Partial Identifiability for Domain Adaptation

Lingjing Kong 1  Shaoan Xie 1  Weiran Yao 1  Yujia Zheng 1  Guangyi Chen 2,1  Petar Stojanov 3  Victor Akinwande 1  Kun Zhang 2,1

Abstract

Unsupervised domain adaptation is critical to many real-world applications where label information is unavailable in the target domain. In general, without further assumptions, the joint distribution of the features and the label is not identifiable in the target domain. To address this issue, we rely on a property of minimal changes of causal mechanisms across domains to minimize unnecessary influences of domain shift. To encode this property, we first formulate the data generating process using a latent variable model with two partitioned latent subspaces: invariant components whose distributions stay the same across domains, and sparse changing components that vary across domains. We further constrain the domain shift to have a restrictive influence on the changing components. Under mild conditions, we show that the latent variables are partially identifiable, from which it follows that the joint distribution of data and labels in the target domain is also identifiable. Given the theoretical insights, we propose a practical domain adaptation framework, called iMSDA. Extensive experimental results reveal that iMSDA outperforms state-of-the-art domain adaptation algorithms on benchmark datasets, demonstrating the effectiveness of our framework.

1. Introduction

Unsupervised domain adaptation (UDA) is a form of prediction setting in which the labeled training data, and the unlabeled test data follow different distributions. UDA can frequently be done in settings where multiple labeled training datasets are available, and this is termed as multiple-source UDA. Formally, given features \( x \), target variables \( y \), and domain indices \( u \), the training (source domain) data follows multiple joint distributions \( p_{x,y|u_1}, p_{x,y|u_2}, \ldots, p_{x,y|u_M} \), and the test (target domain) data follows the joint distribution \( p_{x,y|u_T} \), where \( p_{x,y|u} \) may vary across \( u_1, u_2, \ldots, u_M \). During training, for each \( i \)-th source domain, we are given labeled observations \( (x_k^{(i)}, y_k^{(i)})_{m_i} \) from \( p_{x,y|u_i} \), and target domain unlabeled instances \( (x_k^{T})_{M_T} \) from \( p_{x,y|u_T} \). The main goal of domain adaptation is to make use of the available observed information, to construct a predictor that will have optimal performance in the target domain.

It is apparent that without further assumptions, this objective is ill-posed. Namely, since the only available observations in the target domain are from the marginal distribution \( p_{x|u_T} \), the data may correspond to infinitely many joint distributions \( p_{x,y|u_T} \). This mandates making additional assumptions on the relationship between the source and the target domain distributions, with the hope to be able to reconstruct (identify) the joint distribution in the target domain \( p_{x,y|u_T} \). Typically, these assumptions entail some measure of similarity across all of the joint distributions. One classical assumption is that each joint distribution shares the same conditional distribution \( p_{y|x} \), whereas \( p_{x|u} \) changes, more widely known as the covariate shift setting (Shimodaira, 2000; Sugiyama et al., 2008; Zadrozny, 2004; Huang et al., 2007; Cortes et al., 2010). In addition, assumptions can be made that the source and the target domain distributions are related via the changes in \( p_{y|u} \) or one or more linear transformations of the features \( x \) (Zhang et al., 2013; 2015; Gong et al., 2016).

In order to drop any parametric assumptions regarding the relationship between the domains, one may want to consider a more abstract assumption of minimal change. To make reasoning about this notion easier, it is often useful to consider the data-generating process (Schölkopf et al., 2012; Zhang et al., 2013). For example, if the data generating process \( Y \rightarrow X \), then \( p_{y|u} \) and \( p_{x|y,u} \) change independently across domains, and factoring the joint distribution in terms of these factors makes it possible to represent its changes in the most parsimonious way. Furthermore the changes

\[ p_{x,y|u}(x,y|u') \]

1. With a slight abuse of notation, we use \( p_{x,y|u} \) to represent the conditional distribution for a specific domain \( u' \), i.e. \( p_{x,y|u}(x,y|u') \).
of the distribution \( p_{x|y,u} \) can be represented as lying on a low-dimensional manifold (Stojanov et al., 2019). The generating process of the observed features and its changing parts across domains can also be automatically discovered from multiple-domain data (Zhang et al., 2017b; Huang et al., 2020), and the changes can be captured using conditional GANs (Zhang et al., 2020).

With the advent of deep learning capabilities and their widespread use, another approach is to harness these deep architectures to learn a feature map to some latent variable \( z \), such that \( p_{x|u_1} = p_{x|u_2} = \ldots = p_{x|u_M} = p_{x|u^*} \) (i.e. they are marginally invariant), while at the same time training a classifier \( f_{cls} : Z \rightarrow Y \) on the labeled source domain data, which ensures that \( z \) has predictive information about \( y \) (Ganin & Lempitsky, 2014; Ben-David et al., 2010; Zhao et al., 2018). This could still mean that the joint distributions \( p_{x,y|u} \) may be different across domains \( u_1, \ldots, u_M \), leading to potentially poor prediction performance in the target domain. Solving the problem has also been attempted using deep generative models and enforcing disentanglement of the latent representation (Lu et al., 2020; Cai et al., 2019). However, none of these efforts establishes identifiability of \( p_{x,y|u^*} \).

Since deep architectures are a requirement for strong performance on high-dimensional datasets, it has become essential to combine their representational power with the notion of minimal change of the joint distribution across domain \( u \)’s. In fact, one can make reasonable assumptions about the role of the latent representation \( z \) in the data-generating process, and represent the minimal change of \( p_{x,y|u} \) across domains such that both \( z \) and \( p_{x,y|u} \) are identifiable in the target domain.

In this paper, we represent the variables \( x, y \) and \( z \) in terms of a data-generating process assumption. To enforce minimal change of \( p_{x,y,z|u} \) across domains, we allow only a partition of the latent space to vary over domains and constrain the flexibility of the influence from domain changes. (1) Under this generating process, we show that both the changing and the invariant parts in the generating process are (partially) identifiable, which gives rise to a principled way of aligning source and target domains. (2) Leveraging the identifiability results, we propose an algorithm that identifies the latent representation and utilizes the representation for optimal prediction for the target domain. (3) We show that our algorithm surpasses state-of-the-art UDA algorithms on benchmark datasets.

2. Related Work

Domain adaptation (Cai et al., 2019; Courty et al., 2017; Deng et al., 2019; Tzeng et al., 2017; Wang et al., 2019; Xu et al., 2019; Zhang et al., 2017a; 2013; 2018; 2019; Mao et al., 2021; Stojanov et al., 2021; Wu et al., 2022; Eastwood et al., 2022; Tong et al., 2022; Zhu et al., 2022; Xu et al., 2022; Berthelot et al., 2022; Kirchmeyer et al., 2022) is an actively studied area of research in machine learning and computer vision. One popular direction is by invariant representation learning, which is introduced by (Ganin & Lempitsky, 2014). Specifically, it has been successively refined in several different ways to ensure that the invariant representation \( z \) contains meaningful information about the target domain. For instance, in (Bousmalis et al., 2016), \( z \) was divided into a changing and an invariant part, and a constraint was added that \( z \) has to be able to reconstruct the original data, in addition to predicting \( y \) (from the shared part). Furthermore, pseudo-labels have also been utilized in order to match the joint distribution \( p_{x,y} \) across domains, using kernel methods (Long et al., 2017a;b). Another approach to ensure more structure in \( z \) is to assume that the class-conditional distribution \( p_{z|y} \) has a clustering structure, and regularize the neural network algorithm to prevent decision boundaries from crossing high-density regions (Shu et al., 2018; Xie et al., 2018). In these studies, \( p_{y} \) was typically assumed to stay the same across domains. Dropping this assumption has been investigated in (Wu et al., 2019; Guo et al., 2020). Furthermore, the setting in which the label sets do not exactly match has been also been studied (Saito et al., 2020). While the body of work studying representation learning for domain adaptation is large and extensive, there are still no guarantees that the learned representation will help learn the optimal predictor in the target domain, even in the infinite-sample case. Besides, invariant representation learning also sheds light on adaptation with multiple sources (Xu et al., 2018; Peng et al., 2019; Wang et al., 2020; Li et al., 2021; Park & Lee, 2021). Along the line of work of generative models, studies on generative models and unsupervised learning have made headway into better understanding the properties of identifiability of a latent representation which is meant to capture some underlying factors of variation, from which the data was generated. Namely, independent component analysis (ICA) is the classical approach for learning a latent representation for which there are identifiability results, where the generating process is assumed to consist of a linear mixing function (Comon, 1994; Bell & Sejnowski, 1995; Hyvärinen et al., 2001). A major problem in nonlinear ICA is that, without assumption on either source distribution or generating process, the model is seriously unidentifiable (Hyvärinen & Pajunen, 1999). Recent breakthroughs introduced auxiliary variables, e.g., domain indexes, to advance the identifiability results (Hyvarinen et al., 2019; Sorrenson et al., 2020; Hälvi & Hyvärinen, 2020; Lachapelle et al., 2021; Khemakhem et al., 2020a; von Kügelgen et al., 2021; Lu et al., 2020). These works aim to identify and disentangle the components of the latent representation, while assuming that all of them
are changing across domains.

In the context of self-supervised learning with data augmentation, von Kügelgen et al. (2021) considered the identifiability of shared part, which stays invariant between views, in a block-wise manner. However, this line of work assumes availability of paired instances in two domains. In the context of out-of-distribution generalization, Lu et al. (2020) extend the identifiability result of iVAE (Khemakhem et al., 2020a) to a general exponential family that is not necessarily factorized. However, this study does not take advantage of the fact that the latent representations should contain invariant information that can be disentangled from the part that corresponds to changes across domains. Most importantly, this study resorts to finding a conditionally invariant subpart of \( z \), even though there may be parts of \( z \) that are not conditionally invariant, and yet still relevant for predicting \( y \).

In this paper, we make use of realistic assumptions regarding the data-generating process in order to provably identify the changing and the invariant aspects of the latent representation in both the source and target domains. In doing so, we drop any parametric assumptions about \( z \) and we allow both the changing and invariant parts to have predictive information about \( y \). We show that we can learn align domains and make predictions in a principled manner, and we present an autoencoder algorithm to solve the problem in practice.

### 3. High-level Invariance for Domain Adaptation

In this section, we introduce our data generating process in Figure 1 and Equation 1 and discuss how we could exploit this latent variable model to handle UDA.

\[
z_c \sim p_{z_c}, \quad \tilde{z}_s \sim \tilde{p}_{\tilde{z}_s}, \quad z_s = f_u(\tilde{z}_s), \quad x = g(z_c, z_s). \tag{1}
\]

In the generating process, we assume that data \( x \in \mathcal{X} \) (e.g. images) are generated by latent variables \( z \in \mathcal{Z} \subseteq \mathbb{R}^n \) through an invertible and smooth mixing function \( g : \mathcal{Z} \rightarrow \mathcal{X} \). We denote by \( u \in \mathcal{U} \) the domain embedding, which is a constant vector for a specific domain.

We partition latent variables \( z \) into two parts \( z = [z_c, z_s] \): the invariant part \( z_c \in \mathcal{Z}_c \subseteq \mathbb{R}^{n_c} \) (i.e. content) of which the distribution stays constant over domain \( u \)'s, and the changing part \( z_s \in \mathcal{Z}_s \subseteq \mathbb{R}^{n_s} \) (i.e. style) with varying distributions over domains.

We parameterize the influence of domain \( u \) on \( z_s \) as a simple transformation of some generic form of the changing part, given by \( \tilde{z}_s \). Namely, given a component-wise monotonic function \( f_u \), we let \( z_s = f_u(\tilde{z}_s) \). For example, in image datasets, \( z_s \) can correspond to various kinds of background from the images (sand, trees, sky, etc.), and in this case \( \tilde{z}_s \) can be interpreted as a generic background pattern that can easily be transformed into a domain-specific image background, depending on which function \( f_u \) is used.

Further, we assume that \( y \) is generated by invariant latent variables \( z_c \) and \( z_s \). Thus, this generating process addresses the conditional-shift setting, in which \( p_y|_{u} \) changes across domains, and \( p_y|_{u} = p_y \) stays the same.

We describe below the distinguishing features of our generating process, and illustrate how these features are essential to tackling UDA.

### Partitioned latent space

As discussed in Section 1, the bulk of prior work (Ganin & Lempitsky, 2014; Ben-David et al., 2010; Zhao et al., 2018) focuses on learning an invariant representation over domains so that one classifier can be applied to novel domains in the latent space. Unlike these approaches, we will demonstrate that our parameterization of \( z_c \) allows us to preserve the information of the changing part \( z_s \) for prediction instead of discarding this part altogether by imposing invariance. Similarly, recent work (Lu et al., 2020) allows for the possibility that all latent variables could be influenced by domain changes in their framework to address domain shift, but discards a subset of them which may be relevant for predicting \( y \). In contrast, our goal is to disentangle the changing and invariant parts \( z_s \) and \( z_c \), capture the relationship between \( z_s \) and \( y \) across domains by learning \( f_u \), and use this information to perform prediction in the target domain.

Our parameterization also allows us to implement the minimal change principle by constraining the changing components to be as few as possible. Otherwise, unnecessarily large domain influences (e.g. all components being changing) may lead to loss of semantic information in the invariant \( (z_c, \tilde{z}_s) \) space. For instance, in a scenario where each domain comprises digits 0-9 with a specific rotation degree, unconstrained influence may result in a mapping between 1 and 9 across domains.
**High-level Invariance \( \tilde{z}_s \)**  We note that presence of \( \tilde{z}_s \) is significant, as it allows us to provably learn an optimal classifier over domains without requiring that \( p_{x|u} \) to be invariant over domains as in previous work (Ganin & Lempitsky, 2014; Ben-David et al., 2010; Zhao et al., 2018). With the component-wise monotonic function \( f_u \), we are able to identify \( \tilde{z}_s \) through \( z_s \), which is critical to our ability to learn how the changes across domains are related to each other. If we know \( f_u \) and \( z_s \), we can always map all domains to this high-level \( \tilde{z}_s \), which will have the desired information to predict \( y \) (in addition to \( z_s \)). Additionally, this restrictive function form also contributes to regulating unnecessary changes.

The problem remains whether it is possible to identify the latent variables of interest (i.e., \( z_c, \tilde{z}_s \)) with only observational data \((x, u)\). Because if this is the case, we can leverage labeled data in source domains to learn a classifier according to \( p_{y|x, z_c, \tilde{z}_s} \), that is universally applicable over given domains. In Section 4, we show that we can indeed achieve this identifiability for this generating process, and describe how such a model can be learned. In Section 5, we present the corresponding algorithmic implementation based on Variational Auto-Encoder (VAE) (Kingma & Welling, 2014).

**4. Partial Identifiability of the Latent Variables**

In this section, we show our theoretical results that serve as the foundation of our algorithm. In addition to guiding algorithmic design, we highlight that our theory is a novel contribution to the deep generative modeling literature. In Section 4.1 and Section 5, we show that with unlabeled observational data \((x, u)\) over domains we can partially identify the latent space. In Section 4.3, we discuss how these identifiability properties, together with labeled source domain data, give rise to a classifier useful to the target domain.

We parameterize \( z_s = f_u(\tilde{z}_s) \) with a component-wise monotonic function for each domain \( u \), and learn a model \((\hat{g}, p_{\tilde{z}_s}, p_{z_s|u}) \) \(^2\) that assumes the identical process as in Equation 1 and matches the true marginal data distribution in all domains, that is,

\[ p_{x|u}(x'|u') = p_{x|u}(x'|u'), \quad \forall x' \in X, \ u' \in U, \]

where the random variable \( x \) is generated from the true process \((g, p_{z_c}, p_{z_s|u})\) and \( \hat{x} \) is from the estimated model \((\hat{g}, p_{\tilde{z}_s}, p_{z_s|u})\).

In the following, we show that our estimated model can recover the true changing components and the invariant subspace \( z_c \), and then discuss the implications of our theory for UDA.

\(^2\) We use \( \hat{\cdot} \) to distinguish estimators from the ground truth.

**4.1. Identifiability of Changing Components**

First, we show that the changing components can be identified up to permutation and invertible component-wise transformations. More formally, for each true changing component \( z_{s,j} \), there exists a corresponding estimated changing component \( \hat{z}_{s,j} \) and an invertible function \( h_{s,i}: \mathbb{R} \rightarrow \mathbb{R} \), such that \( \hat{z}_{s,i} = h_{s,i}(z_{s,i}) \). For ease of exposition, we assume that the \( z_c \) and \( z_s \) correspond to components in \( z \) with indices \( \{1, \ldots, n_c\} \) and \( \{n_c+1, \ldots, n\} \) respectively, that is, \( z_c = (z_i)_{i=1}^{n_c} \) and \( z_s = (z_i)_{i=n_c+1}^{n} \).

**Theorem 4.1.** We follow the data generation process in Equation 1 and make the following assumptions:

- **A1 (Smooth and Positive Density):** The probability density function of latent variables is smooth and positive, i.e., \( p_{z_s|u} \) is smooth and \( p_{z_s|u} > 0 \) over \( Z \) and \( U \).
- **A2 (Conditional independence):** Conditioned on \( u \), each \( z_i \) is independent of any other \( z_j \) for \( i, j \in [n] \), \( i \neq j \), i.e., \( \log p_{z|u}(z|u) = \sum_{i=1}^{n} q_i(z_i, u) \) where \( q_i \) is the log density of the conditional distribution, i.e., \( q_i := \log p_{z_i|u} \).
- **A3 (Linear independence):** For any \( z_s \in Z_s \subseteq \mathbb{R}^n_s \), there exist \( 2n_s + 1 \) values of \( u \), i.e., \( u_j \) with \( j = 0, 1, \ldots, 2n_s \), such that the \( 2n_s \) vectors \( w(z_s, u_j) - w(z_s, u_0) \) with \( j = 1, \ldots, 2n_s \), are linearly independent, where vector \( w(z_s, u) \) is defined as follows:

\[
\begin{align*}
w(z_s, u) &= \left( \frac{\partial q_{n_s+1}(z_{n_s+1}, u)}{\partial z_{n_s+1}}, \ldots, \frac{\partial q_{n_s}(z_{n_s}, u)}{\partial z_{n_s}} \right), \\
\frac{\partial^2 q_{n_s+1}(z_{n_s+1}, u)}{\partial z_{n_s+1}^2}, \ldots, \frac{\partial^2 q_{n_s}(z_{n_s}, u)}{\partial z_{n_s}^2} \right). \quad (3)
\end{align*}
\]

By learning \((\hat{g}, p_{\tilde{z}_s}, p_{z_s|u})\) to achieve Equation 2, \( z_s \) is component-wise identifiable.

A proof can be found in Appendix A.1. We note that all three assumptions are standard in the nonlinear ICA literature (Hyvarinen et al., 2019; Khemakhem et al., 2020a). In particular, the intuition of Assumption A3 is that \( p_{z_s|u} \) varies sufficiently over domain \( u \)’s. We demonstrate in Section 6 that this identifiability can be largely achieved with Gaussian distributions of distinct means and variances.

It is noteworthy that Theorem 4.1 manifests the benefits of minimal changes - the smaller \( n_s \), the more easily for the linear independence condition to hold, as we need only \( 2n_s + 1 \) domains with sufficient variability, whereas prior work often require \( 2n + 1 \) sufficiently variable domains. We note that in many DA problems, the relations between domains are explicit, so the dimensionality of the intrinsic change is small. Our results provide insights into whether a fixed number of domains is sufficient for DA with different levels of change.
The identifiability of the changing part is crucial to our method. Only by recovering the changing components \( z_s \) and the corresponding domain influence \( f'_u \), can we correctly align various domains with \( z_s \) and learn a classifier there.

### 4.2. Identifiability of the Invariant Subspace

Building on the identifiability of changing components, we show that the invariant subspace is identifiable in a block-wise fashion. Our goal is to show that the estimated invariant part has preserved information of the true invariant part, and there is no information mixed in from the changing components. Formally, we would like to show that there exists an invertible mapping \( h'_c : Z_c \rightarrow Z_c \) between the estimated invariant part \( \tilde{z}_c \) and the true invariant part \( z_c \), s.t. \( z_c = h'_c(\tilde{z}_c) \). The idea of block-wise identifiability emerges in independent subspace analysis (Casey & Westner, 2000; Hyvärinen & Hoyer, 2000; Le et al., 2011; Theis, 2006), and recent work (von Kügelgen et al., 2021) addresses it in the nonlinear ICA scenario.

In the following theorem, we show that the block-wise identifiability of the invariant subspace can be attained with the estimated data generating process. Note that the block-wise identifiability is a weaker notion of identifiability than the component-wise identifiability achieved for the changing components. Thus, we only claim partial identifiability for the invariant latent variables.

**Theorem 4.2.** We follow the data generation process in Equation 1 and make Assumption A1-A3. In addition, we make the following assumption:

- **A4 (Domain Variability):** For any set \( A_z \subseteq Z \) with the following two properties:
  - a) \( A_z \) has nonzero probability measure, i.e. \( \Pr(|z \in A_z| |u = u'|) > 0 \) for any \( u' \in \mathcal{U} \).
  - b) \( A_z \) cannot be expressed as \( B_{z_c} \times Z_s \) for any \( B_{z_c} \subset Z_c \).

\[ \exists u_1, u_2 \in \mathcal{U}, \text{such that} \]
\[ \int_{z \in A_z} p_{z|u}(z|u_1)dz \neq \int_{z \in A_z} p_{z|u}(z|u_2)dz. \]

With the model \((\hat{g}, p_{z_c}, p_{z_s|u})\) estimated by observing Equation 2, the \( z_c \) is block-wise identifiable.

A proof is deferred to Appendix A.2.

Like Assumption A3, Assumption A4 also requires the distribution \( p_{z_s|u} \) to change sufficiently over domains. Intuitively, the chance of having one such subset \( A_z \) on which all domain distributions have an equal probability measure is very slim when the number of domains is large and the domain distributions differ sufficiently. In Section 6, we verify Theorem 4.2 with Gaussian distributions with varying means and variances.

We note that in comparison to the identifiability theory in recent work (von Kügelgen et al., 2021), our setting is more challenging in a sense that the theory in (von Kügelgen et al., 2021) relies on the pairing of augmented data, whereas ours does not - we only assume the invariant marginal distribution of \( p_{z_c} \) does not vary over domains.

### 4.3. Identifiability of the Joint Distribution in the Shared Space

Equipped with the identifiability of the changing components and the invariant subspace, we now discuss how these can result in the identifiability of the joint distribution \( p_{x,y}|u^T \) for the target domain \( u^T \) with a classifier trained on source domain labels.

Since \( g \) is partially invertible and partially identifiable (Theorem 4.1 and Theorem 4.2), we can recover \( z_c, z_s \) from observation \( x \) in any domain, that is, we can partially identify the joint distribution \( p_{x,z_c,z_s}|u \). As \( z_s = f'_u(\tilde{z}_s) \) and \( f_u \) is constrained to be component-wise monotonic and invertible, we can further identify \( p_{x,z_c,\tilde{z}_s}|u \). Also note that the label \( y \) is conditionally independent of \( x \) and especially \( u \) given \( z_c \) and \( \tilde{z}_s \) as shown in Figure 1, which means that \( z_c \) and \( \tilde{z}_s \) capture all the information in \( x \) that is relevant to \( y \). Therefore, given \( z_c, \tilde{z}_s \) we can make prediction in the target domain with the predictor \( p_{y|z_c,\tilde{z}_s} \), and this predictor can be learned on source domains, as \( p_{x,z_c,z_s}|u \) is identifiable as discussed above. This is exactly the goal of UDA. In fact, we can identify the joint distribution \( p_{x,y}|u^T \) - with the identified \( p_{x,z_c,\tilde{z}_s}|u^T \) for the target domain, we can derive
\[ p_{x,y}|u^T = \int_{\tilde{z}_s} \int_{z_c} p_{x,z_c,\tilde{z}_s}|u^T \cdot p_{y|z_c,\tilde{z}_s} dz_c d\tilde{z}_s. \]

The intuition is that by resorting to the high-level invariant part \( \tilde{z}_s \), we are able to account for the influence of the domains, and thus obtain a classifier useful to all domains participating in the estimator learning, which includes the target domain in the UDA setup. This reasoning motivates our algorithm design presented in Section 5.

### 5. Partially-Identifiable VAE for Domain Adaptation

In this section, we discuss how to turn the theoretical insights in Section 4 into a practical algorithm, iMSDA (identifiable Multi-Source Domain Adaptation), by leveraging expressive deep learning models. As instructed by the theory, we can estimate the underlying causal generating process (Figure 1) and recover the ground-truth latent variables up to certain subspaces. This in turn allows for optimal classification across domains. In the following, we describe
Figure 2. Diagram of our proposed method, iMSDA. We first apply the VAE encoder \( f_u \) to encode \( x \) into \( (\hat{z}_c, \hat{z}_s) \), which is further fed into the decoder \( \hat{g} \) for reconstruction. In parallel, the changing part \( \hat{z}_s \) is passed through the flow model \( f_{u}^{-1} \) to recover the high-level invariant variable \( \hat{z}_s \). We use \( (\hat{z}_c, \hat{z}_s) \) for classification with the classifier \( f_{\text{cls}} \) and for matching \( N(0, I) \) with a KL loss.

how to estimate each component in the casual generating process \((g, p_{x|u}, p_{z|u}, p_{y|z, s})\) under the VAE framework. The architecture diagram is shown in Figure 2.

We use the VAE encoder to obtain the posterior \( q(\hat{z}|x) \) which essentially estimates the inverse of the mixing function \( g^{-1} \). Specifically, we parameterize the mean and the covariance matrix diagonal of posterior \( q_{f_{u}, f_{z}^{0}}(\hat{z}|x) \) with multilayer perceptron’s (MLP) \( f_{u} \) and \( f_{z}^{0} \) respectively (Equation 4). The VAE decoder \( \hat{g} \), which is also a MLP, processes the estimated latent variable \( \hat{z} \) and estimates the input data \( x \) (Equation 5).

\[
\hat{z} \sim q_{f_{u}, f_{z}^{0}}(\hat{z}|x) := \mathcal{N}(f_{u}(\hat{x}), f_{z}^{0}(x)),
\]

\[
\hat{x} = \hat{g}(\hat{z}).
\]

As indicated in Figure 1, the changing part \( z_s \) of the latent variable \( z \) contains the information of a specific domain \( u \), that is, \( z_s = f_u(z_s) \). To enforce the minimal change property, we assume in Section 4 that domain influence is component-wise monotonic. We estimate the domain influence function \( f_u \) by flow-based architectures (Dinh et al., 2017; Huang et al., 2018; Durkan et al., 2019) for each domain \( u \), i.e., \( z_s = f_u(z_s) \). Benefiting from the invertibility of \( f_u \), we can obtain \( \hat{z}_s \) from the estimated \( \hat{z}_s \) sampled from Equation 4 by inverting the estimated function \( f_u^{-1} \):

\[
\hat{z}_s = f_u^{-1}(\hat{z}_s).
\]

Overall, the VAE loss can be employed to enforce the above described generating process:

\[
L_{\text{VAE}}(f_\mu, f_\Sigma; \hat{f}_u, \hat{g}) = E_{z \sim q_{f_{\mu}, f_{\Sigma}}}(z) \frac{1}{2} \| x - \hat{x} \|^2 + \beta \cdot D(q_{f_u,f_\Sigma}, f_u(\hat{z}|x)\| p(\hat{z})) \tag{7}
\]

where \( \beta \) is a control factor introduced in \( \beta \)-VAE (Higgins et al., 2016). We use \( q_{f_\mu,f_\Sigma}(\hat{z}|x) \) to denote the posterior of \( \hat{z} \) which can be obtained by executing Equation 4 and Equation 6. We choose \( p(\hat{z}) \) as a standard Gaussian distribution \( N(0, I) \), following the independence assumption (Assumption 4.1). The KL divergence in Equation 7 is tractable, thanks to the Gaussian form of \( q_{f_\mu,f_\Sigma} \) and the tractability of the Jacobian determinant of the flow model \( f_u \).

Lastly, we enforce the causal relation between \( y \) and \((z_c, \hat{z}_s)\) in our estimated generating process (Figure 1). This can be implemented by a classification branch:

\[
\hat{y} = f_{\text{cls}}(\hat{z}_c, \hat{z}_s),
\]

where \( f_{\text{cls}} \) is parameterized by a MLP. In sources domains, we directly optimize a cross-entropy loss:

\[
L_{\text{cls}}(f_{\text{cls}}, f_\mu, f_\Sigma; \hat{f}_u, \hat{g}) = -E_{\hat{z}_c, \hat{z}_s, y} \left[ y \cdot \log (\hat{y}) \right],
\]

where \( y \) are one-hot ground-truth label vectors available in the source domains. In the target domain, we enforce this relation by maximizing the mutual information \( I((\hat{z}_c, \hat{z}_s); \hat{y}) \) which amounts to minimizing the conditional entropy \( H(\hat{y}|\hat{z}_c, \hat{z}_s) \) as in (Wang et al., 2020; Li et al., 2021). Doing so facilitates learning a latent representation space adhering to the generating process.

\[
L_{\text{ent}}(f_{\text{cls}}, f_\mu, f_\Sigma; \hat{f}_u, \hat{g}) = -E_{\hat{z}_c, \hat{z}_s} \left[ \hat{y} \cdot \log (\hat{y}) \right].
\]

**Algorithm 1 Training iMSDA**

**Require:** \((x, y, u)\) from source domains and \((x, u)\) from the target domain.

**Require:** \(f_\mu, f_\Sigma, \hat{g}, \hat{f}_u,\) and \(f_{\text{cls}}\).

1. get the latent representation \( \hat{z} \sim q_{f_{\mu}, f_{\Sigma}}(\hat{z}|x) \) in Equation 4;
2. reconstruct the observation \( \hat{x} = \hat{g}(\hat{z}) \) in Equation 4;
3. recover the high-level invariant part \( \hat{z}_s = \hat{f}_u^{-1}(\hat{z}) \) in Equation 6;
4. estimate the class label \( \hat{y} = f_{\text{cls}}(\hat{z}_c, \hat{z}_s) \) in Equation 8;
5. compute and minimize \( L \) according to Equation 7, Equation 10, Equation 9, and Equation 11.
Overall, our loss function is
\[
\mathcal{L}(f_{\text{ch}}, \hat{f}_{\mu}, f_{\Sigma}, f_{\text{VAE}}, \hat{g}) = \mathcal{L}_{\text{ch}} + \alpha_1 \mathcal{L}_{\text{ent}} + \alpha_2 \mathcal{L}_{\text{VAE}},
\]
where \(\alpha_1, \alpha_2\) are hyper-parameters that balance the loss terms. The complete procedures are shown in Algorithm 1.

6. Experiments on Synthetic Data

In this section, we present synthetic data experiments to verify Theorem 4.1 and Theorem 4.2 in practice.

6.1. Experimental Setup

Data generation We generate synthetic data following the generating process of \(x\) in Equation 1. We work with latent variables \(z\) of 4 dimensions with \(n_c = n_s = 2\). We sample \(z_c \sim \mathcal{N}(0, I)\) and \(z_s \sim \mathcal{N}(\mu_u, \sigma_u^2 I)\) where for each domain \(u\) we sample \(\mu_u \sim \text{Unif}(−4, 4)\) and \(\sigma_u^2 \sim \text{Unif}(0.01, 1)\). We choose \(g\) to be a 2-layer MLP with the Leaky-ReLU activation function.

Evaluation metrics To measure the component-wise identifiability of the changing components, we compute Mean Correlation Coefficient (MCC) between \(z_c\) and \(\hat{z}_c\) on a test dataset. MCC is a standard metric in the ICA literature. A higher MCC indicates a higher extent of identifiability and MCC reaches 1 when latent variables are perfectly component-wise identifiable. To measure the block-wise identifiability of the invariant subspace, we follow the experimental practice of the work (von Kügelgen et al., 2021) and compute the \(R^2\) coefficient of determination between \(z_c\) and \(\hat{z}_c\), where \(R^2 = 1\) suggests there is a one-to-one mapping between the two. As IVAE and \(\beta\)-VAE do not specify the invariant and the changing subspace in their estimation. At each of their evaluation, we test all possible partitions of their estimated latent variables and report the one with the highest overall score (Avg.). We repeat each experiment over 3 random seeds.

6.2. Results and Discussion

From Table 1, we can observe that both \(R^2\) and MCC of our method grow along with the number of domains with the peak at \(d = 9\) where both the invariant and the changing part attain high identifiability scores. This confirms our theory and the intuition that a certain number of domains are necessary for identifiability. Notably, we can observe that decent identifiability can be attained with a relatively small number of domains (e.g. 5), which shed light on why iMSDA would work even with only a moderate number of domains. We visualize the the disentanglement in Figure 3.

![Figure 3](image)

The scatter plot for the true and the estimated components from our method trained with 9 domains. The \(y\) (resp. \(x\)) axe of each subplot corresponds to a specific estimated (resp. true) latent variable. We can observe that the changing components can be identified in a component-wise manner (subplots "Estimated S" and "True S") which verifies our Theorem 4.1. The true invariant components can be (partial) identified within its own subspace (subplots "Estimated C" and "True C") while influencing the changing components minimally, adhering to Theorem 4.2.

<table>
<thead>
<tr>
<th>(d)</th>
<th>MED</th>
<th>Ours (d = 3)</th>
<th>Ours (d = 5)</th>
<th>Ours (d = 7)</th>
<th>Ours (d = 9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCC</td>
<td>0.67 (\pm) 0.12</td>
<td>0.90 (\pm) 0.12</td>
<td>0.91 (\pm) 0.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(R^2)</td>
<td>0.59 (\pm) 0.05</td>
<td>0.84 (\pm) 0.05</td>
<td>0.85 (\pm) 0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg.</td>
<td>0.63 (\pm) 0.08</td>
<td>0.87 (\pm) 0.09</td>
<td>0.88 (\pm) 0.08</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Identifiability on synthetic data: \(d\) denotes the number of domains and Avg. stands for the average of MCC and \(R^2\).

7. Experiments on Real-world Data

7.1. Experimental Setup

Datasets We evaluate our proposed iMSDA on two benchmark domain adaptation datasets, namely Office-Home (Venkateswara et al., 2017) and PACS (Li et al., 2017). Please see Appendix A.4 for detailed description. In experiments, we designate one domain as the target domain and use all other domain as source domains.

Baselines We compare our approach with state-of-the-art methods to verify its effectiveness. We compare with Source Only and single-source domain adaptation methods: DANN (Ganin et al., 2016), MCD (Saito et al., 2018), DANN+BSP (Chen et al., 2019). We also compare our method with existing multi-source domain adaptation methods, including

---

3 Our experimental design closely follows the practice employed in prior nonlinear ICA work (Hyvarinen & Morioka, 2016; Hyvarinen et al., 2019).
Table 2. Classification results on PACS. We employ ResNet-18 as our encoder backbone. Most baseline results are taken from (Yang et al., 2020). We choose $\alpha_1 = 0.1$ and $\alpha_2 = 5e - 5$. The latent space is partitioned with $n_s = 4$ and $n = 64$.  

<table>
<thead>
<tr>
<th>Methods</th>
<th>$\rightarrow$ Art</th>
<th>$\rightarrow$ Cartoon</th>
<th>$\rightarrow$ Photo</th>
<th>$\rightarrow$ Sketch</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only (He et al., 2016)</td>
<td>74.9 ± 0.88</td>
<td>72.1 ± 0.75</td>
<td>94.5 ± 0.58</td>
<td>64.7 ± 1.53</td>
<td>76.6</td>
</tr>
<tr>
<td>DANN (Ganin et al., 2016)</td>
<td>81.9 ± 1.13</td>
<td>77.5 ± 1.26</td>
<td>91.8 ± 1.21</td>
<td>74.6 ± 1.03</td>
<td>81.5</td>
</tr>
<tr>
<td>MDAN (Zhao et al., 2018)</td>
<td>79.1 ± 0.36</td>
<td>76.0 ± 0.73</td>
<td>91.4 ± 0.85</td>
<td>72.0 ± 0.80</td>
<td>79.6</td>
</tr>
<tr>
<td>WBN (Mancini et al., 2018)</td>
<td>89.9 ± 0.28</td>
<td>89.7 ± 0.56</td>
<td>97.4 ± 0.84</td>
<td>58.0 ± 1.51</td>
<td>83.8</td>
</tr>
<tr>
<td>MCD (Saito et al., 2018)</td>
<td>88.7 ± 1.01</td>
<td>88.9 ± 1.53</td>
<td>96.4 ± 0.42</td>
<td>73.9 ± 3.94</td>
<td>87.0</td>
</tr>
<tr>
<td>M3SDA (Peng et al., 2019)</td>
<td>89.3 ± 0.42</td>
<td>89.9 ± 1.00</td>
<td>97.3 ± 0.31</td>
<td>76.7 ± 2.86</td>
<td>88.3</td>
</tr>
<tr>
<td>CMSS (Yang et al., 2020)</td>
<td>88.6 ± 0.36</td>
<td>90.4 ± 0.80</td>
<td>96.9 ± 0.27</td>
<td>82.0 ± 0.59</td>
<td>89.5</td>
</tr>
<tr>
<td>LtC-MSDA (Wang et al., 2020)</td>
<td>90.19</td>
<td>90.47</td>
<td>97.23</td>
<td>81.53</td>
<td>89.8</td>
</tr>
<tr>
<td>T-SVDNet (Li et al., 2021)</td>
<td>90.43</td>
<td>90.61</td>
<td><strong>98.50</strong></td>
<td>85.49</td>
<td>91.25</td>
</tr>
<tr>
<td>iMSDA (Ours)</td>
<td><strong>93.75 ± 0.32</strong></td>
<td><strong>92.46 ± 0.23</strong></td>
<td><strong>98.48 ± 0.07</strong></td>
<td><strong>89.22 ± 0.73</strong></td>
<td><strong>93.48</strong></td>
</tr>
</tbody>
</table>

Table 3. Classification results on Office-Home. We employ Resnet-50 as our encoder backbone. Baseline results are taken from (Park & Lee, 2021). We choose $\alpha_1 = 0.1$ and $\alpha_2 = 1e - 4$. The latent space is partitioned with $n_s = 4$ and $n = 128$.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$\rightarrow$ Art</th>
<th>$\rightarrow$ Clipart</th>
<th>$\rightarrow$ Product</th>
<th>$\rightarrow$ Realworld</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Only (He et al., 2016)</td>
<td>64.58 ± 0.68</td>
<td>52.32 ± 0.63</td>
<td>77.63 ± 0.23</td>
<td>80.70 ± 0.81</td>
<td>68.81</td>
</tr>
<tr>
<td>DANN (Ganin et al., 2016)</td>
<td>64.26 ± 0.59</td>
<td>58.01 ± 1.55</td>
<td>76.44 ± 0.47</td>
<td>78.80 ± 0.49</td>
<td>69.38</td>
</tr>
<tr>
<td>DANN+BiSP (Chen et al., 2019)</td>
<td>66.10 ± 0.27</td>
<td>61.03 ± 0.39</td>
<td>78.13 ± 0.31</td>
<td>79.92 ± 0.13</td>
<td>71.29</td>
</tr>
<tr>
<td>DAN (Long et al., 2015)</td>
<td>68.28 ± 0.45</td>
<td>57.92 ± 0.65</td>
<td>78.45 ± 0.05</td>
<td>81.93 ± 0.35</td>
<td>71.64</td>
</tr>
<tr>
<td>MCD (Saito et al., 2018)</td>
<td>67.84 ± 0.38</td>
<td>59.91 ± 0.55</td>
<td>79.21 ± 0.61</td>
<td>80.93 ± 0.18</td>
<td>71.97</td>
</tr>
<tr>
<td>M3SDA (Peng et al., 2019)</td>
<td>66.22 ± 0.52</td>
<td>58.55 ± 0.62</td>
<td>79.45 ± 0.52</td>
<td>81.35 ± 0.19</td>
<td>71.39</td>
</tr>
<tr>
<td>DCTN (Xu et al., 2018)</td>
<td>66.92 ± 0.60</td>
<td>61.82 ± 0.46</td>
<td>79.20 ± 0.58</td>
<td>77.78 ± 0.59</td>
<td>71.43</td>
</tr>
<tr>
<td>MIAN-γ (Park &amp; Lee, 2021)</td>
<td>69.88 ± 0.35</td>
<td><strong>64.20 ± 0.68</strong></td>
<td>80.87 ± 0.37</td>
<td>81.49 ± 0.24</td>
<td>74.11</td>
</tr>
<tr>
<td>iMSDA (Ours)</td>
<td><strong>75.4 ± 0.86</strong></td>
<td>61.4 ± 0.73</td>
<td><strong>83.5 ± 0.22</strong></td>
<td><strong>84.47 ± 0.38</strong></td>
<td><strong>76.19</strong></td>
</tr>
</tbody>
</table>

Table 4. Ablation study on PACS dataset. We study the impact of individual loss terms. Except for the ablated terms, the hyper-parameters are identical to those in Table 2.

<table>
<thead>
<tr>
<th>$\mathcal{L}_{\text{cls}}$</th>
<th>$\mathcal{L}<em>{\text{cls}} + \alpha_1 \mathcal{L}</em>{\text{ent}}$</th>
<th>$\mathcal{L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rightarrow$ P</td>
<td>97.2 ± 0.01</td>
<td>98.4 ± 0.02</td>
</tr>
<tr>
<td>$\rightarrow$ A</td>
<td>81.1 ± 0.73</td>
<td>93.44 ± 0.31</td>
</tr>
<tr>
<td>$\rightarrow$ C</td>
<td>79.2 ± 0.17</td>
<td>89.03 ± 2.4</td>
</tr>
<tr>
<td>$\rightarrow$ S</td>
<td>70.9 ± 0.69</td>
<td>87.29 ± 0.8</td>
</tr>
<tr>
<td>Avg.</td>
<td>82.11</td>
<td>92.01</td>
</tr>
<tr>
<td>$\mathcal{L}_{\text{cls}}$</td>
<td>$\mathcal{L}<em>{\text{cls}} + \alpha_1 \mathcal{L}</em>{\text{ent}}$</td>
<td>$\mathcal{L}$</td>
</tr>
<tr>
<td>$\rightarrow$ Ar</td>
<td>64.57 ± 0.04</td>
<td>74.23 ± 0.66</td>
</tr>
<tr>
<td>$\rightarrow$ Pr</td>
<td>77.77 ± 0.03</td>
<td>82.83 ± 0.12</td>
</tr>
<tr>
<td>$\rightarrow$ Cl</td>
<td>47.17 ± 0.20</td>
<td>61.8 ± 0.1</td>
</tr>
<tr>
<td>$\rightarrow$ Rw</td>
<td>79.03 ± 0.08</td>
<td>84.2 ± 0.14</td>
</tr>
<tr>
<td>Avg.</td>
<td>67.2</td>
<td>75.77</td>
</tr>
</tbody>
</table>

Table 5. Ablation study on Office-Home dataset. We study the impact of individual loss terms. Except for the ablated terms, the hyper-parameters are identical to those in Table 3.

<table>
<thead>
<tr>
<th>$\mathcal{L}_{\text{cls}}$</th>
<th>$\mathcal{L}<em>{\text{cls}} + \alpha_1 \mathcal{L}</em>{\text{ent}}$</th>
<th>$\mathcal{L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rightarrow$ P</td>
<td>98.40 ± 0.1</td>
<td>98.48 ± 0.07</td>
</tr>
<tr>
<td>$\rightarrow$ A</td>
<td>92.15 ± 1.74</td>
<td>93.75 ± 0.32</td>
</tr>
<tr>
<td>$\rightarrow$ C</td>
<td>91.48 ± 1.54</td>
<td>92.46 ± 0.23</td>
</tr>
<tr>
<td>$\rightarrow$ S</td>
<td>86.15 ± 5.2</td>
<td>89.22 ± 0.73</td>
</tr>
<tr>
<td>Avg.</td>
<td>92.05 ± 2.14</td>
<td>93.48 ± 0.34</td>
</tr>
</tbody>
</table>

Table 6. The impact of the changing dimension $n_s$: we test several values of $n_s$ on PACS dataset. The other configurations are identical to those in Table 2.

M3SDA (Peng et al., 2019), CMSS (Yang et al., 2020) LtC-MSDA (Wang et al., 2020), MIAN-γ (Park & Lee, 2021) and T-SVDNet (Li et al., 2021). Note that when using single-source adaptation methods, we pool all sources together as a single source and then apply the methods. We defer implementation details to Appendix A.4.

7.2. Results and Discussion

PACS The results for PACS are presented in Table 2. We can observe that for the majority of the transfer directions, iMSDA outperforms the most competitive baseline by a considerable margin of 1.2% - 3%. For the $\rightarrow$ Phone task where it does not, the performance is within margin of error com-
pared to the strongest algorithm T-SVDNet. Notably, when compared with currently proposed T-SVDNet (Li et al., 2021), iMSDA achieves a significant performance gain on the challenging task →Sketch. To aid understanding, we visualize the learned features in Figure 4 (Appendix A.4), which shows that iMSDA can effective align source and target domains while retaining discriminative structures.

Office-Home The results in Table 3 demonstrate that the superior performance of iMSDA on the Office-Home dataset. iMSDA achieves a margin of roughly 3% over other baselines on 3 out of 4 tasks, with an exception for →Clipart. Although iMSDA loses out to MIAN-γ for the →Clipart task, it is still on par with or superior to other baselines.

7.3. Ablation studies

The impact of individual loss term. Table 4 and Table 5 contain the ablation results for individual loss terms. We can observe that \( \mathcal{L}_{\text{ent}} \) greatly boosts the model performance in both cases (around 10% and 8.5%). This shows the directly enforcing the relation between \( \hat{z}, \hat{z}_s \) and \( y \) is an effective solution, if the change is insignificant over domains. The inclusion of the alignment consistently benefits the model performance in each case, which justifies our theoretical insights and design choice.

The impact of the changing part dimension. Table 6 includes results on various numbers of changing components \( n_s \). We can observe the model performance degrades at both large \( n_s \) (i.e. 8) and small \( n_s \) (i.e. 2) choices. This is coherent with our hypothesis: a large \( n_s \) compromises the minimal change principle, leading to the undesirable consequences as discussed in Section 3; on the contrary, an overly small \( n_s \) could prevent the model from handling the domain change, for lack of changing capacity, which also yields suboptimal performance.

8. Conclusion

It is not uncommon to assume observations of the real-world are generated from high-level latent variables and thus the ill-posedness in the problem of UDA can be reduced to obtaining meaningful reconstructions of the those latent variables and mapping distinct domains to a shared space for classification.

In this work, we show that under reasonable assumptions on the data generating process, as well as leveraging the principle of minimality, we can obtain partial identifiability of the changing and invariant parts of the generating process. In particular, by introducing an high-level invariant latent variable that influences the changing variable and the corresponding label across domains, we show identifiability of the joint distribution \( p_{x,y|u^T} \) for the target domain \( u^T \) with a classifier trained on source domain labels. Our proposed VAE combined with a flow model architecture learns disentangled representations that allows us perform multi-source UDA with state-of-the-art results across various benchmarks.

Acknowledgment We thank the anonymous reviewers for their constructive comments. This work was supported in part by the NSF-Convergence Accelerator Track-D award #2134901 and by the National Institutes of Health (NIH) under Contract R01HL159805, and by a grant from Apple. The NSF or NIH is not responsible for the views reported in this paper.

References


Partial Identifiability for Domain Adaptation


A. Appendix

A.1. Proof of the changing part identifiability

We present the proof of the changing part identifiability (Theorem 4.1) in this section.

Proof. We start from the matched marginal distribution condition (Equation 2) to develop the relation between \( z \) and \( \hat{z} \) as follows: \( \forall u \in U, \)

\[
    p_{\hat{z}|u} = p_{z|u} \iff p_{\hat{g}(\hat{z})|u} = p_{g(z)|u} \iff p_{g^{-1}(\hat{g}(\hat{z})|u)} = p_{\hat{z}|u} |J_{g^{-1}}| = p_{\hat{z}|u} \iff p_{h(z)|u} = p_{x|u},
\]

(12)

where \( \hat{g}^{-1} : Z \to Z \) denotes the estimated invertible generating function, and \( h := g^{-1} \circ \hat{g} \) is the transformation between the true latent variable and the estimated one. \( |J_{g^{-1}}| \) stands for the absolute value of Jacobian matrix determinant of \( g^{-1} \). Note that as both \( \hat{g}^{-1} \) and \( g \) are invertible, \( |J_{g^{-1}}| \neq 0 \) and \( h \) is invertible.

According to the independence relations in the data generating process in Equation 1 and A2, we have

\[
    p_{\hat{z}|u} = \prod_{i=1}^{n} p_{z_i|u}(z_i); \quad p_{\hat{z}|u} = \prod_{i=1}^{n} p_{z_i|u}(\hat{z}_i)
\]

Rewriting with the notation \( q_i := \log p_{z_i|u} \) and \( \hat{q}_i := \log p_{\hat{z}_i|u} \) yields

\[
    \log p_{\hat{z}|u} = \sum_{i=1}^{n} q_i(z_i, u); \quad \log p_{\hat{z}|u} = \sum_{i=1}^{n} \hat{q}_i(\hat{z}_i, u)
\]

Applying the change of variables to Equation 12 yields

\[
    p_{\hat{z}|u} = p_{z|u} \cdot |J_{h^{-1}}| \iff \sum_{i=1}^{n} q_i(z_i, u) + \log |J_h| = \sum_{i=1}^{n} \hat{q}_i(\hat{z}_i, u)
\]

(13)

where \( J_{h^{-1}} \) and \( J_h \) are the Jacobian matrix of the transformation associated with \( h^{-1} \) and \( h \) respectively.

For ease of exposition, we adopt the following notation:

\[
    h'_{i,k} := \frac{\partial z_i}{\partial z_k}, \quad h''_{i,k,q} := \frac{\partial^2 z_i}{\partial z_k \partial z_q}; \\
    \eta'_i(z_i, u) := \frac{\partial q_i(z_i, u)}{\partial z_i}, \quad \eta''_i(z_i, u) := \frac{\partial^2 q_i(z_i, u)}{(\partial z_i)^2}.
\]

Differentiating both sides of Equation 13 twice w.r.t. \( \hat{z}_k \) and \( \hat{z}_q \) where \( k, q \in [n] \) and \( k \neq q \) yields

\[
    \sum_{i=1}^{n} \left( \eta''_i(z_i, u) \cdot h'_{i,k}(h''_{i,q}) + \eta'_i(z_i, u) \cdot h''_{i,k,q} \right) + \frac{\partial^2 \log |J_h|}{\partial \hat{z}_k \partial \hat{z}_q} = 0.
\]

(14)

Therefore, for \( u = (u_0, \ldots, u_{2n_x}) \), we have \( 2n_x + 1 \) such equations. Subtracting each equation corresponding to \( u_1, \ldots, u_{2n_x} \) with the equation corresponding to \( u_0 \) results in \( 2n_x \) equations:

\[
    \sum_{i=n_x+1}^{n} \left( (\eta''_i(z_i, u_j) - \eta''_i(z_i, u_0)) \cdot h'_{i,k}(h''_{i,q}) + (\eta'_i(z_i, u_j) - \eta'_i(z_i, u_0)) \cdot h''_{i,k,q} \right) = 0,
\]

(15)

where \( j = 1, \ldots, 2n_x \). Note that as the content distribution is invariant to domains i.e., \( p_{x|u} = p_{x|u} \). Thus, we have \( \eta''_i(z_i, u_j) = \eta''_i(z_i, u_{j'}) \) and \( \eta'_i(z_i, u_j) = \eta'_i(z_i, u_{j'}) \forall j, j' \). Hence only the style components \( i = n_x + 1, \ldots, n \) remain in the summation of each equation.
As the data generating process of the learned model, we use the conclusion in Step 3, the smooth and bijective properties of \(
h_i(z, p_x|u) = 0\) and \(h_i''(z, p_x|u) = 0\) for \(i = n_c + 1, \ldots, n\) and \(k, q \in [n], k \neq q\).

As \(h(\cdot)\) is smooth over \(Z\), its Jacobian can be written as:

\[
J_h = \begin{bmatrix}
A := \frac{\partial z_c}{\partial z_c} & B := \frac{\partial z_c}{\partial z_c} \\
C := \frac{\partial z_c}{\partial z_c} & D := \frac{\partial z_c}{\partial z_c}
\end{bmatrix}
\]

(16)

Note that \(h_i'(z, p_x|u) \neq 0\) implies that for each \(i = n_c + 1, \ldots, n\), \(h_i'(z)\) is a nonzero entry in each row indexed by \(i = n_c + 1, \ldots, n\) in the Jacobian matrix \(J_h\). Further, the invertibility of \(h(\cdot)\) necessitates \(J_h\) to be full-rank which implies that there is exactly one non-zero component in each row of matrices \(C\) and \(D\).

Since for every \(i \in \{n_c + 1, \ldots, n\}\), \(z_i\) has changing distributions over \(u\) and all \(z_k\)'s for \(i \in \{1, \ldots, n_c\}\) (i.e. \(z_n\)) have invariant distributions over \(u\), we can deduce that \(C = 0\) and the only non-zero entry \(z_k\) must reside in \(D\) with \(k \in \{n_c + 1, \ldots, n\}\). Therefore, for each true variable in the changing part \(z_i, i \in \{n_c + 1, \ldots, n\}\), there is exactly one estimated variable in the changing part \(\hat{z}_k, k \in \{n_c + 1, \ldots, n\}\) such that \(z_i = h_i'(\hat{z}_k)\). Further, because \(J_h\) is full-rank (\(h(\cdot)\) being invertible) and \(C\) is a zero matrix, \(D\) must be full-rank, which implies that \(h_i'(\cdot)\) is invertible for each \(i \in \{n_c + 1, \ldots, n\}\). Thus, the changing components \(z_c\) are identified up to permutation and component-wise invertible transformations.

**A.2. Proof of the invariant part identifiability**

In the section, we present the proof of the invariant part identifiability (Theorem 4.2).

**Proof.** We divide our proof into four steps to aid understanding.

In Step 1, we leverage properties of the generating process (Equation 1) and the marginal distribution matching condition (Equation 2) to express the marginal invariance with the indeterminacy transformation \(\hat{h}: Z \rightarrow Z\) between the estimated and the true latent variable. The introduction of \(\hat{h}(\cdot)\) allows us to formalize the block-identifiability condition.

In Step 2 and Step 3, we show that the estimated content variable \(\hat{z}_c\) does not depend on the true style variable \(z_c\), that is, \(\hat{h}_c(z)\) does not depend on the input \(z_c\). To this end, in Step 2, we derive its equivalent statements which can ease the rest of the proof and avert technical issues (e.g. sets of zero probability measure). In Step 3, we prove the equivalent statement by contradiction. Specifically, we show that if \(\hat{z}_c\) depended on \(z_c\), the invariance derived in Step 1 would break.

In Step 4, we use the conclusion in Step 3, the smooth and bijective properties of \(h(\cdot)\), and the conclusion in Theorem 4.1, to show the invertibility of the indeterminacy function between the content variables, i.e., the mapping \(\hat{z}_c = \hat{h}_c(z_c)\) being invertible.

**Step 1.** As the data generating process of the learned model \((\hat{g}, p_{z_c}, p_{z_c|u})\) (Equation 1) establishes the independence between the generating process \(z_c \sim p_{z_c}\) and \(u\), it follows that for any \(A_{z_c} \subseteq Z_c\),

\[
\mathbb{P}\left\{ \{\hat{g}_{1:n_c}^{-1}(\hat{x}) \in A_{z_c}\}\{u = u_1\} \right\} = \mathbb{P}\left\{ \{\hat{g}_{1:n_c}^{-1}(\hat{x}) \in A_{z_c}\}\{u = u_2\} \right\}, \quad \forall u_1, u_2 \in U
\]

\[
\iff
\mathbb{P}\left\{ \{\hat{x} \in (\hat{g}_{1:n_c})^{-1}(A_{z_c})\}\{u = u_1\} \right\} = \mathbb{P}\left\{ \{\hat{x} \in (\hat{g}_{1:n_c})^{-1}(A_{z_c})\}\{u = u_2\} \right\}, \quad \forall u_1, u_2 \in U
\]

where \(\hat{g}_{1:n_c} : X \rightarrow Z_c\) denotes the estimated transformation from the observation to the content variable and \((\hat{g}_{1:n_c})^{-1}(A_{z_c}) \subseteq X\) is the pre-image set of \(A_{z_c}\), that is, the set of estimated observations \(\hat{x}\) originating from content variables \(z_c\) in \(A_{z_c}\).

4 Note that in this proof, we decorate quantities with \(\hat{}\) to indicate that they are associated with the estimated model \((\hat{g}, p_{z_c}, p_{z_c|u})\).
Partial Identifiability for Domain Adaptation

Because of the matching observation distributions between the estimated model and the true model in Equation 2, the relation in Equation 17 can be extended to observation \( x \) from the true generating process \( (g, p_{z*}, p_{z*}|u) \), i.e.,

\[
\mathbb{P} \left[ \{x \in (\hat{g}_{1:n})^{-1}(A_{z*})\}|\{u = u_1\} \right] = \mathbb{P} \left[ \{x \in (\hat{g}_{1:n})^{-1}(A_{z*})\}|\{u = u_2\} \right]
\]

\[
\iff \mathbb{P} \left[ \{\hat{g}_{1:n}(x) \in A_{z*}\}|\{u = u_1\} \right] = \mathbb{P} \left[ \{\hat{g}_{1:n}(x) \in A_{z*}\}|\{u = u_2\} \right].
\]

(18)

Since \( g \) and \( \hat{g} \) are smooth and injective, there exists a smooth and injective \( \bar{h} = \hat{g}^{-1} \circ g : \mathcal{Z} \to \mathcal{Z} \). We note that by definition \( h = \bar{h}^{-1} \) where \( h \) is introduced in the proof of Theorem 4.1 (Appendix A.1). Expressing \( \bar{h}^{-1} = \hat{h} \circ g \) and \( \hat{h}_c(\cdot) : \mathcal{Z} \to \mathcal{Z}_c \) in Equation 18 yields

\[
\mathbb{P} \left[ \{\bar{h}_c(z) \in A_{z*}\}|\{u = u_1\} \right] = \mathbb{P} \left[ \{\bar{h}_c(z) \in A_{z*}\}|\{u = u_2\} \right],
\]

\[
\iff \mathbb{P} \left[ \{z \in \bar{h}_c^{-1}(A_{z*})\}|\{u = u_1\} \right] = \mathbb{P} \left[ \{z \in \bar{h}_c^{-1}(A_{z*})\}|\{u = u_2\} \right],
\]

\[
\int_{z \in \bar{h}_c^{-1}(A_{z*})} p_{z|u}(z|u_1) \, dz = \int_{z \in \bar{h}_c^{-1}(A_{z*})} p_{z|u}(z|u_2) \, dz,
\]

(19)

where \( \bar{h}_c^{-1}(A_{z*}) = \{z \in \mathcal{Z} : \bar{h}_c(z) \in A_{z*}\} \) is the pre-image of \( A_{z*} \), i.e. those latent variables containing content variables in \( A_{z*} \) after the indeterminacy transformation \( h \).

Based on the generating process in Equations 1, we can re-write Equation 19 as follows: \( \forall A_{z*} \subseteq \mathcal{Z}_c \),

\[
\int_{[z_*, z_0]^\top \in \bar{h}_c^{-1}(A_{z*})} p_{z_c}(z_c) \left( p_{z_*|u}(z_*|u_1) - p_{z_*|u}(z_*|u_2) \right) \, dz_c \, dz_* = 0.
\]

(20)

Step 2. In order to show the block-identifiability of \( z_c \), we would like to prove that \( z_c := \bar{h}_c([z_*^\top, z_0^\top]^\top) \) does not depend on \( z_* \). To this end, we first develop one equivalent statement (i.e. Statement 3 below) and prove it in later step instead. By doing so, we are able to leverage the full-supported density function assumption to avert technical issues.

- Statement 1: \( \bar{h}_c([z_*^\top, z_0^\top]^\top) \) does not depend on \( z_* \).
- Statement 2: \( \forall z_c \in \mathcal{Z}_c \), it follows that \( \bar{h}_c^{-1}(z_c) = B_{z_*} \times \mathcal{Z}_0 \) where \( B_{z_*} \neq \emptyset \) and \( B_{z_*} \subseteq \mathcal{Z}_c \).
- Statement 3: \( \forall z_c \in \mathcal{Z}_c, r \in \mathbb{R}^+ \), it follows that \( \bar{h}_c^{-1}((B_r(z_*))) = B_{z_*}^+ \times \mathcal{Z}_0 \) where \( B_r(z_*):= \{z'_c \in \mathcal{Z}_c : \|z'_c - z_c\|^2 < r\}, B_{z_*}^+ \neq \emptyset \), and \( B_{z_*}^+ \subseteq \mathcal{Z}_c \).

Statement 2 is a mathematical formulation of Statement 1. Statement 3 generalizes singletons \( z_c \) in Statement 2 to open, non-empty balls \( B_r(z_*). \) Later, we use Statement 3 in Step 3 to show the contraction to Equation 20.

Leveraging the continuity of \( \bar{h}_c(\cdot) \), we can show the equivalence between Statement 2 and Statement 3 as follows. We first show that Statement 2 implies Statement 3. \( \forall z_c \in \mathcal{Z}_c, r \in \mathbb{R}^+ \), \( \bar{h}_c^{-1}((B_r(z_*))) = \cup_{z'_c \in B_r(z_*)} \bar{h}_c^{-1}(z'_c). \) Statement 2 indicates that every participating sets in the union satisfies \( h_c^{-1}(z'_c) = B_{z_*}^+ \times \mathcal{Z}_0 \), thus the union \( \bar{h}_c^{-1}((B_r(z_*))) \) also satisfies this property, which is Statement 3.

Then, we show that Statement 3 implies Statement 2 by contradiction. Suppose that Statement 2 is false, then \( \exists \hat{z}_c \in \mathcal{Z}_c \) such that there exist \( \hat{z}_c^B \in \{z_{1:n_c} : z \in \bar{h}_c^{-1}(\hat{z}_c)\} \) and \( \hat{z}_c^B \in \mathcal{Z}_c \) resulting in \( \bar{h}_c(\hat{z}_c^B) \neq \hat{z}_c \) where \( \hat{z}_c^B = [\hat{z}_c^B]^\top, (\hat{z}_c^B)^\top \). As \( \bar{h}_c(\cdot) \) is continuous, there exists \( \hat{r} \in \mathbb{R}^+ \) such that \( \bar{h}_c(\hat{z}_c^B) \notin B_{\hat{r}}(z_c) \). That is, \( \hat{z}_c^B \notin h_c^{-1}(B_r(z_c)) \). Also, Statement 3 suggests that \( h_c^{-1}(B_r(z_c)) = B_{z_*}^+ \times \mathcal{Z}_0 \). By definition of \( \hat{z}_c^B \), it is clear that \( z_{1:n_c}^B \in B_{z_*}^+ \). The fact that \( \hat{z}_c^B \notin h_c^{-1}(B_r(z_c)) \) contradicts Statement 3. Therefore, Statement 2 is true under the premise of Statement 3. We have shown that Statement 3 implies Statement 2. In summary, Statement 2 and Statement 3 are equivalent, and therefore proving Statement 3 suffices to show Statement 1.
Step 3. In this step, we prove Statement 3 by contradiction. Intuitively, we show that if $\tilde{h}_c(\cdot)$ depended on $\tilde{z}_s$, the preimage $\tilde{h}_c^{-1}(B_r(z_s))$ could be partitioned into two parts (i.e., $B_r^*$ and $\tilde{h}_c^{-1}(A_{z_s}^*) \setminus B_r^*$ defined below). The dependency between $\tilde{h}_c(\cdot)$ and $\tilde{z}_s$ is captured by $B_r^*$, which would not emerge otherwise. In contrast, $\tilde{h}_c^{-1}(A_{z_s}^*) \setminus B_r^*$ also exists when $\tilde{h}_c(\cdot)$ does not depend on $\tilde{z}_s$. We evaluate the invariance relation Equation 20 and show that the integral over $\tilde{h}_c^{-1}(A_{z_s}^*) \setminus B_r^*$ (i.e., $T_1$) is always 0, however, the integral over $B_r^*$ (i.e., $T_2$) is necessarily non-zero, which leads to the contradiction with Equation 20 and thus shows the $\tilde{h}_c(\cdot)$ cannot depend on $\tilde{z}_s$.

First, note that because $B_r(z_s)$ is open and $\tilde{h}_c(\cdot)$ is continuous, the pre-image $\tilde{h}_c^{-1}(B_r(z_s))$ is open. In addition, the continuity of $\tilde{h}(\cdot)$ and the matched observation distributions $\forall u' \in U$, $\mathbb{P}\left(\{x \in A_x\} | u = u'\right) = \mathbb{P}\left(\{\tilde{x} \in A_x\} | u = u'\right)$ lead to $\tilde{h}(\cdot)$ being bijection as shown in (Klindt et al., 2020), which implies that $\tilde{h}_c^{-1}(B_r(z_s))$ is non-empty. Hence, $\tilde{h}_c^{-1}(B_r(z_s))$ is both non-empty and open. Suppose that $\exists A_{z_s}^* = B_r(z_s)$ where $z_s \in Z_c$, $r^* \in \mathbb{R}^+$, such that $B_r^* := \{z \in Z : z \in \tilde{h}_c^{-1}(A_{z_s}^*), \{z_{1:n}\} \times Z_s \not\subseteq \tilde{h}_c^{-1}(A_{z_s}^*) \not= \emptyset \}$ Intuitively, $B_r^*$ contains the partition of the pre-image $\tilde{h}_c^{-1}(A_{z_s}^*)$ that the style part $z_{n+1:n}$ cannot take on any value in $Z_s$. Only certain values of the style part were able to produce specific outputs of indeterminacy $\tilde{h}_c(\cdot)$. Clearly, this would suggest that $\tilde{h}_c(\cdot)$ depends on $z_s$.

To show contraction with Equation 20, we evaluate the LHS of Equation 20 with such a $A_{z_s}^*$:

$$
\int_{\left[z_1^*, z_2^*\right] \in \tilde{h}_c^{-1}(A_{z_s}^*)} p_{z_c}(z_c) \left(p_{z_s|u}(z_s|u_1) - p_{z_s|u}(z_s|u_2)\right) dz_s dz_c
$$

$$
= \int_{\left[z_1^*, z_2^*\right] \in \tilde{h}_c^{-1}(A_{z_s}^*) \setminus B_r^*} p_{z_c}(z_c) \left(p_{z_s|u}(z_s|u_1) - p_{z_s|u}(z_s|u_2)\right) dz_s dz_c
$$

$$
+ \int_{\left[z_1^*, z_2^*\right] \in B_r^*} p_{z_c}(z_c) \left(p_{z_s|u}(z_s|u_1) - p_{z_s|u}(z_s|u_2)\right) dz_s dz_c.
$$

We first look at the value of $T_1$. When $\tilde{h}_c^{-1}(A_{z_s}^*) \setminus B_r^* = \emptyset$, $T_1$ evaluates to 0. Otherwise, by definition we can rewrite $\tilde{h}_c^{-1}(A_{z_s}^*) \setminus B_r^*$ as $C_{z_c} \times Z_s$ where $C_{z_c} \neq \emptyset$ and $C_{z_c} \subseteq Z_c$. With this expression, it follows that

$$
\int_{\left[z_1^*, z_2^*\right] \in C_{z_c}} p_{z_c}(z_c) \left(p_{z_s|u}(z_s|u_1) - p_{z_s|u}(z_s|u_2)\right) dz_s dz_c
$$

$$
= \int_{z_c \in C_{z_c}} p_{z_c}(z_c) \int_{z_s \in Z_s} \left(p_{z_s|u}(z_s|u_1) - p_{z_s|u}(z_s|u_2)\right) dz_s dz_c
$$

$$
= \int_{z_c \in C_{z_c}} p_{z_c}(z_c) (1 - 1) dz_c = 0.
$$

Therefore, in both cases $T_1$ evaluates to 0 for $A_{z_s}^*$.

Now, we address $T_2$. As discussed above, $\tilde{h}_c^{-1}(A_{z_s}^*)$ is open and non-empty. Because of the continuity of $\tilde{h}_c(\cdot)$, $\forall z_B \in B_r^*$, there exists $r(z_B) \in \mathbb{R}^+$ such that $B_r(z_B) \subseteq B_r^*$. As $p_{z_s|u}(z|u) > 0$ over $(z, u)$, we have $\mathbb{P}\left(\{z \in B_r^*\} | u = u'\right) \geq \mathbb{P}\left(\{z \in B_r(z_B)\} | u = u'\right) > 0$ for any $u' \in U$. Assumption A4 indicates that $\exists u_1^*, u_2^*$, such that

$$
T_2 := \int_{\left[z_1^*, z_2^*\right] \in B_r^*} p_{z_c}(z_c) \left(p_{z_s|u}(z_s|u_1^*) - p_{z_s|u}(z_s|u_2^*)\right) dz_s dz_c \neq 0.
$$

Therefore, for such $A_{z_s}^*$, we would have $T_1 + T_2 \neq 0$ which leads to contradiction with Equation 20. We have proved by contradiction that Statement 3 is true and hence Statement 1 holds, that is, $\tilde{h}_c(\cdot)$ does not depend on the style variable $z_s$.

Step 4. With the knowledge that $\tilde{h}_c(\cdot)$ does not depend on the style variable $z_s$, we now show that there exists an invertible mapping between the true content variable $z_c$ and the estimated version $\tilde{z}_c$.
As \( \tilde{h}(\cdot) \) is smooth over \( Z \), its Jacobian can be written as:

\[
J_{\tilde{h}} = \begin{bmatrix}
A \coloneqq \frac{\partial \hat{z}_c}{\partial c} & B \coloneqq \frac{\partial \hat{z}_c}{\partial s} \\
C \coloneqq \frac{\partial \hat{z}_s}{\partial c} & D \coloneqq \frac{\partial \hat{z}_s}{\partial s}
\end{bmatrix}
\]

(21)

where we use notation \( \hat{z}_c = \tilde{h}(z)_{1:n_c} \) and \( \hat{z}_s = \tilde{h}(z)_{n_c+1:n} \). As we have shown that \( \hat{z}_c \) does not depend on the style variable \( z_s \), it follows \( B = 0 \). On the other hand, as \( \tilde{h}(\cdot) \) is invertible over \( Z \), \( J_{\tilde{h}} \) is non-singular. Therefore, \( A \) must be non-singular due to \( B = 0 \). We note that \( A \) is the Jacobian of the function \( \tilde{h}'_c(z_c) := \tilde{h}_c(z_c) : Z_c \to Z_c \), which takes only the content part \( z_c \) of the input \( z \) into \( h_c \). Also, we note that \( C = 0 \), because of the invertible mapping between \( z_s \) and \( \hat{z}_s \) shown in Theorem 4.1. Together with the invertibility of \( \tilde{h} \), we can conclude that \( \tilde{h}'_c \) is invertible. Therefore, there exists an invertible function \( \tilde{h}'_c \) between the estimated and the true content variables such that \( \hat{z}_c = \tilde{h}'_c(z_c) \), which concludes the proof that \( z_c \) is block-identifiable via \( \tilde{y}^{-1}(\cdot) \).

\[ \square \]

### A.3. Synthetic Data Experiments

We provide additional details of our synthetic experiments in Section 6.

**Architecture** For all methods in our synthetic data experiments, the VAE encoder and decoder are 6-layer MLP’s with a hidden dimension of 32 and Leaky-ReLU (\( \alpha = 0.2 \)) activation functions. For our method, we use component-wise spline flows (Durkan et al., 2019) with monotonic linear rational splines to modulate the change components. We use 8 bins for the linear splines and set the bound to be 5. The \( \beta \)-VAE shares the same VAE model with ours and the iVAE implementation is adopted from (Khemakhem et al., 2020b).

**Training hyper-parameters** We apply AdamW to train VAE and flow models for 100 epochs. We use a learning rate of 0.002 with a batch size of 128. The weight decay parameter of AdamW is set to 0.0001. For VAE training, we set the \( \beta \) parameter of KL loss term to 0.1.

### A.4. Real-world Data Experiments

We provide implementation details and additional visualization of real-world data experiments in Section 7.

**Dataset description** PACS (Li et al., 2017) is a multi-domain dataset containing 9991 images from 4 domains of different styles: Photo, Artpainting, Cartoon, Sketch. These domains share the same seven categories. Office-Home (Venkateswara et al., 2017) dataset consists of 4 domains, with each domain containing images from 65 categories of everyday objects and a total of around 15,500 images. The domains include 1) Art: artistic depictions of objects; 2) Clipart: collection of clipart images; 3) Product: images of objects without a background; 4) Real-World: images of objects captured with a regular camera.

**Implementation Details** For the PACS dataset, we follow the protocols in (Yang et al., 2020; Wang et al., 2020) and use ResNet-18 pre-trained on ImageNet. For the Office-Home dataset, we follow the protocols in (Park & Lee, 2021) and use pre-trained ResNet-50 as our backbone network. For two datasets, we use SGD with Nesterov momentum with learning rate 0.01. Since it can be challenging to train VAE on high-resolution images, we use extracted features as our VAE input. In particular, we use the features after the last pooling layer in the ResNet as our VAE input. We use 2-layer fully connected network as the VAE encoder. For the flow model, we use Deep Sigmoidal Flow (DSF) (Huang et al., 2018) across all our experiments. For hyper-parameters, we fix \( \alpha_1 = 0.1 \) for all our experiments and select \( \alpha_2 \) in \([1e-5, 5e-5, 1e-4, 5e-4] \).

**Feature Quality** In addition, we visualize the learned features by our method in Figure 4 and make comparison with the Source Only baseline. We can observe that, for the baseline the features in different classes are mixed, which renders the classification task significantly harder. In contrast, the features learned by iMSDA are more clustered and discriminative.
Figure 4. The t-SNE visualizations of learned features on the Sketch task in PACS. Red: source domains, Blue: target domain.