**	75.17
RTA [51]	67.20	77.90	89.60	86.60	84.90	35.80	73.60	83.70	78.30
KMUR (ours)	72.80	86.30	79.50	90.20	81.40	33.60	**74.00**	86.40	**79.72**

TABLE VI: OS* (%), UNK (%), and HOS (%) on real world dataset SkinDisease with natural OSDA task P→I.

Metric	OSBP [4]	STA [5]	ROS [15]	JACS [11]	UADAL [7]	ANNA [16]	KMUR (ours)
OS*	29.56	22.28	17.97	28.11	24.10	**42.27**	40.87
UNK	22.54	33.55	46.58	22.92	36.10	26.43	**50.32**
HOS	25.58	26.78	25.93	25.25	28.90	32.52	**45.11**

proves that our joint distribution matching under TD distance and risk reformulation for unknown recognition indeed better solve the OSDA problem. For the comparison with ROS and ANNA, which perform self-supervised learning and causal alignment, the outperformance again testifies that our known joint distribution matching and unknown classification risk reformulation are more advantageous than the other strategies in improving the OSDA performance. In Table II for the benchmark dataset ImageCLEF, our KMUR again outperforms the comparison methods, achieving the best average UNK of 88.42%, and the best average HOS of 83.39%. In Table III for the benchmark dataset Office, our KMUR is among the top performing methods and obtains the second best average HOS of 93.57%, next to the latest method RTA. Since RTA is featured by its exploitation of the inter-class relations as a prior to separate the unknown, we conjecture that incorporating such strategy into our approach may improve its performance in this specific Office dataset. In Tables IV and V for the benchmark datasets DomainNet and VisDA, our KMUR outperforms its counterparts and achieves the best average HOS of 58.97% and 79.72%, proving its suitability for complex and large scale datasets with large domain shifts. Finally in Table VI for the real world dataset SkinDisease, our KMUR is also more effective than its competitors in recognizing the known and unknown skin diseases, and achieves the highest HOS of 45.11% on the natural OSDA task. To further reinforce the advantage of our approach, in Subsection IV-C "**Feature visualization**" and "**Robustness to openness**", we demonstrate that compared with other methods, our KMUR better matches the cross domain distributions regarding the known classes and is more robust against the varying number of known classes.

Statistical tests. We validate the superiority of our approach in a statistical sense by conducting the Friedman test followed by Holm's step-down procedure [62]. The Friedman test uses statistic $F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}$, where N denotes the number of

TABLE VII: Average ranks of the methods.

Method	OSBP [4]	STA [5]	UADAL [7]	ANNA [16]	RTA [51]	KMUR (ours)
Avg rank	4.89	5.78	3.81	2.61	2.42	1.50

TABLE VIII: Holm's step-down test results.

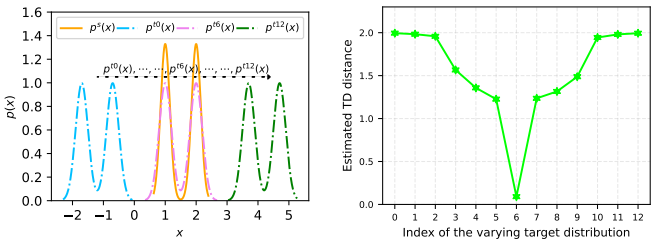
i	Method	z value	p value	$\alpha/(k-i)$	Reject
1	STA [5]	9.7011	0.0000	0.0100	✓
2	OSBP [4]	7.6853	0.0000	0.0125	✓
3	UADAL [7]	5.2285	0.0000	0.0167	✓
4	ANNA [16]	2.5198	0.0117	0.0250	✓
5	RTA [51]	2.0788	0.0376	0.0500	✓

OSDA tasks and k is the number of methods. χ_F^2 is a statistic defined as $\chi_F^2 = \frac{12N}{k(k+1)}\big(\sum_{i=1}^k R_i^2 - \frac{k(k+1)^2}{4}\big)$, where R_i is the average rank of the i-th method. Based on the HOS results from Tables I-IV, we compute the average ranks of the methods and report them in Table VII. Since our calculated $F_F = 106.60$ exceeds the critical value $F(k-1,(k-1)(N-1)) = 2.27$ at a significant level of $\alpha = 0.05$, we conclude that the methods have significantly different performance on the datasets. The subsequent Holm's step-down procedure uses test statistic $z = (R_i - R_j)/\sqrt{\frac{k(k+1)}{6N}}$ to compute the p-values. We report the test results in Table VIII. The results suggest that all the null hypotheses can be rejected since the corresponding p values are below $\alpha/(k-i)$. This implies that our KMUR approach significantly outperforms the comparison methods.

C. Experimental Analysis

TD distance estimation. We verify the reliability of our LSTDE technique for estimating the TD distance. To this end, we prepare in Fig. 3(a) the fixed source distribution $p^s(x)$ and the varying target distributions $p^{t0}(x)$, $\cdots$, $p^{t6}(x)$, $\cdots$, $p^{t12}(x)$, and plot in Fig. 3(b) the estimated TD distance between the source distribution and each target distribution. From Fig. 3(a), we know that as the target distribution varies from left to right, the distance between the source and target distributions should first decrease, then reach a minimal value, and finally increase. From Fig. 3(b), we observe that the TD distance estimated by our LSTDE technique exactly reflects such trend: The estimated TD distance first decreases from the maximal value of 2, then reaches the minimal value close to 0, and finally increases again to the maximal value. Interestingly, the maximal and minimal values also coincide with the fact that the TD distance is upper bounded by 2 and lower bounded by 0. These results effectively confirm that our LSTDE can yield reliable estimated value for the TD distance.

Ablation study. We conduct ablation study on the ImageCLEF tasks to evaluate the contribution of each loss in our objective function (25). As shown in Table IX, removing loss



(a) Source and target distributions (b) Estimated TD distance

Fig. 3: TD distance estimated by our LSTDE technique. (a) Fixed source distribution $p^s(x)$ and varying target distributions $p^{t0}(x)$, $\cdots$, $p^{t6}(x)$, $\cdots$, $p^{t12}(x)$. (b) Estimated TD distance between source and target distributions.

TABLE IX: Ablation study for our KMUR approach on the ImageCLEF tasks. $\mathcal{L}_{\text{known}}$ and $\mathcal{L}_{\text{unknown}}$ are the classification losses for known and unknown classes. $\mathcal{L}_{\text{TD}}$ is the joint distribution matching loss for known classes.

$\mathcal{L}_{\text{known}}$	$\mathcal{L}_{\text{unknown}}$	$\mathcal{L}_{\text{TD}}$	B→C OS*	B→C UNK	B→C **HOS**	C→I OS*	C→I UNK	C→I **HOS**
✓	✓	✓	97.66	97.00	**97.33**	92.66	97.00	**94.78**
×	✓	✓	0.00 ↓	100.00	0.00 ↓	0.00 ↓	100.00	0.00 ↓
✓	×	✓	98.00	44.33 ↓	61.05 ↓	95.67	45.00 ↓	61.21 ↓
✓	✓	×	85.67 ↓	78.67 ↓	82.02 ↓	82.33 ↓	77.00 ↓	79.58 ↓

$\mathcal{L}_{\text{known}}$ (classification loss for known classes) disables our approach's ability to learn the functional relationship between input features and known class labels and severely degrades its OS* performance (from 97.66% to 0.00% for B→C and from 92.66% to 0.00% for C→I). Removing loss $\mathcal{L}_{\text{unknown}}$ (classification loss for unknown class) weakens our approach's ability to classify the unknown class and deteriorates its UNK performance to a large extent (from 97.00% to 44.33% for B→C and from 97.00% to 45.00% for C→I). Finally, removing loss $\mathcal{L}_{\text{TD}}$ (joint distribution matching loss for known classes) disables our approach's ability to match the source to the target known and degrades its OS* (from 97.66% to 85.67% for B→C and from 92.66% to 82.33% for C→I) and UNK (from 97.00% to 78.67% for B→C and from 97.00% to 77.00% for C→I). These findings suggest that all the three losses $\mathcal{L}_{\text{known}}$, $\mathcal{L}_{\text{unknown}}$, and $\mathcal{L}_{\text{TD}}$ are indispensable for the superior performance of our approach.

Pseudo labels under increasing domain shift. We study the performance of our approach relative to the initial pseudo labels generated under increasing domain shift. To this end, we conduct experiments on the face recognition dataset PIE-Multiview [63] in Fig. 4, and choose looking forward (C27) as the source domain and looking towards left in an increasing angle (C05, C37, C25, C02) as the target domains to simulate the increasing domain shift. To measure the shift, we calculate the TD distance between the fixed source C27 and each target and obtain $\widehat{\text{TD}}(\text{C27}, \text{C05}) = 0.47 < \widehat{\text{TD}}(\text{C27}, \text{C37}) = 0.95 < \widehat{\text{TD}}(\text{C27}, \text{C25}) = 1.10 < \widehat{\text{TD}}(\text{C27}, \text{C02}) = 1.20$. These values confirm a controlled, monotonic increase in domain shift. We treat the first 30 classes as known and the remaining 37 classes as unknown, and use a One-Hidden-Layer Neural Network (1HLNN) as the model. Results for both the

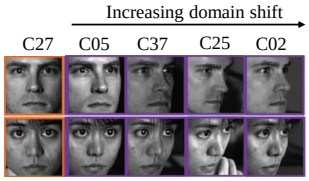


Fig. 4: Sample images from PIE-Multiview dataset, where C27 is used as the source domain and C05, C37, C25, C02 are used as the target domains to simulate the increasing domain shift.

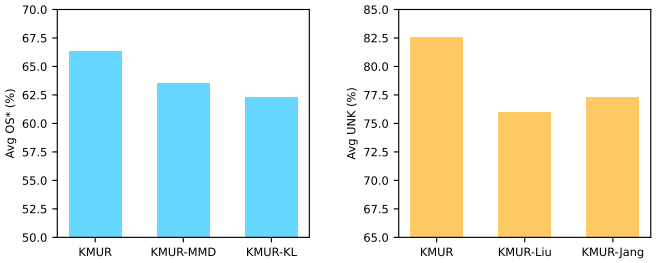
TABLE X: OS* (%), UNK (%), and HOS (%) on face recognition dataset PIE-Multiview (1HLNN).

Method	C27→C05 OS*	C27→C05 UNK	C27→C05 **HOS**	C27→C37 OS*	C27→C37 UNK	C27→C37 HOS	C27→C25 OS*	C27→C25 UNK	C27→C25 **HOS**	C27→C02 OS*	C27→C02 UNK	C27→C02 **HOS**
KMUR-Pretrain	66.03	63.83	64.91	51.26	46.51	48.77	28.88	46.07	35.51	27.77	43.75	33.98
KMUR	**78.41**	**71.94**	**75.03**	**67.77**	**50.90**	**58.14**	**32.38**	**50.96**	**39.60**	**33.65**	**47.87**	**39.52**

pre-training (KMUR-Pretrain) and formal training (KMUR) phases are presented in Table X. We observe that as the domain shift increases: $\widehat{\text{TD}}(\text{C27}, \text{C05}) < \cdots < \widehat{\text{TD}}(\text{C27}, \text{C02})$, the HOS of KMUR-Pretrain keeps declining from 64.91% to 33.98%, indicating that the initial pseudo labels become increasingly noisy with larger shifts. Nevertheless, the formal training phase (KMUR) still improves the HOS across all 4 tasks despite the degraded initial pseudo labels. This suggests that our approach does not excessively rely on the quality of initial pseudo labels, and that the optimization process in the formal training phase remains stable and robust even when starting from low quality labels. We attribute the stability and robustness to the iterative refinement of pseudo labels and model parameters, which mitigates the effect of initial inaccuracies. However, we emphasize that in some extreme cases of large shifts where transfer learning becomes rather difficult or even infeasible, our approach may not succeed since the initial pseudo labels may be too noisy to refine.

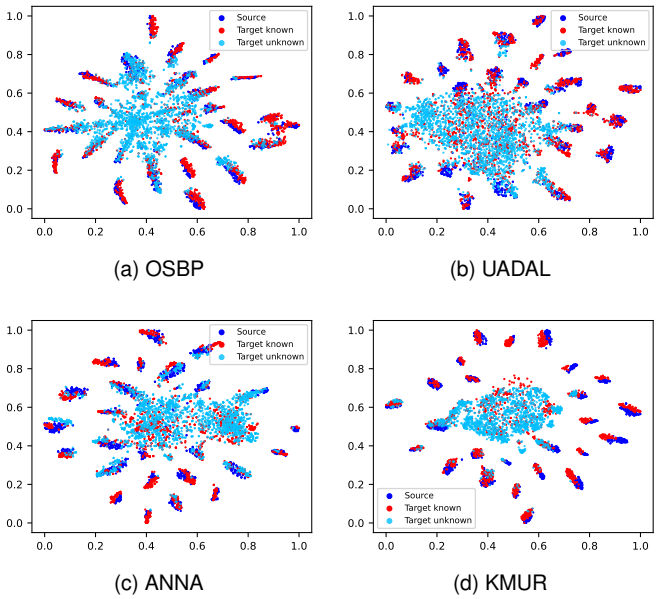
Known joint distribution matching. We compare our known joint distribution matching under TD distance against the distribution matching and segregation under MMD distance [11] and KL divergence [7] by analyzing their impact on classifying the known classes. For this purpose, we replace our original TD based known joint distribution matching by the MMD based and the KL based distribution matching and segregation, and derive 2 variants of our approach: KMUR-MMD and KMUR-KL. Fig. 5(a) plots the average OS* of KMUR and its 2 variants on OfficeHome. We observe that the average OS* of KMUR is higher than KMUR-MMD and KMUR-KL. This indicates that our known joint distribution matching under TD distance is better than the distribution matching and segregation under MMD or KL in [11] or [7].

Unknown classification risk reformulation. We contrast our unknown classification risk reformulation against the strategies in Liu *et al.* [11] and Jang *et al.* [7] for recognizing the unknown class. For this purpose, we replace our original unknown classification loss by the unknown classification losses in [11] and [7], and derive 2 variants of our approach: KMUR-Liu and KMUR-Jang. Fig. 5(b) plots the average UNK of KMUR and its variants on OfficeHome. We find that KMUR clearly outperforms its variants. This suggests our unknown



(a) Known distribution matching (b) Unknown class recognition

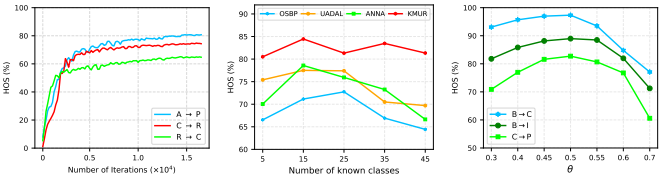
Fig. 5: Comparison of known distribution matching and comparison of unknown class recognition on OfficeHome. (a) Comparison of our known joint distribution matching under TD against the distribution matching and segregation under MMD and KL in [11] and [7]. (b) Comparison of our unknown classification risk reformulation against the strategies in [11] and [7] for recognizing the unknown class.



(a) OSBP (b) UADAL

(c) ANNA (d) KMUR

Fig. 6: Visualization of source data (blue), target known data (red), and target unknown data (skyblue) in the network's feature spaces of OSBP, UADAL, ANNA, and KMUR on the task "P→R" (OfficeHome).



(a) Robustness (b) Convergence (c) Sensitivity

Fig. 7: Convergence, robustness, and sensitivity analysis of our approach. (a) Convergence on the tasks from OfficeHome. (b) Robustness on the task A→R (OfficeHome). (c) Sensitivity on the tasks from ImageCLEF.

classification risk reformulation is more advantageous than the strategies in [11] and [7] for classifying the unknown.

Feature visualization. We demonstrate that our approach can well match the cross domain distributions regarding the known classes. Figs. 6(a)-6(d) plot the t-SNE [64] visualization results of source data (blue), target known data (red), and target unknown data (skyblue) in the network's feature spaces of OSBP, UADAL, ANNA, and KMUR on the task "P→R" (Office-Home). By comparing with the visualization results of other methods, we find that our KMUR approach aligns the source data to the target known data in a better way, and also leaves the target unknown data alone in the center. This proves that in the network's feature space, our approach can well match the cross domain distributions of the known classes.

Convergence analysis. We study the convergence behavior of our approach. Fig. 7(a) plots the convergence of the HOS of our approach on the OfficeHome tasks. We find that as the iteration proceeds, the HOS first rapidly increases, then gradually slows down, and finally reaches a plateau after 1.5×10^4 iterations. This implies that our Algorithm 2 for training the neural network is convergent.

Robustness analysis. We analyze the robustness of our approach to different settings of the number of known classes. Fig. 7(b) plots the HOS of our approach and the comparison methods on the task A→R (OfficeHome) by varying the number of known classes. From Fig. 7(b), we observe that our approach consistently outperforms the comparison methods over all the cases. This suggests that our approach is robust to the change in the number of known classes.

Parameter sensitivity. We analyze the sensitivity of our approach to parameter θ, which is the prior probability of known classes in the target domain, *i.e.*, $\theta = p^t(\mathrm{kn})$. This parameter is associated with the classification loss in our objective function. Fig. 7(c) illustrates the HOS performance of our approach under different values of θ on ImageCLEF tasks. Note that under our experimental setup, the true prior probability of known classes for this dataset is 0.5. As illustrated in Fig. 7(c), our approach achieves the highest HOS when θ is set to 0.5. Performance begins to decline when $\theta < 0.45$ or $\theta > 0.55$, highlighting the importance of setting an appropriate value for θ. As described in the experimental setup, we use the class prior estimation technique from [61] to estimate θ from the source and target data. The estimated value obtained is 0.47, which is close to the true value of 0.5. Therefore, our approach achieves superior HOS results in the main experiments.

V. Conclusion

In this paper, we propose the KMUR approach to address the fundamental challenges in OSDA. Our approach matches the source joint distribution to the target known joint distribution under the TD distance to reduce the distribution difference, and reformulates the unknown classification risk to derive the classification loss for the unknown class. In addition, we also propose the LSTDE technique to estimate the TD distance from data and express the estimated distance as an objective function of the featurizer. The experiments on bench-

mark and real world datasets demonstrate the superiority of our approach over its competitors, and the experimental analysis on benchmark and synthetic datasets testifies the effectiveness of our proposed KMUR and LSTDE. In the future, we plan to explore the related multi-source open set domain adaptation and universal domain adaptation problems. We also plan to apply the LSTDE technique to solving other machine learning problems, including clustering and representation learning.

REFERENCES

[1] V. N. Vapnik, *Statistical learning theory*. New York: Wiley, 1998.

[2] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer, 2009.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, MA: MIT Press, 2016.

[4] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada, "Open set domain adaptation by backpropagation," in *European Conference on Computer Vision*, 2018, pp. 153–168.

[5] H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang, "Separate to adapt: Open set domain adaptation via progressive separation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2927–2936.

[6] Z. Fang, J. Lu, F. Liu, J. Xuan, and G. Zhang, "Open set domain adaptation: Theoretical bound and algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4309–4322, 2020.

[7] J. Jang, B. Na, D. H. Shin, M. Ji, K. Song, and I.-c. Moon, "Unknown-aware domain adversarial learning for open-set domain adaptation," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 16 755–16 767.

[8] J. Jiang, "A literature survey on domain adaptation of statistical classifiers," *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, vol. 3, pp. 1–12, 2008.

[9] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. Cambridge, MA: MIT Press, 2008.

[10] T. Fang, N. Lu, G. Niu, and M. Sugiyama, "Rethinking importance weighting for deep learning under distribution shift," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 996–12 007, 2020.

[11] J. Liu, M. Jing, J. Li, K. Lu, and H. T. Shen, "Open set domain adaptation via joint alignment and category separation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 6186–6199, 2021.

[12] L. Zhong, Z. Fang, F. Liu, B. Yuan, G. Zhang, and J. Lu, "Bridging the theoretical bound and deep algorithms for open set domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 3859–3873, 2021.

[13] J. Liu, X. Guo, and Y. Yuan, "Unknown-oriented learning for open set domain adaptation," in *European Conference on Computer Vision*, 2022, pp. 334–350.

[14] T. Shermin, G. Lu, S. W. Teng, M. Murshed, and F. Sohel, "Adversarial network with multiple classifiers for open set domain adaptation," *IEEE Transactions on Multimedia*, vol. 23, pp. 2732–2744, 2020.

[15] S. Bucci, M. R. Loghmani, and T. Tommasi, "On the effectiveness of image rotation for open set domain adaptation," in *European Conference on Computer Vision*, 2020, pp. 422–438.

[16] W. Li, J. Liu, B. Han, and Y. Yuan, "Adjustment and alignment for unbiased open set domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 110–24 119.

[17] Y. Pan, T. Yao, Y. Li, C.-W. Ngo, and T. Mei, "Exploring category-agnostic clusters for open-set domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 867–13 875.

[18] Y. Xu, L. Chen, L. Duan, I. W. Tsang, and J. Luo, "Open set domain adaptation with soft unknown-class rejection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 3, pp. 1601–1612, 2023.

[19] F. Topsoe, "Some inequalities for information divergence and related measures of discrimination," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1602–1609, 2000.

[20] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[21] D. Acuna, G. Zhang, M. T. Law, and S. Fidler, "f-domain adversarial learning: Theory and algorithms," in *International Conference on Machine Learning*, vol. 139, 2021, pp. 66–75.

[22] Z. Yuan, X. Hu, Q. Wu, S. Ma, C. H. Leung, X. Shen, and Y. Huang, "A unified domain adaptation framework with distinctive divergence analysis," *Transactions on Machine Learning Research*, pp. 1–21, 2022.

[23] L. Wasserman, *All of statistics: A concise course in statistical inference*. New York, NY: Springer, 2004.

[24] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," in *Advances in Neural Information Processing Systems*, 2016, pp. 271–279.

[25] S. Chen, "Decomposed adversarial domain generalization," *Knowledge-Based Systems*, vol. 263, pp. 1–10, 2023.

[26] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, pp. 723–773, 2012.

[27] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "On surrogate loss functions and f-divergences," *The Annals of Statistics*, vol. 37, no. 2, pp. 876–904, 2009.

[28] Q. Que and M. Belkin, "Back to the future: Radial basis function network revisited," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 1856–1867, 2020.

[29] N. Lu, T. Zhang, G. Niu, and M. Sugiyama, "Mitigating overfitting in supervised classification from two unlabeled datasets: A consistent risk correction approach," in *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 1115–1125.

[30] Y. Chen, C. Wei, A. Kumar, and T. Ma, "Self-training avoids using spurious features under domain shift," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 061–21 071, 2020.

[31] Y. Zhu, F. Zhuang, J. Wang, G. Ke, J. Chen, J. Bian, H. Xiong, and Q. He, "Deep subdomain adaptation network for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 4, pp. 1713–1722, 2020.

[32] C.-X. Ren, Y.-W. Luo, and D.-Q. Dai, "Buresnet: Conditional bures metric for transferable representation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4198–4213, 2023.

[33] S. Chen, Z. Hong, M. Harandi, and X. Yang, "Domain neural adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8630–8641, 2023.

[34] L. Wen, S. Chen, M. Xie, C. Liu, and L. Zheng, "Training multi-source domain adaptation network by mutual information estimation and minimization," *Neural Networks*, vol. 171, pp. 353–361, 2024.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015, pp. 1–11.

[36] A. T. Nguyen, T. Tran, Y. Gal, P. H. Torr, and A. G. Baydin, "Kl guided domain adaptation," in *International Conference on Learning Representations*, 2022, pp. 1–12.

[37] P. Ge, C.-X. Ren, X.-L. Xu, and H. Yan, "Unsupervised domain adaptation via deep conditional adaptation network," *Pattern Recognition*, vol. 134, pp. 1–14, 2023.

[38] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," in *Advances in Neural Information Processing Systems*, 2017, pp. 3730–3739.

[39] B. Bhushan Damodaran, B. Kellenberger, R. Flamary, D. Tuia, and N. Courty, "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation," in *European Conference on Computer Vision*, 2018, pp. 447–463.

[40] S. Chen, M. Harandi, X. Jin, and X. Yang, "Domain adaptation by joint distribution invariant projections," *IEEE Transactions on Image Processing*, vol. 29, pp. 8264–8277, 2020.

[41] S. Chen, P. Xuan, and Z. Hao, "Joint distribution weighted alignment for multi-source domain adaptation via kernel relative entropy estimation," *IEEE Transactions on Multimedia*, pp. 1–14, 2025.

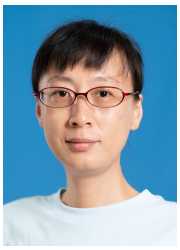
[42] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3614–3631, 2021.

[43] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2317–2324, 2014.

[44] P. R. Mendes Júnior, R. M. De Souza, R. d. O. Werneck, B. V. Stein, D. V. Pazinato, W. R. De Almeida, O. A. Penatti, R. d. S. Torres, and A. Rocha, "Nearest neighbors distance ratio open-set classifier," *Machine Learning*, vol. 106, no. 3, pp. 359–386, 2017.

[45] E. M. Rudd, L. P. Jain, W. J. Scheirer, and T. E. Boult, "The extreme value machine," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 762–768, 2018.
[46] X. Yin, B. Cao, Q. Hu, and Q. Wang, "Rd-openmax: Rethinking openmax for robust realistic open-set recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 4, pp. 7565–7579, 2025.
[47] X. Sun, Z. Yang, C. Zhang, K.-V. Ling, and G. Peng, "Conditional gaussian distribution learning for open set recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 477–13 486.
[48] Q. Wang, F. Meng, and T. P. Breckon, "Progressively select and reject pseudolabeled samples for open-set domain adaptation," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 9, pp. 4403–4414, 2024.
[49] D. Zhang, T. Westfechtel, and T. Harada, "Open-set domain adaptation via joint error based multi-class positive and unlabeled learning," in *European Conference on Computer Vision*, 2024, pp. 105–120.
[50] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5018–5027.
[51] Y. Tong, D. Chang, D. Li, X. Wang, K. Liang, Z. He, Y.-Z. Song, and Z. Ma, "Reserve to adapt: Mining inter-class relations for open-set domain adaptation," *IEEE Transactions on Image Processing*, vol. 34, pp. 1382–1397, 2025.
[52] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European Conference on Computer Vision*, 2010, pp. 213–226.
[53] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *IEEE International Conference on Computer Vision*, 2019, pp. 1406–1415.
[54] K. Saito and K. Saenko, "Ovanet: One-vs-all network for universal domain adaptation," in *IEEE International Conference on Computer Vision*, 2021, pp. 8980–8989.
[55] W. Chang, Y. Shi, H. Tuan, and J. Wang, "Unified optimal transport framework for universal domain adaptation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 512–29 524, 2022.
[56] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, and K. Saenko, "Visda: A synthetic-to-real benchmark for visual domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2021–2026.
[57] D. Chang, A. Sain, Z. Ma, Y.-Z. Song, R. Wang, and J. Guo, "Mind the gap: Open set domain adaptation via mutual-to-separate framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 6, pp. 4159–4174, 2024.
[58] Z. Liu, G. Chen, Z. Li, Y. Kang, S. Qu, and C. Jiang, "Psdc: A prototype-based shared-dummy classifier model for open-set domain adaptation," *IEEE Transactions on Cybernetics*, vol. 53, no. 11, pp. 7353–7366, 2023.
[59] J. Li, L. Yang, Q. Wang, and Q. Hu, "Wdan: A weighted discriminative adversarial network with dual classifiers for fine-grained open-set domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 9, pp. 5133–5147, 2023.
[60] J. Zhu, F. Ye, Q. Xiao, P. Guo, Y. Zhang, and Q. Yang, "A versatile framework for unsupervised domain adaptation based on instance weighting," *IEEE Transactions on Image Processing*, vol. 33, pp. 6633–6646, 2024.
[61] M. C. du Plessis, G. Niu, and M. Sugiyama, "Class-prior estimation for learning from positive and unlabeled data," *Machine Learning*, vol. 106, no. 4, pp. 463–492, 2017.
[62] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
[63] S. Herath, M. Harandi, and F. Porikli, "Learning an invariant hilbert space for domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3845–3854.
[64] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

Dr. Sentao Chen is an Associate Professor at Shantou University and a Researcher in statistical machine learning. His recent research interest is transfer learning. From 2020 to 2025, he has extended the empirical risk minimization principle, proposed the joint distribution matching framework, and developed the kernel statistical distance estimation technique to address the fundamental challenges in transfer learning. These contributions have lead to a series of principled, straightforward, and effective algorithms for various transfer learning problems.



Ping Xuan received the Ph.D. degree in computer science and technology from Harbin Institute of Technology, Harbin, China. She is currently a researcher and professor at the School of Cyberspace Security, Hainan University, Haikou, China. Her research interests include machine learning, deep learning, graph learning, and medical image processing.

Dr. Lifang He is an Associate Professor of Computer Science and Engineering at Lehigh University. She received her Ph.D. in Computer Science from the South China University of Technology and completed postdoctoral training at the University of Pennsylvania and Cornell University's medical schools. Her research interests include machine learning and artificial intelligence, with an emphasis on scalable and interpretable methods for complex biomedical and clinical data analysis. Dr. He has published over 200 papers in leading journals and conferences, including Nature Medicine, IEEE Transactions on Neural Networks and Learning Systems (TNNLS), IEEE Transactions on Knowledge and Data Engineering (TKDE), NeurIPS, and ICML. She currently serves as an Associate Editor for ACM Transactions on Computing for Healthcare and the International Journal on Machine Learning and Cybernetics. Dr. He is an active member of the research community and has served as a reviewer or program committee member for venues including Nature, TNNLS, TKDE, NeurIPS, ICML, AAAI, and CVPR. She has also served as Chair of the IEEE Computer Society Chapter of the IEEE Lehigh Valley Section since 2023.