

Continuously Indexed Domain Adaptation

Hao Wang ^{*1} Hao He ^{*1} Dina Katabi ¹

Abstract

Existing domain adaptation focuses on transferring knowledge between domains with categorical indices (e.g., between datasets A and B). However, many tasks involve continuously indexed domains. For example, in medical applications, one often needs to transfer disease analysis and prediction across patients of different ages, where age acts as a continuous domain index. Such tasks are challenging for prior domain adaptation methods since they ignore the underlying relation among domains. In this paper, we propose the first method for continuously indexed domain adaptation. Our approach combines traditional adversarial adaptation with a novel discriminator that models the encoding-conditioned domain index distribution. Our theoretical analysis demonstrates the value of leveraging the domain index to generate invariant features across a continuous range of domains. Our empirical results show that our approach outperforms the state-of-the-art domain adaption methods on both synthetic and real-world medical datasets¹.

1. Introduction

Machine learning often assumes that training and test data come from the same distribution, so that the trained model generalizes well to the test scenario. This assumption breaks however when the model is trained and tested in distinct domains, i.e., different source and target domains. Domain adaption (DA) leverages labeled data from the source domains and unlabeled data (or a limited amount of labeled data) from the target domains to significantly improve performance (Ben-David et al., 2010; Ganin et al., 2016; Tzeng et al., 2017; Zhang et al., 2019).

Existing DA methods however focus on adaption among

^{*}Equal contribution ¹MIT Computer Science and Artificial Intelligence Laboratory, Massachusetts, USA. Correspondence to: Hao Wang <hoguewang@gmail.com>.

Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

¹Code will soon be available at <https://github.com/hehaodele/CIDA>

categorical domains where the domain index is just a label. A common example would be to adapt a model from one image dataset to another, e.g., adapting from MNIST to SVHN. However, many real-world tasks require adaptation among continuously index domains. For example, in medical applications, one needs to adapt disease diagnosis and prognosis across patients of different ages, where age is a continuous domain index. Treating the age of the source and target domains as domain labels is unlikely to yield the best results because it does not take advantage of the relationship between the disease manifestation and the person's age. Similar issues appear in robotics. For example, underwater robots have to operate at different water depths and viscosity, and one expects that adaptation across datasets from different depths or viscosity (e.g., lake vs. sea) should take into account the relationship between the robot operation and the physical properties of the liquid in which it operates. These examples highlight the limitations of current DA methods when applied to continuously indexed domains.

So, how should we perform domain adaption across continuously indexed domains? We note that in the above examples the domain index plays the role of a distance metric – i.e., it captures a similarity distance between the domains with respect to the task. Thus, one approach for addressing the problem is to modify traditional adversarial adaptation to make the discriminator regress the domain index using a distance-based loss, like the L_2 or L_1 loss. Although this is better than categorical DA, we show analytically that such treatment can lead to equilibriums with relatively poor domain alignments. A better solution is to develop a probabilistic discriminator that models the domain index distribution. We show that such a discriminator not only successfully captures the underlying relation among domains, but also enjoys better theoretical guarantees in terms of domain alignment. We also note that our method can be naturally generalized to handle multi-dimensional domain indices, achieving further performance gain. For example, in medial applications the index can be a vector of age, blood pressure, activity level, etc.

Our contributions are as follows:

- We identify the problem of adaptation across continuously indexed domains and propose continuously indexed domain adaptation (CIDA) as the first general DA method

for addressing this problem. Further, we analyze our method and provide theoretical guarantees that CIDA aligns continuously indexed domains at equilibrium.

- We derive two advanced versions, probabilistic CIDA and multi-dimensional CIDA, to further improve performance and handle multi-dimensional domain indices, with minimal overhead.
- We provide empirical results using both synthetic and real-world medical datasets which show that CIDA and its probabilistic and multi-dimensional variants significantly improve performance over the state-of-the-art DA methods for continuously indexed domains.

2. Related Work

Adversarial Domain Adaptation. Much prior work has focused on the problem of domain adaptation (Zhao et al., 2017; Long et al., 2018; Saito et al., 2018; Sankaranarayanan et al., 2018; Zhang et al., 2019). The key idea is to match the distributions of the source and target domains. This is achieved by matching their distributions’ statistics either directly (Pan et al., 2010; Tzeng et al., 2014; Sun & Saenko, 2016) or with the help of an adversarial loss (Ganin et al., 2016; Zhao et al., 2017; Tzeng et al., 2017; Zhang et al., 2019; Kuroki et al., 2019). Adversarial domain adaptation is particularly popular due to its relatively strong theoretical insights (Goodfellow et al., 2014; Zhao et al., 2018; Zhang et al., 2019; Zhao et al., 2019) and its compatibility with neural networks. It aligns the distributions of the source and target domains by generating an encoding indistinguishable from a perspective of discriminator that is trained to classify the domain of the data. In this paper, we build on adversarial domain adaptation and extend it to address continuously indexed domains.

Incremental Domain Adaptation. Closest to our work are incremental DA approaches. Essentially they assume the domain shifts smoothly over time and try to incrementally adapt the source domain to multiple target domains. Different methods are used to perform categorical DA for each domain pair, such as optimal transport (Jimenez et al., 2019), adversarial loss (Bitarafan et al., 2016), generative adversarial networks (Wulfmeier et al., 2018), and linear transform (Hoffman et al., 2014). Bobu et al. (2018) notices such incremental adaptation procedure is prone to catastrophic forgetting, a tendency to forget the knowledge of previous domains while specializing to a new domain, and therefore proposes a replay technique to tackle the issue. Here we note several key differences between CIDA and the methods above. (1) These approaches incrementally perform pair-wise *categorical DA*. Hence failure in adapting one domain pair can lead to catastrophic failures for all following pairs. (2) They only work on DA tasks with one

single domain shifting dimension (usually ‘time’), while our method naturally generalizes to multi-dimensional settings. Such differences are empirically verified in Sec. 5.

3. Methods

In this section, we formalize the problem of adaptation among continuously indexed domains, and describe our methods for addressing the problem. We then provide theoretical guarantees for the proposed methods in Sec. 4.

Problem. We consider the unsupervised domain adaptation setting and assume a set of continuous domain indices $\mathcal{U} = \mathcal{U}_s \cup \mathcal{U}_t$, where \mathcal{U}_s and \mathcal{U}_t are the domain index sets for the source and the target domains, and \mathcal{U} is part of a metric space (i.e., a metric like the Euclidian distance is defined over the set). The input and labels are denoted as \mathbf{x} and y , respectively. With access to the labeled data $\{(\mathbf{x}_i^s, y_i^s, u_i^s)\}_{i=1}^n$ from source domains ($u_i^s \in \mathcal{U}_s$) and unlabeled data $\{(\mathbf{x}_i^t, u_i^t)\}_{i=1}^m$ from target domains ($u_i^t \in \mathcal{U}_t$), the goal is to accurately predict the labels $\{(y_i^t)\}_{i=1}^m$ for data in the target domains.

Multi-Dimensional Domain Indices. For clarity, we introduce our methods and theory in the context of unidimensional domain indices. However, they can directly apply to multi-dimensional domain indices. Later in Sec. 5, we show that the ability of handling multi-dimensional domain indices brings further performance gains.

3.1. Continuously Indexed Domain Adaptation (CIDA)

To perform adaptation across a continuous range of domains, we leverage the idea of learning domain-invariant encodings with adversarial training. We propose to learn an encoder² E and a predictor F such that the distribution of the encodings $\mathbf{z} = E(\mathbf{x}) \in \mathcal{Z}$ (or $\mathbf{z} = E(\mathbf{x}, u)$) from all domains \mathcal{U} are aligned so that all labels can be accurately predicted by the shared predictor F . Formally, domain-invariant encodings require that $p(\mathbf{z}|u_1) = p(\mathbf{z}|u_2), \forall u_1, u_2 \in \mathcal{U}$. It implies that \mathbf{z} and u are independent ($u \perp\!\!\!\perp \mathbf{z}$), i.e., $p(u|\mathbf{z}) = p(u)$ or equivalently $p(\mathbf{z}|u) = p(\mathbf{z})$. This is achieved with the help of a discriminator D . In continuously indexed domains however, small changes in u should lead to small changes in the encoding. Thus, instead of classifying the encoding into categorical domains, the discriminator D in CIDA regresses the domain index.

Formally, CIDA performs a minimax optimization with the value function $V(E, F, D)$ as:

$$\min_{E, F} \max_D V_p(E, F) - \lambda_d V_d(D, E), \quad (1)$$

²In general the encoder $E(\mathbf{x}, u)$ can be probabilistic. For example, \mathbf{z} can be generated from a Gaussian distribution whose mean and variance are given by $E(\mathbf{x}, u)$.

where we have

$$\begin{aligned} V_p(E, F) &\triangleq \mathbb{E}^s[L_p(F(E(\mathbf{x}, u)), y)] \\ V_d(D, E) &\triangleq \mathbb{E}[L_d(D(E(\mathbf{x}, u)), u)] \end{aligned}$$

where \mathbb{E} and \mathbb{E}^s denote the expectations taken over the entire data distribution $p(\mathbf{x}, y, u)$ and the source data distribution $p^s(\mathbf{x}, y, u)$. Note that the label y is only accessible in the source domains. L_p is the prediction loss (e.g., cross-entropy loss for classification tasks), and L_d is the domain index loss. λ_d is a hyperparameter balancing both losses. The main difference between CIDA and traditional adversarial domain adaptation is that the discriminator loss L_d is a monotonic function of the metric defined over \mathcal{U} .

3.2. Variants of CIDA

Note that there can be various designs for both D and L_d . For example, D can either directly predict the domain index or predict its mean and variance, and L_d can be either the L_2 or L_1 loss. Different designs come with different theoretical guarantees.

Vanilla CIDA. In the vanilla CIDA, D directly predicts the domain index, and correspondingly L_d is the L_2 loss between the predicted and ground-truth domain index,

$$L_d(D(\mathbf{z}), u) = (D(\mathbf{z}) - u)^2, \quad (2)$$

Vanilla CIDA above only guarantees matching the mean of the distribution $p(u|\mathbf{z})$ (see theoretical results in Sec. 4).

Therefore in the following, we introduce an advanced version, dubbed probabilistic CIDA (PCIDA), which enjoys better theoretical guarantees to match both the mean and variance of the distribution $p(u|\mathbf{z})$. We note that PCIDA can be extended to match higher-order moments.

Probabilistic CIDA. The major improvement from CIDA to PCIDA is that in PCIDA, the discriminator predicts the distribution of $p(u|\mathbf{z})$ instead of providing point estimation. We start with the simplest probabilistic model, Gaussian distributions, where the discriminator D outputs the mean and variance of $p(u|\mathbf{z})$ as $D_\mu(\mathbf{z})$ and $D_{\sigma^2}(\mathbf{z})$, respectively. To train such a discriminator, we use the negative log-likelihood as the loss function:

$$L_d(D(\mathbf{z}), u) = \frac{(D_\mu(\mathbf{z}) - u)^2}{2D_{\sigma^2}(\mathbf{z})} + \frac{1}{2} \log D_{\sigma^2}(\mathbf{z}), \quad (3)$$

where $D(\mathbf{z}) = (D_\mu(\mathbf{z}), D_{\sigma^2}(\mathbf{z}))$.

Extension to Gaussian Mixture Models. PCIDA can be naturally extended from a single Gaussian to a Gaussian mixture model (GMM) by using a mixture density network as the discriminator D (Bishop, 1994) and the corresponding negative log-likelihood as $L_d(\cdot, \cdot)$.

4. Theoretical Results

In this section, we provide theoretical guarantees for CIDA and PCIDA. As standard in adversarial domain adaption, we analyze a game in which the encoder aims to fool the discriminator and prevent it from inferring the domain index. We first analyze a simplified game between the encoder and the discriminator (without the predictor) to gain insight of the aligned encodings. We then discuss the full three-player game and show our framework preserves the prediction power while aligning the encodings.

4.1. Analysis for the Simplified Game

We consider a simplified game which does not involve the predictor F , defined by the $V_d(E, D)$ term in Eqn. 1:

$$\max_E \min_D V_d(E, D) = \mathbb{E}[L_d(D(E(\mathbf{x}, u)), u)]. \quad (4)$$

We first analyze the equilibrium of the simplified game for CIDA. Recall that, in CIDA, the discriminator D predicts the domain index u given the encoding \mathbf{z} and the domain index loss L_d is the L_2 loss. We show that in the equilibrium of CIDA, the encoder will align the mean of the conditional domain distribution $p(u|\mathbf{z})$ to the mean of the marginal domain distribution $p(u)$.

Lemma 4.1 below analyzes the discriminator D with the encoder E fixed and states that the optimal discriminator D outputs the mean domain index of all data with the same encoding \mathbf{z} .

Lemma 4.1 (Optimal Discriminator for CIDA). *For E fixed, the optimal D is*

$$D_E^*(E(\mathbf{x}, u)) = \mathbb{E}_{u \sim p(u|\mathbf{z})}[u],$$

where $\mathbf{z} = E(\mathbf{x}, u)$.

Proof. With E fixed, the optimal D

$$\begin{aligned} D_E^* &= \operatorname{argmin}_D \mathbb{E}_{(\mathbf{x}, u) \sim p(\mathbf{x}, u)}[L_d(D(E(\mathbf{x}, u)), u)] \\ &= \operatorname{argmin}_D \mathbb{E}_{(\mathbf{z}, u) \sim p(\mathbf{z}, u)}[\|D(\mathbf{z}) - u\|_2^2] \\ &= \operatorname{argmin}_D \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \mathbb{E}_{u \sim p(u|\mathbf{z})}[\|D(\mathbf{z}) - u\|_2^2] \end{aligned}$$

Notice that

$$\begin{aligned} &\mathbb{E}_{u \sim p(u|\mathbf{z})}[(D(\mathbf{z}) - u)^2] \\ &= \mathbb{E}_{u \sim p(u|\mathbf{z})}[u^2] - 2D(\mathbf{z})\mathbb{E}_{u \sim p(u|\mathbf{z})}[u] + D(\mathbf{z})^2, \end{aligned}$$

is a quadratic form of $D(\mathbf{z})$ which achieves the minimum at $D(\mathbf{z}) = \mathbb{E}_{u \sim p(u|\mathbf{z})}[u]$. \square

Assuming that D always achieves its optimum w.r.t E during the training, the minimax game in Eqn. 4 can be reformulated as maximizing $C_d(E)$ where

$$\begin{aligned} C_d(E) &\triangleq \min_D V_d(E, D) = V_d(E, D_E^*) \\ &= \mathbb{E}_{(\mathbf{x}, u) \sim p(\mathbf{x}, u)} (\mathbb{E}_{u \sim p(u|\mathbf{z})}[u] - u)^2 \\ &= \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \mathbb{E}_{u \sim p(u|\mathbf{z})} (\mathbb{E}_{u \sim p(u|\mathbf{z})}[u] - u)^2 \\ &= \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \mathbb{V}_{u \sim p(u|\mathbf{z})}[u] = \mathbb{E}_{\mathbf{z}} \mathbb{V}[u|\mathbf{z}], \end{aligned}$$

where \mathbb{V} denotes variance.

Next we analyze the virtual training criterion $C_d(E)$ for the encoder and derive the global optimum.

Lemma 4.2 (Uniqueness of Constant Expectation). *If there exists a constant μ_c such that $\mathbb{E}_{u \sim p(u|\mathbf{z})}[u] = \mu_c$ for any \mathbf{z} , we have $\mu_c = \mathbb{E}_{u \sim p(u)}[u]$.*

Theorem 4.1 (Global Optimum for CIDA). *The global maximum of $C_d(E)$ is achieved if and only if the encoder E satisfies that the expectations of the domain index u over the conditional distribution $p(u|\mathbf{z})$ for any given \mathbf{z} are identical to the expectation over the marginal distribution $p(u)$, i.e., $\mathbb{E}[u|\mathbf{z}] = \mathbb{E}[u], \forall \mathbf{z}$.*

Proof. We first show $C_d(E) \leq \mathbb{V}[u]$ and then show the equality is achieved when $\mathbb{E}[u|\mathbf{z}] = \mathbb{E}[u], \forall \mathbf{z}$.

$$\begin{aligned} C_d(E) - \mathbb{V}[u] &= \mathbb{E}_{\mathbf{z}} \mathbb{V}[u|\mathbf{z}] - \mathbb{V}[u] \\ &= \mathbb{E}_{\mathbf{z}} [\mathbb{E}[u^2|\mathbf{z}] - \mathbb{E}[u|\mathbf{z}]^2] - (\mathbb{E}[u^2] - \mathbb{E}[u]^2) \\ &= \mathbb{E}[u]^2 - \mathbb{E}_{\mathbf{z}} [\mathbb{E}[u|\mathbf{z}]^2]. \end{aligned}$$

By the convexity of x^2 and Jensen's inequality, we have $\mathbb{E}[u]^2 = (\mathbb{E}_{\mathbf{z}} [\mathbb{E}[u|\mathbf{z}]]))^2 \leq \mathbb{E}_{\mathbf{z}} [\mathbb{E}[u|\mathbf{z}]^2]$ and the equality is achieved when $\mathbb{E}[u|\mathbf{z}]$ is constant w.r.t. \mathbf{z} . By Lemma 4.2 we have $\mathbb{E}[u|\mathbf{z}] = \mathbb{E}[u], \forall \mathbf{z}$. \square

As Theorem 4.1 states, the vanilla CIDA using the L_2 loss guarantees that the mean of the distribution $p(u|\mathbf{z})$ matches the mean of the marginal distribution $p(u)$. It means that there is a risk the encoder E only aligns the mean of the distributions without exactly matching the entire distributions. However, surprisingly, we find that CIDA often achieves good empirical performance (see Sec. 5 for more details). Next, we analyze PCIDA and show that PCIDA enjoys better theoretical guarantees and matches both the mean and variance of the distribution $p(u|\mathbf{z})$.

Recall that in PCIDA, the discriminator D outputs the mean and variance of $p(u|\mathbf{z})$ as $D_\mu(\mathbf{z})$ and $D_{\sigma^2}(\mathbf{z})$. We use the negative log-likelihood (Eqn. 3) as the domain loss L_d . We start from analyzing the discriminator D when the encoder E is fixed. Lemma 4.3 states that the optimal discriminator D , given the encoding \mathbf{z} , will output the mean and variance of the domain index distribution $p(u|\mathbf{z})$.

Lemma 4.3 (Optimal Discriminator for PCIDA). *With E fixed, the optimal D is*

$$\begin{aligned} D_{\mu, E}^*(\mathbf{z}) &= \mathbb{E}_{u \sim p(u|\mathbf{z})}[u], \\ D_{\sigma^2, E}^*(\mathbf{z}) &= \mathbb{V}_{u \sim p(u|\mathbf{z})}[u], \end{aligned}$$

where $\mathbf{z} = E(\mathbf{x}, u)$, and $D = (D_\mu, D_{\sigma^2})$.

Proof of Lemma 4.3 is similar to that of Lemma 4.1 (see the Supplement for details).

Assuming discriminator D always reaches optimum, the virtual training criterion $C_d(E)$ for the encoder becomes:

$$\begin{aligned} C_d(E) &= \min_D V_d(E, D) = V_d(E, D_E^*) \\ &= \mathbb{E}_{\mathbf{z}, u} \left[\frac{(\mathbb{E}[u|\mathbf{z}] - u)^2}{2\mathbb{V}[u|\mathbf{z}]} + \frac{1}{2} \log(\mathbb{V}[u|\mathbf{z}]) \right]. \end{aligned}$$

Now we analyze $C_d(E)$ and provide PCIDA's global optimum.

Lemma 4.4 (Uniqueness of Constant Expectation and Variance). *If there exist constants μ_c and σ_c^2 such that $\mathbb{E}_{u \sim p(u|\mathbf{z})}[u] = \mu_c$ and $\mathbb{V}_{u \sim p(u|\mathbf{z})}[u] = \sigma_c^2$ for any \mathbf{z} , we have $\mu_c = \mathbb{E}_{u \sim p(u)}[u]$ and $\sigma_c^2 = \mathbb{V}_{u \sim p(u)}[u]$.*

Theorem 4.2 (Global Optimum for PCIDA). *In PCIDA (with the Gaussian model), the global optimum is achieved if and only if the mean and variance of the distribution $p(u|\mathbf{z})$ given any \mathbf{z} are identical to those of the marginal distribution $p(u)$.*

Proof. Given that

$$C_d(E) = \underbrace{\mathbb{E}_{\mathbf{z}, u} \left[\frac{(\mathbb{E}[u|\mathbf{z}] - u)^2}{2\mathbb{V}[u|\mathbf{z}]} \right]}_{C_1} + \underbrace{\mathbb{E}_{\mathbf{z}, u} \left[\frac{1}{2} \log(\mathbb{V}[u|\mathbf{z}]) \right]}_{C_2},$$

we analyze the upper bounds of the two terms separately. For the first term,

$$\begin{aligned} C_1 &= \mathbb{E}_{\mathbf{z}} \mathbb{E}_{u|\mathbf{z}} \left[\frac{(\mathbb{E}[u|\mathbf{z}] - u)^2}{2\mathbb{V}[u|\mathbf{z}]} \right] = \mathbb{E}_{\mathbf{z}} \left[\frac{\mathbb{E}_{u|\mathbf{z}} (\mathbb{E}[u|\mathbf{z}] - u)^2}{2\mathbb{V}[u|\mathbf{z}]} \right] \\ &= \mathbb{E}_{\mathbf{z}} \left[\frac{\mathbb{V}[u|\mathbf{z}]}{2\mathbb{V}[u|\mathbf{z}]} \right] = \mathbb{E}_{\mathbf{z}} \frac{1}{2} = \frac{1}{2}. \end{aligned}$$

For the second term, by the concavity of $\log(x)$ and Jensen's inequality, we have that $2C_2 \leq \log(\mathbb{E}_{\mathbf{z}} [\mathbb{V}[u|\mathbf{z}]])$ the equality holds when $\mathbb{V}[u|\mathbf{z}]$ is constant w.r.t. \mathbf{z} . Further, in the proof of Theorem 4.1, we show that $\mathbb{E}_{\mathbf{z}} [\mathbb{V}[u|\mathbf{z}]] \leq \mathbb{V}[u]$ and the maximum is achieved when $\mathbb{E}[u|\mathbf{z}]$ is constant w.r.t. \mathbf{z} . Together with Lemma 4.4, we then have that $C_d(E)$ reaches the global optimal $0.5 + 0.5 \log(\mathbb{V}[u])$ if and only if $\mathbb{E}[u|\mathbf{z}] = \mathbb{E}[u]$ and $\mathbb{V}[u|\mathbf{z}] = \mathbb{V}[u]$ for all \mathbf{z} . \square

Corollary 4.1. *For both CIDA and PCIDA, the global optimum of $C_d(E)$ is achieved if the encoding of all domains (continuously indexed by u) are totally aligned, i.e., $\mathbf{z} \perp\!\!\!\perp u$.*

Remark 4.1 (Matching Higher-Order Moments). By Theorem 4.1 and Theorem 4.2, we show that CIDA using the L_2 loss matches the mean of $p(u|\mathbf{z})$ while the PCIDA with the Gaussian model matches both the mean and variance of $p(u|\mathbf{z})$. Can we match higher-order moments? We believe our methodology can generalize to match higher-order moments by using PCIDA with more complex parametric probabilistic models. For example, one can use skew-normal distributions (Azzalini, 2013) to match the third moment (skewness).

4.2. Analysis of the Three-player Game

We analyze the equilibrium state of the three-player game of E, F and D as defined in Eqn. 1. We divide the situation into two cases based on whether the domain index u is independent of the label y .

4.2.1. $u \perp\!\!\!\perp y$

The domain index u is independent of the label y when it captures nuisance variations that are irrelevant to the task of predicting the label y . In this case, we prove the following theorem showing that the optimal encoding captures all the information in the input x that is relevant to the predictive tasks while aligning the domain index distributions.

Lemma 4.5 (Optimal Predictor). Given the encoder E , the prediction loss $V_p(F, E) \triangleq L_p(F(E(\mathbf{x}, u)), y) \geq H(y|E(\mathbf{x}, u))$ where $H(\cdot)$ is the entropy. The optimal predictor F^* that minimizes the prediction loss is $F^*(E(\mathbf{x}, u)) = P_y(\cdot|E(\mathbf{x}, u))$.

Assuming the predictor F and the discriminator D are trained to achieve their optimal losses, by Lemma 4.5, the three-player game (Eqn. 1) can be rewritten as following training procedure of the encoder E ,

$$\min_E C(E) \triangleq H(y|E(\mathbf{x}, u)) - \lambda_d C_d(E). \quad (5)$$

Theorem 4.3. If the encoder E , the predictor F and the discriminator D have enough capacity and are trained to reach optimum, any global optimal encoder E^* has the following properties:

$$H(y|E^*(\mathbf{x}, u)) = H(y|\mathbf{x}, u) \quad (6a)$$

$$C_d(E^*) = \max_{E'} C_d(E') \quad (6b)$$

Proof. Since $E(\mathbf{x}, u)$ is a function of \mathbf{x}, u , by the data processing inequality, we have $H(y|E(\mathbf{x}, u)) \geq H(y|\mathbf{x}, u)$.

Hence, $C(E) = H(y|E(\mathbf{x}, u)) - \lambda_d C_d(E) \geq H(y|\mathbf{x}, u) - \lambda_d \max_{E'} C_d(E')$. The equality holds if and only if $H(y|\mathbf{x}, u) = H(y|E(\mathbf{x}, u))$ and $C_d(E) = \max_{E'} C_d(E')$. Therefore, we only need to prove that the optimal value of $C(E)$ is equal to $H(y|\mathbf{x}, u) - \lambda_d \max_{E'} C_d(E')$ in order to

prove that any global encoder E^* satisfies both Eqn. 6a and Eqn. 6b.

We show that $C(E)$ can achieve $H(y|\mathbf{x}, u) - \lambda_d \max_{E'} C_d(E')$ by considering the following encoder E_0 : $E_0(\mathbf{x}, u) = P_y(\cdot|\mathbf{x}, u)$. It can be examined that $H(y|E_0(\mathbf{x}, u)) = H(y|\mathbf{x}, u)$ and $E_0(\mathbf{x}, u) \perp\!\!\!\perp u$ which leads to $C(E_0) = \max_{E'} C(E')$ using Corollary 4.1. \square

Theorem 4.3 shows that, at the equilibrium, the optimal encoder preserves all the information about label y contained in the data \mathbf{x} and the domain index u while aligning the encoding cross domains.

Note that in general the encoder E is a probabilistic encoder that generates \mathbf{z} stochastically. For example, one can use a probabilistic encoder parameterized by a natural-parameter network (Wang et al., 2016) and generate \mathbf{z} using the reparameterization trick (Kingma & Welling, 2013). Empirically, we find that directly using a deterministic encoder also works favorably and therefore keep the encoder deterministic in Sec. 5 for simplicity.

4.2.2. $u \not\perp\!\!\!\perp y$

The domain index u is dependent of the label y when it contains information relevant to predicting y . In this case, discretization of the inherently continuous domain index u is necessary to perform categorical domain adaptation. However, this discretization inevitably loses information in u and could hurt the predictive task since $u \not\perp\!\!\!\perp y$. In contrast, our methods CIDA/PCIDA performs domain adaption with the continuous domain index u , thus, can fully retain information in u that is relevant to the label y .

5. Experiments

We evaluate CIDA and its variants on two toy datasets, one image dataset (*Rotating MNIST*), and three real-world medical datasets. These empirical studies verify our theoretical findings in Sec. 3 and show that:

- Using categorical domain adaption to align continuously indexed domains leads to poor alignment with marginal (or no) performance gain compared to no adaptation.
- CIDA aligns domains with continuous indices and achieves significant performance boost compared to categorical domain adaption methods.
- PCIDA’s ability to predict a distribution rather than a single value is helpful in avoiding bad equilibriums and improving prediction performance.
- The performance gains of CIDA and PCIDA increase with multi-dimensional domain indices.

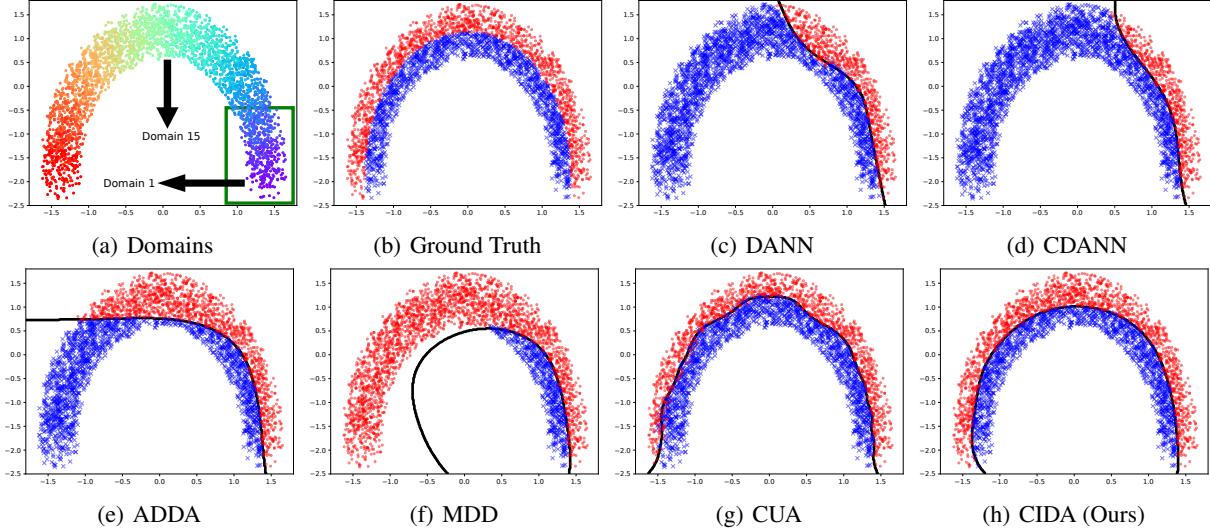


Figure 1. Results on the *Circle* dataset with 30 domains. Fig. 1(a) shows domain index by color. The first 6 domains are source domains, marked by green boxes. Red dots and blue crosses are positive and negative data samples. Black lines show the decision boundaries generated according to model predictions.

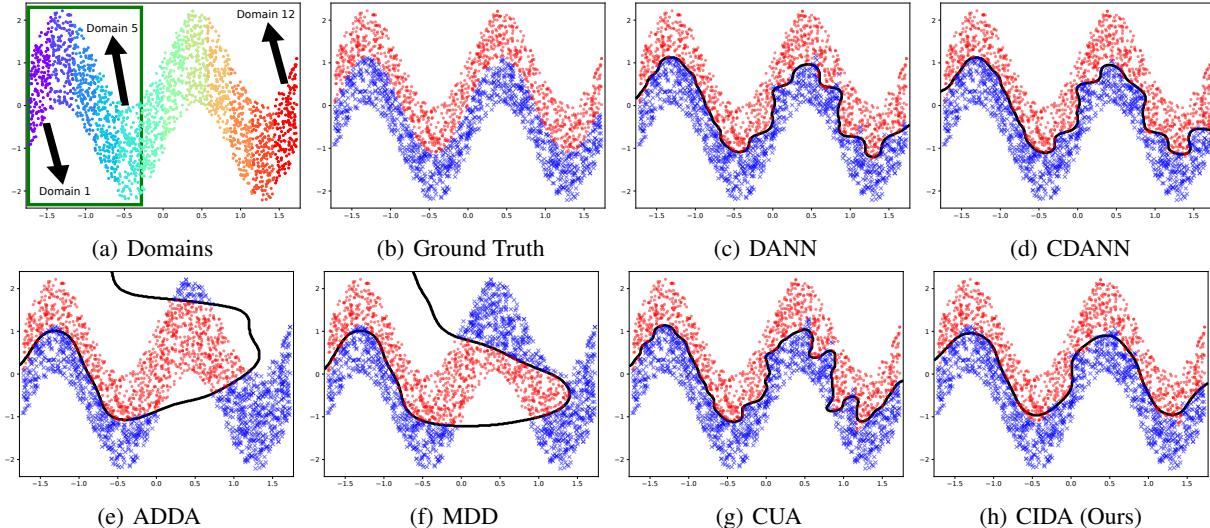


Figure 2. Results on the *Sine* dataset with 12 domains. The first 5 domains are source domains marked by green boxes. Red dots and blue crosses are positive and negative data samples. Black lines show the decision boundaries generated according to model predictions.

5.1. Baselines and Implementations

We compare variants of CIDA with state-of-the-art domain adaptation methods including Domain Adversarial Neural Network (**DANN**) (Ganin et al., 2016), Conditional Domain Adversarial Neural Network (**CDANN**) (Zhao et al., 2017), Adversarial Discriminative Domain Adaptation (**ADDA**) (Tzeng et al., 2017), Margin Disparity Discrepancy (**MDD**) (Zhang et al., 2019), and Continuous Unsupervised Adaptation (**CUA**) (Bobu et al., 2018). ADDA and MDD merge data with different domain indices into one source and one target domains; DANN, CDANN, and CUA divide the continuous domain spectrum into several separate domains and perform adaptation between multiple source and target domains. CUA adapts from the source domains to

each target domain one-by-one from the closest target to the farthest one. For a fair comparison with CIDA, all baselines use both x and the domain index u as inputs to the encoder.

All methods are implemented using PyTorch (Paszke et al., 2019) with the same neural network architecture. λ_d is chosen from $\{0.2, 0.5, 1.0, 2.0, 5.0\}$ and kept the same for all tasks associated with the same dataset (see the Supplement for more details about training).

5.2. Toy Datasets

To gain insight into the differences between CIDA and the baselines, we start with two toy datasets: *Circle* and *Sine*.

Circle Dataset includes 30 domains indexed from 1 to 30.

Table 1. Rotating MNIST accuracy (%) for various adaptation methods. We report the accuracy at the source domain and each target domain. X° denotes the domain whose images are Rotating by X° to $X + 45^\circ$. The last column shows the average accuracy across target domains. We use **bold face** to mark the best results.

Method	# Target Domains	0° (Source)	45°	90°	135°	180°	225°	270°	315°	Average
Source-Only	-	98.4	81.4	29.8	33.6	41.4	39.0	30.4	81.1	48.1
ADDA	1	95.0	70.2	25.5	44.0	59.2	46.2	23.7	61.4	47.2
DANN	1	98.1	80.1	44.8	42.2	43.6	46.8	57.3	79.3	56.3
CUA	7	91.4	73.9	60.1	55.0	52.7	45.1	55.2	88.4	61.5
CIDA (Ours)	∞	99.1	87.2	56.7	79.6	91.2	91.5	96.2	97.5	85.7
PCIDA (Ours)	∞	98.6	90.1	82.2	90.5	91.9	87.1	80.0	88.2	87.1

Fig. 1(a) shows the 30 domains in different colors. We also use arrows to indicates domain 1 and domain 15. Each domain contains data on a circle. The task is binary classification. Fig. 1(b) shows positive samples as red dots and negative samples as blue crosses. As shown in the figure, the ground-truth decision boundary continuously evolves with the domain index. We use domains 1 to 6 as source domains and the rest as target domains. Fig. 1 compares the results of CIDA with the baselines. The figure shows that overall categorical DA methods perform poorly when asked to align domains with continuous indices. CUA is the best performing baseline since it incrementally adapts 24 pairs of domains. Still, CUA’s performance is inferior to CIDA which produces a more accurate decision boundary.

Sine Dataset includes 12 domains as shown in Fig. 2(a). Each domain covers $\frac{1}{6}$ the period of the sinusoid. We consider the first 5 domains as source domains and the rest as target domains. Fig. 2 shows the results. The figure shows that it is very challenging for the baselines to capture the ground-truth decision boundary. The baselines either produce incorrect decision boundaries (ADDA and MDD) or only capture the correct trend with very rugged boundaries (DANN, CDANN and CUA). In contrast, CIDA can successfully recover the ground-truth boundary. We also note that while CUA performed better than the other baselines on the Circle dataset, it performed worse than DANN and CDANN on the Sine dataset. This is because CUA performs incremental pairwise adaptation; it fails on the pair (5, 9), and this failure propagates to the following domains.

Overall, both the results from the Circle and Sine datasets demonstrate that CIDA captures the underlying relationship between the domain index and the classification task and leverages it to improve performance. In contrast, the baselines cannot accurately capture this relationship, and hence yield worse results.

5.3. Rotating MNIST

We further evaluate our methods on the *Rotating MNIST* dataset. The goal is to adapt from regular MNIST digits with mild rotation to significantly Rotating MNIST digits. We designate images that are Rotating by 0° to 45° as the labeled source domain, and assign images Rotating by 45° to 360° to the target domains. Naturally, the

domain index is the rotation angle of the image. Since the target domain has a much larger range of rotation angles, we split the target domain into seven target domains for categorical domain adaptation baselines, DANN and CUA. These seven target domains contain images Rotating by [45, 90), [90, 135), …, [315, 360) degrees, respectively. Table 1 compares the accuracy our proposed CIDA/PCIDA with different baselines. We can see ADDA and DANN hardly improve and sometimes even decrease the accuracy compared to not performing adaptation at all. This is because without capturing the underlying structure, adversarial encoding alignment may harm the transferability of the data. CUA’s performs fairly well in target domains near source domains but poorly in distant domains.³ On the other hand, CIDA and PCIDA can learn such domain structure and successfully adapt the knowledge from source domains to target domains (see the Supplement for more details such as model architectures).

5.4. Healthcare Datasets

Dataset Description. We use three medical datasets, Sleep Heart Health Study (*SHHS*) (Quan et al., 1997), Multi-Ethnic Study of Atherosclerosis (*MESA*) (Zhang et al., 2018) and Study of Osteoporotic Fractures (*SOF*) (Cummings et al., 1990). Each dataset contains full-night breathing signals of subjects and the corresponding sleep stage labels (‘Awake’, ‘Light Sleep’, ‘Deep Sleep’, and ‘Rapid Eye Movement (REM)'). Breathing signals are split into 30-second segments with one label for each segment. We consider the task of sleep stage prediction, i.e., to predict the sleep stage label y given a breathing segment x . This is a natural task in sleep studies and can be performed in the patient home by having them wearing a breathing belt. The breathing signal can then serve to predict sleep stages and also detect apnea (temporary cessation of breathing).

The datasets also contain subjects’ information such as age, which is a natural domain index u . *SHHS*, *MESA*, and *SOF* include 2,651, 2,055, and 453 subjects, respectively. On av-

³We note that CUA’s performance on our Rotating MNIST data is worse than in the original papers, possibly because our Rotating MNIST has images rotated by all angles as opposed to only 8 fixed angles. Also we are using different model architectures. Please refer to the Supplement for details.

Table 2. Accuracy (%) for intra-dataset adaptation. ‘SHHS@Outside → SHHS@(52,75]’ means transferring from age range outside (52,75] (i.e., [44,52] ∪ (75,90]) to (52,75] within SHHS. ‘SO’ is short for ‘Source-Only’. We use **bold face** mark the best results.

	Task	SO	ADDA	DANN	CDANN	MDD	CUA	CIDA	PCIDA
Domain Extrapolation	SHHS@[44,52] → SHHS@(52,90]	77.4	78.0	77.1	77.5	77.7	77.4	79.8	80.6
	MESA@[54,58] → MESA@(58,95]	80.1	80.7	79.9	80.4	80.3	80.1	82.7	82.5
	SOF@[75,82] → SOF@(82,90]	74.7	74.8	74.2	74.4	74.6	74.5	76.7	76.7
Domain Interpolation	SHHS@Outside → SHHS@(52,75]	82.4	81.7	82.5	82.3	82.5	82.4	82.2	83.7
	MESA@Outside → MESA@(58,75]	83.5	83.5	83.2	83.3	83.8	83.4	83.5	84.7
	SOF@Outside → SOF@(79,86]	71.8	71.5	71.4	70.9	71.8	71.5	71.8	73.6

Table 3. Accuracy (%) for cross-dataset adaptation. We use **bold face** to mark the best results.

Task	Source-Only	ADDA	DANN	CDANN	MDD	CUA	CIDA	PCIDA
SOF → SHHS	75.6	76.0	75.2	75.6	75.8	75.3	75.9	80.1
SOF → MESA	74.0	75.1	74.6	75.2	74.9	73.6	74.8	80.0
SHHS → MESA	82.8	83.0	82.6	82.1	83.0	82.1	83.2	85.3
MESA → SHHS	80.7	81.8	80.9	80.9	81.2	81.0	80.8	83.4
SHHS → SOF	78.7	79.5	79.0	79.2	79.7	79.1	81.1	80.9
MESA → SOF	75.9	76.6	77.0	76.9	76.9	76.0	79.3	79.0

erage, there are 1,000 segments (i.e., 8.33 hours of breathing signals) for each subject. Different datasets have different domain index distributions. For example, subjects’ age range in *SHHS* is [44, 90], while the age ranges for *MESA* and *SOF* are [54, 95] and [72, 90], respectively. Apparently *SOF* subjects are much older. *SHHS* subjects and *MESA* subjects have similar age ranges but the distributions are actually different (see the histogram plot in the Supplement).

Intra-Dataset Adaptation. We first evaluate our methods’ performance on adaptation across continuously indexed domains within the same dataset using ‘age’ as the domain index. We cover two cases:

- **Domain Extrapolation.** For example, the source domain has data with a domain index (age) from the range [44,52], while the target domain contains data with a domain index range of (52,90].
- **Domain Interpolation.** For example, in the source domain, the domain index range is [44,52] ∪ (75,90], while in the target domain, the domain index range is (52,75].

The first three rows of Table 2 show the results for domain extrapolation. One observation is that directly using categorical domain adaptation only achieves minimal performance boost compared to models trained only on the source domains (Source-Only). Some methods such as DANN and CUA achieve no or even negative performance improvement. On the other hand, CIDA variants can successfully transfer across subjects with different ages and significantly improve upon all baselines. Similarly, the last three rows in Table 2 show the results for domain interpolation. Note that Source-Only can already achieve satisfactory accuracy in domain interpolation, since the model naturally learns the average of data from both sides (e.g., [44,52] ∪ (75,90]) and performs prediction for the data in the middle (e.g., (52,75]). For example, in the task ‘SHHS@Outside → SHHS@(52,75]’, Source-Only already has a high accuracy of 82.4%, leaving little room for improvement. Interestingly, PCIDA can still

Table 4. Accuracy (%) for the multi-dimensional domain index setting. The task is SHHS@[44,52] → SHHS@(52,90].

# Dimensions	Source-Only	CIDA	PCIDA
1	77.4	79.8	80.6
2	77.6	81.0	81.1
4	77.7	81.2	81.3
11	77.7	82.6	82.6

further improve the accuracy by a tangible margin. This also shows PCIDA’s ability to avoid bad equilibriums by using a discriminator that predicts distributions rather than values.

Cross-Dataset Adaptation. Most clinical trials collect data from a population with a specific medical condition, and exclude people who have other conditions. However in practice many patients have multiple medical conditions and hence doctors need to apply the results of a particular study outside the population for which the data is collected. Thus, in this section we consider cross-dataset adaption. Specifically, we evaluate how different methods perform when transferring among the datasets *SHHS*, *MESA*, and *SOF*. Table 3 shows the accuracy of all methods in these cross-dataset settings. We observe that categorical domain adaptation barely improves upon models trained with only source domains, while CIDA and PCIDA can naturally transfer across continuously indexed domains even in the cross-dataset setting with significant performance improvement. Interestingly, when the task is hard such as transferring from *SOF*, a very old people dataset, to datasets with much more diverse age range, PCIDA becomes a clear winner. But when the task is relatively easier, such as transferring from datasets with diverse age range to a very old dataset, CIDA is marginally better than PCIDA; but, this latter difference in performance is minor, and PCIDA performs well across all scenarios.

We also note that *SHHS* and *MESA* are both diverse datasets with similar age distribution, which is why the Source-Only model already achieves high accuracy. Interestingly, even

in such cases PCIDA can still achieve stable performance gain compared to all baselines.

Multi-Dimensional Indices. As mentioned in Sec. 3, both CIDA and PCIDA naturally generalize to multi-dimensional domain indices. To demonstrate this feature, we leverage that the *SHHS* dataset includes multiple variables per patient in addition to their age, such as their heart rate, their physical and emotional health scores, etc. We combine such variables with the person’s age to create a multi-dimensional domain index. (see more details on different domain indices in the Supplement). Table 4 shows the accuracy for multi-dimensional CIDA/PCIDA. For reference, we report corresponding accuracy for Source-Only. Note that Source-Only takes both the breathing signals (x) and the domain index (u) as input, as is done in all previous experiments. As expected we can observe substantial improvement in accuracy with multi-dimensional domain indices.

Note that in the case of multi-dimensional indices, *domain extrapolation* means that the target domain indices are *outside* the convex hull of the source domain indices. Similarly, *domain interpolation* means that the target domain indices are *inside* the convex hull of the source domain indices.

6. Conclusion

We identify the problem of adaptation across continuously indexed domains, propose a series of methods for addressing it, and provide supporting theoretical analysis and empirical results using both synthetic and real-world medical data. Our work demonstrates the viability of efficient adaptation across continuously indexed domains and its potential impact on important real-world applications. Future work could investigate the possibility of matching higher or even infinite order moments, and the application of the proposed methods to other datasets in robotics or the medical field.

Acknowledgement

We thank Guang-He Lee and Mingmin Zhao for the insightful and tremendously helpful discussions. We are also grateful to Xingjian Shi, Xiaomeng Li, Hongzi Mao as well as other members at NETMIT and CSAIL for their comments to improve this paper. We would also like to thank Daniel R. Mobley and NSRR for their help with the datasets.

References

Azzalini, A. *The skew-normal and related families*, volume 3. Cambridge University Press, 2013.

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.

Bishop, C. M. Mixture density networks. 1994.

Bitarafan, A., Baghshah, M. S., and Gheisari, M. Incremental evolving domain adaptation. *TKDE*, 28(8):2128–2141, 2016.

Bobu, A., Tzeng, E., Hoffman, J., and Darrell, T. Adapting to continuously shifting domains. 2018.

Cummings, S. R., Black, D. M., Nevitt, M. C., Browner, W. S., Cauley, J. A., Genant, H. K., Mascioli, S. R., Scott, J. C., Seeley, D. G., Steiger, P., et al. Appendicular bone density and age predict hip fracture in women. *JAMA*, 263(5):665–668, 1990.

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

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NIPS*, pp. 2672–2680, 2014.

Hoffman, J., Darrell, T., and Saenko, K. Continuous manifold based adaptation for evolving visual domains. In *CVPR*, pp. 867–874, 2014.

Jimenez, G. O., Gheche, M. E., Simou, E., Maretic, H. P., and Frossard, P. Cdots: Continuous domain adaptation using optimal transport. *arXiv preprint arXiv:1909.11448*, 2019.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kuroki, S., Charoenphakdee, N., Bao, H., Honda, J., Sato, I., and Sugiyama, M. Unsupervised domain adaptation based on source-guided discrepancy. In *AAAI*, pp. 4122–4129, 2019.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. In *NIPS*, pp. 1647–1657, 2018.

Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. Domain adaptation via transfer component analysis. *TNN*, 22(2):199–210, 2010.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, pp. 8024–8035, 2019.

- Quan, S. F., Howard, B. V., Iber, C., Kiley, J. P., Nieto, F. J., O’connor, G. T., Rapoport, D. M., Redline, S., Robbins, J., Samet, J. M., et al. The sleep heart health study: design, rationale, and methods. *Sleep*, 20(12):1077–1085, 1997.
- Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 3723–3732, 2018.
- Sankaranarayanan, S., Balaji, Y., Castillo, C. D., and Chellappa, R. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, pp. 8503–8512, 2018.
- Sun, B. and Saenko, K. Deep CORAL: correlation alignment for deep domain adaptation. In *ICCV workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, pp. 443–450, 2016.
- Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
- Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *CVPR*, pp. 7167–7176, 2017.
- Wang, H., Shi, X., and Yeung, D. Natural-parameter networks: A class of probabilistic neural networks. In *NIPS*, pp. 118–126, 2016.
- Wulfmeier, M., Bewley, A., and Posner, I. Incremental adversarial domain adaptation for continually changing environments. In *ICRA*, pp. 1–9, 2018.
- Zhang, G.-Q., Cui, L., Mueller, R., Tao, S., Kim, M., Rueschman, M., Mariani, S., Mobley, D., and Redline, S. The national sleep research resource: towards a sleep data commons. *JAMA*, 25(10):1351–1358, 2018.
- Zhang, Y., Liu, T., Long, M., and Jordan, M. I. Bridging theory and algorithm for domain adaptation. *arXiv preprint arXiv:1904.05801*, 2019.
- Zhao, H., Zhang, S., Wu, G., Moura, J. M. F., Costeira, J. P., and Gordon, G. J. Adversarial multiple source domain adaptation. In *NIPS*, pp. 8568–8579, 2018.
- Zhao, H., des Combes, R. T., Zhang, K., and Gordon, G. J. On learning invariant representations for domain adaptation. In *ICML*, pp. 7523–7532, 2019.
- Zhao, M., Yue, S., Katabi, D., Jaakkola, T. S., and Bianchi, M. T. Learning sleep stages from radio signals: A conditional adversarial architecture. In *ICML*, pp. 4100–4109, 2017.

Supplementary Materials: Continuously Indexed Domain Adaptation

1 Proof

Lemma 1.1 (Uniqueness of Constant Expectation). \mathbf{z} and u are random variables. If $\mathbb{E}_{u \sim p(u|\mathbf{z})}[u]$ is constant w.r.t \mathbf{z} , then $\mathbb{E}_{u \sim p(u|\mathbf{z})}[u] = \mathbb{E}_{u \sim p(u)}[u], \forall \mathbf{z}$.

Proof. Let $\mathbb{E}_{u \sim p(u|\mathbf{z})}[u] = \mu, \forall \mathbf{z}$. We then have $\mathbb{E}_{u \sim p(u)}[u] = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}\mathbb{E}_{u \sim p(u|\mathbf{z})}[u] = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}\mu = \mu$. \square

Lemma 1.2 (Uniqueness of Constant Expectation and Variance). \mathbf{z} and u are random variables. If $\mathbb{E}_{u \sim p(u|\mathbf{z})}[u]$ and $\mathbb{V}_{u \sim p(u|\mathbf{z})}[u]$ are constants w.r.t \mathbf{z} , then $\mathbb{E}_{u \sim p(u|\mathbf{z})}[u] = \mathbb{E}_{u \sim p(u)}[u]$ and $\mathbb{V}_{u \sim p(u|\mathbf{z})}[u] = \mathbb{V}_{u \sim p(u)}[u]$ for any \mathbf{z} .

Proof. Let $\mathbb{E}_{u \sim p(u|\mathbf{z})}[u] = \mu$ and $\mathbb{V}_{u \sim p(u|\mathbf{z})}[u] = \sigma^2$ for any \mathbf{z} . By the previous lemma, we have $\mathbb{E}_{u \sim p(u)}[u] = \mu$. For the variance, we have:

$$\begin{aligned}\mathbb{V}_{u \sim p(u)}[u] &= \mathbb{E}_{u \sim p(u)}[(u - \mathbb{E}[u])^2] = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}\mathbb{E}_{u \sim p(u|\mathbf{z})}[(u - \mathbb{E}[u|\mathbf{z}])^2] \\ &= \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}\mathbb{V}_{u \sim p(u|\mathbf{z})}[u] = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}\sigma^2 = \sigma^2,\end{aligned}$$

concluding the proof. \square

Lemma 1.3 (Optimal Discriminator for PCIDA). With E fixed, the optimal D is

$$\begin{aligned}D_{\mu, E}^*(\mathbf{z}) &= \mathbb{E}_{u \sim p(u|\mathbf{z})}[u], \\ D_{\sigma^2, E}^*(\mathbf{z}) &= \mathbb{V}_{u \sim p(u|\mathbf{z})}[u],\end{aligned}$$

where $\mathbf{z} = E(\mathbf{x}, u)$.

Proof. The optimal D :

$$D_E^*(\mathbf{x}, u) = \operatorname{argmin}_D \mathbb{E}_{(\mathbf{z}, u) \sim p(\mathbf{z}, u)}[L_d(D(\mathbf{z}), u)],$$

where the objective function expands to

$$\begin{aligned}&\mathbb{E}_{(\mathbf{z}, u) \sim p(\mathbf{z}, u)}[L_d((D_\mu(\mathbf{z}), D_{\sigma^2}(\mathbf{z})), u)] \\ &= \mathbb{E}_{(\mathbf{z}, u) \sim p(\mathbf{z}, u)}\left[\frac{(D_\mu(\mathbf{z}) - u)^2}{2D_{\sigma^2}(\mathbf{z})} + \frac{1}{2} \log D_{\sigma^2}(\mathbf{z})\right] \\ &= \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}\mathbb{E}_{u \sim p(u|\mathbf{z})}\left[\frac{(D_\mu(\mathbf{z}) - u)^2}{2D_{\sigma^2}(\mathbf{z})} + \frac{1}{2} \log D_{\sigma^2}(\mathbf{z})\right].\end{aligned}$$

Notice that

$$\begin{aligned}&\mathbb{E}_{u \sim p(u|\mathbf{z})}\left[\frac{(D_\mu(\mathbf{z}) - u)^2}{2D_{\sigma^2}(\mathbf{z})} + \frac{1}{2} \log D_{\sigma^2}(\mathbf{z})\right] \\ &= \frac{\mathbb{E}[u^2|\mathbf{z}]}{2D_{\sigma^2}(\mathbf{z})} - \frac{D_\mu(\mathbf{z})}{D_{\sigma^2}(\mathbf{z})}\mathbb{E}[u|\mathbf{z}] + \frac{D_\mu(\mathbf{z})^2}{2D_{\sigma^2}(\mathbf{z})} + \frac{1}{2} \log D_{\sigma^2}(\mathbf{z}).\end{aligned}$$

Taking the derivative w.r.t. $D(\mathbf{z})$ and setting it to 0, we get the optimal $D_{\mu, E}^*(\mathbf{z}) = \mathbb{E}[u|\mathbf{z}]$ and $D_{\sigma^2, E}^*(\mathbf{z}) = \mathbb{V}[u|\mathbf{z}]$, completing the proof. \square

Table 1: 11 domain indices in the *SHHS* dataset.

u_1	Age
u_2	Resting heart rate
u_3	Gender
u_4	Physical functioning
u_5	Role limitation due to physical health
u_6	General health
u_7	Role limitation due to emotional problems
u_8	Energy/fatigue
u_9	Emotional well being
u_{10}	Social functioning
u_{11}	Pain Level

Table 2: Network structure for the encoder.

Kernel	Stride	Channel In	Channel Middle	Channel Out	Type	Number
13	2	1	-	64	Conv	1
9	1	64	64	64	ResBlock	1
9	2	64	64	128	ResBlock	1
9	1	128	128	128	ResBlock	1
7	1	128	128	256	ResBlock	1
9	5	256	256	256	ResBlock	1
5	1	256	256	512	ResBlock	1
5	1	512	512	512	ResBlock	1
5	1	512	384	384	ResBlock	1
9	5	384	384	384	ResBlock	1
3	1	384	384	384	ResBlock	1
5	1	384	384	384	ResBlock	1

2 Experiments

In this section we provide more details for our experiments. The code is available at <https://github.com/hehaodele/CIDA>.

2.1 Experiment on the Healthcare Datasets

Dataset details. The three real-world medical datasets [?, ?, ?] with detailed information are publicly available¹. They can all be freely accessed upon request and submission of relevant IRB documents. In Fig. 1 we plot the histograms subjects' age in the three medical datasets. All the three datasets contains many health related variables of the subjects. In Table 1, we list all the variables we considered as the domain indices.

Implementations. We use the same neural network architecture in all methods for fair comparison. Table 2 shows the neural network architecture for the encoders taking breathing signals \mathbf{x} as input. ‘Number’ in the tables indicates the number of corresponding blocks stacked in the network. The predictor includes 3 fully connected layers, each with batch normalization and ReLU. Similarly, the discriminator includes 5 fully connected layers. For the baseline models, we explore different λ_d (the hyperparameter for the discriminator term) in the range $\{0.2, 0.5, 1.0, 2.0, 5.0\}$ and find that $\lambda_d = 2.0$ produce stable and the best results in the toy datasets. We follow recommendations from the original papers for other hyperparameters. We set $\lambda_d = 2.0$ for all methods including CIDA/PCIDA. We train all models using the Adam optimizer [?] with a learning rate of 10^{-4} . We run all experiments on a server with four NVIDIA Titan Xp GPUs.

2.2 Experiment on the Rotating MNIST

Dataset details. In our *Rotating MNIST*, there are 60,000 images in each domain index interval spanning 45 degrees from $[0^\circ, 45^\circ]$ to $[315^\circ, 360^\circ]$. They are generated by rotating each of the 60,000 image in the MNIST training set by a

¹<https://sleepdata.org/>

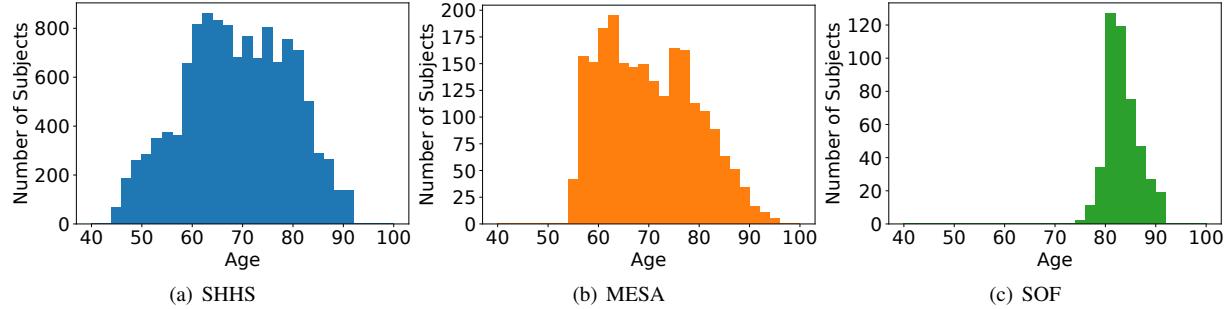


Figure 1: Age histograms for three medical datasets.

angle randomly sampled the corresponding domain index interval. Therefore, this dataset contains images with rotation angles evenly spread in the range $[0, 360^\circ]$. We note that this is different from the *Rotating MNIST* dataset in [?], where the images are Rotating by 8 fixed angles. Another difference is that in our Rotating MNIST, the amount of data in target domains is 7 times as many as that in source domain while in [?], the target domain has the same amount of data as the source domain.

Implementations. We use the same neural network architecture in all methods for fair comparison. Mainly, we use a four-layer convolutional neural networks to encode the image and a three-layer MLP to make the prediction, while the discriminator is a four-layer MLP. In addition, we make two augmentations to provide the model with a stronger inductive bias. First, we add a Spacial Transfer Network (STN) [?] to the image encoder. Basically, the STN will take the image and the domain index as input and output a set of rotation parameters which are then applied to rotate the given image. Second, we add the dropout layers to the STN and the ConvNet backbone. As mentioned in [?], dropout can be viewed as a way of performing Bayesian inference. Here, we use this dropout switch to make image encoder either deterministic or probabilistic. For more details, please refer to our code.

2.3 Experiments on the Toy Datasets

Visualization of the decision boundary (approximately). Unlike shallow models such as logistic regression, plotting deep neural networks’ exact decision boundaries is not straightforward. To generate a virtual decision boundary for visualization, we fit an SVM with the RBF kernel by neural networks’ prediction and draw the decision boundary of the SVM. To be fair, when fitting the SVM, we ensure that the fitting accuracy is the same for all deep learning models. Note that since the generated boundaries are not exact, we can observe some data points on the wrong side of the boundaries.

3 Discussion

3.1 Categorical Domains versus Continuously Indexed Domains

Continuous Indices. As mentioned in the main paper, the hypothesis of ‘continuous indices’ is that input \mathbf{x} and labels y are drawn from $p(\mathbf{x}, y|u)$ given a specific domain index $u \in \mathcal{U}$, and that $p(\mathbf{x}, y|u)$ (and $p(y|\mathbf{x}, u)$) is continuous w.r.t. u . Therefore, CIDA tries to produce correct predictions in a continuous range of target domains by effectively capturing the underlying relation (functional) between $p(y|\mathbf{x}, u)$ and u .

Distance Metrics. Such a hypothesis comes with a distance metric for domain indices, which are captured by the regression loss (e.g., euclidean distance for L_2 loss) in the discriminator. This is a key difference between CIDA and categorical domain adaptation, where any pair of domains effectively has the same distance. This is also true for categorical domain adaptation methods such as [?]. Note that [?] uses a least-square loss as a surrogate for cross-entropy to perform domain *classification* in the discriminator, therefore still treating different domains as equal. This is substantially different from CIDA where the L_2 loss and the Gaussian (or Gaussian Mixture Model) loss are used to regress the domain indices.

3.2 Matching $p(u|z)$ versus Matching $p(z|u)$

In general, matching the entire $p(u|z)$ for any z is equivalent to matching the entire $p(z|u)$ for any u . This is because $p(u|z) = p(u) \iff p(z|u) = p(z) \iff u \perp\!\!\!\perp z$. However, matching the mean and variance of $p(u|z)$ for any z is **different** from matching the mean and variance of $p(z|u)$ for any u . Considering the dimension of z is much higher than that of u , the former is actually **stronger** alignment.

To see this, consider a simplified case where $z \in \{1, 2, 3, 4\}^{100}$ and $u \in \{1, 2, 3, 4\}$. Matching the mean and variance of $p(u|z)$ requires matching the mean and variance of 4^{100} univariate distributions, i.e., 2×4^{100} parameters in total. On the other hand, Matching the mean and variance of $p(z|u)$ only requires matching the mean and variance of 4 100-dimensional distributions, i.e., 400 parameters in total. Therefore the former implies stronger alignment.