AFRL-AFOSR-JP-TR-2022-0017

Deep Generative Models for Learning from Multiple High-Dimensional Data Sources

TRUNG LE
MONASH UNIVERSITY
WELLINGTON RD
CLAYTON, VIC, 3168
AUS

03/31/2022
Final Technical Report

Air Force Research Laboratory
Air Force Office of Scientific Research
Asian Office of Aerospace Research and Development
Unit 45002, APO AP 96338-5002

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
20220331	Final	START DATE 20190822	END DATE 20210821

4. TITLE AND SUBTITLE
Deep Generative Models for Learning from Multiple
High-Dimensional Data Sources

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA2386-19-1-4040		
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Trung Le

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MONASH UNIVERSITY
WELLINGTON RD
CLAYTON, VIC 3168
AUS

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD UNIT 45002 APO AP 96338-5002	AFRL/AFOSR IOA	AFRL-AFOSR-JP-TR-2022-0017

12. DISTRIBUTION/AVAILABILITY STATEMENT
A Distribution Unlimited: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Modern machine learning systems need to handle complex high-dimensional data such as natural images, motion pictures, speeches, dialog texts, and hand-written cursive drawings to name a few originated from multiple sources. Generating, manipulating, and learning from multiple homogeneous high-dimensional data sources are impact capacities of intelligent systems that have a mixed varieties of applications in reality. Appropriate generating methods using multiple homogeneous high-dimensional data sources allow us to interpolate over structural manifolds inside data and generating data that follow constraints. Comparing to the standard setting of generative model wherein it requires generating data examples mimicking a single data source, this generating task is more challenging given the fact that multiple source data in high-dimensional space tend to be located in a great deal of data modes/structural manifolds carried in data. Another way to exploit multiple data sources is to learn common or source-invariant features that can be transferred to another independent data source. This setting is known as single source or multiple source domain adaptation. This research aims to propose efficient and effective methods enabling us to either generating data followed the constraints specified by multiple data sources or learning from multiple sources in a transfer learning scenario.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR	38

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
ALAN LIN	227-7009

Final Report for AOARD Grant FA2386-19-1-4040

Deep Generative Models for Learning from Multiple Data Sources

INVESTIGATORS

Principal Investigator: Dr **Trung Le**
Monash University, Australia
Email: trunglm@monash.edu

Co-Principal Investigator: Professor **Dinh Phung**
Monash University, Australia
Email: dinh.phung@monash.edu

Abstract

Modern machine learning systems need to handle complex high-dimensional data such as natural images, motion pictures, speeches, dialog texts, and hand-written cursive drawings to name a few originated from multiple sources. Generating, manipulating, and learning from multiple homogeneous high-dimensional data sources are impact capacities of intelligent systems that have a mixed varieties of applications in reality. Appropriate generating methods using multiple homogeneous high-dimensional data sources allow us to interpolate over structural manifolds inside data and generating data that follow constraints. Comparing to the standard setting of generative model wherein it requires generating data examples mimicking a single data source, this generating task is more challenging given the fact that multiple source data in high-dimensional space tend to be located in a great deal of data modes/structural manifolds carried in data. Another way to exploit multiple data sources is to learn common or source-invariant features that can be transferred to another independent data source. This setting is known as single source or multiple source domain adaptation. This research aims to propose efficient and effective methods enabling us to either generating data followed the constraints specified by multiple data sources or learning from multiple sources in a transfer learning scenario.

1 Publication Outcomes

These results have been documented in 5 research papers accepted for publication at the top-notch conferences in machine learning and artificial intelligence including IJCAI2019, ICML2020, IJCAI2021, UAI2021, and ICCV2021. In what follows, we provide the details of publications and briefly summarize each publication.

1. **Trung Le**, Quan Hoang, Hung Vu, Tu Dinh Nguyen, Hung Bui, and **Dinh Phung**. "Learning generative adversarial networks from multiple data sources." In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 2823-2829. 2019.
   - This work proposes a novel deep generative model allowing us to generate data which are constrained to follow some characteristics of some data sources and avoid some characteristics of other data sources.

2. Hoang Quan, **Trung Le**, and **Dinh Phung**. "Parameterized Rate-Distortion Stochastic Encoder." In International Conference on Machine Learning, pp. 4293-4303. PMLR, 2020.
   - This work proposes a Parameterized Rate-Distortion Stochastic Encoder developed based on the rate distortion and information-bottleneck theories.

3. Tuan Nguyen, **Trung Le**, He Zhao, Hung Quan Tran, Truyen Nguyen, and **Dinh Phung**. "Tidot: A teacher imitation learning approach for domain adaptation with optimal transport." In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2021.
   - This work develops a novel optimal transport based imitation learning mechanism that can be applied successfully to single source transfer learning to earn state-of-the-art performance.

4. Tuan Nguyen, **Trung Le**, He Zhao, Quan Hung Tran, Truyen Nguyen, and **Dinh Phung**. "Most: Multi-source domain adaptation via optimal transport for student-teacher learning." In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), 2021.
   - This work develops a novel optimal transport based imitation learning mechanism that can be applied successfully to multiple source transfer learning to earn state-of-the-art performance.

5. Van-Anh Nguyen, Tuan Nguyen, **Trung Le**, Quan Hung Tran, **Dinh Phung**. "STEM: An Approach to Multi-Source Domain Adaptation With Guarantees ". In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.
   - This work develops a teacher-student framework with theoretical guarantees which obtains state-of-the-art performance in multiple source transfer learning.

2 Acknowledgment, Collaborations, Partnerships, and Human Involved

We deeply appreciate the US Air Force for sponsoring and supporting this fundamental research. This sponsorship is especially precious and valuable to Dr Trung Le at his early research career. This strongly assists us in promoting and strengthening our research and collaboration with our research partners including Dr Quan Hung Tran at Adobe Research and Dr Truyen Nguyen at University of Akron. We really enjoy the journey of this research and the fruitful outcome from this research opens doors for our further and future research.

Moreover, the US Air Force grant helps us to receive two Adobe gift funds, each of which values 10K USDs through our collaboration with Dr Quan Hung Tran from Adobe Research. Last but not least, the US Air Force grant strongly assists us in promoting new research problems in our research group which involve and engage our PostDoc and PhD students including Dr He Zhao (a PostDoc), Dr Hung Vu (a graduated PhD student), Mr Hoang Quan (a PhD student), and Mr Tuan Nguyen (a PhD student).

3 Learning Generative Adversarial Networks from Multiple Data Sources

3.1 Introduction

Modern machine learning systems need to deal with complex high-dimensional objects such as natural images, motion pictures, speeches, dialog texts and hand-written cursive drawings to name a few. Recent deep generative models, in particular Generative Adversarial Networks (GANs) [28] have quickly become a building block for designing powerful models to work with such high-dimensional objects. The

idea of GAN is to train a generator $G(z)$ where z might come from any arbitrary distribution such that the distribution P_G induced over the values of $G(z)$ (s) as z varies is close to the true data distribution P_{data}. Once trained, generating a new sample is extremely efficient as one can simply draw z then feed it through $G(z)$ where $G(z)$ is a deep neural network (NN). Despite its simplicity, GAN has shown an enormous capacity in dealing with high-dimensional objects and has been enjoying remarkable success from image, video generation [57], image-to-image translation [41] to name a few [27].

However, GAN comes with some important limitations. Central to its formulation [28] is a minimax optimization problem whose Nash equilibrium point minimizes the Jensen-Shanon (JS) divergence between P_{data} and P_G. This JS divergence might be viewed as a 'pulling force' to move generated samples toward data samples. Other variants of GAN have extended this mechanism to different divergences, notably f-divergence family proposed in [69] which generalizes JS divergence via a variational bound whose solution can be characterized tractably. Nonetheless, the ill-posedness of GAN minimax problem and the nature of f-divergence pose the inherent mode collapsing problem where generated samples tend to 'collapse' to a few modes, hence hindering the diversity of generating process [28, 27, 48]. Overcoming this problem has become one of the main research themes in GAN with reasonable success, but still an open problem. Besides f-divergence, WGAN [2, 30, 14] employs the Wasserstein distance and formulates the optimization through the Kantorovich duality, but has its own problems in training due to the constraints encountered in the optimization formulation.

From a practicality viewpoint, while enjoying its research success, GAN is still limited in exploiting data from multiple sources. One particular open and important problem is to extend its current setting to move beyond a single data source to work with multiple data distributions, which currently receives very little research attention. For example, one might generate realistic photos of the beach, but at the same time purposely avoid generating images of a storm whose data collection are available for both beach and storm scenes; or in abnormality detection where one not only has access to normal data, but also partially abnormal data to train with. Real-world scenarios like these have abundant applications and are open to explore. Our main contribution of this paper is to propose a novel approach, leveraging on the success of GAN and recent techniques for this open problem.

Specifically, assume that there exists multiple data distribution with P be the primary and m other auxiliary distributions $P_1, P_2, \ldots, P_m$. Our goal is to recover P via a generative distribution Q as *close* to P as possible (i.e., the pull force), but at the same time, being as *different* from all other P_i(s) as possible (i.e., the push force). It is important to note that Q will *not* be estimated explicitly in a parametric form, but instead a generator function G will be learned such that Q implicitly represents the induced distribution for $G(z)$ (s) where z comes from any arbitrary distribution. This can be then formulated as an optimization problem as in Eq. (1) under a generalized extension for $f-$divergence in the existence of multiple data distributions. Subsequently, we extend the theoretical results in [69], showing that it is still possible to obtain a tractable solutions for the Q's generator, G, and efficient algorithms to train G as well the discriminators. Our proposed model naturally subsumes and extends several important existing variants of GAN, including the original GAN [28], f-GAN [69], and D2GAN [66]. In addition, unlike GAN, the our discriminators at convergence point do not become uniform and redundant, but carry specific meanings which can be exploited for various application uses.

Beside the model contribution, while potentially having a wider application scope, we choose to apply and demonstrate our approach for two specific applications in this paper:

1. *Improving GAN training by overcoming mode collapsing problem.* Interestingly, we show that our approach is flexible enough to be exploited to improve the training of GAN (with a single data distribution), addressing the mode collapsing problem mentioned earlier. To do so, we train the first generator G_1 so that its induced distribution is close to P_{data} as usual. G_1 is anticipated to cover only some modes in P_{data} and known to suffer from the mode collapse problem. To subsequently diversify the generated samples, we need the generator to explore uncovered modes from G_1. Using the proposed approach, we then train the second generator G_2 which is to be as close to P_{data} as possible, but as different from G_1 as possible. This process is repeated until the generators cover sufficient number of modes, resulting in a sequence of generators $\{G_1, G_2, \ldots, G_k\}$. At the generation step, for each sample to be generated, we simply pick a random generator from this

pool.

2. *Generating samples with constraints*. This is an on-going research problem which has been addressed under both supervised and unsupervised setting, notably conditional GAN [59] and InfoGAN [7] respectively. Here we demonstrate that our approach naturally offers an unsupervised solution for generating samples with constraints. Specifically, assume we have m data distributions (supposed to belong to m classes, although we do not need know these class labels explicitly). The goal is to generate samples for just one particular primary data class, and being as different from remainder data classes as possible. For example, we have an *unlabeled* primary data source including images of people with blond, black hairs and wish to generate only images with blond hair; we can find another data source containing images of people with black hair and use it as an auxiliary data source.

We conduct extensive experiments to demonstrate the merits of our simple yet very effective approach. We used the CIFAR-10, STL-10, and ImageNet datasets and computed Frechet Inception Distance (FID) [32] to evaluate our solution to the mode collapsing problem against the baselines. The results show that our approach achieves the best FID scores on these real-world datasets and can generate high-quality images. Beyond addressing the mode collapsing problem, we further demonstrate that our framework is capable of learning from negative examples, e.g., learning to generate faces with non-black hairs while given example of faces with black hair.

3.2 Push and Pull GAN

We now describe how our P2GAN works. Recall that P2GAN maintains an existing set of generators that are currently well occupying some data modes and the principle for sequentially adding a new generator is to encourage this generator seeking for new missing data modes in order to boost the diversity. Guided by this principle, we propose using push forces to push the generated distribution Q of the new generator away those of the previous generators, i.e. $P_1, \ldots, P_m$, while using a pull force to pull the generated distribution Q towards the real data distribution P. We term our model *Push-and-Pull GAN* (P2GAN) and explicitly characterize the pull and push forces as f-divergences. Intuitively, since the previous generators can well occupy some data modes and the new generator is encouraged to generate data samples that mimic the true ones and diverge from the existing ones, this is expected to well explore and occupy some additional missing data modes. In what follows, we present the formulation of P2GAN when adding a new generator based on the existing ones, followed by some theoretical results enabling training P2GAN and a concise discussion on how to generalize P2GAN for other applications, specifically generating images with constraints.

3.2.1 P2GAN Formulation

Let f_i(s) and ϕ be convex, lower semi-continuous functions. At each incremental round, our proposed P2GAN solves the following optimization problem to find a new generator Q

$$\max_{Q \in \mathcal{Q}} \left\{ \sum_{i=1}^{m} \alpha_i D_{f_i}(P_i \| Q) - D_{\phi}(P \| Q) \right\} \tag{1}$$

where $\alpha_1, ..., \alpha_m \geq 0$ are the push parameters and $\mathcal{Q}$ is a suitable class of functions.

This formulation is natural as it is clear that the optimal solution Q^* for Eq. (1) is the closest to the data distribution P while furthermost from P_i (s) in the f-divergence optimization sense. The parameter α_i is a hyper-parameter to adjust how favorable we would like Q^* is to be different from P_i (s). Setting $\alpha_1, \ldots, \alpha_m$ to be small will favor Q^* to be closer to P and vice versa[1]. In its most general form, the formulation allows us to use different kind of f-divergence for different pull and push forces. In our experiments, we will simply use standard JS for all divergences.

[1]Note that we purposely do not specify a hyper-parameter for the pulling term $-D_{\phi}(P\|Q)$ as it is implicitly controlled via all α_i (s). However, in our discussion for a extended version of Eq. (1) later in the supplementary material, such parameters will need to be explicitly introduced.

3.2.2 Training P2GAN

We now assume that the generative distribution Q is formed by a NN-based generator G and the source of randomness $\mathbf{z} \sim P_{\mathbf{z}}$ (i.e., the noise distribution). Let $S(\mathbf{x})$ be the *primary discriminator* specifying a score for $\mathbf{x}$ to be more likely generated from the primary distribution P rather than the generative distribution Q. Likewise, for each auxiliary distribution P_i, denote by $\tilde{S}_i(\mathbf{x})$ the *auxiliary discriminator* scoring the degree to which $\mathbf{x}$ is generated from P_i rather than the generative distribution Q. Once again, all discriminator functions S and $\tilde{S}_i$ (s) are parameterized by deep NNs.

We propose to solve the following optimization problem, which is later on proved in Theorem 2 to be equivalent to the our formulation in Eq (1):

$$\max_{G,\tilde{S}_{1:m}} \min_{S} \mathscr{J}\left(G, S, \tilde{S}_{1:m}\right) \tag{2}$$

where we have defined the objective function $\mathscr{J}\left(G, S, \tilde{S}_{1:m}\right)$ as:

$$-\mathop{\mathbb{E}}_{P}[g(S(\mathbf{x}))] + \mathop{\mathbb{E}}_{P_{\mathbf{z}}}[\phi^*(g(S(G(\mathbf{z}))))]$$
$$+\sum_{i=1}^{m} \alpha_i \left[\mathop{\mathbb{E}}_{P_i}\left[\tilde{g}_i\left(\tilde{S}_i(\mathbf{x})\right)\right] - \mathop{\mathbb{E}}_{P_{\mathbf{z}}}\left[f_i^*\left(\tilde{g}_i\left(\tilde{S}_i(G(\mathbf{z}))\right)\right)\right]\right]$$

and g, $\tilde{g}_i$ (s) are the monotonic increasing wrapping functions mapping from $\mathbb{R}$ to $\mathrm{dom}(\phi^*)$ and $\mathrm{dom}(f_i^*)$ respectively to ensure a valid optimization problem.

We note that with monotonic increasing functions for Fenchel conjugate function in the f-divergences (e.g., as listed in our supplementary material), minimizing the first two terms in above objective function w.r.t S is expected to return high score $S(\mathbf{x})$ for $\mathbf{x} \sim P$ and low for $\mathbf{x} \sim Q$. Likewise, maximizing the last two terms w.r.t $\tilde{S}_i$ returns high score $\tilde{S}_i(\mathbf{x})$ for $\mathbf{x} \sim P_i$ and low score for $\mathbf{x} \sim Q$.

Building upon the result from [69] for the single distribution case, we formally show in Theorem 1 and 2 that optimal solutions $\tilde{S}^*_{1:m}(\mathbf{x})$ and $S^*(\mathbf{x})$ for Eq. (2) when Q is held fixed can be obtained analytically, and that optimizing Eq. (2) is indeed equivalent to solving the optimization problem in Eq. (3). From a high level perspective, this results are natural, given the connection between f-divergence and a suitable optimal loss of the best classifier trying to separate data coming from the two distributions. One thing to remark about Eq. (2) is that despite having to deal with a set of new discriminators for the push forces, these discriminators only appear in the outer max problem, unlike the discriminator for the pull force which has to appear in the inner min problem.

Theorem 1. *Given the generative distribution Q (i.e., G), the optimal solutions* $\tilde{S}^*_{1:m}(\mathbf{x})$ *and* $S^*(\mathbf{x})$ *of the optimization problem in Eq. (2) can be evaluated as*

$$\tilde{S}_i^*(\mathbf{x}) = \tilde{g}_i^{-1}\left(\nabla f_i\left(\frac{p_i(\mathbf{x})}{q(\mathbf{x})}\right)\right), 1 \le i \le m$$
$$and\, S(\mathbf{x}) = g^{-1}\left(\nabla\phi\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)\right)$$

Theorem 2 further establishes the Nash equilibrium point of the minimax problem in Eq. (2) to be a solution of the optimization problem in Eq. (3).

Theorem 2. *The optimization problem in Eq. (2) is equivalent to the following optimization problem:*

$$max_Q\left(\sum_{i=1}^{m} \alpha_i D_{f_i}(P_i \| Q) - D_\phi(P \| Q)\right) \tag{3}$$

3.2.3 Generating Data with Constraints

We can adopt the framework of P2GAN to tackle the problem of generating samples with constraints in an unsupervised manner. To be more precise, assume that we have a primary *unlabeled* data source with m classes: $C_1, \ldots, C_m$. We wish to train a generative model that can generate samples that mimic real data

in the primary data source except those in some classes including $C_{i_1}, \ldots, C_{i_k}$. We further assume that we can find additionally auxiliary data sources of classes $C_{i_1}, \ldots, C_{i_k}$. We train a generative model generated data samples satisfying the aforementioned constraints by pushing from the auxiliary data sources while pulling to the primary data source. In addition, the auxiliary data sources do not need to be a part of the primary data source. For example, assume that we have a primary data source including images of people with black and blond hairs and we wish to generate only images with blond hair; we can find images from another data source which contains images of people with black hair and use it as auxiliary data source.

3.2.4 Addressing the Mode Collapse

The underlying idea of using the auxiliary data sources to address the mode collapse is as follows. We train the generator G_1 by minimizing $D_{f_1}(P\|Q_{G_1})$ where $Q_{G_1}$is the push-forward distribution of the noise distribution via the generator G_1. It is likely that Q_{G_1} can only cover some modes in the real data distribution P. To enable the discovery of other modes in the real data distribution P, we set Q_{G_1} as the first auxiliary data distribution and train the second generator G_2 in such a way that its push-forward distribution Q_{G_2} minimizes $-D_h(Q_{G_1}\|Q_{G_2}) + D_{f_2}(P\|Q_{G_2})$. This minimization encourages the resulting distribution Q_{G_2} to simultaneously move away the distribution Q_{G_1} and toward the real data distribution P, hence pushing Q_{G_2} to cover more other modes in the real data distribution P. This process is proceeded by setting Q_{G_1}, Q_{G_2} as two auxiliary data distributions and until the resulting distributions $Q_{G_1}, Q_{G_2}, \ldots, Q_{G_K}$ can cover most of modes in the real data distribution P.

3.3 Experiment

In this section, we first extensively demonstrate how our P2GAN can be used to address the mode collapsing problem, and achieving the best FID scores on CIFAR-10, STL-10, and ImageNet in comparison with current strongest baselines. This is then followed by another application to generate samples with constraints to demonstrate the flexibility of our P2GAN beyond its use for addressing the mode collapse. Regarding network architectures used in all experiments, the generators share parameters in all layers except for the weights from the input to the first hidden layer; the discriminators share parameters in all layers except for the weights from the penultimate hidden layer to the output layer; auxiliary sources generate the same pushing force, all α_i in Eq. (1) are equal, and their sum is denoted by α. We refer to the supplementary material for more details about model architectures and hyper-parameter setting.

3.4 Addressing the Mode Collapse with P2GAN

We now demonstrate the performance of our proposed model in addressing the mode collapsing problem on both synthetic 2D and real-world large-scale datasets. We compare the results of our method with those of the state-of-the-art GAN's variants by replicating experimental settings in the original work.

3.4.1 Synthetic Data

First, we reuse the experimental design proposed in [58] to investigate how our P2GAN explores multiple data modes. The training data is sampled from a 2D mixture of 8 isotropic Gaussian distributions with a covariance matrix of $0.02I$ and means arranged in a circle of zero centroid and radius of 2.0. The small variance creates low density regions, thus separating the modes.

We start with one generator and sequentially add a new generator after every $1,000$ training epochs until reaching 8 generators, and then train the entire network for $15,000$ epochs in total. The generators have an input layer with 256 noise units drawn from isotropic multivariate Gaussian distribution $\mathcal{N}(0, I)$, and two hidden layers with 128 ReLU units each. The discriminators contain one hidden layer with 128 ReLU units. The pushing parameter α is set to 0.05. Fig. 1 shows the evolution of data generated by P2GAN. It can be seen that each new generator at first generates a cluster at the center of the circle and

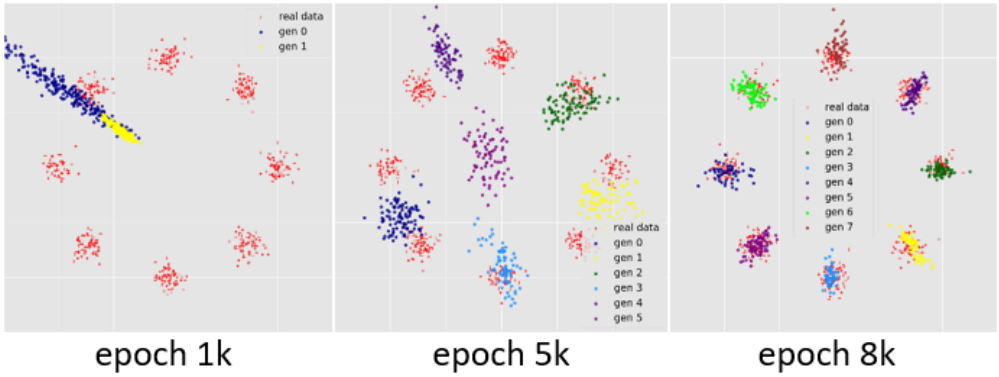
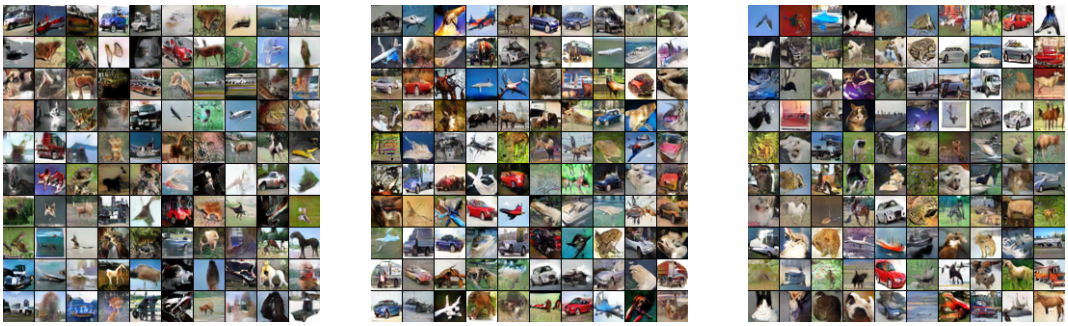


Figure 1: Evolution of data generated by P2GAN. Data samples from the 8 Gaussians are in red, and generated data by each generator are in a different color.



(a) Standard GAN. (b) First gen of 5-gen P2GAN. (c) 5-gen P2GAN.

Figure 2: Random samples generated by GAN models trained on CIFAR-10.

gradually moves to one of the unoccupied modes. Eventually at epoch 8,000, each generator captures one mode, and 8 generators altogether effectively cover all the modes.

3.4.2 Real-world Datasets

In this section, we present our experiments on real-world datasets. We first conduct experiment on the CIFAR-10 dataset [43] to investigate the influence of the number of generators. The result shows that P2GAN model helps stabilize training and improve visual quality of generated samples over the standard GAN with a similar architecture. Next we perform experiments on the STL-10 [8] and ImageNet [80] datasets to prove that P2GAN not only achieves the best quantitative results but also generates highly realistic images.

Data and Evaluation Metric. CIFAR-10 contains 50,000 32×32 training images of 10 classes, including *airplane*, *automobile*, *bird*, *cat*, *deer*, *dog*, *frog*, *horse*, *ship*, and *truck*. STL-10 contains about 100,000 96×96 images, sub-sampled from ImageNet, and is a more diverse dataset than CIFAR-10. ImageNet is the largest and most diverse datasets with more than 1.2 million images from 1,000 classes. We follow the procedure of [44] to resize the STL-10 and ImageNet images down to 48×48 and 32×32, respectively, for a fair comparison with the baselines in [96, 61, 38]. We also resize ImageNet images down to 64×64 to examine P2GAN's capability on higher-resolution data. To quantitatively assess the quality and diversity of generated samples, we adopt Frechet Inception Distance (FID) proposed in [32] that is more advanced than Inception score [83] since it compares the statistics of synthetic samples with those of the real samples, hence capturing the similarity of the two distributions better [32]. FIDs are computed on samples of 50,000 images.

Model Architecture with CNN and ResNet. Our network is designed following the DCGAN's architecture [76], which we refers to as *Standard CNN*, and the *ResNet* architecture used in [30]. For the pulling force, we use the *Jensen-Shannon* (JS) divergence as in the standard GAN. For the pushing

force, we experiment with both JS and KL divergences. The results are similar but the JS is more numerically stable, hence we eventually use it for both pulling and pushing. Discriminators of the same type, either *pull* or *push*, share parameters in all layers except for the weights from the last layer to the output layer. Starting with one generator, we add a new generator every fixed number of epochs until the number of generators reaches a predefined number K , and continue training the entire network. The learning process is terminated after 150 epochs for CIFAR-10, 100 epochs for STL-10, and 50 epochs for ImageNet.

Hyperparameter Setting. We use Adam optimizer with a batch size of 64. The learning rate and the first-order momentum are set at 0.0002 and 0.5, respectively. Regarding the pushing parameter α, we observe that varying α between 0.001 and 0.1 makes only little influence on the quantitative results, but large α can lead to less visually appealing samples as generators push each other too hard. As a result, we employed a gentle force of 0.01 for all experiment. We vary the total number of generators K in $\{1,3,5,10,15\}$ for our experiment on CIFAR-10 to investigate the influence of K. We add a new generator for every 15, 10, 5, 3 epochs when K is $3,5,10,$ and 15, respectively. For STL-10 and ImageNet, we simply set K at 10. For models using the ResNet architecture, we apply noise-reduced regularization [79] and set the regularization parameter γ at 2.0.

The Influence of the Number of Generators. Tab. 1 compares FIDs (lower is better) for P2GAN with different values of K, the number of generators. All models use the ResNet architecture. It should be noted that the P2GAN model with only one generator turns into the standard GAN, so we refer to it as *standard GAN*. The results show that using more generators improves FID and the performance peaks at $K = 5$. However, FID slightly deteriorates when K is 10 or 15. This behavior is consistent with that in our experiment with synthetic data (see Sec. 3.4.1) when the generators are crowded. In general, more diverse data can accommodate more generators.

# Generators	FID
1	26.7
3	25.7
5	**20.1**
10	23.1
15	23.5

Table 1: Comparison of FIDs (lower is better) on CIFAR-10 for P2GAN models with different number of generators.

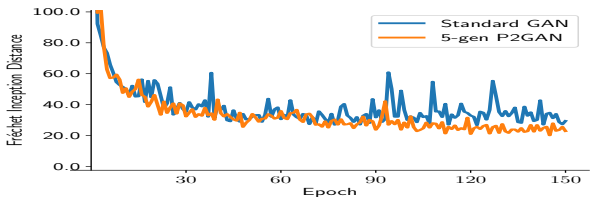


Figure 3: Comparison of FIDs for the P2GAN models with 1 and 5 generators on CIFAR-10. The 5-gen model is better and more stable FIDs.

We further compares the standard GAN model with the 5-gen P2GAN model. Fig. 3 plots FIDs of each model over training epochs. The 5-gen P2GAN achieves better and more stable performance. After 60 epochs, the FID of standard GAN becomes unstable and deteriorates. On the contrary, the FID of 5-gen P2GAN remains stable and keeps improving. Fig. 2 shows random samples generated by the standard GAN (a), the first generator of the 5-gen P2GAN (b) and the 5-gen P2GAN (c). Due to limited space, we refer to the supplementary material for samples generated by other generators of the 5-gen P2GAN. Compared to the standard GAN, the first generator of 5-gen P2GAN generates fewer classes of objects (mostly car, airplanes and birds) but much clearer images. As a result, the 5-gen P2GAN (i.e., P2GAN with 5 generators) produces more diverse and visually appealing samples. This analysis demonstrates that P2GAN has more stable learning, achieves stronger quantitative evaluation and generates better samples.

Frechet Inception Distance Results. Tab. 2 reports the FIDs obtained by our P2GAN and the latest baselines collected from recent work in literature [32, 61]. For fair comparison, we also implement MGAN [38] using the ResNet architecture with noise-reduced regularization. It is worthy to note that in the *standard CNN* group, DCGAN+TTUR and P2GAN share the same architecture similar to DCGAN, while WGAN-GP and SN-GANs employ a similar architecture for the generator, but add three more

(a) SN-GAN. (b) P2GAN

Figure 4: Random 48×48 STL-10 samples generated by SN-GAN (left) and P2GAN (right).

layers to the discriminator [61]. Overall, the P2GAN significantly outperforms other baselines on both CIFAR-10 and STL-10 in terms of FID. On the ImageNet dataset, our P2GAN model with ResNet architecture achieves a FID of *18.1* while MGAN with ResNet obtains *21.8*. Note that we have not found the FIDs of other baselines for this dataset. These impressive results demonstrate the effectiveness of P2GAN in addressing the mode collapse.

Model	CIFAR-10	STL-10
-Standard CNN-		
WGAN-GP	40.2	55.1
DCGAN [76]	37.7	–
DCGAN + TTUR [32]	36.9	–
LayerNorm [61]	33.9	75.6
WeightNorm [61]	34.7	73.4
Orthonormal [61]	29	46
WeightClipping [61]	42.6	64.2
MGAN [38]	26.7	–
SN-GANs [61]	25.5	43.2
P2GAN (ours)	25.2	53.3
-ResNet-		
WGAN-GP [30]	29.3	–
WGAN-GP + TTUR [32]	24.8	–
SN-GANs [61]	21.7	40.1
MGAN [38]	20.9	45.3
P2GAN (ours)	**20.1**	**38.2**

Table 2: Comparison of FIDs (lower is better) with *unsupervised* image generators.

Generated samples. Random CIFAR-10 images generated by our proposed model with ResNet architecture were discussed and presented previously in Fig. 2. For 48×48 STL-10, Fig. 4 compares random samples generated by P2GAN and the closest baseline, SN-GAN [61]. SN-GAN samples sketch the general shapes of some objects but are blurry and fragmented. P2GAN samples are much sharper, more vivid and visually appealing depiction of a variety of objects.

Figure 5: Samples generated by P2GAN model pushed by images of females with black hair. It can be seen that most generated images are of females with blond hair.

3.5 Generating Data with Constraints with P2GAN

In this experiment, we verify the effectiveness of the pushing force in our framework. We consider the scenario where the primary *unlabeled* data source consists of images from two classes, whilst the auxiliary *unlabeled* data source only has images from one of those two classes. The images in auxiliary source, though from the same class, are different from those contained in the primary data. In particular, we conduct the experiment where we have two classes: blond and black hair. Our task is to train generators to generate blond hair. In what follows we present our data construction, model setting and the results.

Data construction. For the blond/black hair setting, we randomly pick $12,000$ female images with black hair, and $12,000$ with blond hair from the CelebA dataset [51], and merge them to create the primary data source. We then randomly select $11,000$ images of females with black hair, which do not appear in the primary data source, from the CelebA dataset to create the auxiliary data source.

Model setting. In generating data with constraints, the generator faces two conflicting goals. The first goal is to driven by the pulling force to generate data for both two classes. The second goal is driven by the pushing force to avoid generating data of the class in auxiliary source. To address this conflict, we train two generators G_1 and G_2 where: the primary source pulls both G_1 and G_2; the auxiliary source pulls G_2 but pushes G_1; and G_1 and G_2 push each other. The idea is that G_2 will cover a part of the primary data that is similar to the auxiliary data, and contribute additional force to push G_1 to generate images as our desired results. Here G_1 and G_2 also follow the same parameter sharing scheme as mentioned above.

Results. Empirically we observe that setting the pulling force produced by the primary data to 1.0 and applying a small pulling force of the auxiliary source of 0.1 and a gentle pushing force (of the auxiliary source, and between $G_1$and G_2) of 0.01 can effectively encourage G_1 and G_2 to generate different data. Fig. 5 shows samples generated by G_1 for blond/black hair experiment. It can be seen that most images generated by G_1 are of females with blond hair, whilst some tend to mix with other colors such as red, green or white.

4 MOST: Multi-Source Domain Adaptation via Optimal Transport for Student-Teacher Learning

4.1 Introduction

Recent advances in deep learning have succeeded in undertaking visual learning tasks under the support of massive annotated data. However, directly transferring knowledge of such a learned model to a novel domain can undesirably degrade its performance due to the existence of *domain shift* [75]. To address this issue, a diverse range of approaches in domain adaptation (DA) has been proposed from shallow domain adaptation [9, 11] to deep domain adaptation [20, 52, 87, 15, 68, 67]. While the conventional DA aims to transfer knowledge from a labeled source domain to an unlabeled target domain, in many real-world contexts, labeled data are collected from multiple domains, for example, images taken under different conditions (e.g., weather, poses, lighting conditions, distinct backgrounds, and etc) [107]. In this paper, we address a challenging but more practical transfer learning problem named multi-source domain adaptation (MSDA) in which we need to transfer knowledge from multiple distinct domains to a single unlabeled target domain.

Imitation learning method has been known as *learning from demonstration*. Specifically, there are two fundamental agents: an expert teacher and a student. The former agent knows how to do its job perfectly, whilst the latter learns a policy to mimic the teacher's behavior. This learning paradigm has been applied in reinforcement learning and sequence prediction [1, 34].

Inspired by the principle of imitation learning, we propose in this paper a novel model for MSDA, named ***M**ulti-Source Domain Adaptation via **O**ptimal Transport for **S**tudent-**T**eacher Learning* (MOST). When applying the teacher-student mechanism in the context of MSDA, we seek solutions for two naturally raised questions: i) how is the teacher determined? and ii) what are the principle and mechanisms to enable the student to mimic its teacher? We address these two questions by developing a rigorous and intuitive theory based on the literature of optimal transport (OT) [93, 85, 73]. Our approach (see Sections 4.4 and 4.5) postulates that the teacher is a combination of domain experts learned perfectly under the support of labeled source samples, and the student aims to predict unlabeled target samples via imitating the prediction of the teacher. We summarize our contributions in this work as follows:

- We propose a rigorous OT-based theory to leverage imitation learning into domain adaptation. Our general paradigm can further apply to many learning problems including reinforcement learning.
- Under imitation learning's perspective, we propose a novel model for MSDA, which utilizes two cooperative agents: teacher and student. The implementation of MOST is also available online[2].
- Comprehensive experiments are conducted on benchmark datasets for multi-source domain adaptation including Digits-five, Office-Caltech10, and Office-31. The experimental results show that our MOST achieves state-of-the-art performance on those benchmark datasets to the best of our knowledge.

4.2 Related Work

4.2.1 Unsupervised Domain Adaptation

A variety of unsupervised domain adaptation (UDA) approaches have been successfully applied to generalize a model learned from a labeled source domain to an unlabeled novel target domain. Several existing methods based on discrepancy-based alignment to minimize a different discrepancy metric to close the gap between the source and target domains [52, 91, 88, 104, 50]. Another branch of UDA methods have leveraged adversarial learning wherein generative adversarial networks [28] were employed to align the source and target domains on either feature-level [20, 90, 53] or pixel-level [23, 4, 84, 101]. On the

[2]https://github.com/tuanrpt/MOST

category-level, some approaches utilized dual classifier [82, 50], or domain prototype [97, 71, 100] to investigate the category relations across domains.

4.2.2 Multi-Source Domain Adaptation

The fundamental study in [56, 3] has shed light upon the wide applications of MSDA, such as in [17, 102]. Based on the above works, [39] gave strong theoretical guarantees for cross-entropy and other similar losses, which is a normalized solution for the MSDA problems. Recently, [107] deployed domain adversarial networks to align the target domain to source domains. [102] proposed a new model to deal with the *category shift*, which is the case where sources may not completely share their categories. [72] introduced a model that aligns moments of source and target feature distributions in latent space. Multi-source distilling model was proposed in [108] to fine-tune the generator and classifier separately and utilized the domain weight to aggregate target prediction. Finally, the work in [94] deployed a graph convolutional network to conduct domain alignment on the category-level.

4.2.3 Optimal Transport

Optimal Transport (OT) has raised interest in various fields including domain adaptation. Many works in single-source domain adaptation have used OT as a tool to mitigate the domain gap via minimizing the cost of complex distributions [93, 10, 85, 105, 15, 65, 47]. Recently, [50] proposed using the sliced Wasserstein distance on the category-level, whereas [98] proposed SPOT in which the optimal transport plan is approximated by a pushforward of a reference distribution, and cast the optimal transport problem into a minimax problem. The OT-based DA work in [103] has leveraged spatial prototypical information and intra-domain structures of image data to reduce the negative transfer caused by target samples near decision boundaries. Notably, [11] developed a new framework to connect the theory of optimal transport and domain adaptation [9], which later inspired an OT-based deep DA method (DeepJDOT) [15] and a learning from multiple data sources method (JCPOT) [77].

4.3 Background

4.3.1 Optimal Transport

Consider two distributions $\mathbb{P}$ and $\mathbb{Q}$ which operate on the domain $\Omega \subseteq \mathbb{R}^d$, let $d(x,y)$ be a non-negative and continuous cost function or metric. In the modern mathematical language, the very first notion of optimal transport (i.e., Monge problem) [93, 85] aims to find the minimum total cost to transport mass from $\mathbb{Q}$ to $\mathbb{P}$ as

$$\mathscr{M}_d(\mathbb{Q},\mathbb{P}) := \min_{T:T_{\#}\mathbb{Q}=\mathbb{P}} \mathbb{E}_{x\sim\mathbb{Q}}\left[d(x,T(x))\right],$$

where $T_{\#}\mathbb{Q}$ is the push-forward distribution of $\mathbb{Q}$ via the transport map T. A relaxation of the Monge problem (MP), a.k.a the Kantorovich problem (KP), is defined as

$$\mathscr{K}_d(\mathbb{Q},\mathbb{P}) := \min_{\gamma\in\Gamma(\mathbb{Q},\mathbb{P})} \mathbb{E}_{(x,y)\sim\gamma}\left[d(x,y)\right], \tag{4}$$

where γ is a coupling admitting $\mathbb{Q},\mathbb{P}$ as marginals.

Under some mild conditions as stated in Theorems 1.32 and 1.33 in [85], KP is identical to MP and for convenience we denote both $\mathscr{M}_d$ and $\mathscr{K}_d$ collectively by $\mathscr{W}_d$ as $\mathscr{W}_d(\mathbb{Q},\mathbb{P}) = \mathscr{K}_d(\mathbb{Q},\mathbb{P}) = \mathscr{M}_d(\mathbb{Q},\mathbb{P})$.

In addition, under some mild conditions as stated in Theorem 5.10 in [93], we can replace the primal form by its corresponding dual form

$$\mathscr{W}_d(\mathbb{Q},\mathbb{P}) = \max_{\phi\in\mathscr{L}_1(\Omega,\mathbb{P})} \left\{\mathbb{E}_{\mathbb{Q}}\left[\phi^c(x)\right] + \mathbb{E}_{\mathbb{P}}\left[\phi(y)\right]\right\}, \tag{5}$$

where $\mathscr{L}_1(\Omega,\mathbb{P}) := \{\psi : \int_\Omega |\psi(y)|\,\mathrm{d}\mathbb{P}(y) < \infty\}$ and ϕ^c is the c-transform of function ϕ defined as $\phi^c(x) := \min_y \{d(x,y) - \phi(y)\}$.

Clustering view of optimal transport. This view of optimal transport has been utilized to study a rich class of hierarchical and multilevel clustering problems [35, 36]. We now present the clustering view of optimal transport which assists us to interpret our method developed in the sequel. Let $\mathbb{P}$ and $\mathbb{Q}$ be two discrete distributions defined as

$$\mathbb{P} := \frac{1}{m}\sum_{i=1}^{m}\delta_{u_i} \text{ and } \mathbb{Q} := \frac{1}{n}\sum_{j=1}^{n}\delta_{v_j},$$

where δ_x indicates a Dirac measure centered at x. Without loss of generality, we can assume that $n \leq m$ and consider the Wasserstein distance $\mathscr{W}_d(\mathbb{P},\mathbb{Q})$ w.r.t. a metric d. The following theorem characterizes the clustering view of OT.

Theorem 3. *Consider the following optimization problem:* $\min_{v_{1:n}} \mathscr{W}_d(\mathbb{P},\mathbb{Q})$. *Let* $v^*_{1:n}$ *and* $\mathbb{Q}^* := \frac{1}{n}\sum_{j=1}^{n}\delta_{v^*_j}$ *be its optimal solution and* T^* *be the optimal transport map as*

$$T^* = argmin_{T:T_{\#}\mathbb{P}=\mathbb{Q}^*}\sum_{i=1}^{m} d(u_i, T(u_i)).$$

Furthermore, let $c^*_{1:n}$ *and* σ^* *denote the optimal solution of the following clustering problem:*

$$\min_{c_{1:n},\sigma\in\Pi(m,n)}\sum_{i=1}^{m} d\left(u_i, v_{\sigma(i)}\right),$$

where $\Pi(m,n)$ *is the set of surjective maps from* $\{1,...,m\}$ *to* $\{1,...,n\}$. *We then have* $c^*_{1:n} = v^*_{1:n}$ *and* $T^*(u_i) = v^*_{\sigma^*(i)}$.

The above theorem states that if we learn the atoms of $\mathbb{Q}$ to minimize $\mathscr{W}_d(\mathbb{P},\mathbb{Q})$ w.r.t. the metric d, the optimal atoms of $\mathbb{Q}$ become the centroids of the clusters formed by the atoms of $\mathbb{P}$ or the atoms of $\mathbb{Q}$ are moving to find the groups of atoms of $\mathbb{P}$ with the aim to minimize the distortion w.r.t. the metric d (see our *supplementary material* for more details).

4.3.2 Entropic Regularized Duality

To enable the application of optimal transport in machine learning and deep learning, Genevay et al. developed an entropic regularized dual form in [22]. First, they proposed to add an entropic regularization term to the primal form in (4)

$$\mathscr{W}^{\varepsilon}_d(\mathbb{Q},\mathbb{P}) := \min_{\gamma\in\Gamma(\mathbb{Q},\mathbb{P})}\left\{\mathbb{E}_{(x,y)\sim\gamma}[d(x,y)] + \varepsilon D_{KL}(\gamma\|\mathbb{Q}\otimes\mathbb{P})\right\}, \tag{6}$$

where ε is the regularization rate, $D_{KL}(\cdot\|\cdot)$ is the Kullback-Leibler (KL) divergence, and $\mathbb{Q}\otimes\mathbb{P}$ represents the specific coupling in which $\mathbb{Q}$ and $\mathbb{P}$ are independent. Note that when $\varepsilon\to 0$, $\mathscr{W}^{\varepsilon}_d(\mathbb{Q},\mathbb{P})$ approaches $\mathscr{W}_d(\mathbb{Q},\mathbb{P})$ and the optimal transport plan γ^*_ε of (6) also weakly converges to the optimal transport plan γ^* of (4). In practice, we set ε to be a small positive number, hence γ^*_ε is very close to γ^*.

Second, using the Fenchel-Rockafellar theorem, they obtained the following dual form w.r.t. the potential ϕ

$$\begin{aligned}\mathscr{W}^{\varepsilon}_d(\mathbb{Q},\mathbb{P}) &= \max_{\phi}\left\{\int \phi^c_\varepsilon(x)\,\mathrm{d}\mathbb{Q}(x) + \int \phi(y)\,\mathrm{d}\mathbb{P}(y)\right\}\\ &= \max_{\phi}\left\{\mathbb{E}_{\mathbb{Q}}[\phi^c_\varepsilon(x)] + \mathbb{E}_{\mathbb{P}}[\phi(y)]\right\},\end{aligned} \tag{7}$$

where $\phi^c_\varepsilon(x) := -\varepsilon\log\left(\mathbb{E}_{\mathbb{P}}\left[\exp\left\{\frac{-d(x,y)+\phi(y)}{\varepsilon}\right\}\right]\right)$.

4.4 Theoretical Developments

4.4.1 Preliminaries

We first examine a general supervised learning setting. Consider a hypothesis h in a hypothesis class $\mathscr{H}$ and a labeling function f (i.e., $f(\cdot)\in\mathscr{Y}_\triangle$ and $h(\cdot)\in\mathscr{Y}_\triangle$ where $\mathscr{Y}_\triangle := \left\{\pi\in\mathbb{R}^M : \|\pi\|_1 = 1 \text{ and } \pi\geq\mathbf{0}\right\}$

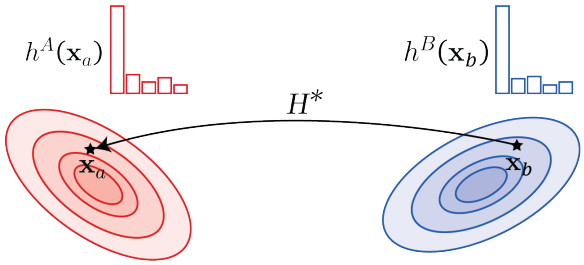


Figure 6: Imitation view explanation. $h^B(x_b)$ for $x_b \sim \mathbb{P}^B$ tries to imitate $h^A(x_a)$ with $x_a = H^*(x_b) \sim \mathbb{P}^A$.

with the number of classes M). Let $d_{\mathscr{Y}}$ be a metric or divergence over $\mathscr{Y}_\triangle$. We further define the general loss of the hypothesis h w.r.t. the data distribution $\mathbb{P}$ and the labeling function f as:

$$\mathscr{L}(h, f, \mathbb{P}) := \int d_{\mathscr{Y}}(h(x), f(x)) \, \mathrm{d}\mathbb{P}(x).$$

It is worth noting that by defining the metric or divergence $d_{\mathscr{Y}}$ as

$$d_{\mathscr{Y}}(h(x), f(x)) := \sum_{i=1}^{M} f_i(x) D_{KL}(\mathbf{1}_i \| h(x)),$$

where $\mathbf{1}_i$ is an one-hot vector, we can recover the cross-entropy loss widely used in deep learning.

Next we consider a domain adaptation setting [20, 9] in which we have a source space $\mathscr{X}^S$ endowed with a distribution $\mathbb{P}^S$ and a target space $\mathscr{X}^T$ endowed with a distribution $\mathbb{P}^T$. Given two pairs $z_1 = \left(x_1, y_1^\triangle\right) \in \mathscr{X}^S \times \mathscr{Y}_\triangle$ and $z_2 = \left(x_2, y_2^\triangle\right) \in \mathscr{X}^T \times \mathscr{Y}_\triangle$, we define the cost (distance) function between them as:

$$d(z_1, z_2) := \lambda d_{\mathscr{X}}(x_1, x_2) + d_{\mathscr{Y}}\left(y_1^\triangle, y_2^\triangle\right), \tag{8}$$

where $d_{\mathscr{X}}$ is a metric over $\mathscr{X}^S \times \mathscr{X}^T$ and $\lambda > 0$.

4.4.2 Optimal Transport based Imitation Learning

In what follows, we present the OT based imitation learning which lays foundation for our proposed MOST. Consider two data domains $\mathscr{X}^A$ and $\mathscr{X}^B$ with two data distributions $\mathbb{P}^A$ and $\mathbb{P}^B$ respectively, and assume that $h^A : \mathscr{X}^A \to \mathscr{Y}_\triangle$ is a well-qualified labeling function (classifier) that gives accurate prediction for data instances on $\mathscr{X}^A$ sampled from $\mathbb{P}^A$. We wish to learn a labeling function (classifier) h^B to predict accurately data instances sampled from $\mathbb{P}^B$ by imitating what is done by h^A on $\left(\mathscr{X}^A, \mathbb{P}^A\right)$. Based on the data distribution $\mathbb{P}^A$ and labeling function h^A, we define a distribution $\mathbb{P}_{A,h^A}$ over $\mathscr{X}^A \times \mathscr{Y}_\triangle$ including sample pair $\left(x, h^A(x)\right)$ by first sampling $x \sim \mathbb{P}^A$ and then computing $h^A(x)$. Similarly, we can define another distribution $\mathbb{P}_{B,h^B}$ over $\mathscr{X}^B \times \mathscr{Y}_\triangle$ using the data distribution $\mathbb{P}^B$ and the labeling function h^B. To allow h^B to imitate the behavior of h^A, we propose to inspect the Wasserstein distance (WS) between $\mathbb{P}_{A,h^A}$ and $\mathbb{P}_{B,h^B}$ w.r.t. the cost (metric) function d defined in (8). The following proposition is crucial for us to derive the fundamental mechanism of OT-based imitation learning.

Proposition 4. *The WS distance of interest* $\mathscr{W}_d\left(\mathbb{P}_{A,h^A}, \mathbb{P}_{B,h^B}\right)$ *can be expressed as:*

$$\begin{aligned}
&\min_{L:L_\# \mathbb{P}^A = \mathbb{P}^B} \mathbb{E}_{x\sim\mathbb{P}^A}\left[\lambda d_{\mathscr{X}}(x, L(x)) + d_{\mathscr{Y}}\left(h^A(x), h^B(L(x))\right)\right] \\
= &\min_{H:H_\# \mathbb{P}^B = \mathbb{P}^A} \mathbb{E}_{x\sim\mathbb{P}^B}\left[\lambda d_{\mathscr{X}}(x, H(x)) + d_{\mathscr{Y}}\left(h^B(x), h^A(H(x))\right)\right].
\end{aligned}$$

As indicated by Proposition 4, the optimal transport $H^* : H^*_\# \mathbb{P}^B = \mathbb{P}^A$ is the optimal mover that moves $\mathbb{P}^B$ to $\mathbb{P}^A$ so as to minimize the difference in the predictions of h^B for $x \sim \mathbb{P}^B$ and h^A for $H^*(x) \sim \mathbb{P}^A$. In other words, given $x \sim \mathbb{P}^B$, the optimal transport H^* finds its closest counterpart in the space of $\mathscr{X}^A$

(i.e., $H^*(x)$) so that h^B can conveniently imitate the prediction of h^A on $H^*(x)$) for predicting x (see Figure 6).

To further elaborate on the proposed OT-based imitation learning, we assume that f^A is the ground-truth labeling function for the domain $\left(\mathscr{X}^A, \mathbb{P}^A\right)$ and theoretically prove that if we minimize the Wasserstein distance $\mathscr{W}_d\left(\mathbb{P}_{A,h^A}, \mathbb{P}_{A,f^A}\right)$, we can obtain the optimal solution $h^A_* = f^A$ and can upper-bound this Wasserstein distance by the general loss of h^A over $\left(\mathscr{X}^A, \mathbb{P}^A\right)$ (the statement (iii) in Theorem 5).

Theorem 5. *The following statements hold*

i) Given $\mathscr{X}^A = \mathscr{X}^B = \mathscr{X}$, $\mathscr{W}_d\left(\mathbb{P}_{A,h^A}, \mathbb{P}_{B,h^B}\right) = 0$ *if only if* $\mathbb{P}^A = \mathbb{P}^B$ *and* $h^A = h^B$.

ii) Consider the problem: $\min_{h_A} \mathscr{W}_d\left(\mathbb{P}_{A,h^A}, \mathbb{P}_{A,f^A}\right)$, *the optimal solution is* $h^A_* = f^A$ *obtained with the optimal mover* $L^* : L^*_{\#}\mathbb{P}^A = \mathbb{P}^A$ *to be the identity map.*

iii) $\mathscr{W}_d\left(\mathbb{P}_{A,h^A}, \mathbb{P}_{A,f^A}\right) \leq \mathscr{L}\left(h^A, f^A, \mathbb{P}^A\right)$.

iv) $\mathscr{W}_d\left(\mathbb{P}_{A,h^A}, \mathbb{P}_{B,h^B}\right) \geq \lambda \mathscr{W}_{d_{\mathscr{X}}}\left(\mathbb{P}^A, \mathbb{P}^B\right)$.

4.5 Our Proposed Method

4.5.1 Problem Formulation

In multi-source domain adaptation, we have K multiple source domains with the collected data and labels, and single target domain with the collected data only. We wish to transfer a model learned on the labeled source domains to the unlabeled target domain. Let us denote the collected data and labels for the source domains by $\mathscr{D}^S_k = \left\{\left(sx^k_i, y^k_i\right)\right\}_{i=1}^{N^S_k}$ with k is the index of a source domain, label $y^k_i \in \{1,2,...,M\}$ and collected data without labels for the target domain $\mathscr{D}^T = \{tx_i\}_{i=1}^{N^T}$.

For the sake of simplification, we denote the common space for source domains by $\mathscr{X}^S$. Note that if source domains have different input spaces, we can resize either input images or use appropriate transformations to map them to a common space. We further equip source domains with data distribution $\mathbb{P}^S_{1:K}$ whose density functions are $p^S_{1:K}(x)$. Let us denote the ground-truth labeling functions for source domains by $f^S_{1:K}(\cdot) \in \mathscr{Y}_\Delta$, implying that $p^S_k(y \mid x) = f^S_k(x,y)$ (i.e., $f^S_k(x,y)$ represents the y-th value of $f^S_k(x)$). Therefore, the joint distribution to generate data instance x and categorical label $y \in \{1,...,M\}$ is $p^S_k(x,y) = p^S_k(x) f^S_k(x,y)$.

Regarding the target domain, we define its data space as $\mathscr{X}^T$, data distribution and density function as $\mathbb{P}^T$ and $p^T(x)$, respectively. We further define the ground-truth labeling function for the target domain by f^T which subsequently implies $p^T(y \mid x) = f^T(x,y)$ for a categorical label $y \in \{1,...,M\}$.

Given a discrete distribution π over $\{1,...,K\}$, we define $\mathbb{P}^S_\pi := \sum_{k=1}^K \pi_k \mathbb{P}^S_k$ which is a mixture of $\mathbb{P}^S_{1:K}$. For a data instance $x \sim \mathbb{P}^S_\pi$ (i.e., we sample a hidden index $t \sim Cat(\pi)$ (i.e., the categorical distribution) and then sample $x \sim \mathbb{P}^S_t$), we further define f^S as a labeling function such that $f^S(x)$ is identical to $f^S_t(x)$. By this definition, f^S can be viewed as the ground-truth labeling function over the mixture distribution $\mathbb{P}^S_\pi$. Finally, the mixing proportion π can be the uniform distribution $[\frac{1}{K}, ..., \frac{1}{K}]$ or proportional to the number of training examples in the source domains (i.e., $N^S_{1:K}$). It is worth noting that the mixing proportion π influences the proportion of samples from the individual data sources in the mini-batches. We conduct an ablation study to compare two aforementioned options for π and observe that they are comparable in terms of the predictive performances (see the *supplementary material*).

4.5.2 Multi-source Expert Teacher

Using the labeled source training sets $\mathscr{D}^S_{1:K}$, we can train qualified domain expert classifiers $h^S_{1:K}$ (i.e., $h^S_k(x) \in \mathscr{Y}_\Delta$ represents the prediction probability of h^S_k for a data instance x in the k^{th} source domain) with

good generalization capacity (e.g., $\mathscr{L}\left(h_k^S, f_k^S, \mathbb{P}_k^S\right) \leq \varepsilon$ for some small $\varepsilon > 0$). The next arising question is how to combine these domain experts to achieve a a multi-source expert teacher h^S that can work well on $\mathbb{P}_\pi^S$ (i.e., $\mathscr{L}\left(h^S, f^S, \mathbb{P}_\pi^S\right) \leq \varepsilon$). To this end, we leverage the weighted ensembling strategy in [56, 39] to achieve

$$h^S(x,y) = \sum_{k=1}^{K} \frac{\pi_k p_k^S(x,y)}{\sum_{j=1}^{K} \pi_j p_j^S(x,y)} h_k^S(x,y), \tag{9}$$

where $y \in \{1,2,...,M\}$, and $h_k^S(x,y)$ and $h^S(x,y)$ specify the y-th values of $h_k^S(x)$ and $h^S(x)$ respectively.

The following theorem shows that the multi-source expert teacher h^S can work well on the mixture joint distribution $\mathbb{P}_\pi^S$. More importantly, it works better than the worst domain expert on its source domain. Hence, if each domain expert is an ε-qualified classifier (i.e., $\mathscr{L}\left(h_k^S, f_k^S, \mathbb{P}_k^S\right) \leq \varepsilon$), the multi-source expert teacher h^S is also an ε-qualified classifier (i.e., $\mathscr{L}\left(h^S, f^S, \mathbb{P}_\pi^S\right) \leq \varepsilon$).

Theorem 6. *If $d_{\mathscr{Y}}$ can be decomposed as $d_{\mathscr{Y}}(\alpha,\beta) := \sum_{i=1}^{M} \beta_i \ell(\alpha_i)$ where $\alpha, \beta \in \mathscr{Y}_\Delta$ and ℓ is a convex function, the following statements hold true:*

i) $\mathscr{L}\left(h^S, f^S, \mathbb{P}_\pi^S\right) \leq \max_{1\leq k\leq K} \mathscr{L}\left(h_k^S, f_k^S, \mathbb{P}_k^S\right)$.

ii) If each domain expert is an ε-qualified classifier (i.e., $\mathscr{L}\left(h_k^S, f_k^S, \mathbb{P}_k^S\right) \leq \varepsilon$), the multi-source expert teacher h^S is also an ε-qualified classifier (i.e., $\mathscr{L}\left(h^S, f^S, \mathbb{P}_\pi^S\right) \leq \varepsilon$).

In what follows, we present how to train the multi-source expert teacher h^S. Our workaround to train h^S comes from the following theoretical observation. Assume that we have K distributions $\mathbb{R}_{1:K}$ with density functions $r_{1:K}(z)$. We form a joint distribution $\mathscr{D}$ of a data instance z and label $t \in \{1,...,K\}$ by sampling an index $t \sim \text{Cat}(\pi)$, sampling $x \sim \mathbb{R}_t$, and collecting (z,t) as a sample from $\mathscr{D}$. With this setting, we have the following corollary.

Corollary 7. *If we train a source domain discriminator $\mathscr{C}$ to classify samples from the joint distribution $\mathscr{D}$ using the cross-entropy loss (i.e., $CE(\cdot,\cdot)$), the optimal source domain discriminator $\mathscr{C}^*$ defined as*

$$\mathscr{C}^* = argmin_{\mathscr{C}} \mathbb{E}_{(z,t)\sim\mathscr{D}}\left[CE\left(\mathscr{C}(z), t\right)\right]$$

satisfies $\mathscr{C}^*(z) = \left[\frac{\pi_i r_i(z)}{\sum_j \pi_j r_j(z)}\right]_{i=1}^{K}$.

Corollary 10 suggests us a way to compute the weights of the domain experts in (15) in which for a given y, the distributions $p_{1:K}^S(x,y)$ play roles of $r_{1:K}(z)$ where $z = (x,y)$. More specifically, for each $m \in \{1,...,M\}$, we sample $t \sim Cat(\pi)$, then sample $(x, y=m)$ from $p_t^S(x, y=m)$, and train a source domain discriminator $\mathscr{C}_m(x, y=m)$ (i.e., only consider (x,y) in which x has label $y=m$) to distinguish the source domain t of $(x, y=m)$. We finally use $\mathscr{C}_m(x, y=m)$ to estimate the weights of the domain experts. In addition, to conveniently train the source domain discriminators $\mathscr{C}_m$, we share their parameters, hence having an unique $\mathscr{C}$ that receives a pair (x,y) and predicts its source domain t. Therefore, in practice, we obtain the expert teacher in (15) as $h^S(x,y) = \sum_{k=1}^{K} \mathscr{C}(x,y,k)\, h_k^S(x,y)$.

4.5.3 Target-domain Imitating Student

Inspired by the statement (ii) in Theorem 5, recall that f^T is the ground-truth labeling function and h^T is the classifier on the target domain, we propose to learn h^T on this domain to further minimize with the aim to obtain $h^T = f^T$:

$$\min_{h^T} \mathscr{W}_d\left(\mathbb{P}_{T,h^T}, \mathbb{P}_{T,f^T}\right).$$

To proceed our theory, we assume that $d_{\mathscr{Y}}$ is a metric over $\mathscr{Y}_\Delta$, which together with the metric $d_{\mathscr{X}}$ forms the metric d (cf. (8)), implying that $\mathscr{W}_d(\mathbb{P}_{.,.}, \mathbb{P}_{.,.})$ is a proper metric. We can thus bound the

quantity of interest $\mathscr{W}_d\left(\mathbb{P}_{T,h^T},\mathbb{P}_{T,f^T}\right)$:

$$\mathscr{W}_d\left(\mathbb{P}_{T,h^T},\mathbb{P}_{T,f^T}\right) \leq \mathscr{W}_d\left(\mathbb{P}_{T,h^T},\mathbb{P}^{\pi}_{S,h^S}\right)$$
$$+\mathscr{W}_d\left(\mathbb{P}^{\pi}_{S,h^S},\mathbb{P}^{\pi}_{S,f^S}\right)+\mathscr{W}_d\left(\mathbb{P}^{\pi}_{S,f^S},\mathbb{P}_{T,f^T}\right) \overset{(1)}{\leq}$$
$$\mathscr{W}_d\left(\mathbb{P}_{T,h^T},\mathbb{P}^{\pi}_{S,h^S}\right)+\mathscr{L}\left(h^S,f^S,\mathbb{P}^S_{\pi}\right)+\mathscr{W}_d\left(\mathbb{P}^{\pi}_{S,f^S},\mathbb{P}_{T,f^T}\right), \tag{10}$$

where $\mathbb{P}^{\pi}_{S,f^S}$, a joint distribution over $\mathscr{X}^S\times\mathscr{Y}_\Delta$, consists of pairs (x,y_Δ) in which $x\sim\mathbb{P}^S_\pi$ and $y_\Delta=f^S(x)$, h^S is a classifier on the mixture of source domains (i.e., $\mathbb{P}^S_\pi$), and the definition of $\mathbb{P}^{\pi}_{S,h^S}$ is similar to $\mathbb{P}^{\pi}_{S,f^S}$ by changing the role of f^S to h^S. Note that we achieve the inequality (1) because $\mathscr{W}_d\left(\mathbb{P}^{\pi}_{S,h^S},\mathbb{P}^{\pi}_{S,f^S}\right)$ is upper-bounded by $\mathscr{L}\left(h^S,f^S,\mathbb{P}^S_\pi\right)$ (thanks to the statement (iii) in Theorem 5).

Moreover, $\mathscr{W}_d\left(\mathbb{P}^{\pi}_{S,f^S},\mathbb{P}_{T,f^T}\right)$ is a constant. Hence, to minimize the upper-bound in (10), we seek a classifier h^S working well on the mixture of source domains with a sufficiently small $\mathscr{L}\left(h^S,f^S,\mathbb{P}^S_\pi\right)$, while encouraging h^T to imitate h^S by minimizing $\mathscr{W}_d\left(\mathbb{P}_{T,h^T},\mathbb{P}^{\pi}_{S,h^S}\right)$. To this end, we employ the multi-source expert teacher h^S as in Section 4.5.2, which can operate well on $\mathbb{P}^S_\pi$ as long as we can train good domain experts $h^S_{1:K}$, hence leading to the following optimization problem:

$$\min_{h^T}\left\{\mathscr{W}_d\left(\mathbb{P}_{T,h^T},\mathbb{P}^{\pi}_{S,h^S}\right)+\mathscr{L}\left(h^S,f^S,\mathbb{P}^S_\pi\right)\right\}. \tag{11}$$

The optimization problem in (11) is in line with the context of imitation learning for which the teacher classifier h^S has been trained effectively on the mixture source domain (i.e., $\mathbb{P}^S_\pi$) and the student classifier h^T tries to imitate the teacher on the target domain. Specifically, Proposition 4 implies finding the optimal transport map H^*: $H^*_\#\mathbb{P}^T=\mathbb{P}^S_\pi$ so that for any $x\sim\mathbb{P}^T$, $h^T(x)$ should mimic the prediction of the expert teacher h^S over $H^*(x)\sim\mathbb{P}^S_\pi$. This observation forms the foundation of our proposed MOST.

Proposition 4 further illustrates that among the transport maps H transporting $\mathbb{P}^T$ to $\mathbb{P}^S_\pi$, we need to seek the map incurring the minimal label shift and enabling the student h^T easiest to imitate its teacher h^S. Inspired by the statement (iv) in Theorem 5 where $\mathscr{W}_d\left(\mathbb{P}^{\pi}_{S,h^S},\mathbb{P}_{T,h^T}\right)$ is lower-bounded by $\lambda\mathscr{W}_{d_\mathscr{X}}\left(\mathbb{P}^S_\pi,\mathbb{P}^T\right)$ (the discrepancy gap between the mixture of source distributions and the target one), to reduce the data shift, we propose to map both $\left(\mathscr{X}^S,\mathbb{P}^S_\pi\right)$ and $\left(\mathscr{X}^T,\mathbb{P}^T\right)$ to a common joint space via two generators G^S and G^T and solve the following optimization problem:

$$\min_{h^T,G^T}\left\{\mathscr{L}\left(h^S\circ G^S,f^S,\mathbb{P}^S_\pi\right)+\mathscr{W}_d\left(\mathbb{Q}_{T,h^T},\mathbb{Q}^{\pi}_{S,h^S}\right)\right\}, \tag{12}$$

where $\mathbb{Q}_{T,h^T}$ is similar to $\mathbb{P}_{T,h^T}$ but on the joint space and consists of the pairs $\left(G^T(x),h^T\left(G^T(x)\right)\right)$ for $x\sim\mathbb{P}^T$ and $\mathbb{Q}^{\pi}_{S,h^S}$ is similar to $\mathbb{P}^{\pi}_{S,h^S}$ but on the joint space and consists of the pairs $\left(G^S(x),h^S\left(G^S(x)\right)\right)$ for $x\sim\mathbb{P}^S_\pi$. Note that both h^S and $h^S_{1:K}$ now act on $G^S(\cdot)$.

Theorem 8. *Let $h^S_*\circ G^S_*$ be the optimal teacher and h^T_*,G^T_* be the optimal solutions of the optimization problem in (12). Assume that G^T,h^T are in the families having infinite capacity (i.e., those can approximate any continuous function up to any level of precision, e.g., neural nets), we have*[3]

$$\min_{h^T,G^T}\mathscr{W}_d\left(\mathbb{P}^{G^T}_{T,h^T},\mathbb{P}^{G^T}_{T,f^{G^T}_T}\right)\leq\mathscr{L}\left(h^S_*\circ G^S_*,f^S,\mathbb{P}^S_\pi\right)+\mathscr{W}_d\left(\mathbb{P}^{G^S_*}_{S,f^S_*},\mathbb{P}^{G^T_*}_{T,f^T_*}\right), \tag{13}$$

where $f^S_*:=f_S^{G^S_*}$ and $f^T_*:=f_T^{G^T_*}$.

In Theorem 8, $\mathbb{P}^{G^T}_{T,h^T}$ is the distribution consisting of samples of pairs $\left(G^T(x),h^T\left(G^T(x)\right)\right)$ where $x\sim\mathbb{P}^T$and same definition for other similar distributions. Theorem 8 demonstrates that our MOST with the support of the generators and the joint space can mitigate data and label shifts as $\mathscr{W}_d\left(\mathbb{P}^{G^S_*}_{S,f^S_*},\mathbb{P}^{G^T_*}_{T,f^T_*}\right)$ is the natural shift between two ground-truth labeling functions f^S and f^T in the joint space.

[3]We define f^G as the induced labeling function over the joint space such that f^G predicts $G(x)$ as same as f predicts x.

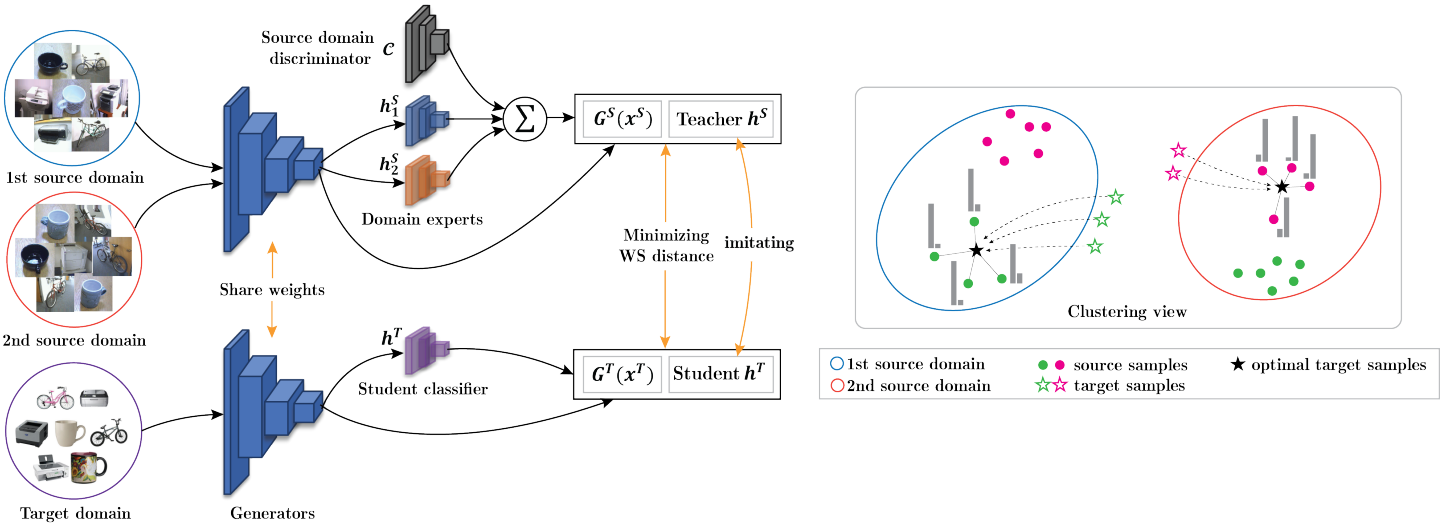


Figure 7: Left: The overall structure of our proposed method for multi-source domain adaptation. MOST consists of two cooperative agents: an expert teacher h^S, a weighted combination of domain experts and a student h^T that tries to imitate the prediction of the teacher via the OT-based imitation learning. Right: Clustering view explanation of the WS distance term.

4.5.4 Training Process of MOST

Training Multi-Source Expert Teacher

To work out the multi-source expert teacher h^S, we simultaneously train domain experts $h^S_{1:K}$ on the labeled training sets $\mathscr{D}^S_{1:K}$ and the source domain discriminator $\mathscr{C}$ to offer the weights of the domain experts. Basically, we minimize: $\sum_{k=1}^K \mathscr{L}_k^{de} + \mathscr{L}^{\mathscr{C}}$, where we define

$$\mathscr{L}_k^{de} = \mathbb{E}_{(x,y)\sim\mathscr{D}_k^s}\left[CE\left(h_k^S\left(G^S\left(x\right)\right),y\right)\right],$$
$$\mathscr{L}^{\mathscr{C}} = \mathbb{E}_{(x,y,t)\sim\mathscr{D}}\left[CE\left(\mathscr{C}\left(x,y\right),t\right)\right]$$

with $\mathscr{D}$ is formed by sampling $t\sim Cat\left(\pi\right)$ and $\left(x,y\right)\sim\mathscr{D}_t^S$ and $CE\left(\cdot,\cdot\right)$ is the cross-entropy loss.

Training Target-Domain Imitating Student

We use the *entropic regularized dual form* in (7) to solve the optimization problem of interest in (12) by minimizing $\mathscr{W}_d^{\varepsilon}\left(\mathbb{Q}_{T,h^T},\mathbb{Q}^{\pi}_{S,h^S}\right)$, hence arriving at the following optimization problem:

$$\min_{h^T,G^T}\mathscr{L}^{WS} = \min_{h^T,G^T}\max_{\phi}\left\{\mathbb{E}_{\mathbb{P}^T}\left[-\varepsilon\log\left(\mathbb{E}_{\mathbb{P}^S_{\pi}}\left[\exp\left\{\frac{1}{\varepsilon}\gamma\left(x^S,x^T\right)\right\}\right]\right)\right]+\mathbb{E}_{\mathbb{P}^S_{\pi}}\left[\phi\left(G^S\left(x^S\right)\right)\right]\right\}$$

, where we have defined

$$\gamma\left(x^S,x^T\right)=\phi\left(G^S\left(x^S\right)\right)-d\left(G^S\left(x^S\right),G^T\left(x^T\right)\right)$$

, ϕ is a neural net named Kantorovich potential network and we have definened

$$d\left(G^S\left(x^S\right),G^T\left(x^T\right)\right)=d_{\mathscr{Y}}\left(h^T\left(G^T\left(x^T\right)\right),h^S\left(G^S\left(x^S\right)\right)\right)+\lambda\left\|G^T\left(x^T\right)-G^S\left(x^S\right)\right\|$$

, while $x^T\sim\mathbb{P}^T, x^S\sim\mathbb{P}^S_{\pi}$.

Clustering view explanation of the WS distance term. More specifically, according to the cluster view of optimal transport $\mathscr{W}_d^{\varepsilon}\left(\mathbb{Q}_{T,h^T},\mathbb{Q}^{\pi}_{S,h^S}\right)$, at the optimal solution, each $G^T\left(x^T\right)$ finds a cluster of $G^S\left(x^S\right)$ (s) to minimize the distortion w.r.t. the metric $d\left(G^T\left(x^T\right),G^S\left(x^S\right)\right)$ defined as

$$d_{\mathscr{Y}}\left(h^T\left(G^T\left(x^T\right)\right),h^S\left(G^S\left(x^S\right)\right)\right)+\lambda\left\|G^T\left(x^T\right)-G^S\left(x^S\right)\right\|$$

, which further implies that $G^T\left(x^T\right)$ should move closely to a cluster of $G^S\left(x^S\right)$(s) with the same predicted label regarding h^S so as to imitate the prediction of h^S (i.e., $\min d_{\mathscr{Y}}\left(h^T\left(G^T\left(x^T\right)\right),h^S\left(G^S\left(x^S\right)\right)\right)$). This certainly helps to mitigate the label shift problem (see Figure 7).

The teacher h^S also offers pseudo labels on source and target examples for the student h^T to imitate, hence we minimize:

$$\mathscr{L}^{pl} = \mathbb{E}_{x\sim\mathbb{P}^S_\pi,\mathbb{P}^T}\left[CE\left(h^S\left(G^S\left(x\right)\right),h^T\left(G^T\left(x\right)\right)\right)\right].$$

Virtual adversarial training (VAT) [60] in conjunction with minimizing entropy of prediction [29] with the aim to ensuring the clustering assumption [6] has been applied successfully to UDA [87]. Inspired by this success, we propose to minimize:

$$\mathscr{L}^{clus} = \mathscr{L}^{ent} + \mathscr{L}^{vat},$$

where we have defined

$$\mathscr{L}^{ent} = \mathbb{E}_{\mathbb{P}^T}\left[\mathbb{H}\left(h^T\left(G^T\left(x\right)\right)\right)\right]$$

, $\mathbb{H}$ is the entropy and

$$\mathscr{L}^{vat} = \mathbb{E}_{x\sim\mathbb{P}^T}\left[\max_{x':\|x'-x\|<\theta} D_{KL}\left(h^T\left(G^T\left(x\right)\right),h^T\left(G^T\left(x'\right)\right)\right)\right]$$

with which D_{KL} represents a Kullback-Leibler divergence and θ is very small positive number.

Simultaneous Training of Student and Teacher

We have two scenarios to train our teacher and student paradigm: (i) sequential training and (ii) simultaneous training of teacher and student. As suggested by the ablation study (see Section 4.6.4), we follow the strategy of simultaneous training of teacher and student in which we minimize:

$$\mathscr{L} = \sum_{k=1}^{K}\mathscr{L}_k^{de} + \mathscr{L}^{\mathscr{C}} + \alpha\mathscr{L}^{WS} + \beta\mathscr{L}^{pl} + \gamma\mathscr{L}^{clus}, \tag{14}$$

where $\alpha,\beta,\gamma > 0$ are trade-off parameters.

We note that the loss $\mathscr{L}^{WS}$ has the form of maximizing over ϕ which is parameterized by a neural net. In training MOST, we update ϕ several times for each mini-batch of data. Due to the effect of the envelope theorem, the term $\mathscr{L}^{WS}$ (hence the total loss $\mathscr{L}$) smoothly decreases (see Figure 8). Finally, we present the overview of our approach in Figure 7.

4.6 Experiments

4.6.1 Model Evaluation

We evaluate our proposed MOST on several commonly-used benchmark domain adaptation datasets, including *Digits-five* [72], *Office-Caltech10* [25] and *Office-31* [81]. The details of datasets and preprocessing steps are described in our *supplementary material*. Similar to [94], we compare our MOST with the MSDA standards: (1) *Single Best*: the best classification accuracy on the test set among single-source transfer results; (2) *Source Combine*: the evaluation on single-source domain adaptation whereas the single-source is combined by all source data; (3) *Multi-Source*: results on the adaptation from multiple source domains to the target domain.

4.6.2 Architecture/Hyperparameters

We follow the training paradigms in [82, 72] where G^S are shared weights with G^T. All the experiments on *Office-Caltech10* and *Office-31* are based on pre-trained ResNet-101 [31] and AlexNet [44], respectively. The network architecture and hyperparameter settings are presented in the *supplementary material*.

4.6.3 Results

We first compare MOST with recent state-of-the-art works on *Digits-five* whose results are reported in Table 3. Our MOST surpasses all transfer tasks, with a sizable margin especially in the following adaptation tasks: "→**mm**", "→**sv**", and "→**sy**". Overall, our proposed method achieves a high average accuracy of 96.0%, which is a 4.2% increase compared to LtC-MSDA [94].

Table 3: Classification results with mean and standard deviation on Digits-five.

Standards	Methods	→**mm**	→**mt**	→**up**	→**sv**	→**sy**	Avg
Single Best	Source-only	59.2±0.6	97.2±0.6	84.7±0.8	77.7±0.8	85.2±0.6	80.8
	DAN [52]	63.8±0.7	96.3±0.5	94.2±0.9	62.5±0.7	85.4±0.8	80.4
	CORAL [88]	62.5±0.7	97.2±0.8	93.5±0.8	64.4±0.7	82.8±0.7	80.1
	DANN [21]	71.3±0.6	97.6±0.8	92.3±0.9	63.5±0.8	85.4±0.8	82.0
	ADDA [90]	71.6±0.5	97.9±0.8	92.8±0.7	75.5±0.5	86.5±0.6	84.8
Source Combine	Source-only	63.4±0.7	90.5±0.8	88.7±0.9	63.5±0.9	82.4±0.6	77.7
	DAN [52]	67.9±0.8	97.5±0.6	93.5±0.8	67.8±0.6	86.9±0.5	82.7
	DANN [21]	70.8±0.8	97.9±0.7	93.5±0.8	68.5±0.5	87.4±0.9	83.6
	JAN [55]	65.9±0.7	97.2±0.7	95.4±0.8	75.3±0.7	86.6±0.6	84.1
	ADDA [90]	72.3±0.7	97.9±0.6	93.1±0.8	75.0±0.8	86.7±0.6	85.0
	MCD [82]	72.5±0.7	96.2±0.8	95.3±0.7	78.9±0.8	87.5±0.7	86.1
Multi-Source	MDAN [107]	69.5±0.3	98.0±0.9	92.4±0.7	69.2±0.6	87.4±0.5	83.3
	DCTN [102]	70.5±1.2	96.2±0.8	92.8±0.3	77.6±0.4	86.8±0.8	84.8
	M^{3}SDA [72]	72.8±1.1	98.4±0.7	96.1±0.8	81.3±0.9	89.6±0.6	87.7
	MDDA [108]	78.6±0.6	98.8±0.4	93.9±0.5	79.3±0.8	89.7±0.7	88.1
	LtC-MSDA [94]	85.6±0.8	99.0±0.4	98.3±0.4	83.2±0.6	93.0±0.5	91.8
	MOST (ours)	**91.5**±1.7	**99.6**±0.0	**98.4**± 0.0	**90.9**±0.6	**96.4**±2.7	**95.4**

Table 4: Classification accuracy (%) on Office-Caltech10 using pretrained ResNet-101.

Standards	Methods	→**W**	→**D**	→**C**	→**A**	Avg
Source Combine	Source-only	99.0	98.3	87.8	86.1	92.8
	DAN [52]	99.3	98.2	89.7	94.8	95.5
Multi-Source	Source-only	99.1	98.2	85.4	88.7	92.9
	DAN [52]	99.5	99.1	89.2	91.6	94.8
	DCTN [102]	99.4	99.0	90.2	92.7	95.3
	JAN [55]	99.4	99.4	91.2	91.8	95.5
	MEDA [95]	99.3	99.2	91.4	92.9	95.7
	MCD [82]	99.5	99.1	91.5	92.1	95.6
	M^{3}SDA [72]	99.5	99.2	92.2	94.5	96.4
	MOST (ours)	**100**	**100**	**96.0**	**96.4**	**98.1**

The experimental results on *Office-Caltech10* are shown in Table 9. Compared to the baselines, our MOST obtains impressive scores on all the settings: 100% on the adaptation tasks from corresponding source domains to **W** and **D**, and significant improvements on "→**C**", and "→**A**" tasks. As a result, our proposed method experiences a rise of 1.7% on average compared to the runner-up method M^{3}SDA [72]. Finally, we report the performance on *Office-31* and compare results in Table 5. MOST continues to perform the best with 1.8% improvement on average over the second. Additionally, on the challenging task "→**A**", MOST significantly surpasses the state-of-the-art method by 3.7%.

4.6.4 Ablation Study

Training Strategy

We consider two training strategies for MOST, which are two-phase training and simultaneous training. In the former, we train a perfect teacher and then train a student to imitate it, while in the latter, we train all in once with the loss in (18). Table 6 shows that simultaneous training is more effective with an improvement of 0.6% on the average accuracy. We hence stick to this strategy for our main experiments.

Table 5: Classification accuracy (%) on Office-31 using pretrained AlexNet.

Standards	Methods	→**D**	→**W**	→**A**	Avg
Single Best	Source-only	99.0	95.3	50.2	81.5
	RevGrad [20]	99.2	96.4	53.4	83.0
	DAN [52]	99.0	96.0	54.0	83.0
	RTN [54]	99.6	96.8	51.0	82.5
	ADDA [90]	99.4	95.3	54.6	83.1
Source Combine	Source-only	97.1	92.0	51.6	80.2
	DAN [52]	98.8	96.2	54.9	83.3
	RTN [54]	99.2	95.8	53.4	82.8
	JAN [55]	99.4	95.9	54.6	83.3
	ADDA [90]	99.2	96.0	55.9	83.7
	MCD [82]	99.5	96.2	54.4	83.4
Multi-Source	MDAN [107]	99.2	95.4	55.2	83.3
	DCTN [102]	99.6	96.9	54.9	83.8
	M^3SDA [72]	99.4	96.2	55.4	83.7
	MDDA [108]	99.2	97.1	56.2	84.2
	LtC-MSDA [94]	99.6	97.2	56.9	84.6
	MOST (ours)	**100**	**98.7**	**60.6**	**86.4**

Table 6: Results (%) on different training strategies.

Methods	→**mm**	→**mt**	→**up**	→**sv**	→**sy**	Avg
Two-phase training	89.7	**99.6**	98.2	**92.0**	97.7	95.4
Simultaneous training	**93.4**	**99.6**	**98.4**	90.9	**97.8**	**96.0**

Effect of Losses

We investigate the effectiveness of the component losses in (18) w.r.t. the source domain (a.k.a. *source only* setting), i.e., $\mathscr{L}_k^{de} + \mathscr{L}^{\mathscr{C}}$, and w.r.t. the target domain to perform domain adaptation, i.e., $\mathscr{L}^{pl}$, $\mathscr{L}^{WS}$, $\mathscr{L}^{ent}$ and $\mathscr{L}^{vat}$. The average results reported in Table 7 show that by only incorporating 2 losses $\mathscr{L}^{pl} + \mathscr{L}^{WS}$ (fourth row) to align the target samples to source samples, MOST already achieves the state-of-the-art results (94.2% on *Digits-five* and 96.9% on *Office-Caltech10*) compared to the runner-up baselines (91.8% on *Digits-five* and 96.4% on *Office-Caltech10*). While the performance is improved further with the help of clustering assumption (the last row).

Table 7: Average accuracies (%) on Digits-five and Office-Caltech10 datasets with different settings.

$\mathscr{L}_k^{de} + \mathscr{L}^{\mathscr{C}}$	$\mathscr{L}^{pl}$	$\mathscr{L}^{WS}$	$\mathscr{L}^{ent}$	$\mathscr{L}^{vat}$	Digits-five	Office-Caltech10
✓					82.6	95.8
✓	✓				89.9	96.7
✓	✓	✓			94.2	96.9
✓	✓	✓	✓		93.8	97.9
✓	✓	✓		✓	94.8	97.4
✓	✓	✓	✓	✓	**96.0**	**98.1**

Wasserstein Distance

We further observe the values of the WS distance between $\mathbb{Q}_{T,h^T}$ and $\mathbb{Q}^{\pi}_{S,h^S}$ in (8) on "→**mm**" task during training. As shown in Figure 8, the WS values tend to go down, which signals the decline of the data shift and label shift between the two domains.

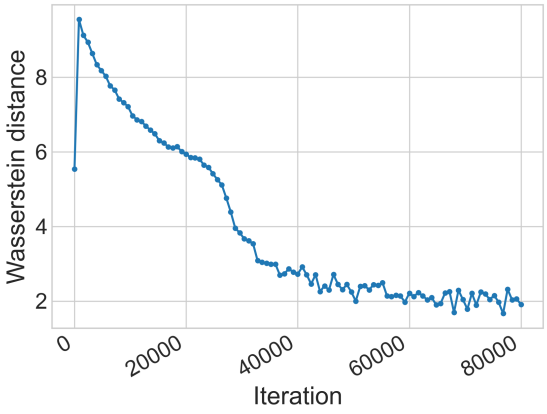


Figure 8: Values of $\mathscr{W}_d^{\varepsilon}\left(\mathbb{Q}_{T,h^T}, \mathbb{Q}^{\pi}_{S,h^S}\right)$ during training.

5 STEM: An approach to Multi-source Domain Adaptation with Guarantees

5.1 Introduction

Recent advances in deep learning have enjoyed great success in performing visual learning tasks under the collection of massive annotated data [44, 99, 78, 86, 5]. However, directly transferring knowledge of a learned model, which is trained on a source domain, to a novel target domain can undesirably degrade its performance due to the existence of *domain and label shifts* [75]. To address these issues, a diverse range of approaches in domain adaptation (DA) has been proposed from shallow domain adaptation [70, 24, 9, 11] to deep domain adaptation [20, 52, 18, 19, 87, 15, 47, 65, 64]. While the conventional DA aims to transfer knowledge from a labeled source domain to an unlabeled target domain, in many real-world contexts, labeled data are collected from multiple domains, for example, images taken under different conditions (e.g., weather, poses, lighting conditions, distinct backgrounds, and etc) [107]. This has arisen a very practical and useful setting for transfer learning named multi-source domain adaptation (MSDA) in which we need to transfer knowledge from multiple distinct source domains to a single unlabeled target domain.

For multi-source domain adaptation, there exist two fundamental challenges: (i) how to deal with the diversity in the labeled source domains and (ii) how to cope with the domain shift between the target domain and the source domains. The first challenge makes it harder to train a single model that is expected to work well on multiple source domains due to the requirement to resolve diverge data complexity imposed on model training. To overcome this challenge, inspired by [56, 39], we propose combining domain experts into a multi-source teacher by mixing the domain expert predictions using the coefficients learned by a domain discriminator. Our rigorous theory demonstrates that the performance of this multi-source teacher expert predicting globally on the mixture source domains is at least better than that of the worst domain expert predicting locally on its domain (see Theorem 9). Therefore, if we can train qualified domain experts, their combination leads to another qualified expert with significantly broader coverage.

To address the second challenge, as suggested by Theorem 11, we employ a joint feature extractor that maps the target domain and the mixture of source domains into the same latent space with the help of adversarial learning. Furthermore, together with closing the divergence of the target domain and mixture of source domains on the latent space, we train a target-domain student to imitate the multi-source teacher on both source and target examples while enforcing the clustering assumption [6] on the target-domain student to strengthen the student's generalization ability.

- We propose an approach named ***S**tudent-**T**eacher **E**nsemble **M**ulti-source Domain Adaptation* (STEM) with theoretical guarantees for multi-source domain adaptation. Not only driving us in devising our

STEM, the rigorous theory developed provides us an insightful understanding of how each model component really influences the transferring performance.

- We conduct extensive experiments on three benchmark datasets including Digits-five, Office-Caltech10, and DomainNet. Experimental results show that our STEM achieves state-of-the-art performances on those three benchmark datasets. More specifically, for Digits-five and Office-Caltech10 datasets, our STEM wins the baselines on all pairs and surpasses the runner-up baselines by 3.2% and 1.5% on average, while for DomainNet dataset, our STEM wins the runner-up baseline on 5 out of 6 pairs and surpasses the runner-up baseline by 6.0% on average.

5.2 Related Work

5.2.1 Unsupervised Domain Adaptation

A variety of unsupervised domain adaptation (UDA) approaches have been successfully applied to generalize a model learned from labeled source domain to unlabeled novel target domain. Several existing methods based on discrepancy-based alignment to minimize a different discrepancy metric to close the gap between source and target domain [52, 91, 88, 104, 50]. Another branch of UDA has leveraged adversarial learning wherein generative adversarial networks [28, 66, 37, 13, 46] were employed to align source and target domain on feature-level [20, 90, 53, 67] or pixel-level [23, 4, 84, 101]. On the category-level, some approaches utilized dual classifier [82, 50], or domain prototype [97, 71, 100] to investigate the category relations across domains.

5.2.2 Multi-Source Domain Adaptation

The aforementioned UDA methods mainly consider single-source domain adaptation, which is less practical than multi-source domain adaptation. The fundamental study in [12, 56, 3] has shed light upon the wide applications of MSDA, such as in [17, 102]. Based on the above works, Hoffman et al. [39] gave strong theoretical guarantees for cross-entropy and other similar losses, which is a normalized solution for MSDA problems. Recently, Zhao et al. [107] deployed domain adversarial networks to align the target domain to source domains. Xu et al. [102] proposed a new model to deal with the *category shift*, which is the case where sources may not completely share their categories. Peng et al. [72] introduced a model that aligned moments of source and target feature distributions in latent space. A multi-source distilling model was proposed in [108] to fine-tune generator and classifier separately and utilized domain weight to aggregate target prediction. Finally, the work in [94] deployed a graph convolutional network to conduct domain alignment on the category-level.

5.3 Our Proposed Framework

5.3.1 Problem Setting

In this paper, we address the problem of multi-source domain adaptation in which we have K source domains with collected data and labels, and a single target domain with only collected data. We wish to transfer a model learned on labeled source domains to an unlabeled target domain. Let us denote the collected data and labels for the source domains by $\mathbb{D}_k^S = \left\{ \left(sx_i^k, y_i^k \right) \right\}_{i=1}^{N_k^S}$ where k is the index of a source domain and label $y_i^k \in \{1, 2, ..., M\}$ with the number of classes M, and collected data without labels for the target domain $\mathbb{D}^T = \{tx_i\}_{i=1}^{N^T}$. We further equip source domains with data distributions $\mathbb{P}_{1:K}^S$ whose density functions are $p_{1:K}^S(x)$. Also, we define $p_{1:K}^S(y \mid x)$ as the conditional distributions that assign labels to each data example x for the source domains. Regarding the target domain, we define its data space as $\mathscr{X}^T$, data distribution and density function as $\mathbb{P}^T$ and $p^T(x)$, respectively. We further define the conditional distribution that assigns labels for the target domain as $p^T(y \mid x)$.

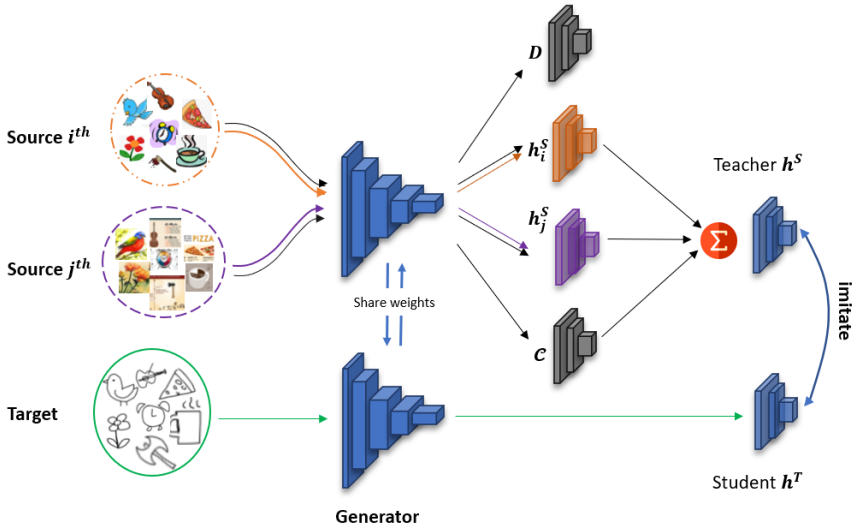


Figure 9: Overall framework of STEM for multi-source domain adaptation, which consists of cooperative agents, namely a multi-source teacher expert h^S and a target-domain student h^T. Our model is trained to implement simultaneously two tasks: (i) achieving the teacher expert h^S by first training to obtain domain experts $h^S_{1:K}$ using their labels (orange and purple arrows), and then output the teacher h^S using a weighted ensembling strategy (black arrows) and (ii) training the student h^T with the aim to mimic the prediction of its teacher expert h^S (green arrows) with the support of D to close the gap between the mixture of source data distributions and the target distribution on the latent space.

Furthermore, we denote $\mathbb{D}$ as a joint distribution with density function $p(x,y)$ used to generate data-label pairs (i.e., $(x,y) \sim \mathbb{D}$). Note that for the sake of notion simplification, we overload the notion $\mathbb{D}$ to denote both joint distribution for generating data-label pairs and a training set sampled from this distribution. Let h be a classifier in which $h(x,y)$ specifies the probability to assign the data example x to a class $y \in \{1,...,M\}$ and $h(x) = [h(x,y)]_{y=1}^{M}$ is the prediction probability vector w.r.t. x. We consider the loss function $\ell(h(x),y)$ and define the general loss w.r.t. the data-label joint distribution $\mathbb{D}$ as follows:

$$\begin{aligned}\mathscr{L}(h,\mathbb{D}) &:= \mathbb{E}_{(x,y)\sim\mathbb{D}}\left[\ell(h(x),y)\right] \\ &= \int \ell(h(x),y)\, p(x,y)\, dxdy.\end{aligned}$$

Finally, given a discrete distribution π over $\{1,...,K\}$, we define $\mathbb{P}^S_\pi := \sum_{k=1}^K \pi_k \mathbb{P}^S_k$ which is a mixture of $\mathbb{P}^S_{1:K}$ with density function $p^S_\pi(x) = \sum_{k=1}^K \pi_k p^S_k(x)$ and $\mathbb{D}^S_\pi := \sum_{k=1}^K \pi_k \mathbb{D}^S_k$ with density function $p^S_\pi(x,y) = \sum_{k=1}^K \pi_k p^S_k(x,y)$. Moreover, the mixing proportion π can be the uniform distribution $[\frac{1}{K},...,\frac{1}{K}]$ or proportional to the number of training examples in the source domains (i.e., $N^S_{1:K}$).

5.3.2 Overall Framework of STEM

Figure 9 illustrates the overall framework of our STEM. Source and target domains are mapped to a latent space via a shared generator or feature extractor G. On the latent space, we train the domain experts $h^S_{1:K}$ and a source domain discriminator $\mathscr{C}$ for which we can combine them to achieve a multi-source teacher expert h^S. Particularly, the source domain discriminator is trained to distinguish the source domains, hence rendering the probabilities to assign an example to the source domains. Therefore, given a source example, the domain experts more relevant to this example contribute more to the final decision. Furthermore, we develop a theory to demonstrate that the multi-source teacher expert h^S can predict well on the mixture of source domains with the performance at least better than the worst domain expert on its source domain. Note that to support the source domain discriminator $\mathscr{C}$ to do its task, the latent representations from the individual source domains are encouraged to be separate, hence increasing their coverage on the latent space. Meanwhile, with the assistance of adversarial learning framework [28], we train G with the support of a discriminator D to bridge the gap between the target distribution and the mixture of source distributions, which enables the multi-source teacher expert h^S to transfer its knowledge to predict well the target examples. Moreover, inspired by the principle of knowledge

distillation [33] in which we can conduct a student to distill knowledge and outperform its teacher, we train an additional target-domain student h^T to mimic the predictions of the multi-source teacher expert h^S on the target and source examples. Finally, we develop a rigorous theory to quantify the loss in performance for this imitating.

5.3.3 Ensemble based Teacher Expert

In what follows, we present how to conduct the multi-source teacher expert h^S, an ensemble expert which leverages knowledge of domain experts. Particularly, using the labeled source training sets $\mathbb{D}^S_{1:K}$, we can train qualified domain expert classifiers $h^S_{1:K}$ with good generalization capacity (i.e., $\mathscr{L}\left(h^S_k, \mathbb{D}^S_k\right) \leq \varepsilon$ for some small $\varepsilon > 0$). The next arising question is how to combine those domain experts to achieve a multi-source teacher expert h^S that can work well on $\mathbb{D}^S_\pi$ (i.e., $\mathscr{L}\left(h^S, \mathbb{D}^S_\pi\right) \leq \varepsilon$). Inspired by [56, 39], we leverage the domain experts to achieve a more powerful multi-source teacher expert by a weighted ensembling as follows:

$$h^S(x,y) = \sum_{k=1}^{K} \frac{\pi_k p^S_k(x,y)}{\sum_{j=1}^{K} \pi_j p^S_j(x,y)} h^S_k(x,y), \tag{15}$$

where $y \in \{1,2,...,M\}$, and $h^S_k(x,y)$ and $h^S(x,y)$ specify the y-th values of $h^S_k(x)$ and $h^S(x)$ respectively.

The following theorem shows that the multi-source domain teacher expert h^S can work well on the mixture joint distribution $\mathbb{D}^S_\pi$. More specifically, it works better than the worst domain expert on its source domain, hence if each domain expert is an ε-qualified classifier (i.e., $\mathscr{L}\left(h^S_k, \mathbb{D}^S_k\right) \leq \varepsilon$), the multi-source teacher expert h^S is also an ε-qualified classifier (i.e., $\mathscr{L}\left(h^S, \mathbb{D}^S_\pi\right) \leq \varepsilon$).

Theorem 9. *If ℓ is a convex function, the following statements hold true (the proof of this theorem is adapted from a proof in [56, 39]):*

i) $\mathscr{L}\left(h^S, \mathbb{D}^S_\pi\right) \leq \max_{1\leq k\leq K} \mathscr{L}\left(h^S_k, \mathbb{D}^S_k\right)$.

ii) If each domain expert is an ε-qualified classifier (i.e., $\mathscr{L}\left(h^S_k, \mathbb{D}^S_k\right) \leq \varepsilon$), the multi-source teacher expert h^S is also an ε-qualified classifier (i.e., $\mathscr{L}\left(h^S, \mathbb{D}^S_\pi\right) \leq \varepsilon$).

So far the question of how to weight the domain experts $h^S_{1:K}$ to form multi-source teacher expert h^S is still left unanswered. Moreover, [39] proposed using DC-programming (i.e., difference of convex) [16] for estimating weights. However, this approach seems to be overly complicated and there is not any convincing evidence of the effectiveness of this work for real-world datasets (i.e., the reported performance for the Office-31 dataset in the context of the standard multiple source setting without any transfer learning is only approximately 84.7%). In this paper, we propose a new approach to weight the domain experts, which is hinted from the following theoretical observation. Assume that we have K distributions $\mathbb{R}_{1:K}$ with density functions $r_{1:K}(z)$. We form a joint distribution $\mathbb{D}$ of a data instance z and label $t \in \{1,...,K\}$ by sampling an index $t \sim \text{Cat}(\pi)$ (i.e., the categorical distribution w.r.t. π), sampling $x \sim \mathbb{R}_t$, and collecting (z,t) as a sample from $\mathbb{D}$. With this equipment, we have the following proposition.

Proposition 10. *If we train a source domain discriminator $\mathscr{C}$ to classify samples from the joint distribution $\mathscr{D}$ using the cross-entropy loss (i.e., $CE(\cdot,\cdot)$), the optimal source domain discriminator $\mathscr{C}^*$ defined as*

$$\mathscr{C}^* = argmin_{\mathscr{C}} \mathbb{E}_{(z,t)\sim\mathscr{D}} \left[CE\left(\mathscr{C}(z), t\right)\right]$$

satisfies $\mathscr{C}^*(z) = \left[\frac{\pi_k r_k(z)}{\sum_j \pi_j r_j(z)}\right]_{k=1}^{K}$.

Proposition 10 suggests us a way to compute the weights of the domain experts in Eq. (15) in which for a given $y = m$, the distributions $p^S_{1:K}(x, y = m)$ play roles of $r_{1:K}(z)$ where $z = (x, y = m)$. More specifically, for each $m \in \{1,...,M\}$, we sample $t \sim Cat(\pi)$, then sample $(x, y = m)$ from $p^S_t(x, y = m)$, and train a source domain discriminator $\mathscr{C}_m(x, y = m)$ (i.e., only consider (x,y) in which x has label $y = m$) to distinguish the source domain of $(x, y = m)$. We finally use $\mathscr{C}_m(x, y = m)$ to estimate the

weights of the domain experts. In addition, to conveniently train the source domain discriminators $\mathscr{C}_m$, we share their parameters, hence having an unique $\mathscr{C}$ that receives a pair (x,y) and predicts its source domain t. Therefore, we obtain the expert teacher

$$h^S(x,y) = \sum_{k=1}^{K} \mathscr{C}(x,y,k)\, h_k^S(x,y). \tag{16}$$

To leverage the information of multiple source domains and encourage learning multiple-source domain-invariant representations for transfer learning in the sequel, we employ a feature extractor G to map multiple source domains and the target domain to a latent space. The domain experts $h_{1:K}^S$ and the source domain discriminator are trained on the latent space. The formula in Eq. (16) is rewritten as:

$$h^S(G(x),y) = \sum_{k=1}^{K} \mathscr{C}(G(x),y,k)\, h_k^S(G(x),y).$$

At the outset, we want to emphasize that our principle to learn representations is different from that in some recent works in MSDA, typically [72]. In [72], the moment distance was used to force the representations of multiple source domains to be identical in the latent space, while ours encourages the representations of the individual source domains to be separate so that the source domain discriminator $\mathscr{C}$ can distinguish them more effectively. By this way, we increase the coverage of the representations from the multiple source domains, which makes the representations from the target domain more conveniently to adapt the source representation in the transfer learning phase.

5.3.4 Performance of The Multi-source Teacher Expert on the Target Domain

We have possessed a qualified multi-source teacher expert h^S that expects to predict well data examples sampled from $\mathscr{D}_\pi^S$ (i.e., a mixture of $\mathscr{D}_{1:K}^S$) as indicated in Theorem 9. It is natural to ask the question of the factors that influence the performance of h^S when *predicting on the target joint distribution* $\mathbb{D}^T$. The following theorem answers this question.

Theorem 11. *If ℓ is a convex function and upper-bounded by a positive constant L, the general loss $\mathscr{L}\left(h^S,\mathbb{D}^T\right)$ is upper-bounded by:*

i) $A\left[\max_k \mathscr{L}\left(h_k^S,\mathbb{D}_k^S\right) + L\max_k \mathbb{E}_{\mathbb{P}_k^S}\left[\left\|\Delta p_k(y\mid x)\right\|_1\right]\right]^{\frac{\alpha-1}{\alpha}}$ *where* $A = \exp\left\{R^\alpha\left(\mathbb{P}^T\|\mathbb{P}_\pi^S\right)\right\}^{\frac{\alpha-1}{\alpha}} L^{\frac{1}{\alpha}}$ *in which* $R^\alpha\left(\mathbb{P}^T\|\mathbb{P}_\pi^S\right)$ *represents the Rényi divergence between those distributions and*

$$\Delta p_k(y\mid x) := \left[\left|p_k^S(y=m\mid x) - p^T(y=m\mid x)\right|\right]_{m=1}^M$$

represents the label shift between the labeling assignment mechanisms of an individual source domain and target domain.

ii) $A\left[\varepsilon + L\max_k \mathbb{E}_{\mathbb{P}_k^S}\left[\left\|\Delta p_k(y\mid x)\right\|_1\right]\right]^{\frac{\alpha-1}{\alpha}}$ *provided that* $\mathscr{L}\left(h_k^S,\mathbb{D}_k^S\right) \le \varepsilon, \forall k = 1,...,K.$

We now interpret Theorem 11 which lays foundation for us to devise our STEM in the sequel. The general loss of interest $\mathscr{L}\left(h^S,\mathbb{D}^T\right)$ is upper-bounded by the construction of three terms, each of which has a specific meaning.

(i) The *expert-loss* term $\max_k \mathscr{L}\left(h_k^S,\mathbb{D}_k^S\right)$ represents the worst general loss of the domain experts $h_{1:K}^S$. Minimizing this term implies training the domain experts to work well on their domains.

(ii) The *label-shift* term $\mathbb{E}_{\mathbb{P}_k^S}\left[\left\|\Delta p_k(y\mid x)\right\|_1\right]$ where $\Delta p_k(y\mid x) := \left[\left|p_k^S(y=m\mid x) - p^T(y=m\mid x)\right|\right]_{m=1}^M$ specifies the label shift indicating the divergence of the ground-truth target labeling function and the ground-truth source labeling function on a source domain. This term is constant and reflects the characteristics of collected data.

(iii) The *domain-shift* term $R^\alpha\left(\mathbb{P}^T\|\mathbb{P}_\pi^S\right)$ expresses the data shift between the mixture source distribution $\mathbb{P}_\pi^S$ and the target distribution $\mathbb{P}^T$.

The observation in (iii) hints us using adversarial learning framework [28] to bridge the gap between the representations of the multiple source domains and the target domain on the latent space using an additional discriminator D (see Section 5.3.6).

5.3.5 Target-Domain Student

The multi-source teacher expert h^S is guaranteed to work well on the mixture of source data distributions $\mathbb{P}^S_\pi$, while the generator G with the support of a discriminator D in adversarial learning framework [28] aims to close the discrepancy gap between the mixture of source data distributions $\mathbb{P}^S_\pi$ and the target distribution $\mathbb{P}^T$ on the latent space. Therefore, the multi-source teacher expert h^S is expected to work well on the target domain. However, the ill-posed problem of GAN (e.g., the mode collapsing problem) could occur during training, so that using directly h^S to predict target samples in latent space is not the best solution, which motivates us to design the student network h^T. Particularly, in Figure 10a, GAN works perfectly, hence both h^S and h^T work equally well. In another case, since GAN does not mix up well the class 1 and 2 of source and target domains (Figure 10b), h^S predicts well on the source domain but unwell on the target one. By enforcing the clustering assumption on h^T [6] (i.e., h^T preserves clusters and is encouraged to give the same prediction for source and target data on the same cluster), the possible ill-posed training of GAN is mitigated. Additionally, inspired by the principle of knowledge distillation [33] in which we can conduct a student to distill knowledge and outperform its teacher, we propose to train h^T which aims to mimic the predictions of the teacher h^S on the mixture source and target domains. This also helps to mitigate the negative impact from possible ill-posed training of GAN, while offering us an opportunity to apply regularization techniques such as VAT [60] and label smoothing [62] to h^T. We note that in our framework, it is hard to apply those regularization techniques directly to the teacher h^S, but it is convenient to apply to h^T. Indeed, we decide to apply VAT to h^T (see Section 5.3.6) and observe its superiority to the teacher in terms of predictive performance (see Section 5.6.4).

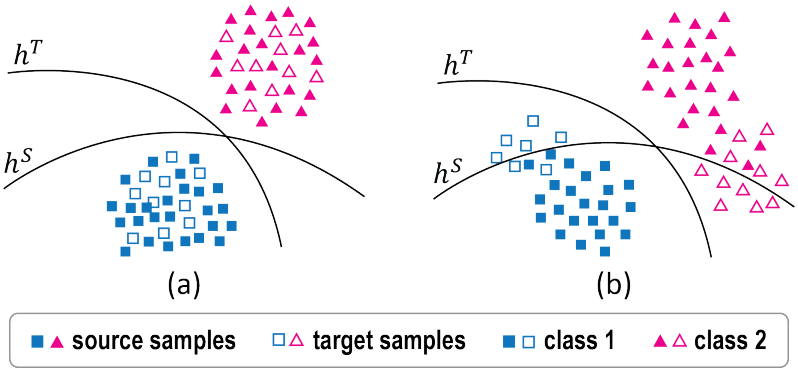


Figure 10: The motivation of the student h^T.

5.3.6 Training Procedure of Our STEM

Training Multi-Source Teacher Expert

To work out the multi-source teacher expert h^S, we simultaneously train domain experts $h^S_{1:K}$ on the labeled training sets $\mathscr{D}^S_{1:K}$ and the source domain discriminator $\mathscr{C}$ to offer the weights for leveraging the domain experts. We propose two workarounds to train $\mathscr{C}$ and ensemble the domain experts. Basically, we minimize: $\sum_{k=1}^K \mathscr{L}^{ie}_k + \alpha \mathscr{L}^{\mathscr{C}}$, where $\alpha > 0$ and consider two variants.

Theoretical oriented version. For the theoretical oriented version, we feed $(G(x), y)$ to source domain discriminator $\mathscr{C}$ with the aim to predict the data source index of x

$$\mathscr{L}^{ie}_k = \mathbb{E}_{(x,y)\sim\mathbb{D}^s_k}\left[CE\left(h^S_k(G(x)), y\right)\right],$$

$$\mathscr{L}^{\mathscr{C}} = \mathbb{E}_{(x,y,t)\sim\mathscr{D}}\left[CE\left(\mathscr{C}(G(x), y), t\right)\right],$$

$$h^S(G(x), y) = \sum_{k=1}^K \mathscr{C}(G(x), y, k)\, h^S_k(G(x), y),$$

where $\mathscr{D}$ is formed by sampling $t \sim Cat(\pi)$ and $(x,y) \sim \mathbb{D}^S_t$ and $CE(\cdot,\cdot)$ is the cross-entropy loss.

Simplified version. For the simplified version, instead of feeding $(G(x), y)$ to the source domain discriminator $\mathscr{C}$, we only feed $G(x)$ to this discriminator with the aim to predict the data source index of x

$$\mathscr{L}_k^{ie} = \mathbb{E}_{(x,y)\sim\mathbb{D}_k^s}\left[CE\left(h_k^S(G(x)), y\right)\right],$$

$$\mathscr{L}^{\mathscr{C}} = \mathbb{E}_{(x,t)\sim\mathscr{D}}\left[CE\left(\mathscr{C}(G(x)), t\right)\right],$$

$$h^S(G(x), y) = \sum_{k=1}^K \mathscr{C}(G(x), k)\, h_k^S(G(x), y),$$

where $\mathscr{D}$ is formed by sampling $t \sim Cat(\pi)$ and $x \sim \mathbb{P}_t^S$. According to our ablation study in Section 5.6.2, the simplified version performs slightly better than the theoretical oriented version, while easier to train due to its simplicity. Therefore, we stick with the simplified version and detail the training of other components based on this version.

Training Target-Domain Student

We train the target domain student h^T to mimic the teacher h^S on the predictions for target and mixture of source examples using the following loss:

$$\mathscr{L}^m = \mathbb{E}_{\mathbb{P}_\pi^S}\left[\ell\left(h^T(G(x)), h^S(G(x))\right)\right] + \mathbb{E}_{\mathbb{P}^T}\left[\ell\left(h^T(G(x)), h^S(G(x))\right)\right].$$

Moreover, Virtual adversarial training (VAT) [60] in conjunction with minimizing entropy of prediction [29] with the aim to ensuring the clustering assumption [6] has been applied successfully to UDA [87, 45, 68]. Inspired by this success, we propose minimizing

$$\mathscr{L}^{clus} = \mathscr{L}^{ent} + \mathscr{L}^{vat},$$

where $\mathbb{H}$ is the entropy,

$$\mathscr{L}^{ent} = \mathbb{E}_{\mathbb{P}^T}\left[\mathbb{H}\left(h^T(G(x))\right)\right],$$

$$\mathscr{L}^{vat} = \mathbb{E}_{x\sim\mathbb{P}^T}\left[\max_{x':\|x'-x\|<\theta} D_{KL}\left(h^T(G(x)), h^T(G(x'))\right)\right]$$

with which D_{KL} represents a Kullback-Leibler divergence and θ is very small positive number. The total loss to train the student h^T is as follows:

$$\mathscr{L}^{stu} = \mathscr{L}^m + \beta\mathscr{L}^{clus},$$

where $\beta > 0$ is a parameter.

Training Discriminator

The discriminator D is employed to distinguish the examples from the mixture of source data distributions $\mathbb{P}_\pi^S$ and the target distribution $\mathbb{P}^T$. The loss to train D is as follows:

$$\mathscr{L}^d = -\mathbb{E}_{\mathbb{P}_\pi^S}\left[\log D(G(x))\right] - \mathbb{E}_{\mathbb{P}^T}\left[\log\left(1 - D(G(x))\right)\right].$$

Training Generator

We train the generator G to bring the target examples to the mixture of source examples and provide appropriate representations for learning h^S and h^T with the following loss:

$$\sum_{k=1}^K \mathscr{L}_k^{ie} + \alpha\mathscr{L}^{\mathscr{C}} + \mathscr{L}^{stu} - \gamma\mathscr{L}^d, \tag{17}$$

where $\gamma > 0$ is are parameters.

Overall Training

We simultaneously update $G, \mathscr{C}, h^S, h^T$ by minimizing:

$$\sum_{k=1}^{K} \mathscr{L}_k^{ie} + \alpha \mathscr{L}^{\mathscr{C}} + \mathscr{L}^m + \beta \mathscr{L}^{clus} - \gamma \mathscr{L}^d. \tag{18}$$

We alternatively update D by minimizing $\mathscr{L}^d$. In addition, the pseudocode of our STEM is presented in Algorithm 1.

Algorithm 1 Pseudocode for training our STEM.

Input: Sources $\mathbb{D}_k^S = \left\{\left(sx_i^k, y_i^k\right)\right\}_{i=1}^{N_k^S}$, target $\mathbb{D}^T = \{tx_i\}_{i=1}^{N^T}$.
Output: Classifiers h^S, h^T, source discriminator $\mathscr{C}$, generator G.

```
for epoch in epochs do
    for iter in iter_per_epoch do
        Sample minibatches of sources {(sx_i^k, y_i^k)}_{i=1}^m and target {tx_i}_{i=1}^m.
        Update G, C, h^S, h^T according to (18).
        Update D by minimizing L^d.
    end for
end for
```

5.4 Experiments

5.5 Experiments on Benchmark Datasets

This section describes our experiment settings. We compare our STEM with the state-of-the-art baselines for MSDA on three benchmark datasets: Digits-five, Office-Caltech10, and DomainNet to demonstrate its merits.

5.5.1 Experimental setup

Implementation detail. In the experiments, we use Adam optimizer ($\beta_1 = 0.5, \beta_2 = 0.999$) [42] with Polyak averaging [74] for Digits-five and Office-Caltech10, and the learning rate is set to 2×10^{-4} and 10^{-4}, respectively. For DomainNet, we apply Stochastic Gradient Decent (SGD) [89] (learning rate = 5×10^{-2}, momentum $= 0.9$, decay rate = 5×10^{-4}) to optimize the model.

For STEM, the trade-off hyper-parameter α is fixed to 1.0 in all experiments, while the parameters (β, γ) (with a recommended range of $[10^{-4}, 1]$ for each parameter) are set to $(0.1, 0.1)$ for Digit-five, $(0.01, 0.1)$ for Office-Caltech10, and $(10^{-4}, 10^{-4})$ for DomainNet.

Performance comparison. Following the previous work [94], we conduct the experiments to evaluate the model performance with the MSDA standards: (1) *Single best*: the highest classification accuracy among single-source domain adaptation results; (2) *Source combine*: the result on single-source domain adaptation where the source domain is a combination of multiple domains; (3) *Multi-source*: the evaluation of the adaptation from multiple source domains to the target domain.

5.5.2 Experiment Results on Digits-five

Digits-five contains five common digit-datasets: MNIST [49], Synthetic Digits [21], MNISTM [21], SVHN [63], and USPS [40]. This is a benchmark dataset in MSDA, with ten classes corresponding to the digits ranging from 0 to 9 in each domain. In each experiment on Digits-five, one domain will be chosen as the target domain and the rest as the source domains.

In Table 8, we report the performance of our STEM compared with the baselines. Our STEM outperforms the baselines on all transfer tasks. As far as we know, LtC-MSDA [94] is the current state-of-the-art on Digits-five. Compared to this baseline, our STEM significantly surpasses some transfer

Standard	Methods	→**mm**	→**mt**	→**up**	→**sv**	→**sy**	Avg
Single Best	Source-only	59.2	97.2	84.7	77.7	85.2	80.8
	DAN [52]	63.8	96.3	94.2	62.5	85.4	80.4
	CORAL [88]	62.5	97.2	93.5	64.4	82.8	80.1
	DANN [21]	71.3	97.6	92.3	63.5	85.4	82.0
	ADDA [90]	71.6	97.9	92.8	75.5	86.5	84.8
Source Combine	Source-only	63.4	90.5	88.7	63.5	82.4	77.7
	DAN [52]	67.9	97.5	93.5	67.8	86.9	82.7
	DANN [21]	70.8	97.9	93.5	68.5	87.4	83.6
	JAN [55]	65.9	97.2	95.4	75.3	86.6	84.1
	ADDA [90]	72.3	97.9	93.1	75.0	86.7	85.0
	MCD [82]	72.5	96.2	95.3	78.9	87.5	86.1
Multi-Source	MDAN [107]	69.5	98.0	92.4	69.2	87.4	83.3
	DCTN [102]	70.5	96.2	92.8	77.6	86.8	84.8
	M^3SDA [72]	72.8	98.4	96.1	81.3	89.6	87.7
	MDDA [108]	78.6	98.8	93.9	79.3	89.7	88.1
	CMSS [106]	75.3	99.0	97.7	88.4	93.7	90.8
	LtC-MSDA [94]	85.6	99.0	98.3	83.2	93.0	91.8
	STEM (ours)	**89.7**	**99.4**	**98.4**	**89.9**	**97.5**	**95.0**

Table 8: Classification accuracy (%) on Digits-five.

Standard	Methods	→**W**	→**D**	→**C**	→**A**	Avg
Source Combine	Source-only	99.0	98.3	87.8	86.1	92.8
	DAN [52]	99.3	98.2	89.7	94.8	95.5
Multi-Source	Source-only	99.1	98.2	85.4	88.7	92.9
	DAN [52]	99.5	99.1	89.2	91.6	94.8
	DCTN [102]	99.4	99.0	90.2	92.7	95.3
	JAN [55]	99.4	99.4	91.2	91.8	95.5
	MEDA [95]	99.3	99.2	91.4	92.9	95.7
	MCD [82]	99.5	99.1	91.5	92.1	95.6
	M^3SDA [72]	99.5	99.2	92.2	94.5	96.4
	CMSS [106]	99.6	99.3	93.7	96.6	97.2
	STEM (ours)	**100**	**100**	**94.2**	**98.4**	**98.2**

Table 9: Classification accuracy (%) on Office-Caltech10 dataset.

tasks, i.e., →**mm**, →**sv**, and →**sy** by sizeable margins of 4.1%, 6.7%, and 4.5% respectively and rank the first on average with a significant gap of 3.2%.

5.5.3 Experimental Results on Office-Caltech10

Office-Caltech10 [26] consists of four domains: Amazon (**A**), Caltech (**C**), DSLR (**D**), and Webcam (**W**). There are ten categories in each domain, and the total number of images is $2,533$. In this experiment, we split the training and testing set with a ratio of 80% and 20%, respectively, and use ResNet-101 [31] pre-trained on ImageNet as a backbone.

In Table 9, we present the results of STEM and the baselines. Overall, it can be seen that our STEM surpasses the baselines in all four settings and achieves 98.2% on average. Since the baselines already achieve impressive performances on all adaptation tasks, it is hard to gain significant improvements. However, on two adaptation tasks (i.e., →**W** and →**D**), our model yields impressive performances with two perfect scores of 100%, while STEM also achieves remarkable improvements on the other tasks.

Standard	Methods	→**clp**	→**inf**	→**pnt**	→**qdr**	→**rel**	→**skt**	Avg
Single Best	Source-only	39.6	8.2	33.9	11.8	41.6	23.1	26.4
	DAN [52]	39.1	11.4	33.3	16.2	42.1	29.7	28.6
	RTN [54]	35.3	10.7	31.7	13.1	40.6	26.5	26.3
	JAN [55]	35.3	9.1	32.5	14.3	43.1	25.7	26.7
	DANN [21]	37.9	11.4	33.9	13.7	41.5	28.6	27.8
	ADDA [90]	39.5	14.5	29.1	14.9	41.9	30.7	28.4
	MCD [82]	42.6	19.6	42.6	3.8	50.5	33.8	32.2
Source Combine	Source-only	47.6	13.0	38.1	13.3	51.9	33.7	32.9
	DAN [52]	45.4	12.8	36.2	15.3	48.6	34.0	32.1
	RTN [54]	44.2	12.6	35.3	14.6	48.4	31.7	31.1
	JAN [55]	40.9	11.1	35.4	12.1	45.8	32.3	29.6
	DANN [21]	45.5	13.1	37.0	13.2	48.9	31.8	32.6
	ADDA [90]	47.5	11.4	36.7	14.7	49.1	33.5	32.2
	MCD [82]	54.3	22.1	45.7	7.6	58.4	43.5	38.5
Multi-Source	MDAN [107]	52.4	21.3	46.9	8.6	54.9	46.5	38.4
	DCTN [102]	48.6	23.5	48.8	7.2	53.5	47.3	38.2
	M^3SDA [72]	58.6	26.0	52.3	6.3	62.7	49.5	42.6
	MDDA [108]	59.4	23.8	53.2	12.5	61.8	48.6	43.2
	CMSS [106]	64.2	28.0	53.6	16.0	63.4	53.8	46.5
	LtC-MSDA [94]	63.1	**28.7**	56.1	16.3	66.1	53.8	47.4
	STEM (ours)	**72.0**	28.2	**61.5**	**25.7**	**72.6**	**60.2**	**53.4**

Table 10: Classification accuracy (%) on DomainNet dataset.

5.5.4 Experimental Results on DomainNet

DomainNet was first introduced in [72] and has become the most challenging dataset in MSDA. It consists of around 0.6 million images of 345 categories from 6 domains: *clipart* (clp), *infograph* (inf), *quickdraw* (qdr), *real* (rel) and *sketch* (skt). Prominently, the high number of classes and enormous noise in this dataset makes it challenging to gain satisfactory performances even when training and testing for supervised classification tasks in an individual domain, especially the *infograph* domain. Moreover, a significant difference in the distribution of each domain causes the domain shift problem when transferring knowledge. For all experiments on this dataset, we utilize ResNet-101 [31] pre-trained on ImageNet as the backbone.

We compare STEM with the current state-of-the-art method which is LtC-MSDA [94]. As shown in Table 10, our STEM exceeds LtC-MSDA on 5 out of 6 transfer tasks with significant improvements of 8.9% on →**clp** task, 9.4% on →**qdr** task, and 6.5% on →**rel** task. Averagely, STEM also yields an impressive improvement of 6.0%.

5.6 Ablation Study

5.6.1 Latent Space Visualization

The crucial factors for the success of our STEM include (i) the mix-up of target domain and the mixture of source domains in the latent space and (ii) the target examples are located in their matching classes in the source domains. To visually demonstrate why STEM can achieve good performances, we utilize t-SNE [92] to visualize the representations of target and source examples in the latent space. It is noticeable that in Figure 11, we visualize the case in which the target domain is USPS and the rest serves as source domains. As shown in Figure 11 (Left) wherein we visualize the mixture of source domains and the target domain when the model is trained with source domains only. In Figure 11 (Right), we show how accurately the target examples match the classes in the source domains when training the model with STEM approach. It is evident that our STEM forms source domains and target domain into the same

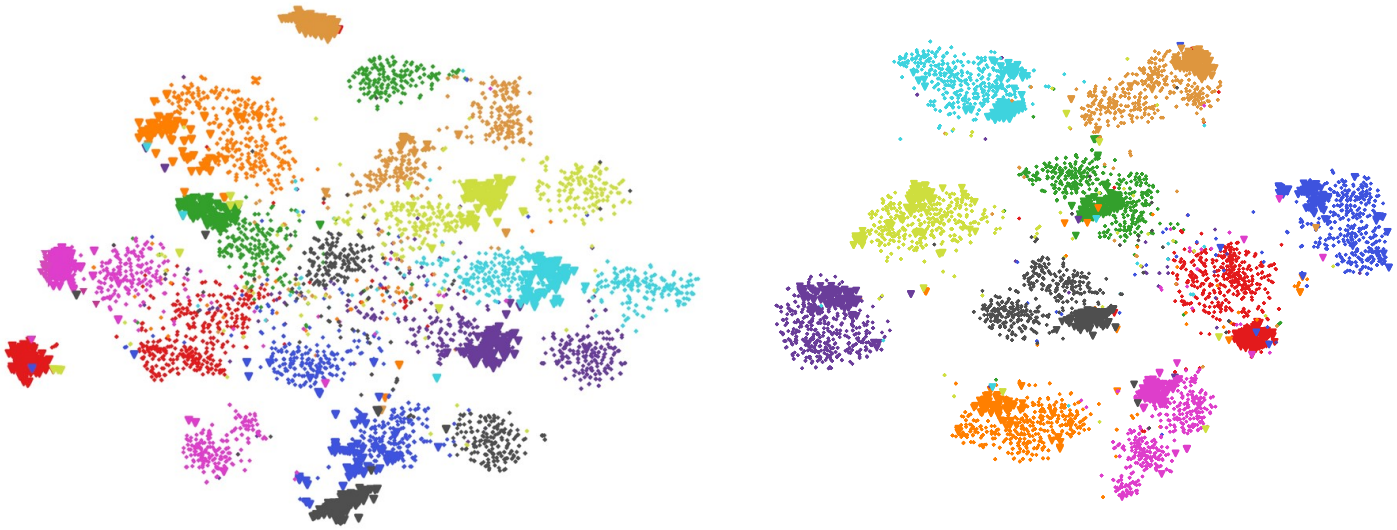


Figure 11: The t-SNE visualization of the transfer task →**up** with label and domain information in two settings: *Source-only* (left) and our *STEM* (right). Each color denotes a class while the circle and triangle markers represent the mixture of source and target data respectively.

clusters and the target examples can find their matching classes in the source domains, hence the label shift is mitigated. This explains the qualified performances of our STEM.

5.6.2 Simplified and Theoretical-Oriented Domain Discriminator $\mathscr{C}$

We conduct an ablation study to compare two variants of the domain discriminator $\mathscr{C}$: theoretical oriented and simplified versions (see Section 5.3.6). As shown in Table 11, the simplified variant performs better than the theoretical oriented one. We conjecture that this is because the simplified variant still keeps the principal spirit of the theoretically oriented one, while much easier to train due to its simplicity. Therefore, we select the simplified variant in all experiments.

Method	→**mm**	→**mt**
Theoretical $\mathscr{C}$	86.8	99.1
Simplified $\mathscr{C}$	89.7	99.4

Table 11: Comparison of the theoretical oriented and simplified version of the proposed method

$\mathscr{L}^{vat}$	$\mathscr{L}^{ent}$	→**mm**	→**up**
		83.04	96.86
✓		86.25	96.11
	✓	86.82	97.11
✓	✓	89.71	98.42

Table 12: Ablation study for the affection of VAT and entropy term.

5.6.3 Clustering Assumption Effect

We now speculate the effect of VAT and conditional entropy terms on our model performance. According to Table 12, adding $\mathscr{L}^{vat}$ (first row) or $\mathscr{L}^{ent}$ (second row) alone improves the performance, while combining these two losses (third row) even boosts the performance further.

Component	Digit-five	Office-Caltech10	DomainNet
h^S	92.7	97.9	51.6
h^T	95.0	97.9	53.4

Table 13: The comparison of teacher and student performance.

5.6.4 Teacher and Student Performances

We observe that the performance of the student h^T totally depends on that of the teacher h^S. In what follows, we compare the performance of the teacher and student on the target domain. We report the average of the *teacher* and *student*'s accuracy scores for all transfer tasks regarding each dataset. As shown in Table 13, the student outperforms its teacher except for Office-Caltech10 dataset. This totally makes sense because the student not only strictly imitates its teacher, but also is strengthened the generalization ability by enforcing the clustering assumption (see Section 5.3.6).

References

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Mach. Learn.*, 2010.

[4] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017.

[5] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[6] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *AISTATS*, 2005.

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[9] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *NIPS*, 2017.

[10] N. Courty, R. Flamary, and D. Tuia. Domain adaptation with regularized optimal transport. In *Machine Learning and Knowledge Discovery in Databases*, 2014.

[11] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE TPAMI*, 2017.

[12] K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. In *NIPS*. 2007.

[13] N. Dam, Q. Hoang, T. Le, T. D. Nguyen, H. Bui, and D. Phung. Three-player wasserstein gan via amortised duality. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2202–2208. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[14] Nhan Dam, Quan Hoang, Trung Le, Tu Dinh Nguyen, Hung Bui, and Dinh Phung. Three-player wasserstein gan via amortised duality. In *Proc. of the 28th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2019.

[15] B. B. Damodaran, B. Kellenberger, R. Flamary, D. Tuia, and N. Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*, 2018.

[16] T. P. Dinh and T. H. A. Le. Convex analysis approach to d.c. programming: Theory, algorithm and applications. 1997.

[17] L. Duan, D. Xu, and S. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *CVPR*, 2012.

[18] K. Saito et al. Asymmetric tri-training for unsupervised domain adaptation. In *ICML*, 2017.

[19] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018.

[20] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[21] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

[22] A. Genevay, M. Cuturi, G. Peyré, and F. Bach. Stochastic optimization for large-scale optimal transport. In *NIPS*. 2016.

[23] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, 2016.

[24] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, 2013.

[25] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[26] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. pages 2066–2073, 06 2012.

[27] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[29] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*. 2005.

[30] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[31] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[32] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6629–6640, 2017.

[33] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[34] J. Ho and S. Ermon. Generative adversarial imitation learning. In *NIPS*, 2016.

[35] N. Ho, V. Huynh, D. Phung, and M. I. Jordan. Probabilistic multilevel clustering via composite transportation distance. In *AISTATS*, 2019.

[36] N. Ho, X. L. Nguyen, M. Yurochkin, H. H. Bui, V. Huynh, and D. Phung. Multilevel clustering via wasserstein means. In *ICML*, 2017.

[37] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. Multi-generator gernerative adversarial nets. *arXiv preprint arXiv:1708.02556*, 2017.

[38] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. MGAN: Training generative adversarial nets with multiple generators. In *International Conference on Learning Representations*, 2018.

[39] J. Hoffman, M. Mohri, and N. Zhang. Algorithms and theory for multiple-source adaptation. In *NeurIPS*. Curran Associates, Inc., 2018.

[40] J. J. Hull. A database for handwritten text recognition research. *IEEE TPAMI*, 1994.

[41] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[42] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[43] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[44] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[45] A. Kumar, P. Sattigeri, K. Wadhawan, L. Karlinsky, R. Feris, B. Freeman, and G. Wornell. Co-regularized alignment for unsupervised domain adaptation. In *NeurIPS*. 2018.

[46] T. Le, Q. Hoang, H. Vu, T. D. Nguyen, H. Bui, and D. Phung. Learning generative adversarial networks from multiple data sources. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2823–2829. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[47] T. Le, T. Nguyen, N. Ho, H. Bui, and D. Phung. Lamda: Label matching deep domain adaptation. In *ICML*, 2021.

[48] Trung Le, Hung Vu, Tu Dinh Nguyen, and Dinh Phung. Geometric enclosing networks. In *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2018.

[49] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[50] C. Lee, T. Batra, M. H. Baig, and D. Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, 2019.

[51] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[52] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[53] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*. 2018.

[54] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*. 2016.

[55] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017.

[56] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *NIPS*. 2009.

[57] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[58] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[59] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[60] T. Miyato, S. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE TPAMI*, 2019.

[61] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations*, 2018.

[62] R. Müller, S. Kornblith, and G. E. Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4694–4703, 2019.

[63] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[64] T. Nguyen, T. Le, H. Zhao, H. Q. Tran, T. Nguyen, and D. Phung. Most: Multi-source domain adaptation via optimal transport for student-teacher learning. In *UAI*, 2021.

[65] T. Nguyen, T. Le, H. Zhao, H. Q. Tran, T. Nguyen, and D. Phung. Tidot: A teacher imitation learning approach for domain adaptation with optimal transport. In *IJCAI*, 2021.

[66] Tu Dinh Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems 29 (NIPS)*, 2017.

[67] V. Nguyen, T. Le, O. De Vel, P. Montague, J. Grundy, and D. Phung. Dual-component deep domain adaptation: A new approach for cross project software vulnerability detection. In *PAKDD*, 2020.

[68] V. Nguyen, T. Le, T. Le, K. Nguyen, O. De Vel, P. Montague, L. Qu, and D. Phung. Deep domain adaptation for vulnerable code function identification. In *IJCNN*, 2019.

[69] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pages 271–279, 2016.

[70] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *IJCAI*, 2009.

[71] Y. Pan, T. Yao, Y. Li, Y. Wang, C. Ngo, and T. Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*, 2019.

[72] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

[73] G. Peyré, M. Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 2019.

[74] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, July 1992.

[75] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[76] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[77] I. Redko, N. Courty, R. Flamary, and D. Tuia. Optimal transport for multi-source domain adaptation under target shift. In *AISTATS*, 2019.

[78] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*. 2015.

[79] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems 30*, pages 2018–2028. 2017.

[80] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[81] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[82] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

[83] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[84] S. Sankaranarayanan, Y. Balaji, Carlos D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. *CVPR*, 2018.

[85] F. Santambrogio. Optimal transport for applied mathematicians. *Birkäuser*, 2015.

[86] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE TPAMI*, 2017.

[87] R. Shu, H. H. Bui, H. Narui, and S. Ermon. A DIRT-t approach to unsupervised domain adaptation. In *ICLR*, 2018.

[88] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

[89] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[90] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

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

[92] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 2008.

[93] C. Villani. *Optimal Transport: Old and New*. Springer Berlin Heidelberg, 2008.

[94] H. Wang, M. Xu, B. Ni, and W. Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *ECCV*, 2020.

[95] J. Wang, Wenjie Feng, Y. Chen, H. Yu, M. Huang, and Philip S. Yu. Visual domain adaptation with manifold embedded distribution alignment. *ACM international conference on Multimedia*, 2018.

[96] David Warde-Farley and Yoshua Bengio. Improving generative adversarial networks with denoising feature matching. 2016.

[97] S. Xie, Z. Zheng, L. Chen, and C. Chen. Learning semantic representations for unsupervised domain adaptation. In *ICML*, 2018.

[98] Y. Xie, M. Chen, H. Jiang, T. Zhao, and H. Zha. On scalable and efficient computation of large scale optimal transport. In *ICML*, 2019.

[99] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. Proceedings of Machine Learning Research, 2015.

[100] M. Xu, H. Wang, B. Ni, Q. Tian, and W. Zhang. Cross-domain detection via graph-induced prototype alignment. In *CVPR*, 2020.

[101] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang. Adversarial domain adaptation with domain mixup. In *AAAI*, 2020.

[102] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *CVPR*, 2018.

[103] R. Xu, P. Liu, L. Wang, C. Chen, and J. Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR*, 2020.

[104] H. Yan, Yukang Ding, P. Li, Qilong Wang, Yong Xu, and W. Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. *CVPR*, 2017.

[105] Y. Yan, W. Li, H. Wu, H. Min, M. Tan, and Q. Wu. Semi-supervised optimal transport for heterogeneous domain adaptation. In *IJCAI*, 2018.

[106] L. Yang, Y. Balaji, S. Lim, and A. Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 608–624, Cham, 2020.

[107] H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J Gordon. Adversarial multiple source domain adaptation. In *NeurIPS*. 2018.

[108] S. Zhao, G. Wang, S. Zhang, Y. Gu, Y. Li, Z. Song, P. Xu, R. Hu, H. Chai, and K. Keutzer. Multi-source distilling domain adaptation. In *AAAI*, 2020.