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 $\mathbf{x}$, target variables $\mathbf{y}$, and domain indices $\mathbf{u}$, the training (source domain) data follows multiple joint distributions $p_{\mathbf{x},\mathbf{y}|\mathbf{u}_1}$, $p_{\mathbf{x},\mathbf{y}|\mathbf{u}_2}$, ..., $p_{\mathbf{x},\mathbf{y}|\mathbf{u}_M}$,[1] and the test (target domain) data follows the joint distribution $p_{\mathbf{x},\mathbf{y}|\mathbf{u}^\mathcal{T}}$, where $p_{\mathbf{x},\mathbf{y}|\mathbf{u}}$ may vary across $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_M$. During training, for each i-th source domain, we are given labeled observations $(\mathbf{x}_k^{(i)}, \mathbf{y}_k^{(i)})_{k=1}^{m_i}$ from $p_{\mathbf{x},\mathbf{y}|\mathbf{u}_i}$, and target domain unlabeled instances $(\mathbf{x}_k^\mathcal{T})_{k=1}^{m_\mathcal{T}}$ from $p_{\mathbf{x},\mathbf{y}|\mathbf{u}^\mathcal{T}}$. 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 $p_{\mathbf{x}|\mathbf{u}^\mathcal{T}}$, the data may correspond to infinitely many joint distributions $p_{\mathbf{x},\mathbf{y}|\mathbf{u}^\mathcal{T}}$. 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 $p_{\mathbf{x},\mathbf{y}|\mathbf{u}^\mathcal{T}}$. 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 $p_{\mathbf{y}|\mathbf{x}}$, whereas $p_{\mathbf{x}|\mathbf{u}}$ 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 $p_{\mathbf{y}|\mathbf{u}}$ or one or more linear transformations of the features $\mathbf{x}$ (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 is $Y \to X$, then $p_{\mathbf{y}|\mathbf{u}}$ and $p_{\mathbf{x}|\mathbf{y},\mathbf{u}}$ 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

[1]Carnegie Mellon University, USA [2]Mohamed bin Zayed University of Artificial Intelligence, UAE [3]Broad Institute of MIT and Harvard, USA. Correspondence to: Kun Zhang <kunz1@cmu.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1] With a slight abuse of notation, we use $p_{\mathbf{x},\mathbf{y}|\mathbf{u}'}$ to represent the conditional distribution for a *specific* domain $\mathbf{u}'$, i.e. $p_{\mathbf{x},\mathbf{y}|\mathbf{u}}(\mathbf{x},\mathbf{y}|\mathbf{u}')$.

of the distribution $p_{\mathbf{x}|\mathbf{y},\mathbf{u}}$ 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 $\mathbf{z}$, such that $p_{\mathbf{z}|\mathbf{u}_1} = p_{\mathbf{z}|\mathbf{u}_2} = ... = p_{\mathbf{z}|\mathbf{u}_M} = p_{\mathbf{z}|\mathbf{u}^\tau}$ (i.e. they are *marginally invariant*), while at the same time training a classifier $f_{\text{cls}} : \mathcal{Z} \to \mathcal{Y}$ on the labeled source domain data, which ensures that $\mathbf{z}$ has predictive information about $\mathbf{y}$ (Ganin & Lempitsky, 2014; Ben-David et al., 2010; Zhao et al., 2018). This could still mean that the joint distributions $p_{\mathbf{z},\mathbf{y}|\mathbf{u}}$ may be different across domains $\mathbf{u}_1$, ..., $\mathbf{u}_M$, 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 $p_{\mathbf{x},\mathbf{y}|\mathbf{u}^\tau}$.

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 $\mathbf{u}$'s. In fact, one can make reasonable assumptions about the role of the latent representation $\mathbf{z}$ in the data-generating process, and represent the minimal change of $p_{\mathbf{x},\mathbf{y}|\mathbf{u}}$ across domains such that both $\mathbf{z}$ and $p_{\mathbf{x},\mathbf{y}|\mathbf{u}}$ are identifiable in the target domain.

In this paper, we represent the variables $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ in terms of a data-generating process assumption. To enforce *minimal change* of $p_{\mathbf{x},\mathbf{y},\mathbf{z}|\mathbf{u}}$ 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 $\mathbf{z}$ contains meaningful information about the target domain. For instance, in (Bousmalis et al., 2016), $\mathbf{z}$ was divided into a changing and an invariant part, and a constraint was added that $\mathbf{z}$ has to be able to reconstruct the original data, in addition to predicting $\mathbf{y}$ (from the shared part). Furthermore, pseudo-labels have also been utilized in order to match the joint distribution $p_{\mathbf{z},\mathbf{y}}$ across domains, using kernel methods (Long et al., 2017a;b). Another approach to ensure more structure in $\mathbf{z}$ is to assume that the class-conditional distribution $p_{\mathbf{x}|\mathbf{y}}$ 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, $p_{\mathbf{y}}$ 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; Hälvä & Hyvarinen, 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

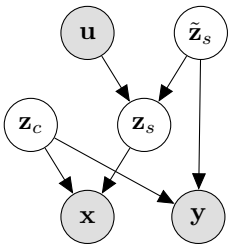


Figure 1. **The generating process**: The gray shade of nodes indicates that the variable is observable.

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 $\mathbf{z}$, even though there may be parts of $\mathbf{z}$ that are not conditionally invariant, and yet still relevant for predicting $\mathbf{y}$.

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 $\mathbf{z}$ and we allow both the changing and invariant parts to have predictive information about $\mathbf{y}$. 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.

$$\mathbf{z}_c \sim p_{\mathbf{z}_c},\ \tilde{\mathbf{z}}_s \sim p_{\tilde{\mathbf{z}}_s},\ \mathbf{z}_s = f_{\mathbf{u}}(\tilde{\mathbf{z}}_s),\ \mathbf{x} = g(\mathbf{z}_c, \mathbf{z}_s). \quad (1)$$

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

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

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

Further, we assume that $\mathbf{y}$ is generated by invariant latent variables $\mathbf{z}_c$ and $\tilde{\mathbf{z}}_s$. Thus, this generating process addresses the conditional-shift setting, in which $p_{\mathbf{x}|\mathbf{y},\mathbf{u}}$ changes across domains, and $p_{\mathbf{y}|\mathbf{u}} = p_{\mathbf{y}}$ 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 $\mathbf{z}_s$ allows us to preserve the information of the changing part $\mathbf{z}_s$ 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 $\mathbf{y}$. In contrast, our goal is to disentangle the changing and invariant parts $\mathbf{z}_s$ and $\mathbf{z}_c$, capture the relationship between $\mathbf{z}_s$ and $\mathbf{y}$ across domains by learning $f_{\mathbf{u}}$, 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 $(\mathbf{z}_c, \tilde{\mathbf{z}}_s)$ 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 $\tilde{\mathbf{z}}_s$ We note that presence of $\tilde{\mathbf{z}}_s$ is significant, as it allows us to provably learn an optimal classifier over domains without requiring that $p_{\mathbf{z}|\mathbf{u}}$ 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 $f_{\mathbf{u}}$, we are able to identify $\tilde{\mathbf{z}}_s$ through $\mathbf{z}_s$ which is critical to our ability to learn how the changes across domains are related to each other. If we know $f_{\mathbf{u}}$ and $\mathbf{z}_s$, we can always map all domains to this high-level $\tilde{\mathbf{z}}_s$, which will have the desired information to predict $\mathbf{y}$ (in addition to $\mathbf{z}_c$). 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. $\mathbf{z}_c$, $\tilde{\mathbf{z}}_s$) with only observational data $(\mathbf{x}, \mathbf{u})$. Because if this is the case, we can leverage labeled data in source domains to learn a classifier according to $p_{\mathbf{y}|\tilde{\mathbf{z}}_s,\mathbf{z}_c}$ 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 $(\mathbf{x}, \mathbf{u})$ 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 $\mathbf{z}_s = f_{\mathbf{u}}(\tilde{\mathbf{z}}_s)$ with a component-wise monotonic function for each domain $\mathbf{u}$, and learn a model $(\hat{g}, p_{\hat{\mathbf{z}}_c}, p_{\hat{\mathbf{z}}_s|\mathbf{u}})$ [2] that assumes the identical process as in Equation 1 and matches the true marginal data distribution in all domains, that is,

$$p_{\mathbf{x}|\mathbf{u}}(\mathbf{x}'|\mathbf{u}') = p_{\hat{\mathbf{x}}|\mathbf{u}}(\mathbf{x}'|\mathbf{u}'), \quad \forall \mathbf{x}' \in \mathcal{X}, \mathbf{u}' \in \mathcal{U}, \tag{2}$$

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

In the following, we show that our estimated model can recover the true changing components and the invariant subspace $\mathbf{z}_c$, 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 $z_{s,i}$, there exists a corresponding estimated changing component $\hat{z}_{s,j}$ and an invertible function $h_{s,i} : \mathbb{R} \to \mathbb{R}$, such that $\hat{z}_{s,j} = h_{s,i}(z_{s,i})$. For ease of exposition, we assume that the $\mathbf{z}_c$ and $\mathbf{z}_s$ correspond to components in $\mathbf{z}$ with indices $\{1, \ldots, n_c\}$ and $\{n_c + 1, \ldots, n\}$ respectively, that is, $\mathbf{z}_c = (z_i)_{i=1}^{n_c}$ and $\mathbf{z}_s = (z_i)_{i=n_c+1}^{n}$.

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. $p_{\mathbf{z}|\mathbf{u}}$ is smooth and $p_{\mathbf{z}|\mathbf{u}} > 0$ over $\mathcal{Z}$ and $\mathcal{U}$.*
- *A2 (Conditional independence): Conditioned on $\mathbf{u}$, each z_i is independent of any other z_j for $i, j \in [n]$, $i \neq j$, i.e. $\log p_{\mathbf{z}|\mathbf{u}}(\mathbf{z}|\mathbf{u}) = \sum_i^n q_i(z_i, \mathbf{u})$ where q_i is the log density of the conditional distribution, i.e., $q_i := \log p_{z_i|\mathbf{u}}$.*
- *A3 (Linear independence): For any $\mathbf{z}_s \in \mathcal{Z}_s \subseteq \mathbb{R}^{n_s}$, there exist $2n_s + 1$ values of $\mathbf{u}$, i.e., $\mathbf{u}_j$ with $j = 0, 1, \ldots, 2n_s$, such that the $2n_s$ vectors $\mathbf{w}(\mathbf{z}_s, \mathbf{u}_j) - \mathbf{w}(\mathbf{z}_s, \mathbf{u}_0)$ with $j = 1, ..., 2n_s$, are linearly independent, where vector $\mathbf{w}(\mathbf{z}_s, \mathbf{u})$ is defined as follows:*

$$\mathbf{w}(\mathbf{z}_s, \mathbf{u}) = \Big(\frac{\partial q_{n_c+1}(z_{n_c+1}, \mathbf{u})}{\partial z_{n_c+1}}, \ldots, \frac{\partial q_n(z_n, \mathbf{u})}{\partial z_n}, \frac{\partial^2 q_{n_c+1}(z_{n_c+1}, \mathbf{u})}{\partial z_{n_c+1}^2}, \ldots, \frac{\partial^2 q_n(z_n, \mathbf{u})}{\partial z_n^2} \Big). \tag{3}$$

By learning $(\hat{g}, p_{\hat{\mathbf{z}}_c}, p_{\hat{\mathbf{z}}_s|\mathbf{u}})$ to achieve Equation 2, $\mathbf{z}_s$ 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 $p_{\mathbf{z}_s|\mathbf{u}}$ varies sufficiently over domain $\mathbf{u}$'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 $\mathbf{z}_s$ and the corresponding domain influence $f_{\mathbf{u}}$, can we correctly align various domains with $\tilde{\mathbf{z}}_s$ 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 $h'_c : \mathcal{Z}_c \to \mathcal{Z}_c$ between the estimated invariant part $\hat{\mathbf{z}}_c$ and the true invariant part $\mathbf{z}_c$, s.t. $\hat{\mathbf{z}}_c = h'_c(\mathbf{z}_c)$. 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 $A_{\mathbf{z}} \subseteq \mathcal{Z}$ with the following two properties:*
  - *a) $A_{\mathbf{z}}$ has nonzero probability measure, i.e. $\mathbb{P}\left[\{\mathbf{z} \in A_{\mathbf{z}}\} | \{\mathbf{u} = \mathbf{u}'\}\right] > 0$ for any $\mathbf{u}' \in \mathcal{U}$.*
  - *b) $A_{\mathbf{z}}$ cannot be expressed as $B_{\mathbf{z}_c} \times \mathcal{Z}_s$ for any $B_{\mathbf{z}_c} \subset \mathcal{Z}_c$.*

  $\exists \mathbf{u}_1, \mathbf{u}_2 \in \mathcal{U}$, such that

$$\int_{\mathbf{z} \in A_{\mathbf{z}}} p_{\mathbf{z}|\mathbf{u}}(\mathbf{z}|\mathbf{u}_1) d\mathbf{z} \neq \int_{\mathbf{z} \in A_{\mathbf{z}}} p_{\mathbf{z}|\mathbf{u}}(\mathbf{z}|\mathbf{u}_2) d\mathbf{z}.$$

With the model $(\hat{g}, p_{\hat{\mathbf{z}}_c}, p_{\hat{\mathbf{z}}_s|\mathbf{u}})$ estimated by observing Equation 2, the $\mathbf{z}_c$ is block-wise identifiable.

A proof is deferred to Appendix A.2.

Like Assumption A3, Assumption A4 also requires the distribution $p_{\mathbf{z}|\mathbf{u}}$ to change sufficiently over domains. Intuitively, the chance of having one such subset $A_{\mathbf{z}}$ 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 $p_{\mathbf{z}_c}$ 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 $p_{\mathbf{x},\mathbf{y}|\mathbf{u}^{\mathcal{T}}}$ for the target domain $\mathbf{u}^{\mathcal{T}}$ 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 $\mathbf{z}_c, \mathbf{z}_s$ from observation $\mathbf{x}$ in any domain, that is, we can partially identify the joint distribution $p_{\mathbf{x},\mathbf{z}_c,\mathbf{z}_s|\mathbf{u}}$. As $\mathbf{z}_s = f_{\mathbf{u}}(\tilde{\mathbf{z}}_s)$ and $f_{\mathbf{u}}$ is constrained to be component-wise monotonic and invertible, we can further identify $p_{\mathbf{x},\mathbf{z}_c,\tilde{\mathbf{z}}_s|\mathbf{u}}$. Also note that the label $\mathbf{y}$ is conditionally independent of $\mathbf{x}$ and especially $\mathbf{u}$ given $\mathbf{z}_c$ and $\tilde{\mathbf{z}}_s$ as shown in Figure 1, which means that $\mathbf{z}_c$ and $\tilde{\mathbf{z}}_s$ capture all the information in $\mathbf{x}$ that is relevant to $\mathbf{y}$. Therefore, given $\mathbf{z}_c, \tilde{\mathbf{z}}_s$ we can make prediction in the target domain with the predictor $p_{\mathbf{y}|\mathbf{z}_c,\tilde{\mathbf{z}}_s}$, and this predictor can be learned on source domains, as $p_{\mathbf{x},\mathbf{z}_c,\mathbf{z}_s|\mathbf{u}}$ is identifiable as discussed above. This is exactly the goal of UDA. In fact, we can identify the joint distribution $p_{\mathbf{x},\mathbf{y}|\mathbf{u}^{\mathcal{T}}}$ - with the identified $p_{\mathbf{x},\mathbf{z}_c,\tilde{\mathbf{z}}_s|\mathbf{u}^{\mathcal{T}}}$ for the target domain, we can derive $p_{\mathbf{x},\mathbf{y}|\mathbf{u}^{\mathcal{T}}} = \int_{\mathbf{z}_c} \int_{\tilde{\mathbf{z}}_s} p_{\mathbf{x},\mathbf{z}_c,\tilde{\mathbf{z}}_s|\mathbf{u}^{\mathcal{T}}} \cdot p_{\mathbf{y}|\mathbf{z}_c,\tilde{\mathbf{z}}_s} d\mathbf{z}_c d\tilde{\mathbf{z}}_s$.

The intuition is that by resorting to the high-level invariant part $\tilde{\mathbf{z}}_s$ 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** (**i**dentifiable **M**ulti-**S**ource **D**omain **A**daptation), 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

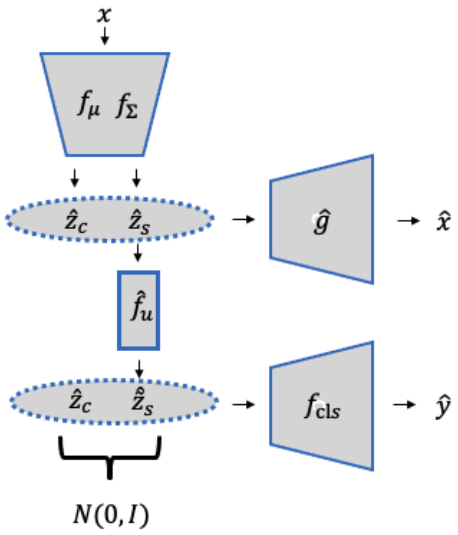


Figure 2. Diagram of our proposed method, **iMSDA**. We first apply the VAE encoder (f_μ, f_Σ) to encode $\mathbf{x}$ into $(\hat{\mathbf{z}}_c, \hat{\mathbf{z}}_s)$, which is further fed into the decoder $\hat{g}$ for reconstruction. In parallel, the changing part $\hat{\mathbf{z}}_s$ is passed through the flow model $f_\mathbf{u}$ to recover the high-level invariant variable $\hat{\tilde{\mathbf{z}}}_s$. We use $(\hat{\mathbf{z}}_c, \hat{\tilde{\mathbf{z}}}_s)$ for classification with the classifier f_{cls} and for matching $\mathcal{N}(\mathbf{0}, \mathbf{I})$ with a KL loss.

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

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

$$\hat{\mathbf{z}} \sim q_{f_\mu,f_\Sigma}(\hat{\mathbf{z}}|\mathbf{x}) := \mathcal{N}(f_\mu(\mathbf{x}), f_\Sigma(\mathbf{x})), \tag{4}$$

$$\hat{\mathbf{x}} = \hat{g}(\hat{\mathbf{z}}). \tag{5}$$

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

$$\hat{\tilde{\mathbf{z}}}_s = \hat{f}_\mathbf{u}^{-1}(\hat{\mathbf{z}}_s). \tag{6}$$

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

$$\mathcal{L}_{\text{VAE}}(f_\mu, f_\Sigma, \hat{f}_\mathbf{u}, \hat{g}) = \mathbb{E}_{(\mathbf{x},\mathbf{u})}\, \mathbb{E}_{\hat{\mathbf{z}}\sim q_{f_\mu,f_\Sigma}} \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \beta \cdot D(q_{f_\mu,f_\Sigma,\hat{f}_\mathbf{u}}(\hat{\tilde{\mathbf{z}}}|\mathbf{x})||p(\tilde{\mathbf{z}})), \tag{7}$$

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

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

$$\hat{\mathbf{y}} = f_{\text{cls}}(\hat{\mathbf{z}}_c, \hat{\tilde{\mathbf{z}}}_s), \tag{8}$$

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

$$\mathcal{L}_{\text{cls}}(f_{\text{cls}}, f_\mu, f_\Sigma, \hat{f}_\mathbf{u}, \hat{g}) = -\mathbb{E}_{\hat{\mathbf{z}}_c,\hat{\mathbf{z}}_s,\mathbf{y}}\Big[\mathbf{y} \cdot \log(\hat{\mathbf{y}})\Big], \tag{9}$$

where $\mathbf{y}$ 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 $\mathbf{I}((\hat{\mathbf{z}}_c, \hat{\tilde{\mathbf{z}}}_s); \hat{\mathbf{y}})$ which amounts to minimizing the conditional entropy $\mathbf{H}(\hat{\mathbf{y}}|\hat{\mathbf{z}}_c, \hat{\tilde{\mathbf{z}}}_s)$ as in (Wang et al., 2020; Li et al., 2021). Doing so facilitates learning a latent representation space adhering to the generating process.

$$\mathcal{L}_{\text{ent}}(f_{\text{cls}}, f_\mu, f_\Sigma, \hat{f}_\mathbf{u}, \hat{g}) = -\mathbb{E}_{\hat{\mathbf{z}}_c,\hat{\mathbf{z}}_s}\Big[\hat{\mathbf{y}} \cdot \log(\hat{\mathbf{y}})\Big]. \tag{10}$$

Algorithm 1 Training **iMSDA**

Require: $(\mathbf{x}, \mathbf{y}, \mathbf{u})$ from source domains and $(\mathbf{x}, \mathbf{u})$ from the target domain.
Require: f_μ, f_Σ, $\hat{g}$, $\hat{f}_\mathbf{u}$, and f_{cls}.
1. get the latent representation $\hat{\mathbf{z}} \sim q_{f_\mu,f_\Sigma}(\hat{\mathbf{z}}|\mathbf{x})$ in Equation 4;
2. reconstruct the observation $\hat{\mathbf{x}} = \hat{g}(\hat{\mathbf{z}})$ in Equation 5;
3. recover the high-level invariant part $\hat{\tilde{\mathbf{z}}}_s = \hat{f}_\mathbf{u}^{-1}(\hat{\mathbf{z}})$ in Equation 6;
4. estimate the class label $\hat{\mathbf{y}} = f_{\text{cls}}(\hat{\mathbf{z}}_c, \hat{\tilde{\mathbf{z}}}_s)$ in Equation 8;
5. compute and minimize $\mathcal{L}$ according to Equation 7, Equation 10, Equation 9, and Equation 11.

Overall, our loss function is

$$\mathcal{L}(f_{\text{cls}}, \hat{f}_\mu, f_\Sigma, f_{\mathbf{u}}, \hat{g}) = \mathcal{L}_{\text{cls}} + \alpha_1 \mathcal{L}_{\text{ent}} + \alpha_2 \mathcal{L}_{\text{VAE}}, \quad (11)$$

where α_1, α_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 $\mathbf{x}$ in Equation 1. We work with latent variables $\mathbf{z}$ of 4 dimensions with $n_c = n_s = 2$. We sample $\mathbf{z}_c \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{z}_s \sim \mathcal{N}(\mu_{\mathbf{u}}, \sigma^2_{\mathbf{u}}\mathbf{I})$ where for each domain $\mathbf{u}$ we sample $\mu_{\mathbf{u}} \sim \text{Unif}(-4, 4)$ and $\sigma^2_{\mathbf{u}} \sim \text{Unif}(0.01, 1)$. We choose g to be a 2-layer MLP with the Leaky-ReLU activation function [3].

Evaluation metrics To measure the component-wise identifiability of the changing components, we compute Mean Correlation Coefficient (MCC) between $\mathbf{z}_s$ and $\hat{\mathbf{z}}_s$ 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 $\mathbf{z}_c$ and $\hat{\mathbf{z}}_c$, where $R^2 = 1$ suggests there is a one-to-one mapping between the two. As iVAE and β-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 sheds light on why **iMSDA**would work even with only a moderate number of domains. We visualize the the disentanglement in Figure 3.

[3] Our experimental design closely follows the practice employed in prior nonlinear ICA work (Hyvarinen & Morioka, 2016; Hyvarinen et al., 2019).

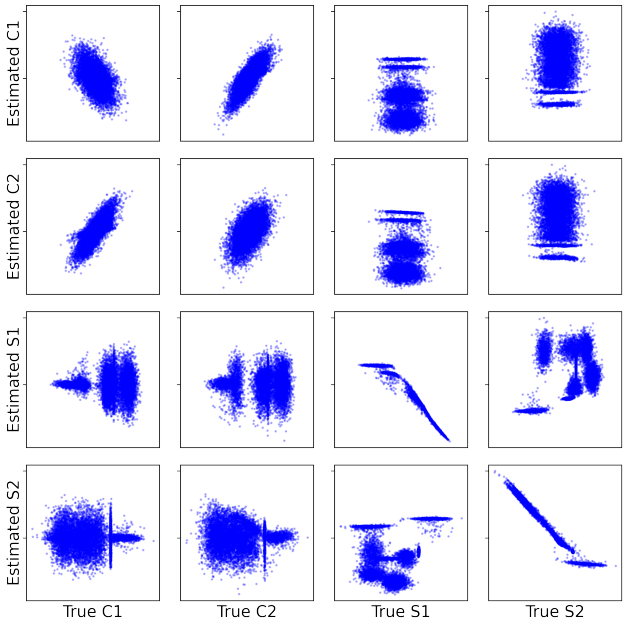


Figure 3. **The scatter plot for the true and the estimated components from our method trained with 9 domains.** The y (resp. x) axe of each subplot corresponds to a specific estimated (resp. true) latent variable. We can observe that the changing components can be identified in a component-wise manner (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.

	ours ($d = 3$)	ours ($d = 5$)	ours ($d = 7$)	ours ($d = 9$)
MCC	0.67 ± 0.12	0.80 ± 0.15	0.90 ± 0.12	$\mathbf{0.91 \pm 0.09}$
R^2	0.59 ± 0.05	0.74 ± 0.09	0.84 ± 0.05	$\mathbf{0.85 \pm 0.06}$
Avg.	0.63 ± 0.08	0.77 ± 0.12	0.87 ± 0.09	$\mathbf{0.88 \pm 0.08}$

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

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

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 $n = 64$.

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

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 $n = 128$.

	$\mathcal{L}_{\text{cls}}$	$\mathcal{L}_{\text{cls}} + \alpha_1\mathcal{L}_{\text{ent}}$	$\mathcal{L}$
→P	97.2 ± 0.01	98.4 ± 0.26	98.48 ± 0.07
→A	81.1 ± 0.73	93.44 ± 0.31	93.75 ± 0.32
→C	79.2 ± 0.17	89.03 ± 2.4	92.46 ± 0.23
→S	70.9 ± 0.69	87.29 ± 0.8	89.22 ± 0.73
Avg.	82.11	92.01	93.48

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.

	$\mathcal{L}_{\text{cls}}$	$\mathcal{L}_{\text{cls}} + \alpha_1\mathcal{L}_{\text{ent}}$	$\mathcal{L}$
→Ar	64.57 ± 0.04	74.23 ± 0.66	75.4 ± 0.86
→Pr	77.77 ± 0.03	82.83 ± 0.12	83.5 ± 0.22
→Cl	47.17 ± 0.20	61.8 ± 0.1	61.4 ± 0.73
→Rw	79.03 ± 0.08	84.2 ± 0.14	84.47 ± 0.38
Avg.	67.2	75.77	76.19

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.

	$n_s = 8$	$n_s = 4$	$n_s = 2$
→P	98.40 ± 0.1	98.48 ± 0.07	98.34 ± 0.07
→A	92.15 ± 1.74	93.75 ± 0.32	93.86 ± 0.46
→C	91.48 ± 1.54	92.46 ± 0.23	92.18 ± 0.34
→S	86.15 ± 5.2	89.22 ± 0.73	86.25 ± 3.05
Avg	92.05 ± 2.14	93.48 ± 0.34	92.66 ± 0.98

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 →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 $(\hat{\mathbf{z}}_c, \hat{\tilde{\mathbf{z}}}_s)$ and $\mathbf{y}$ 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 $p_{\mathbf{x},\mathbf{y}|\mathbf{u}^{\mathcal{T}}}$ for the target domain $\mathbf{u}^{\mathcal{T}}$ 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

Bell, A. J. and Sejnowski, T. J. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

Berthelot, D., Roelofs, R., Sohn, K., Carlini, N., and Kurakin, A. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Q5uh1Nvv5dm.

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. Domain separation networks. In *Advances in Neural Information Processing Systems*, pp. 343–351, 2016.

Cai, R., Li, Z., Wei, P., Qiao, J., Zhang, K., and Hao, Z. Learning disentangled semantic representation for domain adaptation. In *IJCAI: proceedings of the conference*, volume 2019, pp. 2060. NIH Public Access, 2019.

Casey, M. A. and Westner, A. Separation of mixed audio sources by independent subspace analysis. In *ICMC*, pp. 154–161, 2000.

Chen, X., Wang, S., Long, M., and Wang, J. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International conference on machine learning*, pp. 1081–1090. PMLR, 2019.

Comon, P. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Cortes, C., Mansour, Y., and Mohri, M. Learning bounds for importance weighting. In *NIPS 23*, 2010.

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

Deng, Z., Luo, Y., and Zhu, J. Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9944–9953, 2019.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp, 2017.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows, 2019.

Eastwood, C., Mason, I., Williams, C., and Scholkopf, B. Source-free adaptation to measurement shift via bottom-up feature restoration. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1JDiK_TbV4S.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pp. 2839–2848. PMLR, 2016.

Guo, J., Gong, M., Liu, T., Zhang, K., and Tao, D. Ltf: A label transformation framework for correcting label shift. In *International Conference on Machine Learning*, pp. 3843–3853. PMLR, 2020.

Hälvä, H. and Hyvarinen, A. Hidden markov nonlinear ica: Unsupervised learning from nonstationary time series. In *Conference on Uncertainty in Artificial Intelligence*, pp. 939–948. PMLR, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Huang, B., Zhang, K., Zhang, J., Ramsey, J., Sanchez-Romero, R., Glymour, C., and Schölkopf, B. Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research*, 2020.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural autoregressive flows, 2018.

Huang, J., Smola, A., Gretton, A., Borgwardt, K., and Schölkopf, B. Correcting sample selection bias by unlabeled data. In *NIPS 19*, pp. 601–608, 2007.

Hyvärinen, A. and Hoyer, P. Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural computation*, 12(7):1705–1720, 2000.

Hyvarinen, A. and Morioka, H. Unsupervised feature extraction by time-contrastive learning and nonlinear ica, 2016.

Hyvärinen, A. and Pajunen, P. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

Hyvärinen, A., Karhunen, J., and Oja, E. *Independent Component Analysis*. John Wiley & Sons, Inc, 2001.

Hyvarinen, A., Sasaki, H., and Turner, R. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 859–868. PMLR, 2019.

Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217. PMLR, 2020a.

Khemakhem, I., Monti, R., Kingma, D., and Hyvarinen, A. Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. *Advances in Neural Information Processing Systems*, 33:12768–12778, 2020b.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes, 2014.

Kirchmeyer, M., Rakotomamonjy, A., de Bezenac, E., and patrick gallinari. Mapping conditional distributions for domain adaptation under generalized target shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=sPfB2PI87BZ.

Klindt, D., Schott, L., Sharma, Y., Ustyuzhaninov, I., Brendel, W., Bethge, M., and Paiton, D. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

Lachapelle, S., López, P. R., Sharma, Y., Everett, K., Priol, R. L., Lacoste, A., and Lacoste-Julien, S. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ica. *arXiv preprint arXiv:2107.10098*, 2021.

Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR 2011*, pp. 3361–3368. IEEE, 2011.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Li, R., Jia, X., He, J., Chen, S., and Hu, Q. T-svdnet: Exploring high-order prototypical correlations for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9991–10000, 2021.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017a.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2208–2217. JMLR. org, 2017b.

Lu, C., Wu, Y., Hernández-Lobato, J. M., and Schölkopf, B. Invariant causal representation learning. 2020.

Mancini, M., Porzi, L., Bulo, S. R., Caputo, B., and Ricci, E. Boosting domain adaptation by discovering latent domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3771–3780, 2018.

Mao, H., Du, L., Zheng, Y., Fu, Q., Li, Z., Chen, X., Shi, H., and Zhang, D. Source free unsupervised graph domain adaptation. *arXiv preprint arXiv:2112.00955*, 2021.

Park, G. Y. and Lee, S. W. Information-theoretic regularization for multi-source domain adaptation. *arXiv preprint arXiv:2104.01568*, 2021.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1406–1415, 2019.

Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3723–3732, 2018.

Saito, K., Kim, D., Sclaroff, S., and Saenko, K. Universal domain adaptation through self supervision. *arXiv preprint arXiv:2002.07953*, 2020.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. On causal and anticausal learning. In *ICML-12*, Edinburgh, Scotland, 2012.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, 2000.

Shu, R., Bui, H. H., Narui, H., and Ermon, S. A dirt-t approach to unsupervised domain adaptation. In *Proc. 6th International Conference on Learning Representations*, 2018.

Sorrenson, P., Rother, C., and Köthe, U. Disentanglement by nonlinear ica with general incompressible-flow networks (gin). *arXiv preprint arXiv:2001.04872*, 2020.

Stojanov, P., Gong, M., Carbonell, J. G., and Zhang, K. Data-driven approach to multiple-source domain adaptation. *Proceedings of machine learning research*, 89:3487, 2019.

Stojanov, P., Li, Z., Gong, M., Cai, R., Carbonell, J., and Zhang, K. Domain adaptation with invariant representation learning: What transformations to learn? *Advances in Neural Information Processing Systems*, 34, 2021.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60:699–746, 2008.

Theis, F. Towards a general independent subspace analysis. *Advances in Neural Information Processing Systems*, 19: 1361–1368, 2006.

Tong, S., Garipov, T., Zhang, Y., Chang, S., and Jaakkola, T. S. Adversarial support alignment. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=26gKg6x-ie.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

von Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., and Locatello, F. Self-supervised learning with data augmentations provably isolates content from style. *arXiv preprint arXiv:2106.04619*, 2021.

Wang, H., Xu, M., Ni, B., and Zhang, W. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *European Conference on Computer Vision*, pp. 727–744. Springer, 2020.

Wang, J., Chen, Y., Yu, H., Huang, M., and Yang, Q. Easy transfer learning by exploiting intra-domain structures. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1210–1215. IEEE, 2019.

Wu, Y., Winston, E., Kaushik, D., and Lipton, Z. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International Conference on Machine Learning*, pp. 6872–6881. PMLR, 2019.

Wu, Z., Nitzan, Y., Shechtman, E., and Lischinski, D. Stylealign: Analysis and applications of aligned stylegan models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Qg2vi4ZbHM9.

Xie, S., Zheng, Z., Chen, L., and Chen, C. Learning semantic representations for unsupervised domain adaptation. In *International conference on machine learning*, pp. 5423–5432. PMLR, 2018.

Xu, R., Chen, Z., Zuo, W., Yan, J., and Lin, L. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3964–3973, 2018.

Xu, R., Li, G., Yang, J., and Lin, L. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1426–1435, 2019.

Xu, T., Chen, W., WANG, P., Wang, F., Li, H., and Jin, R. CDTrans: Cross-domain transformer for unsupervised domain adaptation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=XGzk5OKWFFc.

Yang, L., Balaji, Y., Lim, S.-N., and Shrivastava, A. Curriculum manager for source selection in multi-source domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 608–624. Springer, 2020.

Zadrozny, B. Learning and evaluating classifiers under sample selection bias. In *ICML-04*, pp. 114–121, Banff, Canada, 2004.

Zhang, J., Li, W., and Ogunbona, P. Joint geometrical and statistical alignment for visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1859–1867, 2017a.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pp. 819–827, 2013.

Zhang, K., Gong, M., and Schölkopf, B. Multi-source domain adaptation: A causal view. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Zhang, K., Huang, B., Zhang, J., Glymour, C., and Schölkopf, B. Causal discovery from nonstationary/heterogeneous data: Skeleton estimation and orientation determination. In *IJCAI*, volume 2017, pp. 1347, 2017b.

Zhang, K., Gong, M., Stojanov, P., Huang, B., LIU, Q., and Glymour, C. Domain adaptation as a problem of inference on graphical models. *Advances in Neural Information Processing Systems*, 33, 2020.

Zhang, W., Ouyang, W., Li, W., and Xu, D. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3801–3809, 2018.

Zhang, Y., Tang, H., Jia, K., and Tan, M. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5031–5040, 2019.

Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P., and Gordon, G. J. Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, pp. 8559–8570, 2018.

Zhu, P., Abdal, R., Femiani, J., and Wonka, P. Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vqGi8Kp0wM.

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 $\mathbf{z}$ and $\hat{\mathbf{z}}$ as follows: $\forall \mathbf{u} \in \mathcal{U}$,

$$p_{\hat{\mathbf{x}}|\mathbf{u}} = p_{\mathbf{x}|\mathbf{u}} \iff p_{\hat{g}(\hat{\mathbf{z}})|\mathbf{u}} = p_{g(\mathbf{z})|\mathbf{u}} \iff p_{g^{-1}\circ\hat{g}(\hat{\mathbf{z}})|\mathbf{u}}|\mathbf{J}_{g^{-1}}| = p_{\mathbf{z}|\mathbf{u}}|\mathbf{J}_{g^{-1}}| \iff p_{h(\hat{\mathbf{z}})|\mathbf{u}} = p_{\mathbf{z}|\mathbf{u}}, \tag{12}$$

where $\hat{g}^{-1} : \mathcal{Z} \to \mathcal{Z}$ 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. $|\mathbf{J}_{g^{-1}}|$ stands for the absolute value of Jacobian matrix determinant of g^{-1}. Note that as both $\hat{g}^{-1}$ and g are invertible, $|\mathbf{J}_{g^{-1}}| \neq 0$ and h is invertible.

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

$$p_{\mathbf{z}|\mathbf{u}}(\mathbf{z}|\mathbf{u}) = \prod_{i=1}^{n} p_{z_i|\mathbf{u}}(z_i); \qquad p_{\hat{\mathbf{z}}|\mathbf{u}}(\hat{\mathbf{z}}|\mathbf{u}) = \prod_{i=1}^{n} p_{\hat{z}_i|\mathbf{u}}(\hat{z}_i)$$

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

$$\log p_{\mathbf{z}|\mathbf{u}}(\mathbf{z}|\mathbf{u}) = \sum_{i=1}^{n} q_i(z_i, \mathbf{u}); \qquad \log p_{\hat{\mathbf{z}}|\mathbf{u}}(\hat{\mathbf{z}}|\mathbf{u}) = \sum_{i=1}^{n} \hat{q}_i(\hat{z}_i, \mathbf{u})$$

Applying the change of variables to Equation 12 yields

$$p_{\mathbf{z}|\mathbf{u}} = p_{\hat{\mathbf{z}}|\mathbf{u}} \cdot |\mathbf{J}_{h^{-1}}| \iff \sum_{i=1}^{n} q_i(z_i, \mathbf{u}) + \log |\mathbf{J}_h| = \sum_{i=1}^{n} \hat{q}_i(\hat{z}_i, \mathbf{u}) \tag{13}$$

where $\mathbf{J}_{h^{-1}}$ and $\mathbf{J}_h$ are the Jacobian matrix of the transformation associated with h^{-1} and h respectively.

For ease of exposition, we adopt the following notation:

$$h'_{i,(k)} := \frac{\partial z_i}{\partial \hat{z}_k}, \qquad h''_{i,(k,q)} := \frac{\partial^2 z_i}{\partial \hat{z}_k \partial \hat{z}_q};$$
$$\eta'_i(z_i, \mathbf{u}) := \frac{\partial q_i(z_i, \mathbf{u})}{\partial z_i}, \qquad \eta''_i(z_i, \mathbf{u}) := \frac{\partial^2 q_i(z_i, \mathbf{u})}{(\partial z_i)^2}.$$

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

$$\sum_{i=1}^{n} \left(\eta''_i(z_i, \mathbf{u}) \cdot h'_{i,(k)} h'_{i,(q)} + \eta'_i(z_i, \mathbf{u}) \cdot h''_{i,(k,q)} \right) + \frac{\partial^2 \log |\mathbf{J}_h|}{\partial \hat{z}_k \partial \hat{z}_q} = 0. \tag{14}$$

Therefore, for $\mathbf{u} = \mathbf{u}_0, \ldots, \mathbf{u}_{2n_s}$, we have $2n_s + 1$ such equations. Subtracting each equation corresponding to $\mathbf{u}_1, \ldots, \mathbf{u}_{2n_s}$ with the equation corresponding to $\mathbf{u}_0$ results in $2n_s$ equations:

$$\sum_{i=n_c+1}^{n} \left((\eta''_i(z_i, \mathbf{u}_j) - \eta''_i(z_i, \mathbf{u}_0)) \cdot h'_{i,(k)} h'_{i,(q)} + (\eta'_i(z_i, \mathbf{u}_j) - \eta'_i(z_i, \mathbf{u}_0)) \cdot h''_{i,(k,q)} \right) = 0, \tag{15}$$

where $j = 1, \ldots 2n_s$. Note that as the content distribution is invariant to domains i.e., $p_{\mathbf{z}_c} = p_{\mathbf{z}_c|\mathbf{u}}$. Thus, we have $\eta''_i(z_i, \mathbf{u}_j) = \eta''_i(z_i, \mathbf{u}_{j'})$ and $\eta'_i(z_i, \mathbf{u}_j) = \eta'_i(z_i, \mathbf{u}_{j'}), \forall j, j'$. Hence only the style components $i = n_c + 1, \ldots, n$ remain in the summation of each equation.

Under the linear independence condition in Assumption A3, the linear system is a $2n_s \times 2n_s$ full-rank system. Therefore, the only solution is $h'_{i,(k)}h'_{i,(q)} = 0$ and $h''_{i,(k,q)} = 0$ for $i = n_c + 1, \dots, n$ and $k, q \in [n], k \neq q$.

As $h(\cdot)$ is smooth over $\mathcal{Z}$, its Jacobian can written as:

$$\mathbf{J}_h = \left[\begin{array}{c|c} \mathbf{A} := \frac{\partial \mathbf{z}_c}{\partial \hat{\mathbf{z}}_c} & \mathbf{B} := \frac{\partial \mathbf{z}_c}{\partial \hat{\mathbf{z}}_s} \\ \hline \mathbf{C} := \frac{\partial \mathbf{z}_s}{\partial \hat{\mathbf{z}}_c} & \mathbf{D} := \frac{\partial \mathbf{z}_s}{\partial \hat{\mathbf{z}}_s}, \end{array}\right] \tag{16}$$

Note that $h'_{i,(k)}h'_{i,(q)} = 0$ implies that for each $i = n_c + 1, \dots, n$, $h'_{i,(k)} \neq 0$ for at most one element $k \in [n]$. Therefore, there is only at most one non-zero entry in each row indexed by $i = n_c + 1, \dots, n$ in the Jacobian matrix $\mathbf{J}_h$. Further, the invertibility of $h(\cdot)$ necessitates $\mathbf{J}_h$ to be full-rank which implies that there is exactly one non-zero component in each row of matrices $\mathbf{C}$ and $\mathbf{D}$.

Since for every $i \in \{n_c + 1, \dots, n\}$, z_i has changing distributions over $\mathbf{u}$ and all $\hat{z}_k$'s for $i \in \{1, \dots, n_c\}$ (i.e. $\hat{\mathbf{z}}_c$) have invariant distributions over $\mathbf{u}$, we can deduce that $\mathbf{C} = \mathbf{0}$ and the only non-zero entry $\frac{z_i}{\hat{z}_k}$ must reside in $\mathbf{D}$ with $k \in \{n_c + 1, \dots, n\}$. Therefore, for each true variable in the changing part $z_i, i \in \{n_c + 1, \dots, n\}$, there exists one estimated variable in the changing part $\hat{z}_k, k \in \{n_c + 1, \dots, n\}$ such that $z_i = h'_i(\hat{z}_k)$. Further, because $\mathbf{J}_h$ is of full-rank ($h(\cdot)$ being invertible) and $\mathbf{C}$ is a zero matrix, $\mathbf{D}$ must be of full-rank, which implies that $h'_i(\cdot)$ is invertible for each $i \in \{n_c + 1, \dots, n\}$. Thus, the changing components $\mathbf{z}_s$ 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 $\bar{h} : \mathcal{Z} \to \mathcal{Z}$ between the estimated and the true latent variable. The introduction of $\bar{h}(\cdot)$ allows us to formalize the block-identifiability condition.

In Step 2 and Step 3, we show that the estimated content variable $\hat{\mathbf{z}}_c$ does not depend on the true style variable $\mathbf{z}_s$, that is, $\bar{h}_c(\mathbf{z})$ does not depend on the input $\mathbf{z}_s$. 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 $\hat{\mathbf{z}}_c$ depended on $\mathbf{z}_s$, 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 $\hat{\mathbf{z}}_c = \bar{h}_c(\mathbf{z}_c)$ being invertible.

Step 1. As the data generating process of the learned model $(\hat{g}, p_{\hat{\mathbf{z}}_s}, p_{\hat{\mathbf{z}}_s|\mathbf{u}})$ (Equation 1) establishes the independence between the generating process $\hat{\mathbf{z}}_c \sim p_{\hat{\mathbf{z}}_c}$ [4] and $\mathbf{u}$, it follows that for any $A_{\mathbf{z}_c} \subseteq \mathcal{Z}_c$,

$$\begin{aligned} &\mathbb{P}\left[\{\hat{g}^{-1}_{1:n_c}(\hat{\mathbf{x}}) \in A_{\mathbf{z}_c}\} | \{\mathbf{u} = \mathbf{u}_1\}\right] = \mathbb{P}\left[\{\hat{g}^{-1}_{1:n_c}(\hat{\mathbf{x}}) \in A_{\mathbf{z}_c}\} | \{\mathbf{u} = \mathbf{u}_2\}\right], \quad \forall \mathbf{u}_1, \mathbf{u}_2 \in \mathcal{U} \\ &\qquad\qquad \Longleftrightarrow \\ &\mathbb{P}\left[\{\hat{\mathbf{x}} \in (\hat{g}^{-1}_{1:n_c})^{-1}(A_{\mathbf{z}_c})\} | \{\mathbf{u} = \mathbf{u}_1\}\right] = \mathbb{P}\left[\{\hat{\mathbf{x}} \in (\hat{g}^{-1}_{1:n_c})^{-1}(A_{\mathbf{z}_c})\} | \{\mathbf{u} = \mathbf{u}_2\}\right], \quad \forall \mathbf{u}_1, \mathbf{u}_2 \in \mathcal{U} \end{aligned} \tag{17}$$

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

[4] Note that in this proof, we decorate quantities with $\hat{\cdot}$ to indicate that they are associated with the estimated model $(\hat{g}, p_{\hat{\mathbf{z}}_c}, p_{\hat{\mathbf{z}}_s|\mathbf{u}})$.

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 $\mathbf{x}$ from the true generating process $(g, p_{\mathbf{z}_c}, p_{\mathbf{z}_s|\mathbf{u}})$, i.e. ,

$$\begin{aligned}
&\mathbb{P}\left[\{\mathbf{x} \in (\hat{g}_{1:n_c}^{-1})^{-1}(A_{\mathbf{z}_c})\}|\{\mathbf{u}=\mathbf{u}_1\}\right] = \mathbb{P}\left[\{\mathbf{x} \in (\hat{g}_{1:n_c}^{-1})^{-1}(A_{\mathbf{z}_c})\}|\{\mathbf{u}=\mathbf{u}_2\}\right] \\
&\iff \\
&\mathbb{P}\left[\{\hat{g}_{1:n_c}^{-1}(\mathbf{x}) \in A_{\mathbf{z}_c}\}|\{\mathbf{u}=\mathbf{u}_1\}\right] = \mathbb{P}\left[\{\hat{g}_{1:n_c}^{-1}(\mathbf{x}) \in A_{\mathbf{z}_c}\}|\{\mathbf{u}=\mathbf{u}_2\}\right]. \qquad (18)
\end{aligned}$$

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 $\bar{h} = h^{-1}$ where h is introduced in the proof of Theorem 4.1 (Appendix A.1). Expressing $\hat{g}^{-1} = \bar{h} \circ g^{-1}$ and $\bar{h}_c(\cdot) := \bar{h}_{1:n_c}(\cdot) : \mathcal{Z} \to \mathcal{Z}_c$ in Equation 18 yields

$$\begin{aligned}
&\mathbb{P}\left[\{\bar{h}_c(\mathbf{z}) \in A_{\mathbf{z}_c}\}|\{\mathbf{u}=\mathbf{u}_1\}\right] = \mathbb{P}\left[\{\bar{h}_c(\mathbf{z}) \in A_{\mathbf{z}_c}\}|\{\mathbf{u}=\mathbf{u}_2\}\right], \\
&\iff \\
&\mathbb{P}\left[\{\mathbf{z} \in \bar{h}_c^{-1}(A_{\mathbf{z}_c})\}|\{\mathbf{u}=\mathbf{u}_1\}\right] = \mathbb{P}\left[\{\mathbf{z} \in \bar{h}_c^{-1}(A_{\mathbf{z}_c})\}|\{\mathbf{u}=\mathbf{u}_2\}\right], \\
&\iff \\
&\int_{\mathbf{z} \in \bar{h}_c^{-1}(A_{\mathbf{z}_c})} p_{\mathbf{z}|\mathbf{u}}(\mathbf{z}|\mathbf{u}_1)\, d\mathbf{z} = \int_{\mathbf{z} \in \bar{h}_c^{-1}(A_{\mathbf{z}_c})} p_{\mathbf{z}|\mathbf{u}}(\mathbf{z}|\mathbf{u}_2)\, d\mathbf{z}, \qquad (19)
\end{aligned}$$

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

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

$$\int_{[\mathbf{z}_c^\top, \mathbf{z}_s^\top]^\top \in \bar{h}_c^{-1}(A_{\mathbf{z}_c})} p_{\mathbf{z}_c}(\mathbf{z_c}) \left(p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_1) - p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_2)\right) d\mathbf{z}_s d\mathbf{z}_c = 0. \qquad (20)$$

Step 2. In order to show the block-identifiability of $\mathbf{z}_c$, we would like to prove that $\mathbf{z}_c := \bar{h}_c([\mathbf{z}_c^\top, \mathbf{z}_s^\top]^\top)$ does not depend on $\mathbf{z}_s$. 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: $\bar{h}_c([\mathbf{z}_c^\top, \mathbf{z}_s^\top]^\top)$ does not depend on $\mathbf{z}_s$.
- Statement 2: $\forall \mathbf{z}_c \in \mathcal{Z}_c$, it follows that $\bar{h}_c^{-1}(\mathbf{z}_c) = B_{\mathbf{z}_c} \times \mathcal{Z}_s$ where $B_{\mathbf{z}_c} \neq \emptyset$ and $B_{\mathbf{z}_c} \subseteq \mathcal{Z}_c$.
- Statement 3: $\forall \mathbf{z}_c \in \mathcal{Z}_c, r \in \mathbb{R}^+$, it follows that $\bar{h}_c^{-1}(\mathcal{B}_r(\mathbf{z}_c)) = B_{\mathbf{z}_c}^+ \times \mathcal{Z}_s$ where $\mathcal{B}_r(\mathbf{z}_c) := \{\mathbf{z}_c' \in \mathcal{Z}_c : ||\mathbf{z}_c' - \mathbf{z}_c||^2 < r\}$, $B_{\mathbf{z}_c}^+ \neq \emptyset$, and $B_{\mathbf{z}_c}^+ \subseteq \mathcal{Z}_c$.

Statement 2 is a mathematical formulation of Statement 1. Statement 3 generalizes singletons $\mathbf{z}_c$ in Statement 2 to open, non-empty balls $\mathcal{B}_r(\mathbf{z}_c)$. 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. $\forall \mathbf{z}_c \in \mathcal{Z}_c, r \in \mathbb{R}^+$, $\bar{h}_c^{-1}((\mathcal{B}_r(\mathbf{z}_c))) = \cup_{\mathbf{z}_c' \in \mathcal{B}_r(\mathbf{z}_c)} h_c^{-1}(\mathbf{z}_c')$. Statement 2 indicates that every participating sets in the union satisfies $h_c^{-1}(\mathbf{z}_c') = B_{\mathbf{z}_c}' \times \mathcal{Z}_s$, thus the union $\bar{h}_c^{-1}((\mathcal{B}_r(\mathbf{z}_c)))$ 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 $\exists \hat{\mathbf{z}}_c \in \mathcal{Z}_c$ such that there exist $\hat{\mathbf{z}}_c^B \in \{\mathbf{z}_{1:n_c} : \mathbf{z} \in \bar{h}_c^{-1}(\hat{\mathbf{z}}_c)\}$ and $\hat{\mathbf{z}}_s^B \in \mathcal{Z}_s$ resulting in $\bar{h}_c(\hat{\mathbf{z}}^B) \neq \hat{\mathbf{z}}_c$ where $\hat{\mathbf{z}}^B = [(\hat{\mathbf{z}}_c^B)^\top, (\hat{\mathbf{z}}_s^B)^\top]^\top$. As $\bar{h}_c(\cdot)$ is continuous, there exists $\hat{r} \in \mathbb{R}^+$ such that $\bar{h}_c(\hat{\mathbf{z}}^B) \notin \mathcal{B}_{\hat{r}}(\hat{\mathbf{z}}_c)$. That is, $\hat{\mathbf{z}}^B \notin h_c^{-1}(\mathcal{B}_{\hat{r}}(\hat{\mathbf{z}}_c))$. Also, Statement 3 suggests that $h_c^{-1}(\mathcal{B}_{\hat{r}}(\hat{\mathbf{z}}_c)) = \hat{B}_{\mathbf{z}_c} \times \mathcal{Z}_s$. By definition of $\hat{\mathbf{z}}^B$, it is clear that $\hat{\mathbf{z}}_{1:n_c}^B \in \hat{B}_{\mathbf{z}_c}$. The fact that $\hat{\mathbf{z}}^B \notin h_c^{-1}(\mathcal{B}_{\hat{r}}(\hat{\mathbf{z}}_c))$ 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 $\bar{h}_c(\cdot)$ depended on $\hat{z}_s$, the preimage $\bar{h}_c^{-1}(\mathcal{B}_r(\mathbf{z}_c))$ could be partitioned into two parts (i.e. $B^*_{\mathbf{z}}$ and $\bar{h}_c^{-1}(A^*_{\mathbf{z}_c}) \setminus B^*_{\mathbf{z}}$ defined below). The dependency between $\bar{h}_c(\cdot)$ and $\hat{z}_s$ is captured by $B^*_{\mathbf{z}}$, which would not emerge otherwise. In contrast, $\bar{h}_c^{-1}(A^*_{\mathbf{z}_c}) \setminus B^*_{\mathbf{z}}$ also exists when $\bar{h}_c(\cdot)$ does not depend on $\hat{z}_s$. We evaluate the invariance relation Equation 20 and show that the integral over $\bar{h}_c^{-1}(A^*_{\mathbf{z}_c}) \setminus B^*_{\mathbf{z}}$ (i.e. T_1) is always 0, however, the integral over $B^*_{\mathbf{z}}$ (i.e. T_2) is necessarily non-zero, which leads to the contraction with Equation 20 and thus shows the $\bar{h}_c(\cdot)$ cannot depend on $\hat{z}_s$.

First, note that because $\mathcal{B}_r(\mathbf{z}_c)$ is open and $\bar{h}_c(\cdot)$ is continuous, the pre-image $\bar{h}_c^{-1}(\mathcal{B}_r(\mathbf{z}_c))$ is open. In addition, the continuity of $h(\cdot)$ and the matched observation distributions $\forall \mathbf{u}' \in \mathcal{U},\ \mathbb{P}\left[\{\mathbf{x} \in A_{\mathbf{x}}\} | \{\mathbf{u} = \mathbf{u}'\}\right] = \mathbb{P}\left[\{\hat{\mathbf{x}} \in A_{\mathbf{x}}\} | \{\mathbf{u} = \mathbf{u}'\}\right]$ lead to $h(\cdot)$ being bijection as shown in (Klindt et al., 2020),which implies that $\bar{h}_c^{-1}(\mathcal{B}_r(\mathbf{z}_c))$ is non-empty. Hence, $\bar{h}_c^{-1}(\mathcal{B}_r(\mathbf{z}_c))$ is both non-empty and open. Suppose that $\exists A^*_{\mathbf{z}_c} := \mathcal{B}_{r^*}(\mathbf{z}^*_c)$ where $\mathbf{z}^*_c \in \mathcal{Z}_c$, $r^* \in \mathbb{R}^+$, such that $B^*_{\mathbf{z}} := \{\mathbf{z} \in \mathcal{Z} : \mathbf{z} \in \bar{h}_c^{-1}(A^*_{\mathbf{z}_c}),\ \{\mathbf{z}_{1:n_c}\} \times \mathcal{Z}_s \not\subseteq \bar{h}_c^{-1}(A^*_{\mathbf{z}_c})\} \neq \emptyset$. Intuitively, $B^*_{\mathbf{z}}$ contains the partition of the pre-image $\bar{h}_c^{-1}(A^*_{\mathbf{z}_c})$ that the style part $\mathbf{z}_{n_c+1:n}$ cannot take on any value in $\mathcal{Z}_s$. Only certain values of the style part were able to produce specific outputs of indeterminacy $\bar{h}_c(\cdot)$. Clearly, this would suggest that $\bar{h}_c(\cdot)$ depends on $\mathbf{z}_c$.
To show contraction with Equation 20, we evaluate the LHS of Equation 20 with such a $A^*_{\mathbf{z}_c}$:

$$
\begin{aligned}
&\int_{[\mathbf{z}_c^\top, \mathbf{z}_s^\top]^\top \in \bar{h}_c^{-1}(A^*_{\mathbf{z}_c})} p_{\mathbf{z}_c}(\mathbf{z_c}) \left(p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_1) - p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_2)\right) d\mathbf{z}_s d\mathbf{z}_c \\
&= \underbrace{\int_{[\mathbf{z}_c^\top, \mathbf{z}_s^\top]^\top \in \bar{h}_c^{-1}(A^*_{\mathbf{z}_c}) \setminus B^*_{\mathbf{z}}} p_{\mathbf{z}_c}(\mathbf{z_c}) \left(p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_1) - p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_2)\right) d\mathbf{z}_s d\mathbf{z}_c}_{T_1} \\
&\quad + \underbrace{\int_{[\mathbf{z}_c^\top, \mathbf{z}_s^\top]^\top \in B^*_{\mathbf{z}}} p_{\mathbf{z}_c}(\mathbf{z_c}) \left(p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_1) - p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_2)\right) d\mathbf{z}_s d\mathbf{z}_c}_{T_2}.
\end{aligned}
$$

We first look at the value of T_1. When $\bar{h}_c^{-1}(A^*_{\mathbf{z}_c}) \setminus B^*_{\mathbf{z}} = \emptyset$, T_1 evaluates to 0. Otherwise, by definition we can rewrite $\bar{h}_c^{-1}(A^*_{\mathbf{z}_c}) \setminus B^*_{\mathbf{z}}$ as $C^*_{\mathbf{z}_c} \times \mathcal{Z}_s$ where $C^*_{\mathbf{z}_c} \neq \emptyset$ and $C^*_{\mathbf{z}_c} \subset \mathcal{Z}_c$. With this expression, it follows that

$$
\begin{aligned}
&\int_{[\mathbf{z}_c^\top, \mathbf{z}_s^\top]^\top \in C^*_{\mathbf{z}_c}} p_{\mathbf{z}_c}(\mathbf{z_c}) \left(p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_1) - p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_2)\right) d\mathbf{z}_s d\mathbf{z}_c \\
&\quad = \int_{\mathbf{z}_c \in C^*_{\mathbf{z}_c}} p_{\mathbf{z}_c}(\mathbf{z_c}) \int_{\mathbf{z}_s \in \mathcal{Z}_s} \left(p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_1) - p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}_2)\right) d\mathbf{z}_s d\mathbf{z}_c \\
&\quad = \int_{\mathbf{z}_c \in C^*_{\mathbf{z}_c}} p_{\mathbf{z}_c}(\mathbf{z_c}) \left(1 - 1\right) d\mathbf{z}_c = 0.
\end{aligned}
$$

Therefore, in both cases T_1 evaluates to 0 for $A^*_{\mathbf{z}_c}$.

Now, we address T_2. As discussed above, $\bar{h}_c^{-1}(A^*_{\mathbf{z}_c})$ is open and non-empty. Because of the continuity of $\bar{h}_c(\cdot)$, $\forall \mathbf{z}_B \in B^*_{\mathbf{z}}$, there exists $r(\mathbf{z}_B) \in \mathbb{R}^+$ such that $\mathcal{B}_{r(\mathbf{z}_B)}(\mathbf{z}_B) \subseteq B^*_{\mathbf{z}}$. As $p_{\mathbf{z}|\mathbf{u}}(\mathbf{z}|\mathbf{u}) > 0$ over $(\mathbf{z}, \mathbf{u})$, we have $\mathbb{P}\left[\{\mathbf{z} \in B^*_{\mathbf{z}}\} | \{\mathbf{u} = \mathbf{u}'\}\right] \geq \mathbb{P}\left[\{\mathbf{z} \in \mathcal{B}_{r(\mathbf{z}_B)}(\mathbf{z}_B) | \{\mathbf{u} = \mathbf{u}'\}\}\right] > 0$ for any $\mathbf{u}' \in \mathcal{U}$. Assumption A4 indicates that $\exists \mathbf{u}^*_1, \mathbf{u}^*_2$, such that

$$
T_2 := \int_{[\mathbf{z}_c^\top, \mathbf{z}_s^\top]^\top \in B^*_{\mathbf{z}}} p_{\mathbf{z}_c}(\mathbf{z_c}) \left(p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}^*_1) - p_{\mathbf{z}_s|\mathbf{u}}(\mathbf{z_s}|\mathbf{u}^*_2)\right) d\mathbf{z}_s d\mathbf{z}_c \neq 0.
$$

Therefore, for such $A^*_{\mathbf{z}_c}$, 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, $\bar{h}_c(\cdot)$ does not depend on the style variable $\mathbf{z}_s$.

Step 4. With the knowledge that $\bar{h}_c(\cdot)$ does not depend on the style variable $\mathbf{z}_s$, we now show that there exists an invertible mapping between the true content variable $\mathbf{z}_c$ and the estimated version $\hat{\mathbf{z}}_c$.

As $\bar{h}(\cdot)$ is smooth over $\mathcal{Z}$, its Jacobian can written as:

$$\mathbf{J}_{\bar{h}} = \left[\begin{array}{c|c} \mathbf{A} := \frac{\partial \hat{\mathbf{z}}_c}{\partial \mathbf{z}_c} & \mathbf{B} := \frac{\partial \hat{\mathbf{z}}_c}{\partial \mathbf{z}_s} \\ \hline \mathbf{C} := \frac{\partial \hat{\mathbf{z}}_s}{\partial \mathbf{z}_c} & \mathbf{D} := \frac{\partial \hat{\mathbf{z}}_s}{\partial \mathbf{z}_s}, \end{array}\right] \tag{21}$$

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

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 decoer 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 β-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 β parameter of KL loss term to 0.1.

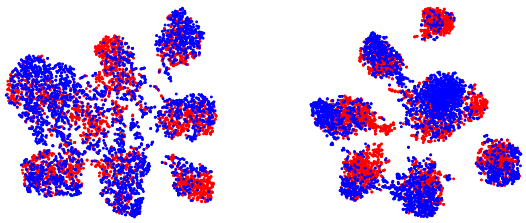
A.4. Real-world Data Experiments

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 α_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.



(a) Source Only (b) **iMSDA**

Figure 4. The t-SNE visualizations of learned features on the → Sketch task in PACS. Red: source domains, Blue: target domain.