TOWARD UNDERSTANDING SUPERVISED REPRESENTATION LEARNING WITH RKHS AND GAN

Anonymous authors
Paper under double-blind review

ABSTRACT

The success of deep supervised learning depends on its automatic data representation abilities. A good representation of high-dimensional complex data should enjoy low-dimensionally and disentanglement while losing as little information as possible. This work gives a statistical understanding of how a deep representation goal can be achieved via reproducing kernel Hilbert spaces (RKHS) and generative adversarial networks (GAN). At the population level, we formulate the ideal representation learning task as that of finding a nonlinear map that minimizes the sum of losses characterizing conditional independence (via RKHS) and disentanglement (via GAN). We estimate the target map at the sample level with deep neural networks. We prove the consistency in terms of the population objective function value. We validate the proposed methods via comprehensive numerical experiments and real data analysis in the context of regression and classification. The resulting prediction accuracies are better than state-of-the-art methods.

1 INTRODUCTION

Over the past decade, deep learning has achieved remarkable successes in many fields such as computer vision and natural language processing (Krizhevsky et al., 2012; Graves et al., 2013; LeCun et al., 2015). A key factor for the success of deep learning is its automatic data representation capabilities (Bengio et al., 2013). For all desired characteristics of an ideal representation for a high-dimensional complex data, information preservation, low dimensionality and disentanglement are among the topmost (Achille & Soatto, 2018). A representation with these characteristics not only makes the model more interpretable but also facilitates the subsequent supervised learning tasks. First, information preservation requires that the learned features should be sufficient statistics for both estimation and prediction. This can be quantified via the concept of conditional independence. Second, low dimensionality means that we should use as few features as possible to represent the underlying structure of data, and the number of features should be fewer than the ambient dimension. Third, the learned features in the representation can often be interpreted as corresponding to the hidden causes of the observed data; thus disentanglement is an essential characteristic that distinguishes cause from others (Goodfellow et al., 2016).

It is widely believed that deep neural networks trained in supervised learning learn effective data representation automatically. For example, in the context of classification, the last layer of a deep neural network is a linear classifier and the preceding layers serve as a feature extractor to the classifier. However, in the supervised training of deep neural networks, the objective functions do not explicitly impose any conditions that guarantee the desired characteristics of learned representations. It appears that there are not any explicit guiding principles for understanding the black-box nature of deep neural networks for representation learning (Alain & Bengio, 2017).

In this paper, we propose a novel supervised representation learning approach (NSRL) to achieve the ideal representation, which in turn demystifies the success of deep neural networks. The key idea of NSRL is that we seek a nonlinear map from the high-dimensional input space to a lower-dimensional feature space such that the data and its label/response are conditionally independent given the value of the nonlinear map. Meanwhile, we regularize the nonlinear map by matching its pushforward distribution to a reference distribution with independent components such as standard Gaussian to achieve disentanglement in the representation. Therefore, the proposed NSRL is guaranteed to enjoy the three desired characteristics of an ideal data representation described above. For measuring the

conditional independence, we use the conditional covariance operator in reproducing kernel Hilbert space (RKHS) (Baker, 1973; Fukumizu et al., 2004) that can be estimated easily with samples. To further match the pushforward distribution under the target nonlinear map with the reference distribution, we use the GAN loss such as the the f-divergence for f-GAN (Nowozin et al., 2016) or the 1-Wasserstein distance for the WGAN (Arjovsky et al., 2017). Our main contributions are as follows:

- At the population level, we formulate the ideal representation learning task equivalently as finding a nonlinear map that minimizes the loss characterizing both information preservation and disentanglement via conditional covariance operator in RKHS and GAN, respectively.
- We estimate the target nonlinear map at the sample level nonparametrically with deep neural networks, and propose a novel supervised representation learning (NSRL) algorithm.
- We validate the proposed NSRL via comprehensive numerical simulations and a number of real datasets, i.e. Life Expectancy and Pole for regression, and MNIST, Kuzushiji-MNIST and CIFAR-10 for classification. We use the learned features from our NSRL as inputs for linear regression and nearest neighbor classification. The resulting prediction accuracies are better than those state-of-the-art methods, that is, linear dimension reduction models for regression and deep learning models for classification.

2 SETUP AND BACKGROUND

2.1 SETUP

Suppose we have a sample of n input-response/label observations $\{(X_i, Y_i) \subseteq \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times \mathbb{R}^1\}_{i=1}^n$ that are i.i.d. copies of (X, Y) with an unknown law $\mu_{X,Y}$. In many applications, high-dimensional complex data X such as images, texts and natural languages, tend to have low-dimensional representations (Bengio et al., 2013). Mathematically, we model this feature of high-dimensional complex data by assuming the existence of a nonlinear map $g^* : \mathbb{R}^d \to \mathbb{R}^{d^*}$ with $d^* \ll d$ such that the information of X can be completely encoded by g^* in the sense

$$X \perp\!\!\!\perp Y | g^*(X). \tag{1}$$

That is, Y and X are conditionally independent given $g^*(X)$. The representation $g^*(X)$ has much lower dimensionality than X and captures all the information about the statistical dependence of Y on X. Moreover, we would like to have the disentanglement property for the representation $g^*(X)$, that is, the potential hidden causes of the observed data. This can be achieved by transforming the distribution of $g^*(X)$ into standard Gaussian. To this end, we first recall a result in optimal transport theory (Brenier, 1991; McCann et al., 1995; Villani, 2008).

Lemma 2.1 *Let μ be a probability measures on $\mathbb{R}^{d^*}$ with second order moment and absolutely continuous with respect to the the Gaussian measure γ_{d^*}. Then it admits a unique optimal transportation map $\mathcal{T}^* : \mathbb{R}^{d^*} \to \mathbb{R}^{d^*}$ such that ${\mathcal{T}^*}_{\#}\mu = \gamma_{d^*} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{d^*})$. Moreover, $\mathcal{T}^*$ is injective μ-a.e.*

Thanks to Lemma 2.1, the map $\mathcal{T}^*$ in Lemma 2.1 transforms the distribution of $g^*(X)$ satisfying equation 1 to the standard normal distribution. Specifically, define

$$F^* = \mathcal{T}^* \circ g^* : \mathbb{R}^d \to \mathbb{R}^{d^*}.$$

Then we also have

$$X \perp\!\!\!\perp Y | F^*(X), \quad F^*(X) \sim \gamma_{d^*} = \mathcal{N}(\mathbf{0}, \mathbf{I}_{d^*}), \tag{2}$$

that is, F^* is a target nonlinear map that preserves both disentanglement and conditional independence.

Next we recall some background on conditional covariance operator in RKHS that is used to characterize conditional independence, and GAN losses that are used to measure the discrepancy of two probability measures.

2.2 CONDITIONAL COVARIANCE OPERATORS

Let $(\mathcal{H}_\mathcal{X}, k_\mathcal{X})$ and $(\mathcal{H}_\mathcal{Y}, k_\mathcal{Y})$ be RKHS of functions on $\mathcal{X}$ and $\mathcal{Y}$, respectively, with measurable positive definite kernels $k_\mathcal{X}$ and $k_\mathcal{Y}$. The cross-covariance operator of (X, Y) from $\mathcal{H}_\mathcal{X}$ to $\mathcal{H}_\mathcal{Y}$ can be defined so that

$$\langle h, \Sigma_{YX} g\rangle_{\mathcal{H}_\mathcal{Y}} = \mathbb{E}_{XY}\left[(g(X) - \mathbb{E}_X[g(X)])\,(h(Y) - \mathbb{E}_Y[h(Y)])\right] \tag{3}$$

holds for all $g \in \mathcal{H}_\mathcal{X}$ and $h \in \mathcal{H}_\mathcal{Y}$. Conditional covariance operator $\Sigma_{YY|X}$ can be defined as (Baker, 1973; Fukumizu et al., 2004; 2009)

$$\Sigma_{YY|X} = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}. \tag{4}$$

Let $F : \mathbb{R}^d \to \mathbb{R}^{d^*}$ be a map and $\mathcal{Z} \subseteq \mathbb{R}^{d^*}$ and $(\mathcal{H}_\mathcal{Z}, k_\mathcal{Z})$ be a RKHS on $\mathcal{Z}$ with kernel $k_\mathcal{Z}$. Define $k_\mathcal{X}^F$ on $\mathcal{X}$ as $k_\mathcal{X}^F(x, \tilde{x}) = k_\mathcal{Z}(F(x), F(\tilde{x}))$. We denote the RKHS that is related to $k_\mathcal{X}^F$ as $\mathcal{H}_\mathcal{X}^F$. We define a new conditional covariance operator that is related to F as

$$\Sigma_{YY|X}^F = \Sigma_{YY} - \Sigma_{YX}^F\left(\Sigma_{XX}^F\right)^{-1}\Sigma_{XY}^F, \tag{5}$$

where Σ_{YX}^F (Σ_{XX}^F) is the cross-covariance operator of (X, Y) defined in (3) with $k_\mathcal{X}$ being replaced by $k_\mathcal{X}^F$. Under some mild regularity conditions (Fukumizu et al., 2004; 2009), we have

$$\Sigma_{YY|X}^F \geq \Sigma_{YY|X}, \tag{6}$$

where the inequality refers to the order of self-adjoint operators. Moreover,

$$\Sigma_{YY|X} = \Sigma_{YY|X}^F \quad \Longleftrightarrow \quad Y \perp\!\!\!\perp X \mid F(X). \tag{7}$$

Define

$$\mathcal{F}_1 = \arg\min_F \Sigma_{YY|X}^F, \tag{8}$$

where the minimization refers to the minimal operators in the partial order of self-adjoint operators. Then, It follows from (6)-(7) that the target representation map F^* in (2) is the minimizer of (8), i.e., $F^* \in \mathcal{F}_1$. The loss in minimization problems (8) can be consistently estimated with samples $\{(X_i, Y_i) \subseteq \mathcal{X} \times \mathcal{Y}\}_{i=1}^n$ (Fukumizu et al., 2009) as follows

$$\mathrm{Tr}\left[G_Y\left(G_X^F + n\varepsilon_n \mathbf{I}_n\right)^{-1}\right], \tag{9}$$

where ε_n is a regularization parameter, and $G_X^F \in \mathbb{R}^{n\times n}$ with the (i, j)th entry is defined as

$$(G_X^F)_{i,j} = k_\mathcal{Z}(Z_i, Z_j) - \frac{1}{n}\sum_{b=1}^n k_\mathcal{Z}(Z_i, Z_b) - \frac{1}{n}\sum_{a=1}^n k_\mathcal{Z}(Z_a, Z_j) + \frac{1}{n^2}\sum_{a=1}^n\sum_{b=1}^n k_\mathcal{Z}(Z_a, Z_b),$$

where $Z_i = F(X_i)$, and $G_Y \in \mathbb{R}^{n\times n}$ can be computed similarly with Z_i replaced by Y_i.

2.3 f-GAN LOSS

Denote $X \sim \mu_X$. Let $Z = F(X)$ and μ_Z be its law. The f-divergence (Ali & Silvey, 1966) between μ_Z and γ_{d^*} with $\mu_Z \ll \gamma_{d^*}$ is defined as

$$\mathbb{D}_f(\mu_Z \| \gamma_{d^*}) = \int_{\mathbb{R}^{d^*}} f\left(\frac{\mathrm{d}\mu_Z}{\mathrm{d}\gamma_{d*}}\right)\mathrm{d}\gamma_{d^*}, \tag{10}$$

where $f : \mathbb{R}^+ \to \mathbb{R}$ is a twice-differentiable convex function satisfying $f(1) = 0$. The KL divergence and JS divergence correspond to $f(t) = t\log t$ and $f(t) = -(t+1)\log\frac{1+t}{2} + t\log t$, respectively. By Jensen's inequality, $\mathbb{D}_f(\mu_Z\|\gamma_{d^*}) = 0$ implies $\mu_Z = \gamma_{d^*}$ almost everywhere. Denote $\mathfrak{F}$ as the Fenchel conjugate of f (Rockafellar, 1970). Then the f-divergence can be recast as the following f-GAN loss (Nowozin et al., 2016).

Lemma 2.2

$$\mathbb{D}_f(\mu_Z\|\gamma_{d^*}) = \max_{D:\mathbb{R}^{d^*}\to\mathrm{dom}(\mathfrak{F})} \mathbb{E}_{Z\sim\mu_Z}[D(Z)] - \mathbb{E}_{W\sim\gamma_{d^*}}[\mathfrak{F}(D(W))], \tag{11}$$

where the maximum is attained when $D(x) = f'(\frac{\mathrm{d}\mu_Z}{\mathrm{d}\gamma}(x))$.

Let

$$\mathcal{F}_2 = \arg\min_F \mathbb{D}_f(\mu_{F(X)} \| \gamma_{d^*}) = \min_F \max_D \mathbb{E}_{X\sim\mu_X}[D(F(X))] - \mathbb{E}_{W\sim\gamma_{d^*}}[\mathfrak{F}(D(W))] \tag{12}$$

By Lemma 2.2, we have $F^* \in \mathcal{F}_2$. If we have $Z_i = F(X_i), W_i$ i.i.d drawn from $\mu_Z = \mu_{F(X)}$ and γ_{d^*}, respectively. We can estimate the f-GAN loss (11) as

$$\widehat{\mathbb{D}}_f(\mu_{F(X)} \| \gamma_{d^*}) = \max_D \frac{1}{n} \sum_{i=1}^{n} [D(F(X_i)) - \mathfrak{F}(D(W_i))]. \tag{13}$$

Other GAN loss such as the 1-Wasserstein distance for the WGAN (Arjovsky et al., 2017) can also be used here.

3 NSRL Algorithm

From the above section, we have the target representation map F^* in (2) satisfying $F^* \in \mathcal{F}_1 \cap \mathcal{F}_2$. And the loss for $\mathcal{F}_1$ and $\mathcal{F}_2$ can be estimated via (9) and (13), respectively. Thus, we can estimate F^* with a neural network F_θ (denoted as a reducer) that minimizes the following criterion

$$\mathrm{Tr}\left[G_Y \left(G_X^{F_\theta} + n\varepsilon_n \mathbf{I}_n\right)^{-1}\right] + \lambda \max_{D_\phi} \frac{1}{n} \sum_{i=1}^{n} [D_\phi(F_\theta(X_i)) - \mathfrak{F}(D_\phi(W_i))], \tag{14}$$

where $\lambda > 0$ is a tuning parameter, and D_ϕ is another neural network (denoted as a discriminator) to estimate the optimal D in (13). Then, we train F_θ according to the loss in equation 14 via in two steps iteratively as follows:

(i) Update the discriminator D_ϕ: Fix θ and calculate the loss for ϕ in (14) and ascending this loss by SGD on ϕ.

(ii) Update the reducer F_θ: Compute the loss for θ in (14) with the updated ϕ in (i) and descend this loss by SGD on θ.

To visualize the framework, we depict it as a flowchart in Figure 1 and give a detailed algorithm below with the "Log-D" trick GAN (Goodfellow et al., 2014) as an example.

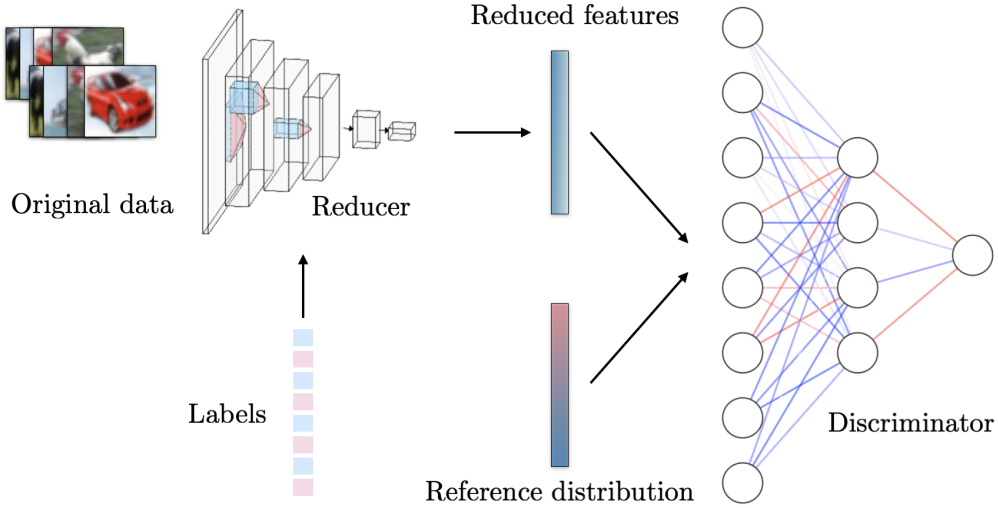


Figure 1: Flowchart for NSRL

- Novel Supervise Representation Learning (NSRL) Algorithm
- Input $\{X_i, Y_i\}_{i=1}^n$. Tuning parameters: $m, \lambda, \varepsilon, d^*$.
- **Outer loop for** θ

- **Inner loop for ϕ**
  - $*$ Sample $\{W_i\}_{i=1}^n \sim \gamma_{d^*}$
  - $*$ Update ϕ with stochastic gradient with batch size m
    $\nabla_\phi \{\sum_{i=1}^m \frac{1}{m} (\log (D_\phi (W_i)) + \log (1 - D_\phi (F_\theta (W_i))))\}$
- **End inner loop**
- Update θ with stochastic gradient with batch size m
  $\nabla_\theta \{\text{Tr}\left[G_Y \left(G_X^{F_\theta} + m\varepsilon_m I_m\right)^{-1}\right] - \lambda \sum_{i=1}^m \frac{1}{m} \log (D_\phi (F_\theta (W_i)))\}$

• **End outer loop**

4 Related works

Supervised dimension reduction: The seminal paper of Li (1991) proposed sufficient dimension reduction via sliced inverse regression, where the aim is to find a minimum subspace (Cook, 1998) such that the orthogonal projection of the data on to which preserves the dependency of the response and the predictors. There is an extensive literature on sufficient dimension reduction via a linear map (Li, 1992; Cook, 1998; Li et al., 2005). Alternative approaches have been developed to estimate the central space (or its subspace) based on nonparametric estimation of conditional independence (Xia et al., 2002; Fukumizu et al., 2004; 2009; Suzuki & Sugiyama, 2013; Vepakomma et al., 2018). See also the review paper (Cook et al., 2007) and monograph (Li, 2018) and the references therein. The methods mentioned above focus on linear dimension reduction (LDR). However, LDR may not be effective for high-dimensional complex data such as images and natural languages since the relationship between the raw data and the underlying features can be highly nonlinear.

Representation learning: Tishby & Zaslavsky (2015); Shwartz-Ziv & Tishby (2017); Saxe et al. (2019) proposed to study the internal mechanism of supervised deep learning from the perspective of information theory, where they showed that training a deep neural network that optimizes the information bottleneck (Tishby & Pereira) is a trade-off between the representation and the prediction at each layer. To make the information bottleneck idea more practical, a deep variational approximation of information bottleneck (VIB) is considered in Alemi et al. (2017). Numerical experiments suggest that the learned representations obtained via VIB are favored by the subsequent supervised learning task and robust to adversarial inputs. Information-theoretic objectives describing conditional independence such as mutual information are utilized as loss functions to train a representation-learning function, i.e., an encoder in the unsupervised setting (Hjelm et al., 2019; Oord et al., 2018; Tschannen et al., 2020; Locatello et al., 2020; Srinivas et al., 2020). Unsupervised models such as VAEs (Kingma & Welling, 2014) and its variants (Kim & Mnih, 2018; Higgins et al., 2017; Tolstikhin et al., 2018; Makhzani et al., 2017) also learn a representation via its encoder as a by-product.

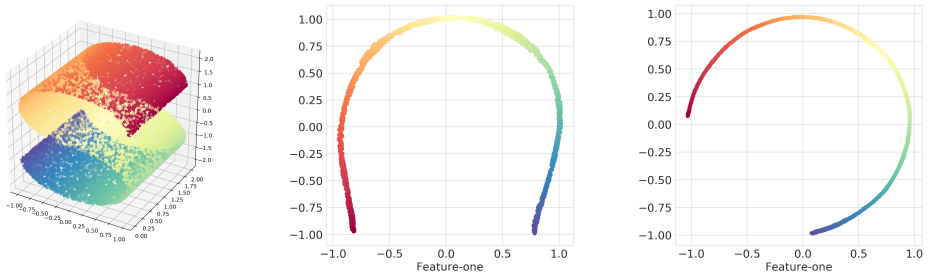
5 Experimental results

We evaluate our proposed NSRL using simulated data and real benchmark data in the setting of regression and classification. Some details on the network structures and hyperparameters used on our experiments are included in the appendix. Our experiments were conducted on Nvidia DGX Station workstation using a single Tesla V100 GPU unit. The PyTorch code of NSRL is available at `https://github.com/anonymous/NSRL`.

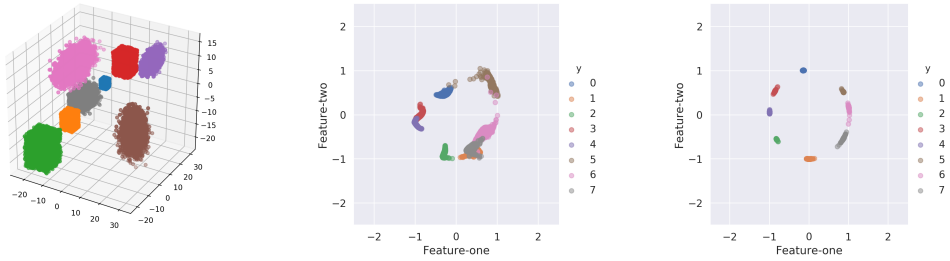
5.1 Toy examples and visualization

Visualization. We visualize the learned manifold of NSRL on two simulated data. We first generate (1) 5,000 data points from 3-dimensional S curve dataset on regression setting as shown in Figure 2 (a); (2) 5,000 data points for each class from 3-dimensional mixed Uniform and Gaussian data on classification setting as shown in Figure 2 (d). We next map these data points into the ones from 400-dimensional space by multiplying matrices with entries i.i.d $\text{Unifrom}([0, 1])$. Finally, these 400-dimensional datasets with their responses are trained by NSRL to learn 2-dimensional features. In detail, a 20-layer dense convolutional network (DenseNet) (Huang et al., 2017; 2019) as F_θ and a 4-layer network with Leaky ReLU activation as D_ϕ are employed. We set the reference distribution

γ_{d^*} uniformly distributed on the surface of the unit sphere to disentangle the representation. Scatter plots of the evolving learned 2-dimensional features are shown in Figure 2.



(a) S curve (b) Epoch = 10 (c) Epoch = 200



(d) Mixed 3D (e) Epoch = 10 (f) Epoch = 200

Figure 2: Scatter plots of the evolving learned representation.

Simulation on regression. We generate 5000 data points from the following simulated models:

$$\begin{cases} \text{Model } A: \mathbf{Y} = e^{\frac{1}{2}(\mathbf{X}_1+\mathbf{X}_2)} \log\left(\mathbf{X}_1^2\right) + \epsilon; \\ \text{Model } B: \mathbf{Y} = \left(\mathbf{X}_1^2+\mathbf{X}_2^2+\mathbf{X}_3^2\right)^{\frac{1}{2}} \log\left(\mathbf{X}_1^2+\mathbf{X}_2^2+\mathbf{X}_3^2\right)^{\frac{1}{2}} + \epsilon; \quad \epsilon \perp\!\!\!\perp \mathbf{X}, \epsilon \sim \mathcal{N}(\mathbf{0}, 0.25 \cdot \mathbf{I}_{10}) \\ \text{Model } C: \mathbf{Y} = \sin\left(\frac{\pi\left(\mathbf{X}_1+\mathbf{X}_2^2+\mathbf{X}_3^3\right)}{10}\right) + \epsilon \end{cases}$$

For the distribution of the 10-dimensional predictor $\mathbf{X}$, we consider three following scenarios:

- Scenario I: $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{10})$, independent Gaussian predictors;
- Scenario II: $\mathbf{X} \sim \frac{1}{3}\mathcal{N}(-\mathbf{1}_{10}, \mathbf{I}_{10}) + \frac{1}{3}\text{Unifrom}([-1,1]^{10}) + \frac{1}{3}\mathcal{N}(\mathbf{1}_{10}, \mathbf{I}_{10})$, independent non-Gaussian predictors;
- Scenario III: $\mathbf{X} \sim \mathcal{N}\left(\mathbf{0}, 0.4 \cdot \mathbf{I}_{10} + 0.6 \cdot \mathbf{1}_{10}\mathbf{1}_{10}^\top\right)$, correlated Gaussian predictors.

where, the notation of Scenario II is the mixture distribution of $\mathcal{N}(-\mathbf{1}_{10}, \mathbf{I}_{10})$, $\text{Unifrom}([-1,1]^{10})$ and $\mathcal{N}(\mathbf{1}_{10}, \mathbf{I}_{10})$ with mixing probabilities $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. These models and the distributional scenarios are modified from Lee et al. (2013); Li (2018).

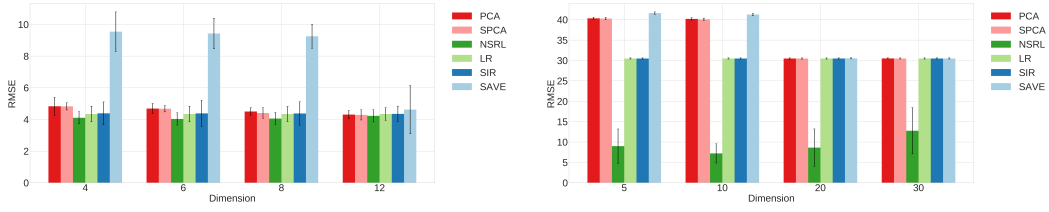
We adopt a 4-layer network for F_θ and a 3-layer network for D_ϕ with Leaky ReLU activation. We compare NSRL with two linear dimension reduction methods: sliced inverse regression (SIR) (Li, 1991), sliced average variance estimation (SAVE) (Shao et al., 2007); two nonlinear dimension reduction methods: generalized sliced inverse regression (GSIR) (Li, 2018) and generalized sliced average variance estimation (GSAVE) (Li, 2018); and linear regression with original data (LR). Finally, we fit a linear regression model between the learned features and the response. As shown in Table 1, we report the prediction error and conditional Hilbert-Schmidt independence criterion (cHSIC) (Fukumizu et al., 2008) that measures conditional dependence between the learned features and the response variable. We can see that NSRL outperforms these traditional linear and nonlinear sufficient dimension reduction methods in terms of the prediction error and cHSIC. Thus, NSRL not only excels in prediction but also can obtain central subspaces more accurately.

Table 1: Averaged prediction errors (RMSE), conditional Hilbert-Schmidt independence criterion (cHSIC) and their standard errors (based on 5-fold validation).

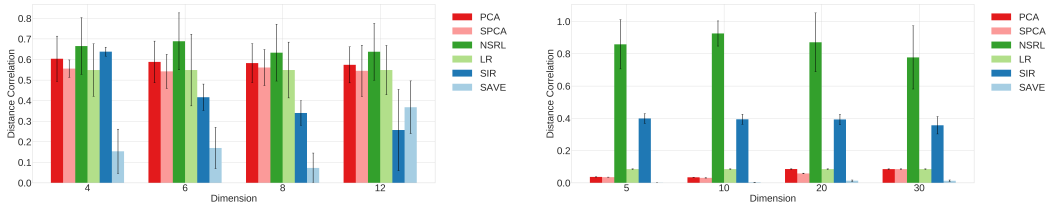
		Model A		Model B		Model C	
Scenario	Method	RMSE	cHSIC	RMSE	cHSIC	RMSE	cHSIC
I	NSRL	$3.10 \pm .2$	60.80 ± 3.7	$0.68 \pm .1$	111.26 ± 24.2	$0.44 \pm .02$	367.89 ± 38.5
	SIR (Li, 1991)	$3.13 \pm .2$	71.85 ± 3.4	$1.05 \pm .0$	208.56 ± 3.6	$0.46 \pm .02$	557.54 ± 2.8
	SAVE (Shao et al., 2007)	$3.13 \pm .2$	70.90 ± 3.7	$1.05 \pm .0$	177.19 ± 4.5	$0.46 \pm .02$	542.11 ± 5.2
	GSIR (Li, 2018)	$3.11 \pm .2$	73.19 ± 3.3	$0.89 \pm .0$	191.56 ± 4.8	$0.46 \pm .02$	575.18 ± 1.5
	GSAVE (Li, 2018)	$3.17 \pm .2$	184.94 ± 13.5	$0.90 \pm .0$	221.89 ± 5.2	$0.49 \pm .01$	608.33 ± 6.4
	LR	$3.13 \pm .2$	$190.16 \pm .8$	$1.05 \pm .0$	192.64 ± 1.4	$0.46 \pm .02$	193.91 ± 1.2
II	NSRL	$3.91 \pm .6$	57.41 ± 9.8	$0.50 \pm .1$	93.96 ± 16.2	$0.53 \pm .02$	280.91 ± 54.7
	SIR (Li, 1991)	$4.39 \pm .3$	73.47 ± 3.9	$1.62 \pm .0$	283.72 ± 12.3	$0.54 \pm .02$	483.13 ± 2.4
	SAVE (Shao et al., 2007)	$4.45 \pm .3$	80.32 ± 6.6	$1.62 \pm .0$	225.54 ± 9.5	$0.55 \pm .02$	464.74 ± 9.7
	GSIR (Li, 2018)	$4.36 \pm .3$	73.9 ± 5.4	$1.10 \pm .1$	$99.26 \pm .1$	$0.55 \pm .02$	419.96 ± 9.4
	GSAVE (Li, 2018)	$4.40 \pm .3$	148.7 ± 9.1	$1.11 \pm .0$	96.58 ± 4.7	$0.56 \pm .02$	454.99 ± 7.9
	LR	$4.39 \pm .3$	161.31 ± 2.6	$1.62 \pm .0$	173.08 ± 1.9	$0.54 \pm .02$	283.84 ± 2.9
III	NSRL	$2.96 \pm .7$	62.12 ± 10.1	$0.54 \pm .2$	108.85 ± 27.6	$0.44 \pm .02$	245.85 ± 56.3
	SIR (Li, 1991)	$3.87 \pm .5$	110.4 ± 8.9	$1.34 \pm .1$	407.10 ± 17.6	$0.47 \pm .02$	443.50 ± 12.2
	SAVE (Shao et al., 2007)	$3.87 \pm .5$	121.04 ± 8.1	$1.34 \pm .1$	221.42 ± 15.5	$0.47 \pm .02$	397.25 ± 10.3
	GSIR (Li, 2018)	$3.48 \pm .5$	68.81 ± 2.6	$0.69 \pm .0$	109.04 ± 6.4	$0.45 \pm .01$	352.96 ± 9.0
	GSAVE (Li, 2018)	$3.44 \pm .5$	73.13 ± 3.3	$0.65 \pm .0$	107.73 ± 5.9	$0.46 \pm .01$	352.28 ± 12.6
	LR	$3.88 \pm .5$	191.57 ± 5.2	$1.34 \pm .1$	$207.79 \pm .9$	$0.47 \pm .02$	267.06 ± 1.2

5.2 Performances on real-world settings

Regression. We consider two datasets: *Life Expectancy* (`https://www.kaggle.com/kumarajarshi/life-expectancy-who`) collected from World Health Organization (WHO) and *Pole* (Weiss & Indurkhya, 1995) collected from a large telecommunications application. *Life Expectancy* dataset has 2938 observations and 20 covariates to predict life expectancy in age. *Pole* dataset consists of 15000 observations with 48 predictors. A 2-layer network and a 3-layer network with Leaky ReLU activation are used for D_ϕ and F_θ in both datasets, respectively. To obtain the performance of predictions, a linear regression is adopted to fit the learned representation against the response variable. As shown in Figure 3, experimental results are reported in terms of root mean square error (RMSE) and distance correlation (DC), as measured using 5-fold cross-validation. We compare NSRL with sliced inverse regression (SIR) (Li, 1991), sliced average variance estimation (SAVE) (Shao et al., 2007), principal component analysis (PCA), sparse principal component analysis (SPCA) and linear regression with original data (LR). As a result, NSRL outperforms not only unsupervised representation methods - PCA and SPCA, but also supervised methods based on sufficient dimension reduction - SIR and SAVE.



(a) RMSE for *Life Expectancy* (b) RMSE for *Pole*



(a) DC for *Life Expectancy* (b) DC for *Pole*

Figure 3: Prediction errors and distance correlation between the representation and the response variable based on 5-fold cross-validation.

Classification. We compare NSRL with a feature extractor based on cross entropy loss (CN) on MINST (LeCun & Cortes, 2010) and Kuzushiji-MNIST (Clanuwat et al., 2018) for handwritten digits and Japanese letter recognition. MINST and Kuzushiji-MNIST both contain 60k 28×28 grayscale images from 10 classes for training and testing, respectively. To demonstrate that NSRL is compatible with various GAN frameworks, we utilize the vanilla GAN (Goodfellow et al., 2014) based on $\log D$ trick (Heusel et al., 2017) and Wasserstein GAN (WGAN) (Arjovsky et al., 2017) on our experiments. We employ the VGG-5 with Spinal FC architecture (Kabir et al., 2020) for F_θ and 4-layer networks for D_ϕ.

We apply the transfer learning technique to the combination of NSRL and CN compared with CN on STL-10 (Coates et al., 2011) and CIFAR-10 (Krizhevsky et al., 2009). The STL-10 dataset consists of 5k and 8k 96×96 color images from 10 classes for training and testing, respectively. CIFAR-10 contains 50k and 10k color images with 32×32 pixels for training and testing, respectively. The pretrained WideResnet-101 model (Zagoruyko & Komodakis, 2016) on the Imagenet dataset with Spinal FC (Kabir et al., 2020) is adopted for F_θ. The discriminator D_ϕ is a 4-layer network with Leaky ReLU activation.

Finally, k-nearest neighbor ($k = 5$) classifier on the learned features is used to obtain classification accuracy for all methods. Classification accuracies for MNIST and Kuzushiji-MNIST are reported in Table 2, and those for STL-10 and CIFAR-10 are reported in Table 3. As Table 2 shows, NSRL with different GAN frameworks are stable and comparable with CN in terms of classification accuracy. For both STL-10 and CIFAR-10 that use transfer learning, CN leveraging NSRL is comparable to CN on STL-10 and outperforms CN on CIFAR-10.

Table 2: Classification accuracy on MINST and Kuzushiji-MNIST.

Datasets	MNIST			Kuzushiji-MNIST		
Model	$d = 8$	$d = 16$	$d = 32$	$d = 8$	$d = 16$	$d = 32$
CN	99.62	99.70	99.64	98.60	98.80	98.84
NSRL ($\log D$)	**99.67**	99.66	**99.62**	**98.61**	**98.81**	98.63
NSRL (WGAN)	**99.67**	**99.70**	**99.66**	**98.68**	98.66	98.72

Table 3: Classification accuracy on STL-10 and CIFAR-10.

Datasets	STL-10			CIFAR-10		
Model	$d = 32$	$d = 64$	$d = 128$	$d = 32$	$d = 64$	$d = 128$
CN	98.17	98.36	98.45	97.79	97.78	97.74
NSRL+CN	**98.34**	98.24	98.36	**97.99**	**97.82**	**97.75**

6 CONCLUSION

In this work, we propose a novel approach to achieving a good data representation for supervised learning with certain desired characteristics including information preservation, low-dimensionality and disentanglement. We formulate the ideal representation learning task as that of finding a non-linear map that minimizes the sum of losses characterizing conditional independence and disentanglement via conditional covariance operator in RKHS and GAN. The proposed method is validated via comprehensive numerical experiments and real data analysis in the context of regression and classification. For the future work, it would be interesting to consider other measures of conditional independence and generalize the proposed method to semi-supervised learning problems.

REFERENCES

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *ICLR Workshop*, 2017.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2017.

Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1): 131–142, 1966.

Brandon Amos and J. Zico Kolter. A PyTorch Implementation of DenseNet. `https://github.com/bamos/densenet.pytorch`. Accessed: [20 Feb 2020].

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

Charles R Baker. Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 186:273–289, 1973.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

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

R Dennis Cook. *Regression graphics: ideas for studying regressions through graphics*, volume 482. 1998.

R Dennis Cook et al. Fisher lecture: Dimension reduction in regression. *Statistical Science*, 22(1): 1–26, 2007.

Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *Advances in neural information processing systems*, pp. 489–496, 2008.

Kenji Fukumizu, Francis R Bach, Michael I Jordan, et al. Kernel dimension reduction in regression. *The Annals of Statistics*, 37(4):1871–1905, 2009.

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

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. 2016.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, 2013.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

Gao Huang, Zhuang Liu, Geoff Pleiss, Laurens Van Der Maaten, and Kilian Weinberger. Convolutional networks with dense connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

HM Kabir, Moloud Abdar, Seyed Mohammad Jafar Jalali, Abbas Khosravi, Amir F Atiya, Saeid Nahavandi, and Dipti Srinivasan. Spinalnet: Deep neural network with gradual input. *arXiv preprint arXiv:2007.03347*, 2020.

Amor Keziou. Dual representation of φ-divergences and applications. *Comptes rendus mathématique*, 336(10):857–862, 2003.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *ICML*, 2018.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Kuang-Yao Lee, Bing Li, and Francesca Cuiaromonte. A general theory for non-linear sufficient dimension reduction: formulation and estimation. *The Annals of Statistics*, 41(1):221–249, 2013.

Bing Li. *Sufficient dimension reduction: Methods and applications with R*. 2018.

Bing Li, Hongyuan Zha, Francesca Chiaromonte, et al. Contour regression: a general approach to dimension reduction. *The Annals of Statistics*, 33(4):1580–1616, 2005.

Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

Ker-Chau Li. On principal hessian directions for data visualization and dimension reduction: Another application of stein's lemma. *Journal of the American Statistical Association*, 87(420): 1025–1039, 1992.

Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. Disentangling factors of variation using few labels. In *ICLR*, 2020.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *ICLR*, 2017.

Robert J McCann et al. Existence and uniqueness of monotone measure-preserving maps. *Duke Mathematical Journal*, 80(2):309–324, 1995.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Guido Philippis. *Regularity of optimal transport maps and applications*, volume 17. Springer Science & Business Media, 2013.

R Tyrrell Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1970.

Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

Yongwu Shao, R Dennis Cook, and Sanford Weisberg. Marginal tests with sliced average variance estimation. *Biometrika*, 94(2):285–296, 2007.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

T Suzuki and M Sugiyama. Sufficient dimension reduction via squared-loss mutual information estimation. *Neural computation*, 25(3):725–758, 2013.

N Tishby and F Pereira. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop*, 2015.

I Tolstikhin, O Bousquet, S Gelly, and B Schölkopf. Wasserstein auto-encoders. In *ICLR*, 2018.

Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *ICLR*, 2020.

Praneeth Vepakomma, Chetan Tonde, Ahmed Elgammal, et al. Supervised dimensionality reduction via distance correlation maximization. *Electronic Journal of Statistics*, 12(1):960–984, 2018.

Cédric Villani. *Optimal transport: old and new*, volume 338. 2008.

Sholom M Weiss and Nitin Indurkhya. Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research*, 3:383–403, 1995.

Yingcun Xia, Howell Tong, WK Li, and Li-Xing Zhu. An adaptive estimation of dimension reduction space. *Journal of the Royal Statistical Society Series B*, 64(3):363–410, 2002.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

A Appendix: Experimental details

In this section, we give the details of the experimental implementations, including hyper-parameters, network architectures, leaning optimizers, and so on. The values of the hyper-parameters are presented in Table A1, where d is the dimension of the learned features, m is the mini-batch size during training, T_1 is the number of inner loops for training D_ϕ to learn the target distribution, T_2 is the number of outer loops for training F_θ.

A.1 Simulation studies

The detailed architectures of dense convolutional network (DenseNet) (Huang et al., 2017; Amos & Kolter) deployed for F_θ on S curve and Mixed 3D are shown in Table A2. A multilayer perceptron (MLP) is adopted for D_ϕ as shown in A3. As shown in Table A4, MLP is used for the neural network structures of D_ϕ and F_θ in the regression setting. For all settings, the Adam (Kingma & Ba, 2014) optimizer is utilized with an initial learning rate of 0.001 and weight decay of 0.0001.

Table A1: Hyper-parameters for all experiments.

Dataset	d	λ	m	T_1	T_2
S curve	2	10^3	64	1	200
Mixed 3D	2	10^4	64	1	200
Simulated regression	2, 3	10^{-3}	256	1	300
Life Expectancy	4, 6, 8, 12	10^{-3}	64	1	200
Pole	5, 10, 20, 30	10^{-3}	64	1	200
MNIST	8, 16, 32	10^{-4}	128	1	200
Kuzushiji-MNIST	8, 16, 32	10^{-4}	128	1	200
STL-10	32, 64, 128	10^{-4}	128	1	50
CIFAR-10	32, 64, 128	10^{-4}	128	1	50

Table A2: 20-layer DenseNet architecture for F_θ for visualization experiments.

Layers	Details	Output size
Convolution	3×3 Conv	$24 \times 20 \times 20$
Dense Block 1	$\begin{bmatrix} \text{BN, } 1 \times 1 \text{ Conv} \\ \text{BN, } 3 \times 3 \text{ Conv} \end{bmatrix} \times 2$	$48 \times 20 \times 20$
Transition Layer 1	BN, ReLU, 2×2 Average Pool,1×1 Conv	$24 \times 10 \times 10$
Dense Block 2	$\begin{bmatrix} \text{BN, } 1 \times 1 \text{ Conv} \\ \text{BN, } 3 \times 3 \text{ Conv} \end{bmatrix} \times 2$	$48 \times 10 \times 10$
Transition Layer 2	BN, ReLU, 2×2 Average Pool, 1×1 Conv	$24 \times 5 \times 5$
Dense Block 3	$\begin{bmatrix} \text{BN, } 1 \times 1 \text{ Conv} \\ \text{BN, } 3 \times 3 \text{ Conv} \end{bmatrix} \times 2$	$48 \times 5 \times 5$
Pooling	BN, ReLU, 5×5 Average Pool, Reshape	48
Fully connected	Linear	2

A.2 Real datasets

Regression: In the regression problems, we utilize the MLP architecture for D_ϕ and F_θ as shown in A5. We adopt the Adam optimizer with an initial learning rate of 0.001 and weight decay of 0.0001.

Classification: the details of 4-layer MLP architecture for D_ϕ are shown in Table A3. The VGG-5 with Spinal FC architecture (Kabir et al., 2020) for F_θ is presented in Table A6. For training MNIST and Kuzushiji-MNIST datasets, Adam optimizer with learning rate of 0.005 is adopted for F_θ. For F_θ of STL-10 and CIFAR-10 datasets, we use customized SGD optimizer with initial learning rate of 0.001 and momentum of 0.9 and decay the learning rate by 0.1 every 7 epochs.

B Appendix: Proofs

B.1 Proof of Lemma 2.1

Proof B.1 *By assumption μ and γ_{d^*} are both absolutely continuous with respect to the Lebesgue measure. The desired result holds since it is a spacial case of the well known results on the exis-*

Table A3: MLP architecture for D_ϕ of toy visualization examples and classification settings.

	D_ϕ for visualization		D_ϕ for classification	
Layers	Details	Output size	Details	Output size
Layer 1	Linear, LeakyReLU	64	Linear, LeakyReLU	32
Layer 2	Linear, LeakyReLU	128	Linear, LeakyReLU	64
Layer 3	Linear, LeakyReLU	64	Linear, LeakyReLU	32
Layer 4	Linear	1	Linear	1

Table A4: MLP architectures for D_ϕ and F_θ for simulated regression.

	D_ϕ		F_θ	
Layers	Details	Output size	Details	Output size
Layer 1	Linear, LeakyReLU	16	Linear, LeakyReLU	32
Layer 2	Linear, LeakyReLU	8	Linear, LeakyReLU	16
Layer 3	Linear	1	Linear, LeakyReLU	8
Layer 4			Linear	d

Table A5: MLP architectures for D_ϕ and F_θ for the real regression setting.

	D_ϕ		F_θ	
Layers	Details	Output size	Details	Output size
Layer 1	Linear, LeakyReLU	8	Linear, LeakyReLU	16
Layer 2	Linear	1	Linear, LeakyReLU	32
Layer 3			Linear, LeakyReLU	8
Layer 4			Linear	d

tence of optimal transport Brenier (1991); McCann et al. (1995), see, Theorem 1.28 on page 24 of Philippis (2013) for details.

B.2 Proof of Lemma 2.2

Proof B.2 *Our proof follows Keziou (2003). Since $f(t)$ is a convex function, then $\forall t \in \mathbb{R}$ $f(t) = f^{**}(t)$, where*

$$f^{**}(t) = \sup_{s \in \mathbb{R}} \{st - \mathfrak{F}(s)\}$$

is the Fenchel conjugate of $\mathfrak{F}$. By Fermat's rule, the maximizer s^* satisfies

$$t \in \partial \mathfrak{F}(s^*),$$

i.e.,

$$s^* \in \partial f(t)$$

Plugging the above display with $t = \frac{\mathrm{d}\mu_Z}{\mathrm{d}\gamma}(x)$ into the definition of f-divergence, we obtain (8).

Table A6: Network architecture for F_θ on MNIST and Kuzushiji-MNIST.

Layers	Details	Output size
VGG-5 Block	$\left[\begin{pmatrix} 3\times 3 \text{ Conv} \\ \text{BN, ReLU} \end{pmatrix} \times 2 \right] \times 2, \left[\begin{pmatrix} 3\times 3 \text{ Conv} \\ \text{BN, ReLU} \end{pmatrix} \times 3 \right] \times 2$	$1 \times 28 \times 28$
	3×3 Max Pool	
Fully Connected Spinal Block	$\left[\begin{matrix} \text{Dropout, Linear} \\ \text{BN, ReLU} \end{matrix} \right] \times 4$	512
Fully connected	Dropout, Linear	d