INVARIANT REPRESENTATION LEARNING VIA FUNCTIONS IN REPRODUCING KERNEL HILBERT SPACES

By

Bashir Sadeghi

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2023

ABSTRACT

Many applications of representation learning, such as privacy preservation and algorithmic fairness, desire explicit control over some unwanted information being discarded. This goal is formulated as satisfying two objectives: maximizing utility for predicting a target attribute while simultaneously being invariant (independent) to a known sensitive attribute (like gender or race). Solutions to invariant representation learning (IRepL) problems lead to a trade-off between utility and invariance when they are competing. Most existing works are empirical and implicitly look for single or multiple points on the utility-invariance trade-off. They do not explicitly seek to characterize the entire trade-off front optimally and do not provide invariance and convergence guarantees.

In this thesis, we address the shortcoming mentioned above by considering simple linear modeling and building upon them. As a first step, we derive a closed-form solution for the global optima of the underlying linear IRepL optimization problem. In further development, we consider neural network-based encoders, where we model the utility of the target task and the invariance to the sensitive attribute via kernelized ridge regressors. This setting leads to a stable iterative optimization scheme toward global/local optima(s). However, such a setting cannot guarantee universal invariance. This drawback motivated us to further study the case where the invariance measure is modeled universally via functions in some reproducing kernel Hilbert spaces (RKHS)s. By modeling the encoder and target networks via functions in some RKHS, too, we derive a closed formula for a near-optimal trade-off, corresponding optimal representation dimensionality, and the associated encoder(s). Our findings have an immediate application to fairness in terms of demographic parity.

To my wife: *Katharina*
To my parents: *Farajollah* and *Farrokh*
To my sister's family: *Atekeh*, *Karen*, and *Hanzaleh*
To my brother's family: *Farhad* and *Fatemeh*

ACKNOWLEDGMENTS

I am thankful to have Dr. Vishnu Naresh Boddeti as my PhD advisor. His attention, expectations, and encouragement helped me learn more than I could have imagined. With his meticulous attention to detail and critical assessment, and setting himself as an example, Dr. Bodetti made me decide what kind of researcher I want to be: A solid researcher who formulates scientific problems principally, who observes carefully, who evaluates/designs experiments efficiently and critically, and finally, presents the findings understandably and intuitively.

I am honored to have Dr. Arun Ross, Dr. Sijia Liu, and Dr. Hamidreza Modares on my thesis committee. I am grateful that I could attend Dr. Sijia Liu's 'Adversarial Machine Learning' course. His deep insight into the topic made this course enjoyable to me. I am also very thankful to Dr. Mathew Hirn for his interesting 'Mathematics of Deep Learning' course. This course contributed to filling some gaps between the application and theory of deep learning for me.

I want to thank Gautam Sreekumar for his generous support and encouragement during the challenging times of my PhD. Likewise, I am also grateful for the encouragement I received from my other labmates, Hamed Bolandi, Lan Wang, Xiaoxue Wang, Shihua Huang, Rahul Dey, Sepehr Dehdashtian, and Wei Ao. Their valuable comments during group meetings and research discussions helped me to improve on what I could do alone.

Last but not least, I am very fortunate and grateful that Dr. Runyi Yu was my MSc advisor. He set himself as an example of a principal scientist who approaches research problems fundamentally. Without any exaggeration, I could not see myself pursuing a PhD without his encouragement and support during my MSc study.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Acronyms / Abbreviations

ARL	Adversarial Representation Learning
COCO	Constrained Covariance
DNN	Deep Neural Network
GAN	Generative Adversarial Network
HSIC	Hilbert-Schmidt Independence Criterion
IRepL	Invariant Representation Learning
iff	If and Only If
KCC	Kernelized Canonical Correlation
MLP	Multi-Layer Perceptrons
MSE	Mean Square Error
NN	Neural Network
PCA	Principal Component Analysis
RFF	Random Fourier Features
RKHS	Reproducing Kernel Hilbert Space
RV	Random Variable
SGD	Stochastic Gradient Descent
SGDA	Stochastic Gradient Descent Ascent
w.r.t.	With Respect To

Chapter 1

Introduction To Invariant Representation Learning

1.1 Introduction

Real-world applications of representation learning often have to contend with objectives beyond predictive performance. These include cost functions pertaining to, invariance (e.g., to photometric or geometric variations), semantic independence (e.g., to age or race for face recognition systems), privacy (e.g., mitigating leakage of sensitive information [1]), algorithmic fairness (e.g., demographic parity [2]), and generalization across multiple domains [3], to name a few.

At its core, the goal of the aforementioned formulations of representation learning is to satisfy two competing objectives: Extracting as much information necessary to predict a target label Y (e.g., face identity), while *intentionally* and *permanently* suppressing information about a given sensitive attribute S (e.g., age or gender). See Figure 1.1 for an illustration. An encoder f produces a representation $Z = f(X)$ from the input data X. A target predictor g_Y operates on the representation Z to predict the target attribute Y. A parametric or non-parametric dependence measure $\text{Dep}(Z, S)$ measures the statistical dependency of the representation Z on the sensitive attribute S. For example, $\text{Dep}(Z, S)$ can be measured by a hypothetical adversary loss that aims to predict the sensitive attribute S. Even though randomized encoder and target predictor can also be consid-

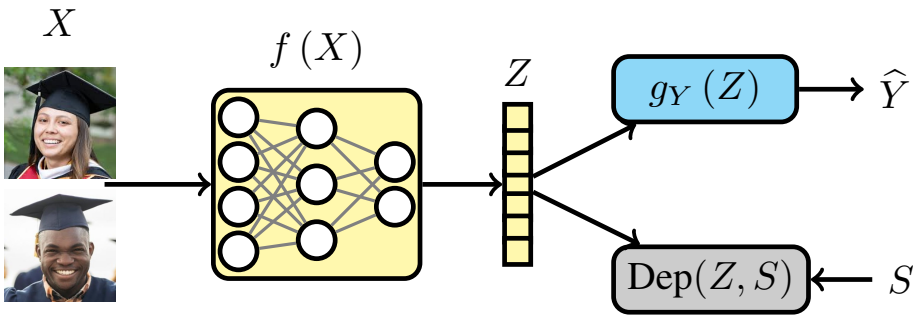


Figure 1.1: An encoder f in the form of a Borel function produces a representation $Z = f(X)$ from the input data X. A target predictor g, in the form of a Borel function, operates on the representation Z to predict the target attribute Y. A parametric or non-parametric dependence measure $\mathrm{Dep}(Z, S)$ quantifies the statistical dependency of the representation Z on the sensitive attribute S. Invariant representation learning seeks a representation $Z = f(X)$ that contains as much information necessary for the downstream target predictor g_Y while being independent of the sensitive attribute S.

ered, however in this dissertation, we assume that both f and g_Y are deterministic Borel functions. When the statistical dependency between Y and S is not negligible, learning a representation Z that is invariant to the sensitive attribute S (i.e., $Z \perp\!\!\!\perp S$) will necessarily degrade the performance of the target prediction, i.e., there exists a trade-off between utility and invariance. The primary application of invariant representation learning (IRepL) is invariant prediction. This is because if Z is independent of S, then the target prediction $\widehat{Y} = g_Y(Z)$ is independent of S, regardless of the target predictor g_Y. As a result, to be robust to the choice of target predictor [4], it is preferred to deploy IRepL for invariant prediction rather than enforcing invariance on $\widehat{Y}$ directly.

The existence of a trade-off between utility and invariance has been well established, both theoretically and empirically, under various contexts of representation learning such as fairness [5, 6, 7, 8], invariance [9], and domain adaptation [10]. However, the central aspect of IRepL is still challenging: *A learning algorithm that achieves any point on the utility-invariance trade-off, optimally or via local optima(s), and how can we estimate them from training data*. A vast majority of existing works are empirical in nature. They implicitly look for single or multiple points on the trade-off between utility and invariance to the sensitive information and do not explicitly seek to

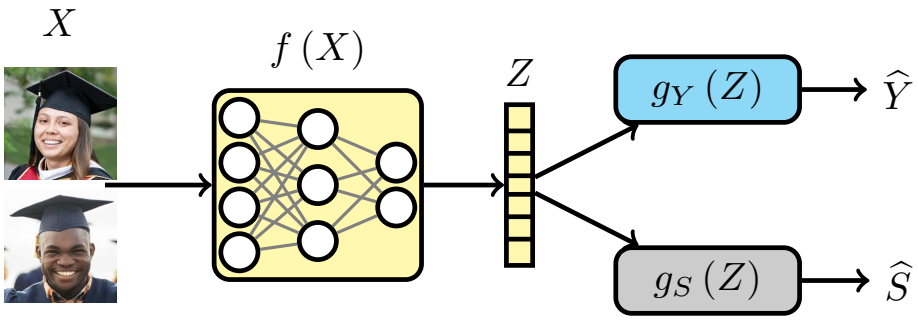


Figure 1.2: **Adversarial Representation Learning** consists of three entities, an encoder f that obtains a compact representation Z of the input data X, a predictor g_Y that predicts a desired target attribute Y and an adversary g_S that seeks to extract a sensitive attribute S, both from the embedding Z.

characterize the entire trade-off front optimally. This dissertation aims to address the mentioned shortcoming of existing IRepL approaches by employing functions in some reproducing kernel Hilbert spaces (RKHS)s to model target predictor g_Y and the dependence measure $\text{Dep}(Z, S)$. Under the case where the encoder f is also modeled via functions in some RKHSs, we are able to find a closed-form solution for the optimal encoder. For encoders modeled by neural networks (NN)s, we are able to provide some stability for the underlying iterative optimization problem.

1.2 Prior Work

1.2.1 Adversarial Representation Learning

Most practical approaches for learning fair, invariant, domain adaptive, or privacy-preserving representations discussed above are based on adversarial representation learning (ARL). At the core of ARL is the idea of modeling $\text{Dep}(Z, S)$ via a proxy adversary that seeks to extract the sensitive attribute S. See Figure 1.2 for an illustration. In the context of image classification, adversarial learning has been utilized to obtain representations that are invariant across domains [3, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]. Such representations allow classifiers that are trained on a source

domain to generalize to a different target domain. In the context of learning fair and unbiased representations, a number of approaches [1, 2, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35] have used and argued for explicit adversarial networks, to extract sensitive attributes from the encoded data. All these methods are usually set up as a minimax game between the encoder, a target task, and a proxy adversary. The encoder is set up to achieve invariance by maximizing the loss of the proxy adversary, i.e., minimizing the negative log-likelihood or mean square error (MSE) of sensitive variables as measured by the proxy adversary. Roy *et al.* [1] identify and address the instability of the optimization in the zero-sum minimax formulation of ARL and propose an alternate non-zero-sum solution, demonstrating improved empirical performance. All the above approaches use deep neural networks (DNN)s to represent the ARL entities, optimize their parameters through stochastic gradient descent ascent (SGDA), and rely on empirical validation. However, none of them seek to study the nature of the ARL formulation itself, i.e., in terms of decoupling the role of the expressiveness of the models and convergence/stability properties of the optimization tools for learning the parameters of the corresponding models. This shortcoming motivates us to take some steps towards filling this gap by studying simpler forms of ARL from an optimal optimization perspective in Chapter 3 and build upon it in Chapters 4, 5.

1.2.2 Invariant Representation Learning

The basic idea of representation learning that discards unwanted semantic information has been explored under many contexts like invariant, fair, or privacy-preserving learning. The concept of learning fair representations was first introduced by Zemel *et al.* [36]. The goal was to learn a representation of data by "fair clustering" while maintaining the discriminative features of the prediction task. Building upon this work, many techniques have been proposed to learn an unbiased representation of data while retaining its effectiveness for a prediction task. These include the Varia-

tional Fair Autoencoder [37] and the more recent information bottleneck-based objective by Moyer *et al.* [38]. In domain adaptation [11, 12, 39], the goal is to learn features that are independent of the data domain. In fair learning [2, 21, 40, 36, 41, 42, 43, 23, 24, 22, 44, 26, 45, 46, 47, 48, 49, 50], the goal is to discard the demographic information that leads to unfair outcomes. Similarly, there is growing interest in mitigating unintended leakage of private information from representations [51, 52, 53, 54, 55, 45, 56, 57, 58, 59].

A vast majority of this body of work is empirical in nature. They implicitly look for single or multiple points on the trade-off between utility and sensitive information and do not explicitly seek to characterize the entire trade-off front. Overall, these approaches are not concerned with or aware of the inherent utility-invariance trade-off. In contrast, using functions in some RKHSs, we near-optimally characterize the trade-off and propose a practical learning algorithm that achieves this trade-off in Chapter 5.

1.2.3 Trade-Offs in Invariant Representation Learning

Prior work has established the existence of trade-offs in IRepL, both empirically and theoretically. In the following, we categorize them based on properties of interest.

Restricted Class of Attributes: A majority of existing work considers IRepL trade-offs under restricted settings, i.e., binary and/or categorical attributes Y and S. For instance, [60] uses information-theoretic tools and characterizes the utility-fairness trade-off in terms of lower bounds when both Y and S are binary labels. Later [55] provided both upper and lower bounds for binary labels. By leveraging Chernoff bound, [61] proposed a construction method to generate an ideal representation beyond the input data to achieve perfect fairness while maintaining the best performance on the target task. In the case of categorical features, a lower bound on utility-fairness trade-off has been provided by [6] for the total invariance scenario (i.e., $Z \perp\!\!\!\perp S$). In contrast to this

body of work, our trade-off analysis applies to multidimensional continuous/discrete attributes. To the best of our knowledge, the only prior work with a general setting is [9]. However, in [9], both S and Y are restricted to be continuous/discrete or binary at the same time (e.g., it is not possible to have Y binary while S is continuous).

Characterizing Exact versus Bounds on Trade-Off: To the best of our knowledge, all existing approaches characterize the trade-off in terms of upper and/or lower bounds. In contrast, we *exactly* characterize a near-optimal trade-off with closed-form expressions for encoders belonging to some RKHSs.

Optimal Encoder and Representation: Another property of practical interest is the optimal encoder that achieves the desired point on the utility-invariance trade-off and the corresponding representation(s). Existing works which only study bounds on the trade-off do not obtain the encoder that achieves those bounds. Hilbert-Schmidt independent criterion (HSIC), a universal measure of dependence, has been adopted by prior work (e.g., [62]) to quantify all types of dependencies between Z and S. However, these methods adopt stochastic gradient descent (SGD) for optimizing the underlying non-convex optimization problem. As such, they fail to guarantee that the representation learning problem converges to a global optima. In contrast, we obtain a closed-form solution for the optimal encoder and its corresponding representation while detecting all modes of dependence between Z and S in Chapter 5.

1.2.4 Optimization Theory for Adversarial Learning

A growing class of learning algorithms, including ARL, generative adversarial networks (GAN)s, etc., involve more than one objective and are trained via games played by cooperating or dueling NNs. An overview of the challenges presented by such algorithms and a plausible solution in general n-player games can be found in [63]. In the context of two-player minimax games such

as GANs, several solutions [64, 65, 66, 67, 68, 69, 70, 71] have been proposed to improve the optimization dynamics, many of them relying on the idea of taking an extrapolation step [72]. The non-convex nature of the ARL formulation poses unique challenges from an optimization perspective. Practically, the parameters of the models in ARL are optimized through SGD, either jointly [21, 64] or alternatively [11], with the former being a generalization of gradient descent and is known as SGDA. While the convergence properties of gradient descent and its variants are well understood, there is relatively little work on the convergence and stability of SGDA in adversarial minimax problems. Recently, Mescheder *et al.* [64] and Nagarajan *et al.* [65] both leveraged tools from non-linear systems theory [73] to analyze the convergence properties of SGDA, in the context of GANs, around a given equilibrium. They show that without the introduction of additional regularization terms to the objective of the zero-sum game, SGDA does not converge. However, their analysis is restricted to the two-player GAN setting and is not concerned with its global/local optima. In contrast, using kernelized ridge regressors for target and proxy adversary networks, we are able to optimize these networks optimally for any given representation $Z = f(X)$ that turns the unstable SGDA optimization scheme into the stable SGD scheme.

1.2.5 Dimensionality Reduction

The technical machinery of Chapters 3, 5 in this dissertation is closely related to principal component analysis (PCA) [74] and its kernelized version [75] for dimensionality reduction. PCA can provide a compact disentangled representation (i.e., different elements of the representation vector are uncorrelated to each other) of the input data, which is efficient for downstream classification, regression, and clustering tasks. In particular, we deploy supervised PCA in this thesis [76]. Kernel methods have also been previously used for fair dimensionality reduction by [77], where the Rayleigh quotient is employed to only search for a single point in the utility-invariance trade-off.

In contrast to this work, our approaches in Chapters 3, 5 aims to characterize the entire trade-off front.

1.2.6 Dependence Measure

In 1959, Rényi introduced dependence measures as a quantifier of the statistical dependence between two random variables (RV)s Z and S by a non-negative value, where zero indicates that Z and S are independent and with larger values indicating greater degrees of dependence [78]. A possible such dependence measure can be defined as the maximum Pearson correlation (aka correlation coefficients) between $\alpha(Z)$ and $\beta(S)$ over all Borel functions α and β [78]. Such a measure is not computationally tractable if Z and/or S are continuous [76]. To circumvent this difficulty, [79] demonstrated that any universal RKHS is sufficient as a search space for α and β to detect all modes of dependence[1] between Z and S. Later, [80] employed the maximum of covariance as a measure of independence for α and β belonging to a unit-ball in some universal RKHSs. Further, [81] proposed HSIC, where they demonstrated that considering covariance for only elements of any basis set in the involved RKHSs is sufficient for a universal dependence measure.

1.3 Overview of the Thesis

In the second chapter, we introduce the notations, definitions, mathematical background, and machinery required for this dissertation. The mathematical background of this dissertation includes linear algebra, probability theory, and functional analysis. In particular, we sometimes deploy functions in some RKHSs to model the encoder f and/or the target predictor g_Y. Further, we

[1] By 'all modes of dependence', all types of linear or non-linear relations in contrast to only linear or monotonic relations.

sometimes model the dependence measure between the representation Z and the sensitive attribute S via kernel measures of independence.

In the third chapter, we study the simplest ARL, where all players 1) encoder f, 2) target predictor g_Y, and 3) proxy adversary g_S are modeled linearly. Under this scenario, we obtain a closed-form solution (both empirically and in population) for the optimal encoder in terms of the eigenvectors of the projection of input data into the space that is as close as possible to the target label space while lying on the least explanatory space for the sensitive attribute. We then generalize our formulation and closed-form solutions to encoders in RKHSs while target and proxy adversary networks remain linear. Moreover, we theoretically obtain an optimal embedding dimensionality (i.e., $\dim(Z)$) as a function of the user-defined trade-off parameter.

In the fourth chapter, we let the encoder be DNN and aim to circumvent the instability and lack of convergence guarantees induced by SGDA optimization in ARL by modeling adversary and target networks by kernelized ridge regressors. This, in turn, yields a closed-form solution for the optimal adversary and target predictors for any given representation $Z = f(X)$. Therefore, the SGDA optimization strategy reduces to a simple SGD to learn the encoder parameters. Moreover, we theoretically obtain an upper bound for the optimal embedding dimensionality.

In the fifth chapter, motivated by the fact that proxy adversary loss may not account for all modes of dependence, we model the invariance measure via a near-universal dependence measure rather than a proxy adversary loss. By modeling the target loss the same, we are able to find a closed-form solution (both empirically and in population) for encoders in RKHSs. The closed-form solution also leads to the determination of optimal embedding dimensionality. The utility-invariance trade-off induced by the optimal encoder can be interpreted as an inherent trade-off arising from the triplet of the input data X, the target label Y, and the sensitive attribute S.

We conclude the thesis in the sixth chapter, where we discuss limitations, future work, and the

broader impact of this thesis.

1.4 Contributions of the Thesis

- We obtain a closed-form solution for the global optima of ARL with linear/kernelized encoder and linear target and proxy adversary networks under MSE loss in Chapter 3. This closed-form solution can interestingly be interpreted as a generalization of supervised PCA (and its kernelized version) when there is a sensitive attribute to discard. Consequently, we are able to obtain an exact optimal embedding dimensionality as a function of the user-defined utility-invariance trade-off parameter under the mentioned scenario.

- We deploy kernelized ridge regressors for modeling proxy adversary and target networks while the encoder can be DNN in Chapter 4. In turn, the unstable SGDA optimization involved in ARL reduces to SGD, which is a stable optimization scheme. Furthermore, we obtain an upper bound for the optimal embedding dimensionality.

- We introduce a simplified version of HSIC to measure the dependence between the representation Z and the sensitive attribute S for encoders in RKHSs in Chapter 5. We demonstrate that our proposed dependence measure is near-universal and lends itself to a closed-form solution for the IRepL problem where the optimal embedding dimensionality can be precisely obtained as a function of the trade-off parameter. The introduced utility-invariance trade-off can be interpreted as a near-optimal trade-off induced by the triplet (X, Y, S).

Chapter 2

Mathematical Background and Preliminaries

2.1 Notations and Definitions

Scalars are denoted by regular lowercase letters, e.g., r, λ. Deterministic vectors are denoted by boldface lowercase letters, e.g., $\boldsymbol{x}$, $\boldsymbol{s}$. The L_2-norm of the vector $\boldsymbol{x}$ is denoted by $\|\boldsymbol{x}\|$, and the inner product between the vectors $\boldsymbol{x}$ and $\boldsymbol{s}$ of the same size is denoted by $\langle \boldsymbol{x}, \boldsymbol{s} \rangle$. We denote n-tuple vectors of ones and zeros by $\mathbf{1}_n$ and $\mathbf{0}_n$, respectively. Finite or infinite sets are denoted by calligraphy letters, e.g., $\mathcal{H}$, $\mathcal{A}$. The indicator function is denoted by $\mathbf{1}_{\mathcal{B}}(\cdot)$, where

$$\mathbf{1}_{\mathcal{B}}(m) = \begin{cases} 1 & \text{if } m \in \mathcal{B} \\ 0 & \text{if } m \notin \mathcal{B} \end{cases} \tag{2.1}$$

We denote deterministic matrices by boldface upper case letters, e.g., $\boldsymbol{M}$, $\boldsymbol{K}$. The element at i-th row and j-th column of any matrix $\boldsymbol{M}$ is denoted by $(\boldsymbol{M})_{ij}$ or m_{ij}; its transpose is denoted by $\boldsymbol{M}^T$; its inverse is denoted by $\boldsymbol{M}^{-1}$, and its Moore-Pensore pseudo-inverse is denoted by $\boldsymbol{M}^{\dagger}$. Centering, i.e., mean subtraction with respect to (w.r.t.) columns, is denoted by "˜", e.g., $\tilde{\boldsymbol{M}}$,

which can be obtained as

$$\tilde{\boldsymbol{M}} = \boldsymbol{M}\boldsymbol{H} \quad \text{where } \boldsymbol{M} \in \mathbb{R}^{m\times n} \quad \text{and } \boldsymbol{H} := \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T. \tag{2.2}$$

The subspace spanned by the columns of $\boldsymbol{M}$ is denoted by $\mathcal{R}(\boldsymbol{M})$ or simply $\mathcal{M}$; the orthogonal complement of $\mathcal{M}$ is denoted by $\mathcal{M}^\perp$, and the null space of $\boldsymbol{M}$ is denoted by $\mathcal{N}(\boldsymbol{M})$. The orthogonal projection onto $\mathcal{M}$ is denoted by $P_{\mathcal{M}}$ and can be obtained as

$$P_{\mathcal{M}} = \boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{M}\right)^\dagger \boldsymbol{M}^T. \tag{2.3}$$

We denote an $n \times n$ identity matrix by $\boldsymbol{I}_n$ or simply $\boldsymbol{I}$. The trace of any square matrix $\boldsymbol{K}$ (i.e., the sum of diagonal elements) is denoted by $\text{Tr}\left[\boldsymbol{K}\right]$. The Frobenius norm of any matrix $\boldsymbol{M}$ is denoted by $\|\boldsymbol{M}\|_F$, which is related to the trace as

$$\|\boldsymbol{M}\|_F^2 = \text{Tr}\left[\boldsymbol{M}\boldsymbol{M}^T\right] = \text{Tr}\left[\boldsymbol{M}^T\boldsymbol{M}\right].$$

We denote both scalar-valued and multidimensional random variables (RV)s by regular upper-case letters, e.g., X, S. The expectation of the RV X is denoted by $\mathbb{E}[X]$; and its covariance matrix is denoted by $\boldsymbol{C}_X$, where

$$\boldsymbol{C}_X := \mathbb{E}\left[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T\right].$$

Similarly, denoted by $\boldsymbol{C}_{XY}$, the cross-covariance between the RVs X and S is defined as

$$\boldsymbol{C}_{XS} := \mathbb{E}\left[(X - \mathbb{E}[X])(S - \mathbb{E}[S])^T\right].$$

For a positive definite matrix $\boldsymbol{C}$ (denoted by $\boldsymbol{C} \succ 0$), its Cholesky factorization results in

$$\boldsymbol{C} \succ 0 \Rightarrow \boldsymbol{C} = \boldsymbol{Q}\boldsymbol{Q}^T, \ \boldsymbol{Q} \text{ is full rank.} \tag{2.4}$$

If $\boldsymbol{C}$ is a positive semi-definite matrix (denoted by $\boldsymbol{C} \succeq 0$), then its incomplete Cholesky factorization is

$$\boldsymbol{C} \succeq 0 \Rightarrow \boldsymbol{C} = \boldsymbol{L}\boldsymbol{L}^T, \ \boldsymbol{L} \text{ is full column-rank.} \tag{2.5}$$

Consider the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the sample space, $\mathcal{F}$ is a $\sigma-$algebra on Ω, and $\mathbb{P}$ is a probability measure on $\mathcal{F}$. We assume that the joint RV, (X, Y, S) containing the input data $X \in \mathbb{R}^{d_X}$, the target label $Y \in \mathbb{R}^{d_Y}$, and the sensitive attribute $S \in \mathbb{R}^{d_S}$, is an RV on $(\Omega, \mathcal{F})$ with joint distribution $\boldsymbol{p}_{X,Y,S}$. Furthermore, Y and S can also belong to any finite set, like a categorical set. This setting enables us to work with classification and multidimensional regression tasks, where the sensitive attribute can be either categorical or multidimensional continuous/discrete RV. We let $\boldsymbol{D} := \{(\boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{s}_1), \cdots, (\boldsymbol{x}_n, \boldsymbol{y}_n, \boldsymbol{s}_n)\}$ be the training data, containing n i.i.d. samples from the joint distribution $\boldsymbol{p}_{X,Y,S}$. We also separately define the input, the label, and the sensitive data, respectively, as follows.

$$\boldsymbol{X} := [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d_X \times n}$$

$$\boldsymbol{Y} := [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n] \in \mathbb{R}^{d_Y \times n}$$

$$\boldsymbol{S} := [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n] \in \mathbb{R}^{d_S \times n}.$$

2.2 Preliminaries

2.2.1 Kernelization

Let $f \in \mathcal{H}_X$, where $\mathcal{H}_X$ is an RKHS of functions from $\mathbb{R}^{d_X}$ to $\mathbb{R}$ with kernel function $k_X(\cdot,\cdot)$. Invoking the representer theorem [82], it follows that

$$f(X) = \sum_{i=1}^{n} \theta_i k_X(\boldsymbol{x}_i, X) = \boldsymbol{\theta}\left[k_X(\boldsymbol{x}_1, X), \cdots, k_X(\boldsymbol{x}_n, X)\right]^T,$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n\times 1}$ and $(\boldsymbol{\theta})_i = \theta_i$. Moreover, let $\boldsymbol{f}(X) := [f_1(X), \cdots, f_r(X)]^T$. Similarly, we have

$$\boldsymbol{f}(X) = \boldsymbol{\Theta}\left[k_X(\boldsymbol{x}_1, X), \cdots, k_X(\boldsymbol{x}_n, X)\right]^T, \tag{2.6}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{r\times n}$ and $(\boldsymbol{\Theta})_{ji} = \theta_{ji}$.

2.2.2 Kernelized Dependence Measures

Principally, two RVs Z and S are independent (denoted by $Z \perp\!\!\!\perp S$) if and only if (iff) [83]

$$\mathbb{C}\text{ov}(\alpha(Z), \beta(S)) := \mathbb{E}\left[\alpha(Z)\,\beta(S)\right] - E\left[\alpha(Z)\right]\, E\left[\beta(S)\right] \tag{2.7}$$

is zero for all Borel functions $\alpha : \mathbb{R}^r \to \mathbb{R}$ and $\beta_s : \mathbb{R}^{d_S} \to \mathbb{R}$ belonging to the universal RKHSs $\mathcal{H}_Z$ and $\mathcal{H}_S$, respectively. Note that universality ensures that RKHSs can approximate any Borel function with arbitrary precision [84]. In the remainder of this thesis, we consider the following assumption unless otherwise stated.

Assumption 2.1. We assume that any RKHS $\mathcal{H}$ (from $\mathbb{R}^d$ to $\mathbb{R}$) is universal and separable and the

corresponding kernel function, $k(\cdot, \cdot)$ is bounded:

$$\mathbb{E}\left[k(U, U)\right] < \infty \quad \text{for any square-integrable } d\text{-dimensional } U. \tag{2.8}$$

Note that a Hilbert space is separable iff it has a countable orthonormal basis set and U is square-integrable iff $\mathbb{E}\left[\|U\|^2\right] < \infty$.

Now consider the following bi-linear functional:

$$h : \mathcal{H}_Z \times \mathcal{H}_S \to \mathbb{R},\ h(\alpha, \beta) \mapsto \mathbb{C}\text{ov}(\alpha(Z), \beta(S)),$$

where $\mathcal{H}_Z$ and $\mathcal{H}_S$ are RKHSs. This bi-linear functional is bounded due to Assumption 2.1 [85]. Invoking Riesz representation theorem [86], there exist a unique and bounded operator $\Sigma_{SZ} : \mathcal{H}_Z \to \mathcal{H}_S$ such that

$$\mathbb{C}\text{ov}(\alpha(Z), \beta(S)) = h(\alpha, \beta) = \langle \Sigma\, \alpha, \beta \rangle_{\mathcal{H}_S} \quad \forall\, \alpha \in \mathcal{H}_Z, \beta \in \mathcal{H}_S. \tag{2.9}$$

Consequently, it follows that

$$Z \perp\!\!\!\perp S \iff \Sigma_{SZ} = 0. \tag{2.10}$$

Notice that $\Sigma_{SZ} = 0$ iff the norm of Σ_{SZ} is zero for any valid norm like spectral norm:

$$\|\Sigma_{SZ}\|_{\text{Spectral}} := \sup_{\alpha \in \mathcal{H}_Z} \frac{\|\Sigma_{SZ}\, \alpha\|_{\mathcal{H}_S}}{\|\alpha\|_{\mathcal{H}_Z}}, \tag{2.11}$$

or Hilbert-Schmidt norm:

$$\|\Sigma_{SZ}\|_{\text{HS}}^2 := \sum_{\alpha_i \in \mathcal{U}_Z,\, \beta_j \in \mathcal{U}_S} \left\langle \Sigma_{SZ}\, \alpha_i, \beta_j \right\rangle_{\mathcal{H}_S}^2, \tag{2.12}$$

where $\mathcal{U}_Z$ and $\mathcal{U}_S$ are countable orthonormal basis sets for the separable universal RKHSs $\mathcal{H}_Z$ and $\mathcal{H}_S$, respectively. These norms have been deployed in constrained covariance (COCO) [80] and HSIC, respectively, which are universal measures of dependence [81]. Moreover, kernelized canonical covariance (KCC) introduced by [79]:

$$\text{KCC}(Z, S) := \sup_{\alpha \in \mathcal{H}_Z, \beta \in \mathcal{H}_S} \frac{\mathbb{C}\text{ov}(\alpha(Z), \beta(S))}{\sqrt{\mathbb{V}\text{ar}(\alpha(Z))\, \mathbb{V}\text{ar}(\beta(S))}}, \tag{2.13}$$

has also widely been used as a universal measure of dependence.

Chapter 3

Adversarial Representation Learning Under Linear Invariance

3.1 Introduction

Adversarial representation learning is a promising framework for training image representation models that can control the information encapsulated within it. ARL is practically employed for learning representations for a variety of applications, including unsupervised domain adaptation of images [87], censoring sensitive information from images [21], learning fair and unbiased representations [37, 2], learning representations that are controllably invariant to sensitive attributes [24] and mitigating unintended information leakage [1], amongst others.

At the core of the ARL formulation is the idea of jointly optimizing three entities: (i) An encoder f that seeks to distill the information from the input data X and retains the information relevant to the target attribute Y while *intentionally* and *permanently* eliminating the information corresponding to the sensitive attribute S, (ii) a predictor g_Y that seeks to predict Y, and (iii) a proxy adversary g_S, playing the role of an unknown adversary, that seeks to extract the sensitive information S. Figure 3.1 shows a pictorial illustration of the ARL problem.

Typical instantiations of ARL represent these entities through non-linear functions in the form of NNs and formulate parameter learning as a minimax optimization problem. Practically, opti-

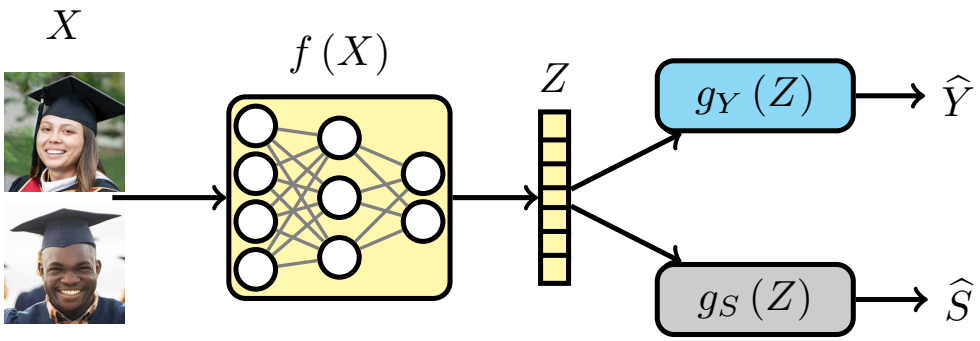


Figure 3.1: **Adversarial Representation Learning** consists of three entities, an encoder f that obtains a compact representation Z of the input data X, a predictor g_Y that predicts a desired target attribute Y and an adversary g_S that seeks to extract a sensitive attribute S, both from the embedding Z.

mization is performed through SGDA, wherein small gradient steps are taken simultaneously in the parameter space of the encoder, target predictor, and proxy adversary. The solutions thus obtained have been effective in learning data representations with controlled invariance across applications such as image classification [1], multi-lingual machine translation [24], and domain adaptation [87].

Despite its practical promise, the aforementioned ARL setup suffers from a number of drawbacks:

– Representation learning under adversarial settings is challenging in its own right. The minimax formulation of the problem leads to an optimization problem that is non-convex in the parameter space, both due to the adversarial loss function as well as due to the non-linear nature of modern NNs. As we show in this chapter, even for simple instances of ARL where each entity is characterized by a linear function, the problem remains non-convex in the parameter space. Similar observations [65] have been made in a different but related context of adversarial learning in generative adversarial networks (GAN)s [88].

– Current paradigm of SGDA to solve the ARL problem provides no provable guarantees while suffering from instability and poor convergence [1, 2]. Again, similar observations [64, 65] have

been made in the context of GANs, demonstrating the difficulty posed by the minimax formulation of the optimization and exposing the limitations of standard simultaneous optimization (i.e., SGDA).

– In applications of ARL related to fairness, accountability, and transparency of machine learning models, it is critically important to be able to provide performance bounds in addition to empirical evidence of their efficacy. A major shortcoming of existing works is the difficulty and lack of performance analysis and provable guarantees of unfairness or information leakage.

In this chapter, we take a step back and analytically study the simplest version of the ARL problem from an optimization perspective with the goal of addressing the aforementioned limitations. Doing so enables us to delineate the contributions of the expressivity of the entities in ARL (i.e., shallow versus DNNs) and the challenges of optimizing the parameters (i.e., local optima through SGDA versus global optima). We first consider the "linear" form of ARL, where the encoder is a linear transformation, the target predictor is a linear regression, and the proxy adversary is also a linear regressor. We show that this linear ARL leads to an optimization problem that is both non-convex and non-differentiable. Despite this fact, by reducing it into a set of trace problems on a Stiefel manifold, we obtain an exact closed-form solution for the global optima. As part of our solution, we also determine the optimal dimensionality of the embedding space. We then obtain analytical bounds (lower and upper) on the target and adversary objectives and prescribe a procedure to control the maximal leakage of sensitive information explicitly. Finally, we extend the linear-ARL formulation to allow non-linear functions in some RKHSs while still enjoying an exact closed-form solution for the global optima. Numerical experiments on multiple datasets, both small and large scale, indicate that the global optima solution for the linear and kernel formulations of ARL are competitive and sometimes even outperform DNN-based ARL trained through SGDA. Practically, we also demonstrate the utility of linear ARL and kernel-ARL for "imparting" prov-

able invariance to any biased pre-trained data representation. We refer to our proposed algorithm for obtaining the global optima as spectral-ARL and abbreviate it as SARL.

3.2 Adversarial Representation Learning

3.2.1 Problem Setting

The adversarial representation learning problem is formulated with the goal of learning parameters of an embedding function $f(\cdot;\boldsymbol{\Theta}) : X \mapsto Z$ with two objectives: (i) aiding a target predictor $g_Y(\cdot;\boldsymbol{\Theta}_Y)$ to accurately predict the target attribute Y from Z, and (ii) preventing an adversary $g_S(\cdot;\boldsymbol{\Theta}_S)$ from inferring the sensitive attribute S from Z. The ARL problem can be formulated as

$$\min_{\boldsymbol{\Theta}} \min_{\boldsymbol{\Theta}_Y} \mathbb{E}_{X,Y}\left[L_Y\left(g_Y\left(f(X;\boldsymbol{\Theta});\boldsymbol{\Theta}_Y\right),Y\right)\right] \quad \text{s.t.} \quad \min_{\boldsymbol{\Theta}_S} \mathbb{E}_{X,S}\left[L_S\left(g_S\left(f(X;\boldsymbol{\Theta});\boldsymbol{\Theta}_S\right),S\right)\right] \geq \alpha, \tag{3.1}$$

where L_Y and L_S are the loss functions for the target and the adversary predictors, respectively; $\alpha \in [0,\infty)$ is a user-defined value that determines the minimum tolerable loss for the adversary on the sensitive attribute. For example, $\alpha = 0$ corresponds to ignoring the adversary loss and resulting in standard representation learning, while $\alpha \to \infty$ corresponds to no tolerance for adversary performance. The minimization in the constraint is equivalent to the encoder operating against an optimal adversary. Existing instances of this problem adopt DNNs to represent f, g_Y, and g_S and learn their respective parameters $\{\boldsymbol{\Theta}, \boldsymbol{\Theta}_Y, \boldsymbol{\Theta}_S\}$ through SGDA. See Figure 3.2 for an illustration.

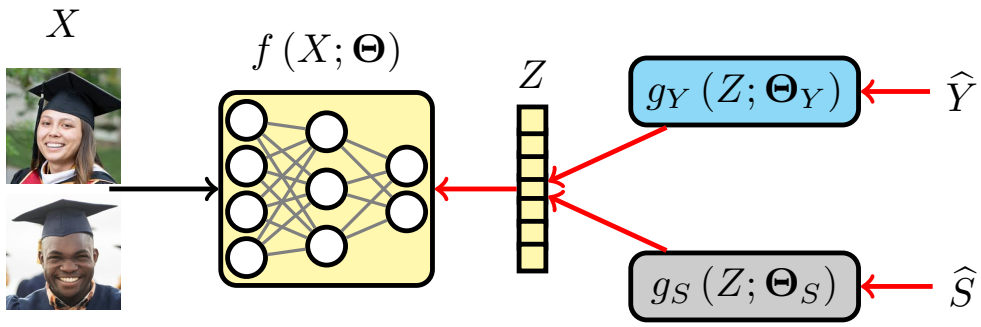
figure 

Figure 3.2: **ARL-SGDA:** Illustration of training adversarial representation learning through stochastic gradient descent ascent. i) At first, the target predictor parameters $\boldsymbol{\Theta}_Y$ are updated while the encoder and adversary are frozen. ii) Then, the adversary parameters $\boldsymbol{\Theta}_S$ are updated while the encoder and $\boldsymbol{\Theta}_Y$ are frozen. iii) Finally, the encoder parameters $\boldsymbol{\Theta}$ get updated while $\boldsymbol{\Theta}_Y$ and $\boldsymbol{\Theta}_S$ are frozen. SGDA does not provide any convergence guarantees.

3.2.2 The Linear Case

We first consider the simplest form of the ARL problem and analyze it from an optimization perspective. We model both the adversary and the target predictors as linear regressors

$$\widehat{Y} = \boldsymbol{\Theta}_Y Z + \boldsymbol{b}_Y, \qquad \widehat{S} = \boldsymbol{\Theta}_S Z + \boldsymbol{b}_S, \tag{3.2}$$

where Z is an encoded version of X, and $\widehat{Y}$ and $\widehat{S}$ are the predictions corresponding to the target and sensitive attributes, respectively. We also model the encoder through a linear mapping

$$\boldsymbol{\Theta} \in \mathbb{R}^{r \times d_X} \quad : \quad X \mapsto Z = \boldsymbol{\Theta} X, \tag{3.3}$$

where r is the dimensionality of the projected space. While existing NN-based solutions select r on an ad-hoc basis, our approach for this problem determines r as part of our solution to the ARL problem. See Figure 3.3 for an illustration. For both adversary and target predictors, we adopt the

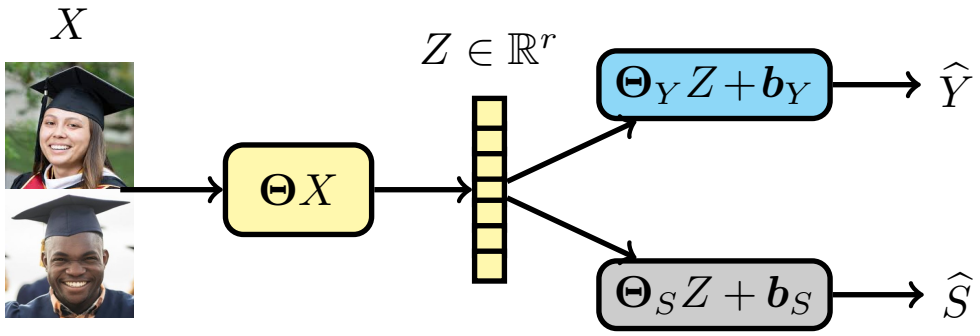


Figure 3.3: **Linear-ARL:** Illustration of linear adversarial representation learning for learning a fair representation. An encoder f, in the form of a linear mapping, produces a new representation $Z = \boldsymbol{\Theta} X$. A target predictor g_Y and an adversary g_S, in the form of linear regressors, operate on the representation Z. We analytically analyze this ARL setup to obtain a closed-form solution for the globally optimal parameters of the encoder $\boldsymbol{\Theta}$. Provable bounds on the trade-off between utility and fairness of the representation are also derived.

MSE to assess the quality of their respective predictions, i.e.,

$$L_Y(Y, \widehat{Y}) = \mathbb{E}\left[\|Y - \widehat{Y}\|^2\right], \qquad L_S(S, \widehat{S}) = \mathbb{E}\left[\|S - \widehat{S}\|^2\right].$$

3.2.2.1 Optimization Problem

For any given encoder $\boldsymbol{\Theta}$ the following lemma gives the minimum MSE for a linear regressor in terms of covariance matrices and $\boldsymbol{\Theta}$. The following Lemma assumes that the RV X is zero-mean and the covariance matrix $\boldsymbol{C}_X$ is positive definite. These assumptions are not restrictive since we can always remove the mean and dependent features from X.

Lemma 3.1. Let X and U be two RVs with $\mathbb{E}[X] = 0$, $\mathbb{E}[U] = \boldsymbol{b}$, where $\boldsymbol{C}_X \succ 0$. Consider a linear regressor, $\widehat{U} = \boldsymbol{W} Z + \boldsymbol{b}$, where $\boldsymbol{W} \in \mathbb{R}^{d_U \times r}$ is the parameter matrix, and $Z \in \mathbb{R}^r$ is an encoded version of X for a given $\boldsymbol{\Theta}$: $X \mapsto Z = \boldsymbol{\Theta} X, \quad \boldsymbol{\Theta} \in \mathbb{R}^{r \times d_X}$. The minimum MSE that can be achieved by designing $\boldsymbol{W}$ is

$$\min_{\boldsymbol{W}} \mathbb{E}\left[\|U - \widehat{U}\|^2\right] = \mathrm{Tr}\left[\boldsymbol{C}_U\right] - \left\|P_{\mathcal{M}} \boldsymbol{Q}_X^{-T} \boldsymbol{C}_{XU}\right\|_F^2,$$

where $\boldsymbol{M} = \boldsymbol{Q}_X \boldsymbol{\Theta}^T \in \mathbb{R}^{d_X \times r}$, and $\boldsymbol{C}_X = \boldsymbol{Q}_X^T \boldsymbol{Q}_X$ (Cholesky factorization).

Proof. See Appendix A.1 □

Applying this Lemma to the target and adversary regressors, we obtain their minimum MSEs as

$$J_Y(\boldsymbol{\Theta}) := \min_{\boldsymbol{\Theta}_Y} L_Y\left(g_Y\left(f(X;\boldsymbol{\Theta});\boldsymbol{\Theta}_Y\right), Y\right) = \text{Tr}\left[\boldsymbol{C}_Y\right] - \left\| P_{\mathcal{M}} \boldsymbol{Q}_X^{-T} \boldsymbol{C}_{XY} \right\|_F^2 \tag{3.4}$$

$$J_S(\boldsymbol{\Theta}) := \min_{\boldsymbol{\Theta}_S} L_S\left(g_S\left(f(X;\boldsymbol{\Theta});\boldsymbol{\Theta}_S\right), S\right) = \text{Tr}\left[\boldsymbol{C}_S\right] - \left\| P_{\mathcal{M}} \boldsymbol{Q}_X^{-T} \boldsymbol{C}_{XS} \right\|_F^2. \tag{3.5}$$

Given the encoder $\boldsymbol{\Theta}$, $J_Y(\boldsymbol{\Theta})$ is related to the performance of the target predictor, whereas $J_S(\boldsymbol{\Theta})$ corresponds to the amount of sensitive information that an adversary is able to extract. Note that the linear model for g_Y and g_S enables us to obtain their respective optimal solutions for a given encoder $\boldsymbol{\Theta}$. On the other hand, when g_Y and g_S are modeled as NNs, doing the same is analytically infeasible and potentially impractical.

The orthogonal projector $P_{\mathcal{M}}$ in Lemma 3.1 is a function of two factors: a data-dependent term $\boldsymbol{Q}_X$ and the encoder parameters $\boldsymbol{\Theta}$. While the former is fixed for a given dataset, the latter is our object of interest. Pursuantly, we decompose $P_{\mathcal{M}}$ in order to separably characterize the effect of these two factors. Let the columns of $\boldsymbol{L}_X \in \mathbb{R}^{d_X \times d_X}$ be an orthonormal basis for the column space of $\boldsymbol{Q}_X$, and $\boldsymbol{G} \in \mathbb{R}^{d_X \times r}$ be an arbitrary matrix. consider $\boldsymbol{L}_X \boldsymbol{G} := \boldsymbol{Q}_X \boldsymbol{\Theta}^T$. Due to the bijection

$$\boldsymbol{G} = \boldsymbol{L}_X^{-1} \boldsymbol{Q}_X \boldsymbol{\Theta}^T \Leftrightarrow \boldsymbol{\Theta} = \boldsymbol{G}^T \boldsymbol{L}_X^T \boldsymbol{Q}_X^{-T},$$

determining the encoder parameters $\boldsymbol{\Theta}$ is equivalent to determining $\boldsymbol{G}$. The projector $P_{\mathcal{M}}$ can now be expressed in terms of $P_{\mathcal{G}}$, which is only dependent on the free parameter $\boldsymbol{G}$:

$$P_{\mathcal{M}} = \boldsymbol{M}\left(\boldsymbol{M}^T \boldsymbol{M}\right)^{\dagger} \boldsymbol{M}^T = \boldsymbol{L}_X P_{\mathcal{G}} \boldsymbol{L}_X^T, \tag{3.6}$$

where we used the equality $\boldsymbol{M} = \boldsymbol{Q}_X \boldsymbol{\Theta}^T$ and the fact that $\boldsymbol{L}_X^T \boldsymbol{L}_X = \boldsymbol{I}$.

Now, we turn back to the ARL setup and see how the above decomposition can be leveraged. The optimization problem in (3.1) reduces to

$$\begin{aligned} &\min_{\boldsymbol{G}} \; J_Y(\boldsymbol{G}) \\ &\text{s.t.} \quad J_S(\boldsymbol{G}) \geq \alpha, \end{aligned} \tag{3.7}$$

where the minimum MSE measures of (3.4) and (3.5) are now expressed in terms of $\boldsymbol{G}$ instead of $\boldsymbol{\Theta}$.

Before solving this optimization problem, we will first interpret it geometrically. Consider a simple example where X is a white RV, i.e., $\boldsymbol{C}_X = \boldsymbol{I}$. Under this setting, $\boldsymbol{Q}_X = \boldsymbol{L}_X = \boldsymbol{I}$ and $\boldsymbol{G} = \boldsymbol{\Theta}$. As a result, the optimization problem in (3.7) can alternatively be solved in terms of $\boldsymbol{G} = \boldsymbol{\Theta}$, where $J_Y(\boldsymbol{G}) = \text{Tr}\left[\boldsymbol{C}_Y\right] - \left\|P_{\mathcal{G}} \boldsymbol{C}_{XY}\right\|_F^2$ and $J_S(\boldsymbol{G}) = \text{Tr}\left[\boldsymbol{C}_S\right] - \left\|P_{\mathcal{G}} \boldsymbol{C}_{XS}\right\|_F^2$.

The constraint $J_S(\boldsymbol{G}) \geq \alpha$ implies $\left\|P_{\mathcal{G}} \boldsymbol{C}_{XS}\right\|_F^2 \leq \left(\text{Tr}\left[\boldsymbol{C}_S\right] - \alpha\right)$ which is geometrically equivalent to the subspace $\mathcal{G}$ being outside (or tangent to) the cone around $\boldsymbol{C}_{XS}$. Similarly, minimizing $J_Y(\boldsymbol{G})$ implies maximizing $\left\|P_{\mathcal{G}} \boldsymbol{C}_{XY}\right\|_F^2$, which in turn is equivalent to minimizing the angle between the subspace $\mathcal{G}$ and the vector $\boldsymbol{C}_{XY}$. Therefore, the global optima of (3.7) are any hyperplane $\mathcal{G}$ which is outside the cone around $\boldsymbol{C}_{XS}$ while subtending the smallest angle to $\boldsymbol{C}_{XY}$. An illustration of this setting and its solution is shown in Figure 3.4 for $d_X = 3$, $r = 2$, and $d_Y = d_S = 1$.

Constrained optimization problems such as (3.7) are commonly solved through their respective unconstrained scalarization [89] formulations as shown below

$$\min_{\boldsymbol{G} \in \mathbb{R}^{d_X \times r}} \left\{(1-\lambda)\, J_Y(\boldsymbol{G}) - \lambda\, J_s(\boldsymbol{G})\right\} \tag{3.8}$$

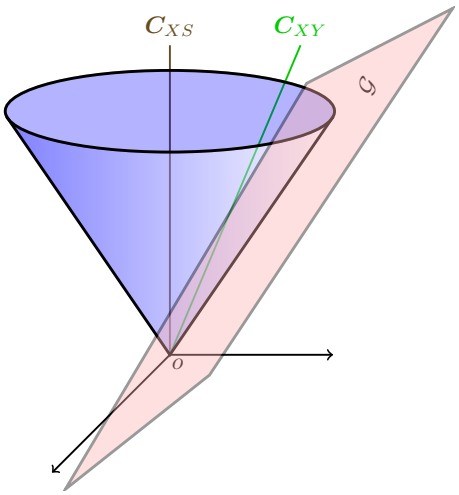


Figure 3.4: **Geometric Interpretation:** An illustration of a three-dimensional input space X and one-dimensional target and adversary regressors. Therefore, both $\boldsymbol{C}_{XS}$ and $\boldsymbol{C}_{XY}$ are one-dimensional. We locate the y-axis in the same direction as $\boldsymbol{C}_{XS}$. The feasible space for the solution $\boldsymbol{G} = \boldsymbol{\Theta}$ imposed by the constraint $J_S(\boldsymbol{\Theta}) \geq \alpha$ corresponds to the region *outside* the cone (specified by $\boldsymbol{C}_S$ and α) around $\boldsymbol{C}_{XS}$. The non-convexity of the problem stems from the non-convexity of this feasible set. The objective $\min J_Y(\boldsymbol{\Theta})$ corresponds to minimizing the angle between the line $\boldsymbol{C}_{XY}$ and the plane $\mathcal{G}$. When $\boldsymbol{C}_{XY}$ is outside the cone, the line $\boldsymbol{C}_{XY}$ itself or any plane that contains the line $\boldsymbol{C}_{XY}$ and does not intersect with the cone, is a valid solution. When $\boldsymbol{C}_{XY}$ is inside the cone, the solution is either a line or, as we illustrate, a tangent hyperplane to the cone that is closest to $\boldsymbol{C}_{XY}$. The non-differentiability stems from the fact that the solution can either be a plane or a line.

for some parameter $0 \leq \lambda < 1$. Such an approach affords two main advantages and one disadvantage: (a) A direct and closed-form solution can be obtained. (b) Framing (3.8) in terms of λ and $(1-\lambda)$ allows explicit control between the two extremes of *no fairness* ($\lambda = 0$) and *only fairness* ($\lambda \to 1$). As a consequence, it can be shown that for every $\lambda \in [0, 1)$, $\exists\, \alpha \in [\alpha_{\min}, \alpha_{\max}]$ (see Appendix A.2 for a proof). (c) The vice-versa, on the other hand, does not necessarily hold, i.e., for a given tolerable loss α, there may not be a corresponding $\lambda \in [0, 1)$. This is the theoretical limit of solving a scalarized problem instead of the constrained problem.

Before we obtain the solution to the scalarization formulation (3.8), we characterize the nature of the optimization problem in the following theorem.

Theorem 3.2. *As a function of $\boldsymbol{G} \in \mathbb{R}^{d_X \times r}$, the objective function in (3.8) is neither convex nor differentiable.*

Proof. See Appendix A.3 □

3.2.2.2 Learning

Despite the difficulty associated with the objective in (3.8), we derive a closed-form solution for its global optima. Our key insight lies in partitioning the search space $\mathbb{R}^{d_X \times r}$ based on the rank of the matrix $\boldsymbol{G}$ (i.e., the number of independent rows or columns of $\boldsymbol{G}$). For a given rank i, let $\mathcal{S}_i$ be the set containing all matrices $\boldsymbol{G}$ of rank i,

$$\mathcal{S}_i = \left\{ \boldsymbol{G} \in \mathbb{R}^{d_X \times r} \quad | \quad \text{rank}(\boldsymbol{G}) = i \right\}, \quad i = 0, 1, \cdots, r.$$

Since $\bigcup_{i=0}^{r} \mathcal{S}_i = \mathbb{R}^{d_X \times r}$, the optimization problem in (3.8) can be solved by considering r mini-

mization problems, one for each possible rank of $\boldsymbol{G}$:

$$\min_{i\in\{1,\ldots,r\}} \left\{ \min_{\boldsymbol{G}\in\mathcal{S}_i} (1-\lambda)\, J_Y(\boldsymbol{G}) - \lambda\, J_S(\boldsymbol{G}) \right\} \tag{3.9}$$

We observe from (3.4), (3.5), and (3.6) that the optimization problem in (3.8) is dependent only on a subspace $\mathcal{G}$. As such, the solution $\boldsymbol{G}$ is not unique since many different matrices can span the same subspace. Hence, it is sufficient to solve for any particular $\boldsymbol{G}$ that spans the optimal subspace $\mathcal{G}$. Without loss of generality, we seek an orthonormal basis spanning the optimal subspace $\mathcal{G}$ as our desired solution. We constrain $\boldsymbol{G} \in \mathbb{R}^{d_X \times i}$ to be an orthonormal matrix i.e., $\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}_i$, where i is the dimensionality of $\mathcal{G}$. Ignoring the constant terms in J_Y and J_S, for each $i = 1, \ldots, r$, the minimization problem over $\mathcal{S}_i$ in (3.9) reduces to

$$\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}_i} J_\lambda(\boldsymbol{G}), \tag{3.10}$$

where

$$J_\lambda(\boldsymbol{G}) := \lambda \left\|\boldsymbol{L}_X\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{L}_X^T\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XS}\right\|_F^2 - (1-\lambda)\left\|\boldsymbol{L}_X\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{L}_X^T\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XY}\right\|_F^2.$$

From basic properties of trace, we have, $J_\lambda(\boldsymbol{G}) = \text{Tr}\left[\boldsymbol{G}^T\boldsymbol{B}\boldsymbol{G}\right]$ where $\boldsymbol{B} \in \mathbb{R}^{d_X \times d_X}$ is a symmetric matrix:

$$\boldsymbol{B} = \boldsymbol{L}_X^T\boldsymbol{Q}_X^{-T}\left(\lambda\, \boldsymbol{C}_{SX}^T\boldsymbol{C}_{SX} - (1-\lambda)\, \boldsymbol{C}_{YX}^T\boldsymbol{C}_{YX}\right)\boldsymbol{Q}_X^{-1}\boldsymbol{L}_X. \tag{3.11}$$

The optimization problem in (3.10) is a trace minimization on a Stiefel manifold which has closed-form solution(s) (see [90] and [91]).

In view of the above discussion, the solution to the optimization problem in (3.8) or equivalently (3.9) can be stated in the next theorem.

Theorem 3.3. *Assume that the number of negative eigenvalues (β) of $\boldsymbol{B}$ in (3.11) is j. Denote $\gamma = \min\{r, j\}$. Then, the minimum value in (3.9) is given as,*

$$\beta_1 + \beta_2 + \cdots + \beta_\gamma \tag{3.12}$$

where $\beta_1 \leq \beta_2 \leq \ldots \leq \beta_\gamma < 0$ are the γ smallest eigenvalues of $\boldsymbol{B}$. And the minimum can be attained by $\boldsymbol{G} = \boldsymbol{V}$, where the columns of $\boldsymbol{V}$ are eigenvectors corresponding to all the γ negative eigenvalues of $\boldsymbol{B}$.

Proof. Consider the inner optimization problem of (3.10) in (3.9). Using the trace optimization problems and their solutions in [90], we get

$$\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}_i} J_\lambda(\boldsymbol{G}) = \min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}_i} \mathrm{Tr}\Big[\boldsymbol{G}^T\boldsymbol{B}\boldsymbol{G}\Big] = \beta_1 + \beta_2 + \cdots + \beta_i,$$

where $\beta_1, \beta_2, \ldots, \beta_i$ are i smallest eigenvalues of $\boldsymbol{B}$ and minimum value can be achieved by the matrix $\boldsymbol{V}$ whose columns are corresponding eigenvectors. If the number of negative eigenvalues of $\boldsymbol{B}$ is less than r, then the optimum i in (3.9) is j, otherwise the optimum i is r. □

Note that including the eigenvectors corresponding to zero eigenvalues of $\boldsymbol{B}$ into our solution $\boldsymbol{G}$ in Theorem 3.3 does not change the minimum value in (3.12). But, considering only the eigenvectors corresponding to negative eigenvalues results in $\boldsymbol{G}$ with the least rank and, thereby, an encoder that is less likely to contain sensitive information for an adversary to exploit. Once $\boldsymbol{G}$ is constructed, we can obtain our desired encoder as $\boldsymbol{\Theta} = \boldsymbol{G}^T\boldsymbol{L}_X^T\boldsymbol{Q}_X^{-T}$. Recall that the solution in Theorem 3.3 is under the assumption that the covariance $\boldsymbol{C}_X$ is a full-rank matrix.

3.3 Empirical Solution for Linear Encoder

In many practical scenarios, we only have access to data samples but not to the population mean vectors and covariance matrices. Therefore, the population solution might not be feasible in such as case. In this section, we provide an approach to solve the optimization problem in (3.3), which relies on empirical moments and is valid even if the covariance matrix $\boldsymbol{C}_X$ is not full-rank.

Firstly, for a given $\boldsymbol{\Theta}$, we find

$$J_Y = \min_{\boldsymbol{W}_Y, \boldsymbol{b}_Y} \text{MSE}\,(\widehat{Y} - Y).$$

Note that the above optimization problem can be separated over $\boldsymbol{W}_Y$ and $\boldsymbol{b}_Y$. Therefore, for a given $\boldsymbol{W}_Y$, we first minimize over $\boldsymbol{b}_Y$:

$$\begin{aligned}\min_{\boldsymbol{b}_Y} \mathbb{E}\left\{\|\boldsymbol{W}_Y\boldsymbol{\Theta}X + \boldsymbol{b}_Y - Y\|^2\right\} &= \min_{\boldsymbol{b}_Y} \frac{1}{n}\sum_{k=1}^{n} \|\boldsymbol{W}_Y\boldsymbol{\Theta}\boldsymbol{x}_k + \boldsymbol{b}_Y - \boldsymbol{y}_k\|^2 \\ &= \frac{1}{n}\sum_{k=1}^{n} \|\boldsymbol{W}_Y\boldsymbol{\Theta}\boldsymbol{x}_k + \boldsymbol{c} - \boldsymbol{y}_k\|^2,\end{aligned}$$

where we used the empirical expectation and the minimizer $\boldsymbol{c}$ is

$$\begin{aligned}\boldsymbol{c} = \frac{1}{n}\sum_{k=1}^{n}(\boldsymbol{y}_k - \boldsymbol{W}_Y\boldsymbol{\Theta}\boldsymbol{x}_k) &= \frac{1}{n}\sum_{k=1}^{n}\boldsymbol{y}_k - \boldsymbol{W}_Y\boldsymbol{\Theta}\frac{1}{n}\sum_{k=1}^{n}\boldsymbol{x}_k \\ &= \mathbb{E}\{Y\} - \boldsymbol{W}_Y\boldsymbol{\Theta}\,\mathbb{E}\{X\}.\end{aligned} \tag{3.13}$$

Let all the columns of matrix $\boldsymbol{C}$ be equal to $\boldsymbol{c}$. We now have

$$
\begin{aligned}
J_Y &= \min_{\boldsymbol{W}_Y, \boldsymbol{b}_Y} \text{MSE}\,(\widehat{Y} - Y) \\
&= \min_{\boldsymbol{W}_Y} \frac{1}{n} \left\| \boldsymbol{W}_Y \boldsymbol{\Theta} \boldsymbol{X} + \boldsymbol{C} - \boldsymbol{Y} \right\|_F^2 \\
&= \min_{\boldsymbol{W}_Y} \frac{1}{n} \left\| \boldsymbol{W}_Y \boldsymbol{\Theta} \tilde{\boldsymbol{X}} - \tilde{\boldsymbol{Y}} \right\|_F^2 \\
&= \min_{\boldsymbol{W}_Y} \frac{1}{n} \left\| \tilde{\boldsymbol{X}}^T \boldsymbol{\Theta}^T \boldsymbol{W}_Y^T - \tilde{\boldsymbol{Y}}^T \right\|_F^2 \\
&= \min_{\boldsymbol{W}_Y} \frac{1}{n} \left\| \boldsymbol{M} \boldsymbol{W}_Y^T - P_{\mathcal{M}} \tilde{\boldsymbol{Y}}^T \right\|_F^2 + \frac{1}{n} \left\| P_{\mathcal{M}^\perp} \tilde{\boldsymbol{Y}}^T \right\|_F^2 \\
&= \frac{1}{n} \left\| \underbrace{\boldsymbol{M} \boldsymbol{M}^\dagger}_{P_{\mathcal{M}}} P_{\mathcal{M}} \tilde{\boldsymbol{Y}}^T - P_{\mathcal{M}} \tilde{\boldsymbol{Y}}^T \right\|_F^2 + \frac{1}{n} \left\| P_{\mathcal{M}^\perp} \tilde{\boldsymbol{Y}}^T \right\|_F^2 \\
&= \frac{1}{n} \left\| P_{\mathcal{M}^\perp} \tilde{\boldsymbol{Y}}^T \right\|_F^2 \\
&= \frac{1}{n} \left\| \tilde{\boldsymbol{Y}}^T \right\|_F^2 - \frac{1}{n} \left\| P_{\mathcal{M}} \tilde{\boldsymbol{Y}}^T \right\|_F^2,
\end{aligned}
$$

where in the third step we used (3.13), $\boldsymbol{M} = \tilde{\boldsymbol{X}}^T \boldsymbol{\Theta}^T$ and the fifth step is due to orthogonal decomposition. Using the same approach, we get

$$
J_S = \frac{1}{n} \left\| \tilde{\boldsymbol{S}}^T \right\|_F^2 - \frac{1}{n} \left\| P_{\mathcal{M}} \tilde{\boldsymbol{S}}^T \right\|_F^2. \tag{3.14}
$$

Now, assume that the columns of $\boldsymbol{L}_x$ are the orthogonal basis for the column space of $\tilde{\boldsymbol{X}}^T$. Therefore, for any $\boldsymbol{M}$, there exists a $\boldsymbol{G}$ such that $\boldsymbol{L}_X \boldsymbol{G} = \boldsymbol{M}$. In general, there is no bijection between $\boldsymbol{\Theta}$ and $\boldsymbol{G}$ in the equality $\tilde{\boldsymbol{X}}^T \boldsymbol{\Theta} = \boldsymbol{L}_X \boldsymbol{G}$. But, there is a bijection between $\boldsymbol{G}$ and $\boldsymbol{\Theta}$ when constrained to $\boldsymbol{\Theta}$'s in which $\mathcal{R}(\boldsymbol{\Theta}^T) \subseteq \mathcal{N}(\tilde{\boldsymbol{X}}^T)^\perp$. This restricted bijection is sufficient to be considered since for any $\boldsymbol{\Theta}^T \in \mathcal{N}(\tilde{\boldsymbol{X}}^T)$, we have $\boldsymbol{M} = \boldsymbol{0}$. Once $\boldsymbol{G}$ is determined, $\boldsymbol{\Theta}^T$ can be

obtained as

$$\boldsymbol{\Theta}^T = (\tilde{\boldsymbol{X}}^T)^\dagger \boldsymbol{L}_X \boldsymbol{G} + \boldsymbol{\Theta}_0, \;\; \boldsymbol{\Theta}_0 \subseteq \mathcal{N}(\tilde{\boldsymbol{X}}^T).$$

However, since

$$\|\boldsymbol{\Theta}\|_F^2 = \left\|\boldsymbol{\Theta}^T\right\|_F^2 = \left\|(\tilde{\boldsymbol{X}}^T)^\dagger \boldsymbol{L}_X \boldsymbol{G}\right\|_F^2 + \|\boldsymbol{\Theta}_0\|_F^2 \,,$$

choosing $\boldsymbol{\Theta}_0 = \mathbf{0}$ results in minimum $\|\boldsymbol{\Theta}\|_F$, which is favorable in terms of robustness to noise.

By choosing $\boldsymbol{\Theta}_0 = \mathbf{0}$, determining the encoder $\boldsymbol{\Theta}$ would be equivalent to determining $\boldsymbol{G}$. Similar to (3.6), we have $P_{\mathcal{M}} = \boldsymbol{L}_X P_{\mathcal{G}} \boldsymbol{L}_X^T$. If we assume that the rank of $P_{\mathcal{G}}$ is i, $J_\lambda(\boldsymbol{G})$ in (3.10) can be expressed as

$$J_\lambda(\boldsymbol{G}) = \lambda \left\|\boldsymbol{L}_X \boldsymbol{G}\boldsymbol{G}^T \boldsymbol{L}_X^T \tilde{\boldsymbol{S}}^T\right\|_F^2 - (1-\lambda) \left\|\boldsymbol{L}_X \boldsymbol{G}\boldsymbol{G}^T \boldsymbol{L}_X^T \tilde{\boldsymbol{Y}}^T\right\|_F^2$$

where $\boldsymbol{G}\boldsymbol{G}^T = P_{\mathcal{G}}$ for some orthogonal matrix $\boldsymbol{G} \in \mathbb{R}^{d_X \times i}$. This resembles the optimization problem in (3.9) and therefore it has the same solution as Theorem 3.3 with modified $\boldsymbol{B}$ given by

$$\boldsymbol{B} = \boldsymbol{L}_X^T \left(\lambda \, \tilde{\boldsymbol{S}}^T \tilde{\boldsymbol{S}} - (1-\lambda) \, \tilde{\boldsymbol{Y}}^T \tilde{\boldsymbol{Y}}\right) \boldsymbol{L}_X \tag{3.15}$$

Once $\boldsymbol{G}$ is determined, $\boldsymbol{\Theta}$ can be obtained as $\boldsymbol{G}^T \boldsymbol{L}_X^T \tilde{\boldsymbol{X}}^\dagger$.

3.3.1 Non-Linear Extension Through Kernelization

We extend the "linear" version of the ARL problem studied thus far to a "non-linear" version through kernelization. We model the encoder in the ARL problem as a linear function over the non-linear mapping of inputs as illustrated in Figure 3.5. Let the data matrix $\boldsymbol{X}$ be mapped non-linearly by a possibly unknown and infinite dimensional function $\phi_X(\cdot)$ and the corresponding

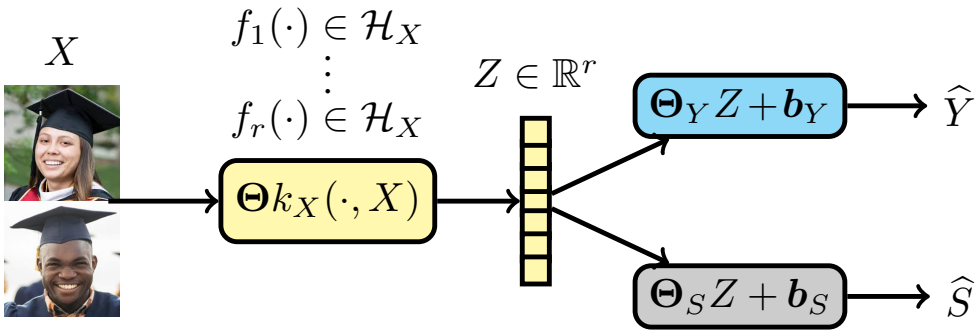


Figure 3.5: **Kernel-ARL:** Illustration of kernel adversarial representation learning for learning a fair representation. An encoder f, in the form of a linear mapping on top of kernelized input, produces a new representation $Z = \boldsymbol{\Theta}\left[k_X(\boldsymbol{x}_1, X), \cdots, k_X(\boldsymbol{x}_n, X)\right]^T$. A target predictor g_Y and an adversary g_S, in the form of linear regressors, operate on the representation Z.

reproducing kernel function be $k_X(\cdot, \cdot)$.

From (2.6), it follows that the representation Z can be expressed as

$$Z = \boldsymbol{\Theta}\left[k_X(\boldsymbol{x}_1, X), \cdots, k_X(\boldsymbol{x}_n, X)\right]^T. \tag{3.16}$$

The scalarization formulation of this kernel-ARL setup and its solution share the same form as that of the linear case (3.8). The objective function remains non-convex and non-differentiable, while the matrix $\boldsymbol{B}$ is now dependent on the kernel matrix $\boldsymbol{K}_X$ as opposed to the covariance matrix $\boldsymbol{C}_X$ (see Appendix A.5 for details):

$$\boldsymbol{B} = \boldsymbol{L}_X^T \left(\lambda\, \tilde{\boldsymbol{S}}^T \tilde{\boldsymbol{S}} - (1-\lambda)\, \tilde{\boldsymbol{Y}}^T \tilde{\boldsymbol{Y}}\right) \boldsymbol{L}_X, \tag{3.17}$$

where the columns of $\boldsymbol{L}_X$ are the orthonormal basis for $\boldsymbol{H}\boldsymbol{K}_X$. Once $\boldsymbol{G}$ is obtained through the eigendecomposition of $\boldsymbol{B}$, we can obtain the optimal encoder as $\boldsymbol{\Theta} = \boldsymbol{G}^T \boldsymbol{L}_X^T \boldsymbol{K}_X^\dagger$. This non-linear extension in the form of kernelization serves to study the ARL problem under a setting where the encoder possesses greater representational capacity while still being able to obtain the global optima and bounds on the objectives of the target predictor and the adversary.

3.4 Analytical Bounds

In this section, we introduce bounds on the utility and invariance of the representation learned by SARL. We define four bounds $\alpha_{\min}$, $\alpha_{\max}$, $\gamma_{\min}$ and $\gamma_{\max}$.

$\gamma_{\min}$: A lower bound on the minimum achievable target loss, or equivalently an upper bound on the best achievable target performance. This bound can be expressed as the minimum target MSE across all possible encoders $\mathbf{\Theta}$ and is attained at $\lambda = 0$:

$$\gamma_{\min} = \min_{\mathbf{\Theta}} J_Y(\boldsymbol{\theta})$$

$\alpha_{\max}$: A upper bound on the maximum achievable adversary loss, or equivalently a lower bound on the minimum leakage of the sensitive attribute. This bound can be expressed as the maximum adversary MSE across all possible encoders $\mathbf{\Theta}$ and is attained at $\lambda = 1$:

$$\alpha_{\max} = \max_{\mathbf{\Theta}} J_S(\mathbf{\Theta})$$

$\gamma_{\max}$: An upper bound on the maximum achievable target loss, or equivalently a lower bound on the minimum achievable target performance. This bound corresponds to the scenario where the encoder is constrained to hinder the adversary maximally. In all other cases, one can obtain higher target performance by choosing a better encoder. This bound is attained in the limit $\lambda \to 1$ and can be expressed as

$$\gamma_{\max} = \min_{\arg\max J_S(\mathbf{\Theta})} J_Y(\mathbf{\Theta})$$

$\alpha_{\min}$: A lower bound on the minimum achievable adversary loss, or equivalently an upper bound on the maximum leakage of the sensitive attribute. The absolute lower bound is obtained in the

scenario where the encoder is neither constrained to aid the target nor hinder the adversary, i.e.,

$$\alpha^*_{min} = \min_{\boldsymbol{\Theta}} J_S(\boldsymbol{\Theta})$$

However, this is an unrealistic scenario since in the ARL problem, by definition, the encoder is explicitly designed to aid the target. Therefore, a more realistic lower bound can be defined under the constraint that the encoder maximally aids the target, i.e.,

$$\bar{\alpha}_{\min} = \min_{\arg\min J_y(\boldsymbol{\Theta})} J_S(\boldsymbol{\Theta})$$

However, even this bound is not realistic since, among all the encoders that aid the target, one can always choose the encoder that minimizes the leakage of the sensitive attribute. The bound corresponding to such an encoder can be expressed as

$$\alpha_{\min} = \max_{\arg\min J_Y(\boldsymbol{\Theta})} J_S(\boldsymbol{\Theta})$$

This bound is attained in the limit $\lambda \to 0$. It is easy to see that these bounds are ordinally related as

$$\alpha^*_{\min} \leq \bar{\alpha}_{\min} \leq \alpha_{\min}$$

To summarize, in each of these cases, there exists an encoder that achieves the respective bound. Therefore, given a choice, the encoder that corresponds to $\alpha_{\min}$ is the most desirable.

The following Lemma defines these bounds and their respective closed-form expressions as a function of data.

Theorem 3.4. *Let the columns of $\boldsymbol{L}_X$ be the orthonormal basis for $\boldsymbol{H}\boldsymbol{K}_X$. Further, assume that*

the columns of $\boldsymbol{V}_S$ are the singular vectors corresponding to zero singular values of $\tilde{\boldsymbol{S}}\boldsymbol{L}_X$ and the columns of $\boldsymbol{V}_Y$ are the singular vectors corresponding to non-zero singular values of $\tilde{\boldsymbol{Y}}\boldsymbol{L}_X$. Then, we have

$$\begin{aligned}
\gamma_{\min} &:= \min_{\boldsymbol{\Theta}} J_Y(\boldsymbol{\Theta}) = \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\|\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\|_F^2 \\
\gamma_{\max} &:= \min_{\arg\max J_S(\boldsymbol{\Theta})} J_Y(\boldsymbol{\Theta}) = \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\boldsymbol{V}_S\right\|_F^2 \\
\alpha_{\min} &:= \max_{\arg\min J_Y(\boldsymbol{\Theta})} J_S(\boldsymbol{\Theta}) = \frac{1}{n}\left\|\tilde{\boldsymbol{S}}^T\right\|_F^2 - \frac{1}{n}\left\|\tilde{\boldsymbol{S}}\boldsymbol{L}_X\boldsymbol{V}_Y\right\|_F^2 \\
\alpha_{\max} &:= \max_{\boldsymbol{\Theta}} J_S(\boldsymbol{\Theta}) = \frac{1}{n}\left\|\tilde{\boldsymbol{S}}^T\right\|_F^2
\end{aligned}$$

Proof. See Appendix A.6 □

Under the special case of one-dimensional data, i.e., X, Y, and S are scalars, the above bounds can be related to the correlation coefficients (i.e., normalized correlations) of the variables involved. Specifically, the normalized bounds $\gamma_{\min}$ and $\alpha_{\min}$ can be expressed as,

$$\begin{aligned}
\frac{\gamma_{\min}}{\sigma_S^2} &= 1 - \rho^2(X, Y) \\
\frac{\alpha_{\min}}{\sigma_S^2} &= 1 - \rho^2(X, S)
\end{aligned}$$

where $\rho(\cdot, \cdot)$ denotes the correlation coefficient between two RVs and $\sigma_Y^2 := \mathbb{V}\text{ar}[Y]$ (σ_S^2 is similarly defined). Similarly, the upper bounds $\gamma_{\max}$ and $\alpha_{\max}$ can be expressed in terms of the variance of the label space as

$$\frac{\gamma_{\max}}{\sigma_y^2} = \frac{\alpha_{\max}}{\sigma_s^2} = 1.$$

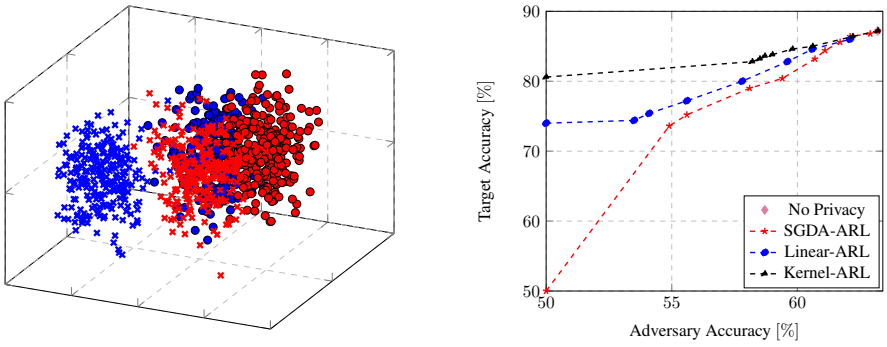
Therefore, in the one-dimensional setting, the achievable bounds are related to the underlying alignment between the subspace spanned by the data $\boldsymbol{X}$, and the respective subspaces spanned by the labels $\boldsymbol{S}$ and $\boldsymbol{Y}$.

3.5 Computational Complexity

In the case of linear-SARL, calculating the covariance matrices $\boldsymbol{C}_X$, $\boldsymbol{C}_{YX}$ and $\boldsymbol{C}_{SX}$ requires $\mathcal{O}(d_X^2 n)$, $\mathcal{O}(d_Y^2 n)$, and $\mathcal{O}(d_S^2 d_X)$ multiplications, respectively. Next, the complexity of Cholesky factorization $\boldsymbol{C}_X = \boldsymbol{Q}_X^T \boldsymbol{Q}_X$ and calculating its inverse $\boldsymbol{Q}_X^{-1}$ is $\mathcal{O}(d_X^3)$ each. Finally, solving the optimization problem has a complexity of $\mathcal{O}(d_X^3)$ to eigendecompose the $d_X \times d_X$ matrix $\boldsymbol{B}$. In the case of kernel-SARL, the eigendecomposition of $\boldsymbol{B}$ requires $\mathcal{O}(n^3)$ operations. However, for scalability, i.e., large n (e.g., CIFAR-100), the Nyström method with data sampling [92] can be adopted. To summarize, the complexity of the linear and kernel formulations is $\mathcal{O}(d_X^3)$ and $\mathcal{O}(n^3)$, respectively.

3.6 Numerical Experiments

We evaluate the efficacy of the proposed Spectral-ARL (SARL) algorithm in finding the global optima and compare it with other ARL baselines that are based on the standard SGDA optimization. In all experiments, we refer to our solution for "linear" ARL as Linear-SARL and the solution to the "kernel" version of the encoder with linear classifiers for the predictor and adversary as Kernel-SARL.



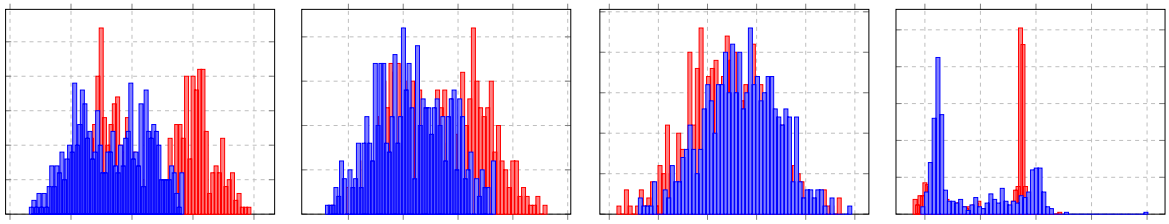
(a) (b)

Figure 3.6: (a) Samples from a mixture of four Gaussians. Each sample has two attributes, shape and color. (b) The trade-off between target performance and leakage of a sensitive attribute by an adversary.

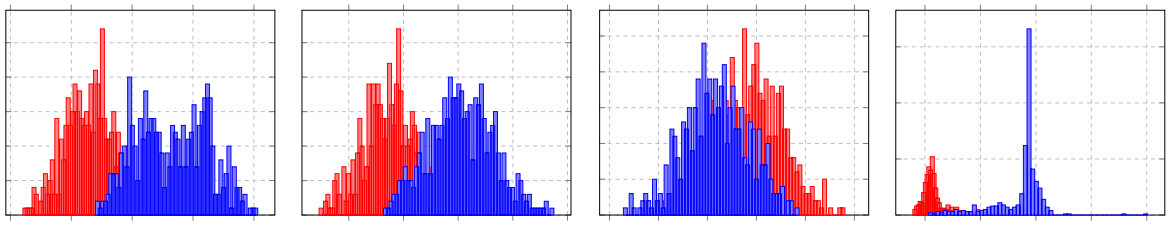
3.6.1 Mixture of Four Gaussians

We first consider a simple example in order to visualize and compare the learned embeddings from different ARL solutions. We consider a three-dimensional problem where each data sample consists of two attributes, color and shape. Specifically, the input data $\boldsymbol{X}$ is generated from a mixture of four different Gaussian distributions corresponding to different possible combinations of the attributes, i.e., $\{\bullet, \bullet, \times, \times\}$ with means at $\boldsymbol{\mu}_1 = (1, 1, 0)$, $\boldsymbol{\mu}_2 = (2, 2, 0)$, $\boldsymbol{\mu}_3 = (2, 2.5, 0)$, $\boldsymbol{\mu}_4 = (2.5, 3, 0)$ and identical covariance matrices $\boldsymbol{C}_X = \text{diag}\left(0.3^2, 0.3^2, 0.3^2\right)$. The shape attribute is the target, while color is the sensitive attribute, as illustrated in Figure 3.6 (a). The goal of the ARL problem is to learn an encoder that projects the data such that it remains separable with respect to the shape and non-separable with respect to the color attribute.

We sample 4000 points to learn linear and non-linear (RBF Gaussian kernel) encoders across $\lambda \in [0, 1]$. To train the encoder, the one-hot encoding of target and sensitive labels are treated as the regression targets. Then, we freeze the encoder and train logistic regressors for the adversary and target task for each λ. We evaluate their classification performance on a separate set of 1000 samples. The resulting trade-off front between target and adversary performance is shown



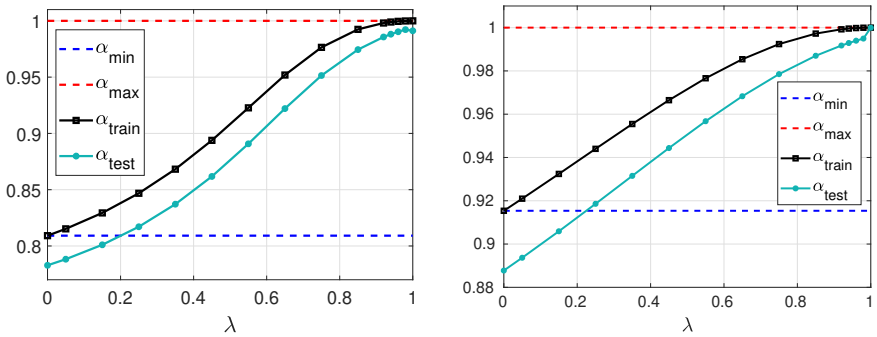
(a) Color ($\lambda = 0$) (b) Color ($\lambda = 0.5$) (c) Color ($\lambda = 1$) (d) Color ($\lambda = 0.5$)



(e) Shape ($\lambda = 0$) (f) Shape ($\lambda = 0.5$) (g) Shape (Kernel, $\lambda = 1$) (h) Shape (Kernel, $\lambda = 0.5$)

Figure 3.7: **Gaussian Mixture:** The optimal dimensionality of embedding Z is 1. Visualization of the embedding histograms w.r.t each attribute for different relative emphasis, λ, on the target (shape) and sensitive attributes (color). The top row is color and the bottom row is shape. The first three columns show results for a linear encoder. At $\lambda = 0$, the weight on the adversary is 0, so the color is still separable. As the value of λ increases, we observe that the colors are less and less separable. The last column shows results for a kernel encoder for $\lambda = 0.5$. We observe that the target attribute is quite separable while the sensitive attribute is entangled.

in Figure 3.6 (b). We make the following observations, (1) For $\lambda = 1$, all methods achieve an accuracy of 50% for the adversary, which indicates complete removal of features corresponding to the sensitive attribute via our encoding, (2) At small values of λ the objective of Linear-ARL is close to being convex, hence the similarity in the trade-off fronts of Linear-SARL and SGDA-ARL in that region. However, everywhere else, due to the iterative nature of SGDA, it is unable to find the global solution and achieve the same trade-off as Linear-SARL. (3) The non-linear encoder in the Kernel-SARL solution significantly outperforms both Linear-SARL and SGDA-ARL. The non-linear nature of the encoder enables it to strongly entangle the color attribute (50% accuracy) while simultaneously achieving a higher target accuracy than the linear encoder. Figure 3.7 visualizes the learned embedding space Z for different trade-offs between the target and adversary objectives.



(a) Linear-SARL (b) Kernel-SARL

Figure 3.8: **Gaussian Mixture:** Lower and upper bounds on adversary loss, $\alpha_{\min}$ and $\alpha_{\max}$, computed on the training set. The loss achieved by our solution as we vary λ is shown on the training and testing sets, α_{train} and α_{test}, respectively.

Figure 3.8 shows the MSE of the adversary as we vary the relative trade-off parameter λ between the target and adversary objectives. The plot illustrates (1) the lower and upper bounds $\alpha_{\min}$ and $\alpha_{\max}$ respectively calculated on the training dataset, (2) achievable adversary MSE computed on the training set α_{train}, and finally, (3) achievable adversary MSE computed on the test set α_{test}. Observe that on the training dataset, all values of $\alpha \in [\alpha_{\min}, \alpha_{\max}]$ are reachable as we sweep through $\lambda \in [0, 1]$. This is, however, not the case on the test set as the bounds are computed through empirical moments as opposed to the population covariance matrices.

3.6.2 Fair Classification

We consider the task of learning representations that are invariant to a sensitive attribute on two datasets, Adult and German, from the UCI ML-repository [93]. For comparison, apart from the raw features $\boldsymbol{X}$, we consider several baselines that use NNs and are trained through SGDA; LFR [36], VAE [94], VFAE [37], ML-ARL [24] and MaxEnt-ARL [1].

The Adult dataset contains 14 attributes. There are $30,163$ and $15,060$ instances in the training and test sets, respectively. The target task is the binary classification of annual income, i.e., more or less than 50K, and the sensitive attribute is gender. Similarly, the German dataset contains 1000

Table 3.1: Fair Classification Performance (in %)

	Adult Dataset			German Dataset		
Method	Target (income)	Sensitive (gender)	Δ^*	Target (credit)	Sensitive (age)	Δ^*
Raw Data	85.0	85.0	17.6	80.0	87.0	6.0
LFR [36]	82.3	67.0	0.4	72.3	80.5	0.5
VAE [94]	81.9	66.0	1.4	72.5	79.5	1.5
VFAE [37]	81.3	67.0	0.4	72.7	79.7	1.3
ML-ARL [24]	84.4	67.7	0.3	74.4	80.2	0.8
MaxEnt-ARL [1]	84.6	65.5	1.9	72.5	80.0	1.0
Linear-SARL	84.1	67.4	0.0	76.3	80.9	0.1
Kernel-SARL	84.1	67.4	0.0	76.3	80.9	0.1

* Absolute difference between adversary accuracy and random guess

instances of individuals with 20 different attributes. The target is to classify the creditworthiness of individuals as good or bad, with the sensitive attribute being age.

We learn encoders on the training set, after which, following the baselines, we freeze the encoder and train the target (logistic regression) and adversary (2-layer network with 64 units) classifiers on the training set. Table 3.1 shows the performance of the target and adversary on both datasets. Both Linear-SARL and Kernel-SARL outperform all NN-based baselines. For either of these tasks, the Kernel-SARL does not afford any additional benefit over Linear-SARL. For the adult dataset, the linear encoder maps the 14 input features to just one dimension. The weights assigned to each feature are shown in Figure 3.9. Notice that the encoder assigns almost zero weight to the gender feature in order to be fair with respect to the gender attribute.

3.6.3 Illumination Invariant Face Classification

This task pertains to face classification under different illumination conditions on the Extended Yale B dataset [95]. It comprises of face images of 38 people under five different light source directions, namely, upper right, lower right, lower left, upper left, and front. The target task is to

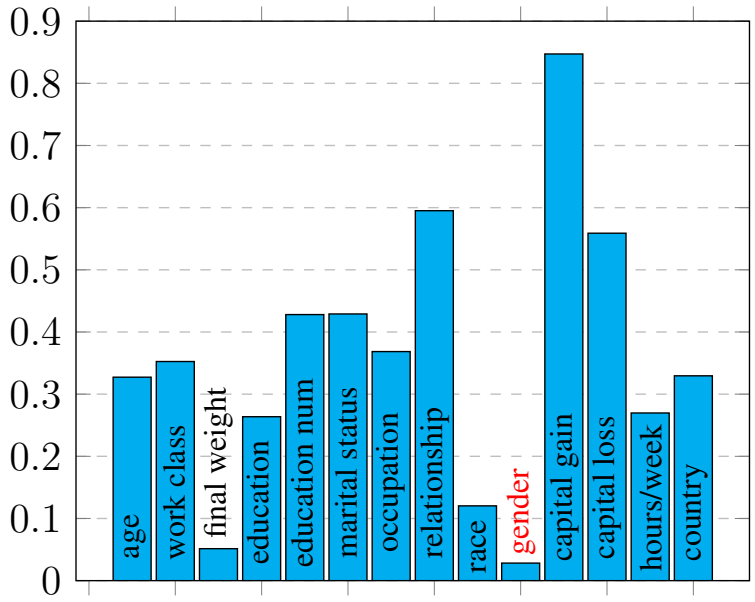


Figure 3.9: **Adult Dataset:** Magnitude of learned encoder weights Θ for each semantic input feature.

establish the identity of the person in the image, with the direction of the light being the sensitive attribute. Since the direction of lighting is independent of identity, the ideal ARL solution should obtain a representation Z that is devoid of any sensitive information. We first followed the experimental setup of Xie *et al.* [24] in terms of the train/test split strategy, i.e., 190 samples (5 from each class) for training and 1096 images for testing. Our global solution was able to completely remove illumination from the embedding resulting in the adversary accuracy being 20%, i.e., random chance. To investigate further, we consider different variations of this problem, flipping target and sensitive attributes and exchanging training and test sets. The complete set of results, including NN-based baselines, are reported in Table 3.2 ([EX] corresponds to exchanging training and testing sets). In all these cases, our solution was able to completely remove the sensitive features resulting in adversary performance that is no better than random chance. Simultaneously, the embedding is also competitive with the baselines on the target task.

Table 3.2: Extended Yale B Performance (in %)

Method	Adversary (illumination)	Target (identity)	Adversary (identity)	Target (illumination)
Raw Data	96	78	-	-
VFAE [37]	57	85	-	-
ML-ARL [24]	57	89	-	-
MaxEnt-ARL [1]	40	89	-	-
Linear-SARL	21	81	3	94
Linear-SARL [EX]	20	86	3	97
Kernel-SARL	20	86	3	96
Kernel-SARL [EX]	20	88	3	96

3.6.4 CIFAR-100

The CIFAR-100 dataset [96] consists of $50,000$ images from 100 classes that are further grouped into 20 superclasses. Each image is therefore associated with two attributes, a "fine" class label and a "coarse" superclass label. We consider a setup where the "coarse" and "fine" labels are the target and sensitive attributes, respectively. For Linear-SARL and Kernel-SARL (degree five polynomial kernel) and SGDA, we use features (64-dimensional) extracted from a pre-trained ResNet-110 model as an input to the encoder instead of raw images. From these features, the encoder is tasked with aiding the target predictor and hindering the adversary. This setup serves as an example to illustrate how invariance can be "imparted" to an existing biased pre-trained representation. We also consider two NN-baselines, ML-ARL [24] and MaxEnt-ARL [1]. Unlike our scenario, where the pre-trained layers of ResNet-18 are not adapted, the baselines optimize the entire encoder for the ARL task. For evaluation, once the encoder is learned and frozen, we train a discriminator and adversary as 2-layer networks with 64 neurons each. Therefore, although our approach uses linear regressor as an adversary at training, we evaluate against stronger adversaries at test time. In contrast, the baselines train and evaluate against adversaries with equal capacity.

Figure 3.10 shows the trade-off in accuracy between the target predictor and adversary. We

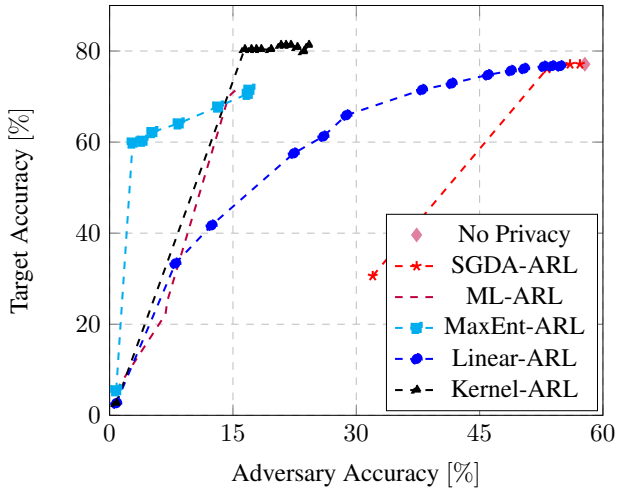
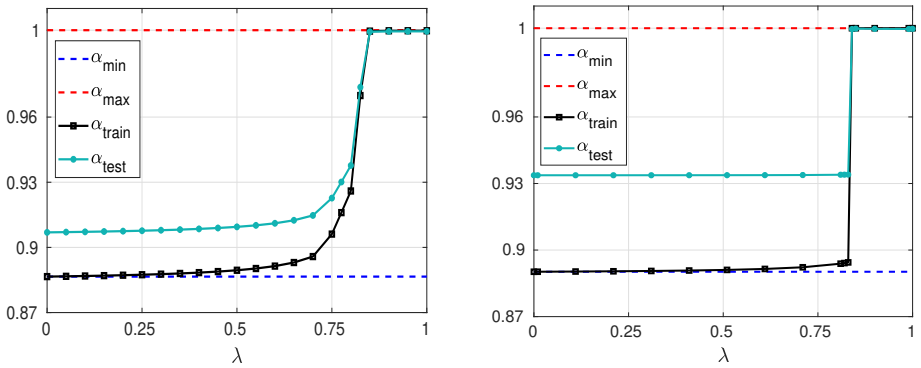


Figure 3.10: **CIFAR-100:** Trade-off between target performance and leakage of sensitive attribute by adversary.

observe that (1) Kernel-ARL significantly outperforms Linear-SARL. Since the former implicitly maps the data into a higher dimensional space, the sensitive features are potentially disentangled sufficiently for the linear encoder in that space to discard such information. Therefore, even for large values of λ, Kernel-SARL is able to simultaneously achieve high target accuracy while keeping the adversary performance low. (2) Despite being handicapped by the fact that Kernel-SARL is evaluated against stronger adversaries than it is trained against, its performance is comparable to that of the NN baselines. In fact, it outperforms both ML-ARL and MaxEnt-ARL with respect to the target task. (3) Despite repeated attempts with different hyper-parameters and choice of optimizers, SGDA was highly unstable across most datasets and often got stuck in a local optimum and failed to find good solutions.

Figure 3.11 shows the MSE of the adversary as we vary the relative trade-off λ between the target and adversary objectives. The plot illustrates (1) the lower and upper bounds $\alpha_{\min}$ and $\alpha_{\max}$ respectively calculated on the training dataset, (2) achievable adversary MSE computed on the training set α_{train}, and finally, (3) achievable adversary MSE computed on the test set α_{test}. Observe that on the training dataset, all values of $\alpha \in [\alpha_{\min}, \alpha_{\max}]$ are reachable as we sweep



(a) Linear-SARL (b) Kernel-SARL

Figure 3.11: **CIFAR-100:** Lower and upper bounds on adversary loss, $\alpha_{\min}$ and $\alpha_{\max}$, computed on the training set. The loss achieved by our solution as we vary λ is shown on the training and testing sets, α_{train} and α_{test}, respectively.

through $\lambda \in [0, 1]$. This is, however, not the case on the test set, as the bounds are computed through empirical moments as opposed to the true covariance matrices.

Figure 3.12 plots the optimal embedding dimensionality provided by SARL as a function of the trade-off parameter λ. At small values of λ, the objective favors the target task, i.e., 20 class predictions. Thus, SARL does indeed determine the optimal dimensionality of 19 for a 20-class linear target regressor. However, at large values of λ, the objective only seeks to hinder the sensitive task, i.e., 100 class prediction. In this case, the ideal embedding dimensionality from the perspective of the linear adversary regressor is at least 99. The SARL ascertained dimensionality of one is, thus, optimal for maximally mitigating the leakage of the sensitive attribute from the embedding. However, unsurprisingly, the target task also suffers significantly.

3.7 Summary

We studied the "linear" form of adversarial representation learning (ARL), where all the entities are linear functions. We showed that the optimization problem, even for this simplified version, is both non-convex and non-differentiable. Using tools from spectral learning, we obtained a

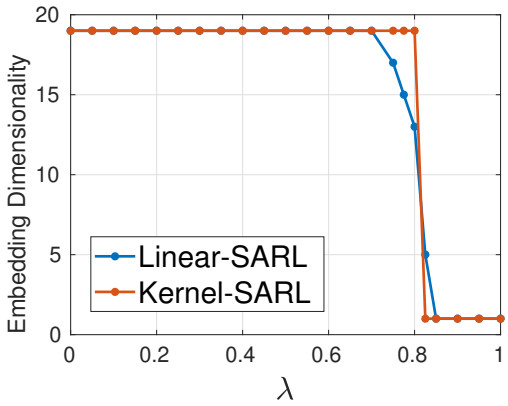


Figure 3.12: **CIFAR-100:** Optimal embedding dimensionality learned by SARL. At small values of λ, the objective favors the target task, which predicts 20 classes. Thus, an embedding dimensionality of 19 is optimal for a linear target regressor. At large values of λ, the objective only seeks to hinder the adversary. Thus, SARL determines the optimal dimensionality of the embedding as one.

closed-form expression for the global optima and derived analytical bounds on the achievable utility and invariance. We also extended these results to non-linear parameterizations through kernelization. Numerical experiments on multiple datasets indicated that the global optima solution of the "kernel" form of ARL is able to obtain a trade-off between utility and invariance that is comparable to that of local optima solutions of NN-based ARL. At the same time, unlike NN-based solutions, the proposed method can (1) analytically determine the achievable utility and invariance bounds and (2) provide explicit control over the trade-off between utility and invariance.

Admittedly, the results presented in this chapter do not extend directly to NN-based formulations of ARL. However, we believe it sheds light on the nature of the ARL optimization problem and aids our understanding of the ARL problem. It helps delineate the role of the optimization algorithm and the choice of embedding function, highlighting the trade-off between the expressivity of the functions and our ability to obtain the global optima of the adversarial game. We consider our contribution as the first step towards controlling the non-convexity that naturally appears in game-theoretic representation learning.

Chapter 4

Adversarial Representation Learning With Closed-Form Solvers

4.1 Introduction

In this chapter, we revisit the ARL problem and look at it from the NN-based optimization point of view. The vanilla algorithm for learning the parameters of the encoder, target, and adversary networks is SGDA [24, 1], where the players take a gradient step simultaneously. See Figure 4.1 for an illustration. However, applying SGDA is not an optimal strategy for ARL and is known to suffer from some drawbacks. Firstly, SGDA has undesirable convergence properties; it fails to converge to a local minimum and can converge to fixed points that are not local minimax while being very unstable and slow in practice [67, 68]. Secondly, SGDA exhibits strong rotation around fixed points, which requires using very small learning rates [64, 66] to converge. Numerous solutions [64, 65, 69] have been proposed recently to address the aforementioned computational challenges. These approaches, however, seek to obtain solutions to the minimax optimization problem in the general case, where each player is modeled as a complex neural network.

We take a different perspective and propose an alternative optimization algorithm for ARL. Our key insight is to replace the shallow NNs with other analytically tractable models with similar capacities. We propose to adopt simple learning algorithms that admit closed-form solutions,

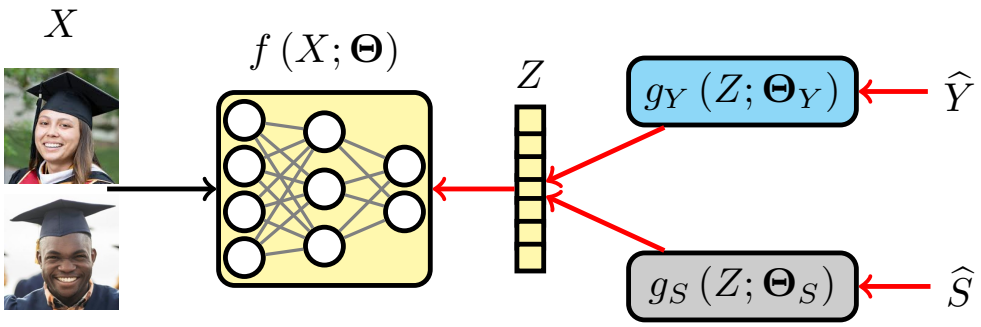


Figure 4.1: **ARL-SGDA:** Illustration of training adversarial representation learning through stochastic gradient descent ascent. i) At first, the target predictor parameters $\boldsymbol{\Theta}_Y$ are updated while the encoder and adversary parameters are frozen. ii) Then, the adversary parameters $\boldsymbol{\Theta}_S$ are updated while the encoder and target parameters are frozen. iii) Finally, the encoder parameters $\boldsymbol{\Theta}$ get updated while target and adversary parameters are frozen. SGDA does not provide any convergence guarantees.

such as linear or kernel ridge regressors for the target and adversary, while modeling the encoder as a DNN. Crucially, such models are particularly suitable for ARL and afford numerous advantages, including (1) closed-form solution allows learning problems to be optimized globally and efficiently, (2) analytically obtaining upper bound on optimal dimensionality of the embedding, (3) the simplicity and differentiability allows us to backpropagate through the closed-form solution, (4) practically it resolves the notorious rotational behavior of iterative minimax gradient dynamics, resulting in a simple optimization that is empirically stable, reliable, converges faster to a local optimum, and ultimately results in a more effective encoder.

We demonstrate the practical effectiveness of our approach, dubbed OptNet-ARL, through numerical experiments on an illustrative toy example, fair classification on UCI Adult and German datasets, and mitigating information leakage on the CelebA dataset. We consider two scenarios where the target and sensitive attributes are (a) dependent and (b) independent. Our results indicate that, in comparison to existing ARL solutions, OptNet-ARL is more stable and converges faster while also outperforming them in terms of accuracy, especially in the latter scenario.

A number of recent approaches have integrated differentiable solvers, both iterative as well as

closed-form, within end-to-end learning systems. Structured layers for segmentation and higher-order pooling were introduced by [97]. Similarly, [98] proposed an asymmetric architecture that incorporates a correlation filter as a differentiable layer. Differential optimization as a layer in NNs was introduced by [99, 100]. More recently, differentiable solvers have also been adopted for meta-learning [101, 102] as well. The primary motivation for all the aforementioned approaches is to endow DNNs with differential optimization and ultimately achieve faster convergence of the end-to-end systems. In contrast, our inspiration for using differential closed-form solvers is to control the non-convexity of the optimization in ARL in terms of stability, reliability, and effectiveness.

4.2 Problem Setting

Recall the ARL optimization problem in (3.1) and denote the global minimums of the adversary and target estimators as

$$\begin{aligned} J_Y(\boldsymbol{\Theta}) &:= \min_{\boldsymbol{\Theta}_Y} \mathbb{E}_{X,Y} \left[L_Y \left(g_Y(f\left(X; \boldsymbol{\Theta}\right); \boldsymbol{\Theta}_Y), Y \right) \right] \\ J_S(\boldsymbol{\Theta}) &:= \min_{\boldsymbol{\Theta}_S} \mathbb{E}_{X,S} \left[L_S \left(g_S(f\left(X; \boldsymbol{\Theta}\right); \boldsymbol{\Theta}_S), S \right) \right]. \end{aligned} \tag{4.1}$$

Similar to Chapter 3, instead of solving the constrained optimization problem in (3.1), we solve its scalarization version:

$$\min_{\boldsymbol{\Theta}} \left\{ (1-\lambda)\, J_Y(\boldsymbol{\Theta}) - \lambda\, J_S(\boldsymbol{\Theta}) \right\},\ 0 \leq \lambda \leq 1, \tag{4.2}$$

where λ is the trade-off parameter between utility and the leakage of the sensitive information.

4.2.1 Motivating Exact Solvers

Most state-of-the-art ARL algorithms cannot solve the optimization problems in (4.1) optimally (e.g., SGDA). For any given $\boldsymbol{\Theta}$, denote any non-optimal adversary and target predictors' loss functions by $J_Y^{\text{approx}}(\boldsymbol{\Theta})$ and $J_S^{\text{approx}}(\boldsymbol{\Theta})$, respectively. It is obvious that for any given $\boldsymbol{\Theta}$, it holds that

$$J_Y^{\text{approx}}(\boldsymbol{\Theta}) \geq J_Y(\boldsymbol{\Theta}) \quad \text{and} \quad J_S^{\text{approx}}(\boldsymbol{\Theta}) \geq J_S(\boldsymbol{\Theta}).$$

Note that the optimization problem raised from a non-optimal adversary and target predictors is

$$\min_{\boldsymbol{\Theta}} \left\{ (1-\lambda)\, J_Y^{\text{approx}}(\boldsymbol{\Theta}) - \lambda\, J_S^{\text{approx}}(\boldsymbol{\Theta}) \right\} \,, 0 \leq \lambda \leq 1. \tag{4.3}$$

Intuitively, solution(s) of (4.3) do not outperform that of (4.2). We now formulate this intuition more concretely.

Definition 4.1. Let (a_1, a_2) and (b_1, b_2) be two arbitrary points in $\mathbb{R}^2$. We say (b_1, b_2) dominates (a_1, a_2) iff $b_1 > a_1$ and $b_2 < a_2$ hold simultaneously.

The following lemma states that any solutions obtained by a sub-optimal adversary and target predictors cannot dominate that of exact solvers.

Lemma 4.2. For any $\lambda_1,\ \lambda_2 \in [0, 1)$, consider the following optimization problems

$$\boldsymbol{\Theta}^{\text{exact}} = \arg\min_{\boldsymbol{\Theta}} \left\{ (1-\lambda_1)\, J_Y(\boldsymbol{\Theta}) - \lambda_1\, J_S(\boldsymbol{\Theta}) \right\} \tag{4.4}$$

and

$$\boldsymbol{\Theta}^{\text{approx}} = \arg\min_{\boldsymbol{\Theta}} \left\{ (1-\lambda_2)\, J_Y^{\text{approx}}(\boldsymbol{\Theta}) - \lambda_2\, J_S^{\text{approx}}(\boldsymbol{\Theta}) \right\}.$$

Then, any adversary-target objective trade-off generated by $\left(J_S(\boldsymbol{\Theta}^{\text{exact}}), J_Y(\boldsymbol{\Theta}^{\text{exact}})\right)$ cannot be dominated by the trade-off generated by $(J_S(\boldsymbol{\Theta}^{\text{approx}}), J_Y(\boldsymbol{\Theta}^{\text{approx}}))$.

Proof. It is enough to show that

if (i) $J_S(\boldsymbol{\Theta}^{\text{approx}}) > J_S(\boldsymbol{\Theta}^{\text{exact}})$ then $J_Y(\boldsymbol{\Theta}^{\text{approx}}) \geq J_Y(\boldsymbol{\Theta}^{\text{exact}})$,

and if (ii) $J_Y(\boldsymbol{\Theta}^{\text{approx}}) < J_Y(\boldsymbol{\Theta}^{\text{exact}})$ then $J_S(\boldsymbol{\Theta}^{\text{approx}}) \leq J_S(\boldsymbol{\Theta}^{\text{exact}})$.

The key point is to observe from (4.4) that regardless of λ_2, J_y^{approx} and J_S^{approx}, we have

$$(1-\lambda_1)\, J_Y(\boldsymbol{\Theta}^{\text{exact}}) - \lambda_1\, J_S(\boldsymbol{\Theta}^{\text{exact}}) \;\; \leq \;\; (1-\lambda_1)\, J_Y(\boldsymbol{\Theta}^{\text{approx}}) - \lambda_1\, J_S(\boldsymbol{\Theta}^{\text{approx}}).$$

Now, consider three possible cases for λ_1:

a) $\lambda_1 = 0$: In this case we have $J_Y(\boldsymbol{\Theta}^{\text{exact}}) \leq J_Y(\boldsymbol{\Theta}^{\text{approx}})$ and therefore regardless of $J_s(\boldsymbol{\Theta}^{\text{exact}})$ and $J_S(\boldsymbol{\Theta}^{\text{approx}})$, (ii) cannot happen and (i) holds under its assumption.

b) $\lambda_1 = 1$: In this case we have $J_S(\boldsymbol{\Theta}^{\text{exact}}) \geq J_Y(\boldsymbol{\Theta}^{\text{approx}})$ and therefore regardless of $J_Y(\boldsymbol{\Theta}^{\text{exact}})$ and $J_Y(\boldsymbol{\Theta}^{\text{approx}})$, (i) cannot happen and (ii) holds under its assumption.

c) $0 < \lambda_1 < 1$:

(i) If $J_S(\boldsymbol{\Theta}^{\text{approx}}) > J_S(\boldsymbol{\Theta}^{\text{exact}})$, then

$$0 < \lambda_1 \left(J_s(\boldsymbol{\Theta}^{\text{approx}}) - J_S(\boldsymbol{\Theta}^{\text{exact}})\right) \leq (1-\lambda_1) \left(J_Y(\boldsymbol{\Theta}^{\text{approx}}) - J_Y(\boldsymbol{\Theta}^{\text{exact}})\right).$$

This implies that $J_Y(\boldsymbol{\Theta}^{\text{approx}}) \geq J_Y(\boldsymbol{\Theta}^{\text{exact}})$.

(ii) If $J_Y(\boldsymbol{\Theta}^{\text{approx}}) < J_Y(\boldsymbol{\Theta}^{\text{exact}})$, then

$$0 < (1-\lambda_1) \left(J_Y(\boldsymbol{\Theta}^{\text{exact}}) - J_Y(\boldsymbol{\Theta}^{\text{approx}})\right) \leq \lambda_1 \left(J_S(\boldsymbol{\Theta}^{\text{exact}}) - J_S(\boldsymbol{\Theta}^{\text{approx}})\right).$$

This implies that $J_S(\boldsymbol{\Theta}^{\text{approx}}) < J_s(\boldsymbol{\Theta}^{\text{exact}})$. □

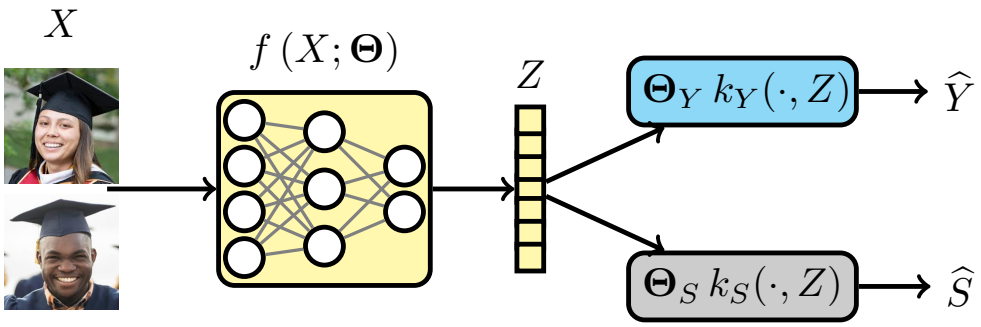


Figure 4.2: ARL with kernelized ridge regressors for adversary and target predictors. This setting turns typical SGDA optimization of ARLs into a simple SGD optimization.

4.3 Exact Adversary and Target Predictor Solvers

Existing instances of ARL adopt NNs to represent f, g_Y, and g_S and learn their respective parameters $\{\boldsymbol{\Theta}, \boldsymbol{\Theta}_Y, \boldsymbol{\Theta}_S\}$ through SGDA. Consequently, the target and adversary in equation (4.1) are not solved to optimality, thereby resulting in a sub-optimal encoder.

4.3.1 Closed-Form Adversary and Target Predictor

The machine learning literature offers a wealth of methods with exact solutions appropriate for modeling adversary and target predictors. In this section, we argue for and adopt simple, fast, and differentiable methods such as kernel ridge regressors as shown in Figure 4.2. Such modeling allows us to obtain the optimal estimators globally for any given encoder $f(\cdot; \boldsymbol{\Theta})$.

On the other hand, kernelized ridge regressors can be stronger than the shallow NNs that are used in many ARL-based solutions(e.g., [24, 103, 2, 1]). Although it is not the focus of this dissertation, it is worth noting that even DNNs in the infinite-width limit reduce to linear models with a kernel called the neural tangent kernel [104], and as such can be adopted to increase the capacity of our regressors.

Consider two RKHSs of functions, $\mathcal{H}_S$ and $\mathcal{H}_Y$, for the adversary and target networks, respec-

tively. Let a possible corresponding pair of feature maps be $\phi_S(\cdot) \in \mathbb{R}^{r_S}$ and $\phi_Y(\cdot) \in \mathbb{R}^{r_Y}$ where r_S and r_Y are the dimensionality of the resulting features and can potentially approach infinity. The respective kernel functions for $\mathcal{H}_S$ and $\mathcal{H}_Y$ can be represented as $k_S(\boldsymbol{z}, \boldsymbol{z}') = \langle \phi_S(\boldsymbol{z}), \phi_S(\boldsymbol{z}') \rangle_{\mathcal{H}_S}$ and $k_Y(\boldsymbol{z}, \boldsymbol{z}') = \langle \phi_Y(\boldsymbol{z}), \phi_Y(\boldsymbol{z}') \rangle_{\mathcal{H}_Y}$. Under this setting, we can relate the target and sensitive attributes to any given embedding Z as

$$\begin{aligned} \widehat{Y} &= \boldsymbol{\Theta}_Y \ [k_Y(\boldsymbol{z}_1, Z), \cdots, k_Y(\boldsymbol{z}_n, Z)]^T \\ \widehat{S} &= \boldsymbol{\Theta}_S \ [k_S(\boldsymbol{z}_1, Z), \cdots, k_S(\boldsymbol{z}_n, Z)]^T , \end{aligned} \tag{4.5}$$

where $\boldsymbol{\Theta}_Y \in \mathbb{R}^{d_Y \times n}$ and $\boldsymbol{\Theta}_S \in \mathbb{R}^{d_S \times n}$, and n is the number of data samples. Let the entire embedding of input data be denoted as $\boldsymbol{Z} := [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n] \in \mathbb{R}^{r \times n}$, where $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$ for $i = 1, \cdots, n$. Consequently, it follows that

$$\begin{aligned} \widehat{\boldsymbol{Y}} &:= [\widehat{\boldsymbol{y}}_1, \cdots, \widehat{\boldsymbol{y}}_n] = \boldsymbol{\Theta}_Y \, \boldsymbol{K}_Y \\ \widehat{\boldsymbol{S}} &:= [\widehat{\boldsymbol{s}}_1, \cdots, \widehat{\boldsymbol{s}}_n] = \boldsymbol{\Theta}_S \, \boldsymbol{K}_S. \end{aligned} \tag{4.6}$$

In a typical ARL setting, once an encoder is learned (i.e., for a given fixed embedding $\boldsymbol{Z}$), we evaluate against the best possible adversary and target predictors. In the following lemma, we obtain the minimum MSE for the kernelized adversary and target predictors for any given embedding $\boldsymbol{Z}$.

Lemma 4.3. Let $J_Y(\boldsymbol{Z})$ and $J_S(\boldsymbol{Z})$ be regularized minimum MSEs for adversary and target:

$$\begin{aligned} J_Y(\boldsymbol{Z}) &= \min_{\boldsymbol{\Theta}_Y} \left\{ \mathbb{E} \left\{ \left\| \widehat{Y} - Y \right\|^2 \right\} + \gamma_Y \left\| \boldsymbol{\Theta}_Y \right\|_F^2 \right\}, \\ J_S(\boldsymbol{Z}) &= \min_{\boldsymbol{\Theta}_S} \left\{ \mathbb{E} \left\{ \left\| \widehat{S} - S \right\|^2 \right\} + \gamma_S \left\| \boldsymbol{\Theta}_S \right\|_F^2 \right\} \end{aligned}$$

where γ_Y and γ_S are regularization parameters for target and adversary regressors, respectively.

Then, for any given embedding matrix $\boldsymbol{Z}$, the minimum MSE for the kernelized adversary and target can be obtained as

$$\begin{aligned} J_Y(\boldsymbol{Z}) &= \frac{1}{n}\|\boldsymbol{Y}\|_F^2 - \frac{1}{n}\left\| P_{\mathcal{M}_Y} \begin{bmatrix} \boldsymbol{Y}^T \\ \boldsymbol{0}_{n\times d_Y} \end{bmatrix} \right\|_F^2 \\ J_S(\boldsymbol{Z}) &= \frac{1}{n}\|\boldsymbol{S}\|_F^2 - \frac{1}{n}\left\| P_{\mathcal{M}_S} \begin{bmatrix} \boldsymbol{S}^T \\ \boldsymbol{0}_{n\times d_S} \end{bmatrix} \right\|_F^2, \end{aligned} \tag{4.7}$$

where

$$\boldsymbol{M}_Y := \begin{bmatrix} \boldsymbol{K}_Y \\ \sqrt{n\gamma_Y}\boldsymbol{I}_n \end{bmatrix}, \qquad \boldsymbol{M}_S := \begin{bmatrix} \boldsymbol{K}_S \\ \sqrt{n\gamma_S}\boldsymbol{I}_n \end{bmatrix}$$

are both full-column rank matrices, and the orthogonal projection matrix for any full-column rank matrix $\boldsymbol{M}$ can be obtained as

$$P_{\mathcal{M}} = \boldsymbol{M}(\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{M}^T.$$

Proof. Using the empirical mean, we have

$$\begin{aligned} J_Y(\boldsymbol{Z}) &= \min_{\boldsymbol{\Theta}_Y} \left\{ \frac{1}{n}\sum_{k=}^{n} \|\boldsymbol{\Theta}_Y\boldsymbol{K}_Y - \boldsymbol{Y}\|_F^2 + \gamma_Y\|\boldsymbol{\Theta}_Y\|_F^2 \right\} \\ &= \frac{1}{n}\min_{\boldsymbol{\Theta}_Y} \left\| \underbrace{\begin{bmatrix} \boldsymbol{K}_Y \\ \sqrt{n\gamma_Y}\boldsymbol{I}_n \end{bmatrix}}_{\boldsymbol{M}_Y} \boldsymbol{\Theta}_Y^T - \begin{bmatrix} \boldsymbol{Y}^T \\ \boldsymbol{0}_{n\times d_Y} \end{bmatrix} \right\|_F^2 \\ &= \frac{1}{n}\min_{\boldsymbol{\Theta}_Y} \left\| \boldsymbol{M}_Y\boldsymbol{\Theta}_Y^T - P_{\mathcal{M}_Y} \begin{bmatrix} \boldsymbol{Y}^T \\ \boldsymbol{0}_{n\times d_Y} \end{bmatrix} \right\|_F^2 + \frac{1}{n}\left\| P_{\mathcal{M}_Y^\perp} \begin{bmatrix} \boldsymbol{Y}^T \\ \boldsymbol{0}_{n\times d_Y} \end{bmatrix} \right\|_F^2 \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{n}\left\|\underbrace{\boldsymbol{M}_Y\boldsymbol{M}_Y^{\dagger}}_{P_{\mathcal{M}_Y}}\begin{bmatrix}\boldsymbol{Y}^T\\ \\ \boldsymbol{0}_{n\times d_Y}\end{bmatrix} - P_{\mathcal{M}_Y}\begin{bmatrix}\boldsymbol{Y}^T\\ \\ \boldsymbol{0}_{n\times d_Y}\end{bmatrix}\right\|_F^2 + \frac{1}{n}\left\|P_{\mathcal{M}_Y^{\perp}}\begin{bmatrix}\boldsymbol{Y}^T\\ \\ \boldsymbol{0}_{n\times d_Y}\end{bmatrix}\right\|_F^2\\
&= \frac{1}{n}\left\|P_{\mathcal{M}_Y^{\perp}}\begin{bmatrix}\boldsymbol{Y}^T\\ \\ \boldsymbol{0}_{n\times d_Y}\end{bmatrix}\right\|_F^2\\
&= \frac{1}{n}\|\boldsymbol{Y}\|_F^2 - \frac{1}{n}\left\|P_{\mathcal{M}_Y}\begin{bmatrix}\boldsymbol{Y}^T\\ \\ \boldsymbol{0}_{n\times d_Y}\end{bmatrix}\right\|_F^2, \qquad (4.8)
\end{aligned}
$$

where we used orthogonal decomposition w.r.t $\mathcal{M}_Y$ in the third and last steps and a possible minimizer used in the forth step is $\boldsymbol{\Theta}_Y^T = \boldsymbol{M}_Y^{\dagger}\begin{bmatrix}\boldsymbol{Y}^T\\ \\ \boldsymbol{0}_{n\times d_Y}\end{bmatrix}$. Using the same approach, we get

$$
J_S(\boldsymbol{Z}) = \frac{1}{n}\|\boldsymbol{S}\|_F^2 - \frac{1}{n}\left\|P_{\mathcal{M}_S}\begin{bmatrix}\boldsymbol{S}^T\\ \\ \boldsymbol{0}_{n\times d_S}\end{bmatrix}\right\|_F^2,
$$

where $\boldsymbol{M}_S := \begin{bmatrix}\boldsymbol{K}_S\\ \\ \sqrt{n\gamma_S}\boldsymbol{I}_n\end{bmatrix}$. □

It is quite straightforward to generalize this method to the case of multiple target and adversary predictors through equation (4.3). In this case, we will have multiple λs to trade-off between utility and the leakage of sensitive information.

4.3.2 Optimal Embedding Dimensionality

The ability to effectively optimize the parameters of the encoder is critically dependent on the dimensionality of the embedding as well. Higher dimensional embeddings can inherently absorb

unnecessary extraneous information in the data. Existing ARL applications, where the target and adversary are non-linear NNs, select the dimensionality of the embedding on an ad-hoc basis.

Adopting closed-form solvers for the target and adversary enables us to analytically determine an upper bound on the optimal dimensionality of the embedding for OptNet-ARL. To obtain the upper bound, we rely on the observation that a non-linear target predictor and adversary, by virtue of greater capacity, can learn non-linear decision boundaries. As such, in the context of ARL, the optimal dimensionality required by non-linear models is lower than the optimal dimensionality of linear target predictor and adversary. Therefore, we analytically determine the optimal dimensionality of the embedding in the following theorem.

Theorem 4.4. *Let Z in Figure 4.2 be disconnected from the encoder and be a free vector in $\mathbb{R}^r$. Further, assume that both adversary and target predictors are linear regressors, and $\gamma_S = \gamma_Y = 0$. Then, for any $0 \leq \lambda \leq 1$ the optimal dimensionality of the embedding vector, r is the number of negative eigenvalues of*

$$\boldsymbol{B} \;=\; \lambda \tilde{\boldsymbol{S}}^T \tilde{\boldsymbol{S}} - (1-\lambda)\tilde{\boldsymbol{Y}}^T \tilde{\boldsymbol{Y}}. \tag{4.9}$$

Proof. Recall that for linear regressor adversary and target predictors, we have

$$\widehat{\boldsymbol{Y}} = \boldsymbol{\Theta}_Y \boldsymbol{Z} + \boldsymbol{b}_Y, \qquad \widehat{\boldsymbol{S}} = \boldsymbol{\Theta}_S \boldsymbol{Z} + \boldsymbol{b}_S. \tag{4.10}$$

Following the proof in Lemma 4.3, we have

$$J_Y(\boldsymbol{Z}) = \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}\right\|_F^2 - \frac{1}{n}\left\|P_{\mathcal{M}}\tilde{\boldsymbol{Y}}\right\|_F^2, \qquad J_S(\boldsymbol{Z}) = \frac{1}{n}\left\|\tilde{\boldsymbol{S}}\right\|_F^2 - \frac{1}{n}\left\|P_{\mathcal{M}}\tilde{\boldsymbol{S}}\right\|_F^2$$

where $\mathcal{M}$ is the column space of $\tilde{\boldsymbol{Z}}^T\tilde{\boldsymbol{Z}}$ or equivalently the column space of $\tilde{\boldsymbol{Z}}$. Consequently, it

follows that

$$(1-\lambda)\, J_Y(\boldsymbol{Z}) - \lambda\, J_S(\boldsymbol{Z}) = \\ \frac{1}{n}\left((1-\lambda)\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \lambda\left\|\tilde{\boldsymbol{S}}^T\right\|_F^2 - (1-\lambda)\left\|P_{\mathcal{M}}\tilde{\boldsymbol{Y}}^T\right\|_F^2 + \lambda\left\|P_{\mathcal{M}}\tilde{\boldsymbol{S}}^T\right\|_F^2\right). \tag{4.11}$$

Now, consider $\left\|P_{\mathcal{M}}\tilde{\boldsymbol{Y}}^T\right\|_F^2$:

$$\begin{aligned}\left\|P_{\mathcal{M}}\tilde{\boldsymbol{Y}}^T\right\|_F^2 &= \text{Tr}\left\{\tilde{\boldsymbol{Y}}\underbrace{P_{\mathcal{M}}^T P_{\mathcal{M}}}_{P_{\mathcal{M}}}\tilde{\boldsymbol{Y}}^T\right\} \\ &= \text{Tr}\left\{P_{\mathcal{M}}\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}\right\}\end{aligned}$$

Similarly,

$$\left\|P_{\mathcal{M}}\tilde{\boldsymbol{S}}^T\right\|_F^2 = \text{Tr}\left\{P_{\mathcal{M}}\tilde{\boldsymbol{S}}^T\tilde{\boldsymbol{S}}\right\}.$$

The terms $\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2$ and $\left\|\tilde{\boldsymbol{S}}^T\right\|_F^2$ on the right side of (4.11) are constants with respect to $\boldsymbol{Z}$, and hence can be ignored. We now get

$$\lambda\left\|P_{\mathcal{M}}\tilde{\boldsymbol{S}}^T\right\|_F^2 - (1-\lambda)\left\|P_{\mathcal{M}}\tilde{\boldsymbol{Y}}^T\right\|_F^2 = \text{Tr}\left\{P_{\mathcal{M}}\boldsymbol{B}\right\},$$

where

$$\boldsymbol{B} = \lambda\,\tilde{\boldsymbol{S}}^T\tilde{\boldsymbol{S}} - (1-\lambda)\,\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}.$$

Noting that any projection matrix $P_{\mathbb{M}} \in \mathbb{R}^{n\times n}$ of rank $i \leq n$ can be decomposed as $\boldsymbol{V}\boldsymbol{V}^T$ for some orthogonal matrix $\boldsymbol{V} \in \mathbb{R}^{n\times i}$, we get

$$\begin{aligned}
&\min_{r\in\{1,\cdots,n\}} \min_{\boldsymbol{Z}\in\mathbb{R}^{r\times n}} \left\{\lambda \left\|P_{\mathcal{M}}\tilde{\boldsymbol{S}}^T\right\|_F^2 - (1-\lambda)\left\|P_{\mathcal{M}}\tilde{\boldsymbol{Y}}^T\right\|_F^2\right\} \\
&\qquad = \min_{r\in\{1,\cdots,n\}} \min_{\dim\mathcal{M}\leq r} \text{Tr}\left\{P_{\mathcal{M}}\boldsymbol{B}\right\} \\
&\qquad = \min_{r\in\{1,\cdots,n\}} \min_{i\in\{1,\cdots,r\}} \min_{\boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{I}_i} \text{Tr}\left\{\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{B}\right\} \\
&\qquad = \min_{i=r\in\{1,\cdots,n\}} \min_{\boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{I}_i} \text{Tr}\left\{\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{B}\right\}.
\end{aligned}$$

From trace optimization problems and their solution in [90], we have

$$\begin{aligned}
\min_{r\in\{1,\cdots,n\}} \min_{\boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{I}_r} \text{Tr}\left\{\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{B}\right\} &= \min_{r\in\{1,\cdots,n\}} \{\beta_1+\beta_2+\cdots+\cdots\beta_r\} \\
&= \beta_1+\beta_2+\cdots+\beta_j
\end{aligned}$$

where $\beta_1,\cdots,\beta_r$ are the r smallest eigenvalues of $\boldsymbol{B}$, j denotes the number of negative eigenvalues of $\boldsymbol{B}$ and a possible minimizer is matrix $\boldsymbol{Z}$ in which the columns space of $\tilde{\boldsymbol{Z}}^T$ (i.e., $\mathcal{M}$) is the span of eigenvectors corresponding to all negative eigenvalues of $\boldsymbol{B}$. □

Given a dataset with the target and sensitive labels, $\boldsymbol{Y}$ and $\boldsymbol{S}$, respectively, the matrix $\boldsymbol{B}$ in (4.9) and its eigenvalues can be computed offline to determine the upper bound on the optimal dimensionality. By virtue of the greater capacity, the optimal dimensionality required by non-linear models is lower than the optimal dimensionality of linear predictors and therefore, Theorem 2 is a tight upper bound for the optimal embedding dimensionality. On large datasets where $\boldsymbol{B} \in \mathbb{R}^{n\times n}$ can be a very large matrix, the Nyström method with data sampling [92] can be adopted.

4.3.3 Gradient of Closed-Form Solution

In order to find the gradient of the encoder loss function in (4.3) with J_Y and J_S given in (4.7), we can ignore the constant terms, $\|\boldsymbol{Y}\|_F$ and $\|\boldsymbol{S}\|_F$. Then, the optimization problem in (4.3) would be equivalent to

$$\begin{aligned}\min_{\boldsymbol{\Theta}} &\left\{ (1-\lambda) \left\| P_{\mathcal{M}_S} \begin{bmatrix} \boldsymbol{S}^T \\ \mathbf{0}_{n\times d_S} \end{bmatrix} \right\|_F^2 - \lambda \left\| P_{\mathcal{M}_Y} \begin{bmatrix} \boldsymbol{Y}^T \\ \mathbf{0}_{n\times d_Y} \end{bmatrix} \right\|_F^2 \right\} \\ &= \min_{\boldsymbol{\Theta}} \left\{ (1-\lambda) \sum_{k=1}^{d_S} \left\| P_{\mathcal{M}_S} \boldsymbol{u}_S^k \right\|^2 - \lambda \sum_{m=1}^{d_Y} \left\| P_{\mathcal{M}_Y} \boldsymbol{u}_Y^m \right\|^2 \right\}, \end{aligned} \tag{4.12}$$

where the vectors $\boldsymbol{u}_S^k$ and $\boldsymbol{u}_Y^m$ are the k-th and m-th columns of $\begin{bmatrix} \boldsymbol{S}^T \\ \mathbf{0}_{n\times d_S} \end{bmatrix}$ and $\begin{bmatrix} \boldsymbol{Y}^T \\ \mathbf{0}_{n\times d_Y} \end{bmatrix}$, respectively. Let $\boldsymbol{M}$ be an arbitrary matrix function of $\boldsymbol{\Theta}$, and θ be an arbitrary scalar element of $\boldsymbol{\Theta}$. Then, from [105] we have

$$\frac{\partial \|P_{\mathcal{M}} \boldsymbol{u}\|^2}{\partial \theta} = 2\, \boldsymbol{u}^T P_{\mathcal{M}^\perp} \frac{\partial \boldsymbol{M}}{\partial \theta}\, \boldsymbol{M}^\dagger \boldsymbol{u}, \tag{4.13}$$

where

$$\left[\frac{\partial \boldsymbol{M}}{\partial \theta}\right]_{ij} = \begin{cases} \nabla_{\boldsymbol{z}_i}^T \left([\boldsymbol{M}]_{ij} \right) \nabla_\theta(\boldsymbol{z}_i) + \nabla_{\boldsymbol{z}_j}^T \left([\boldsymbol{M}]_{ij} \right) \nabla_\theta(\boldsymbol{z}_j), & i \le n \\ 0, & \text{else.} \end{cases}$$

Equation (4.13) can be directly used to obtain the gradient of objective function in (4.12).

Directly computing the gradient in equation (4.13) requires a pseudo-inverse of the matrix $\boldsymbol{M} \in \mathbb{R}^{2n\times n}$, which has a complexity of $\mathcal{O}(n^3)$. For large datasets, this computation can get prohibitively expensive. Therefore, we approximate the gradient using a single batch of data as we

optimize the encoder end-to-end. Similar approximations [92] are in fact, commonly employed to scale up kernel methods. Thus, the computational complexity of computing the loss for OptNet-ARL reduces to $\mathcal{O}(b^3)$, where b is the batch size. Since maximum batch sizes in training NNs are of the order of 10 to 1000, computing the gradient is practically feasible. We note that the procedure presented in this section is a simple SGD in which its stability can be guaranteed under Lipschitz and smoothness assumptions on encoder network [106].

4.4 Experiments

In this section, we will evaluate the efficacy of our proposed approach, OptNet-ARL, on three different tasks; Fair Classification on UCI [107] dataset, mitigating leakage of private information on the CelebA dataset, and ablation study on a Gaussian mixture example. We also compare OptNet-ARL with other ARL baselines in terms of stability of optimization, the achievable trade-off front between the target and adversary objectives, convergence speed, and the effect of embedding dimensionality. We consider three baselines, (1) **SGDA-ARL:** vanilla stochastic gradient descent ascent that is employed by multiple ARL approaches including [24, 103, 2, 1, 108] etc., (2) **ExtraSGDA-ARL:** a state-of-the-art of stochastic gradient descent ascent that uses an extra gradient step [72] for optimizing minimax games. Specifically, we use the ExtraAdam algorithm from [69], and (3) **SARL:** a global optimum solution for a kernelized regressor encoder and linear target and adversary [49]. Specifically, **hypervolume** (HV) [109], a metric for stability and goodness of trade-off (comparing algorithms under multiple objectives) is also utilized. A larger HV indicates a better Pareto front achieved, and the standard deviation of the HV represents stability.

In the training stage, the encoder, a DNN, is optimized **end-to-end** against kernel (RBF Gaussian kernel) ridge regressors in the case of OptNet-ARL and multi-layer perceptrons (MLP)s for

the baselines. Table 4.1 summarizes the network architecture of all experiments. We note that the optimal embedding dimensionality, r, for the binary target is equal to one, which is consistent with Fisher's linear discriminant analysis [110]. The embedding is instance normalized (unit norm). So we adopted a fixed value of $\sigma = 1$ for the Gaussian Kernel in all the experiments. We let the regression regularization parameter be 10^{-4} for all experiments. The learning rate is 3×10^{-4} with weight decay of 2×10^{-4}, and we use Adam as an optimizer for all experiments.

At the inference stage, the encoder is frozen, features are extracted, and a new target predictor and adversary are trained. At this stage, for both OptNet-ARL and the baselines, the target and adversary have the same model capacity. Furthermore, each experiment on each dataset is repeated five times with different random seeds (except for SARL, which has a closed-form solution for the encoder) and several different trade-off parameters $\lambda \in [0, 1)$. We report the median and standard deviation across the five repetitions.

4.4.1 Fair Classification

We consider fair classification on two different tasks. **UCI Adult Dataset:** It includes 14 features from $45,222$ instances. The task is to classify the annual income of each person as high (50K or above) or low (below 50K). The sensitive feature we wish to be fair with respect to is the gender of each person. **UCI German Dataset:** It contains 1000 instances of individuals with 20 different attributes. The target task is to predict their creditworthiness while being unbiased with respect to age. The correlation between the target and sensitive attributes is 0.03 and 0.02 for the Adult and German datasets, respectively. This indicates that the target attributes are almost orthogonal to the sensitive attributes. Therefore, the sensitive information can be totally removed with only a negligible sacrifice on the performance of the target task.

Stability and Performance: Since there is no trade-off between the two attributes, we compare

Table 4.1: Network Architectures in Experiments.

Method (ARL)	Encoder	Embd Dim	Target (Train)	Adversary (Train)	Target (Test)	Adversary (Test)
			Adult			
SGDA [24, 2]	MLP-4-2	1	MLP-4	MLP-4	MLP-4-2	MLP-4-2
ExtraSGDA [72]	MLP-4-2	1	MLP-4	MLP-4	MLP-4-2	MLP-4-2
SARL [49]	RBF krnl	1	linear	linear	MLP-4-2	MLP-4-2
OptNet-ARL (ours)	MLP-4-2	1	RBF krnl	RBF krnl	MLP-4-2	MLP-4-2
			German			
SGDA [24, 2]	MLP-4	1	MLP-2	MLP-2	logistic	logistic
ExtraSGDA [72]	MLP-4	1	MLP-2	MLP-2	logistic	logistic
SARL [49]	RBF krnl	1	linear	linear	logistic	logistic
OptNet-ARL (ours)	MLP-4	1	RBF krnl	RBF krnl	logistic	logistic
			CelebA			
SGDA [24, 2]	ResNet-18	128	MLP-64	MLP-64	MLP-32-16	MLP-32-16
ExtraSGDA [72]	ResNet-18	128	MLP-64-32	MLP-64-32	MLP-32-16	MLP-32-16
OptNet-ARL (ours)	ResNet-18	$[1, 128]$	RBF krnl	RBF krnl	MLP-32-16	MLP-32-16
			Gaussian Mixture			
SGDA [24, 2]	MLP-8-4	2	MLP-8-4	MLP-8-4	MLP-4-4	MLP-4-4
ExtraSGDA [72]	MLP-8-4	2	MLP-8-4	MLP-8-4	MLP-4-4	MLP-4-4
SARL [49]	RBF krnl	2	linear	linear	MLP-4-4	MLP-4-4
RBF-OptNet-ARL (ours)	MLP-8-4	2	RBF krnl	RBF krnl	MLP-4-4	MLP-4-4
IMQ-OptNet-ARL (ours)	MLP-8-4	$[1, \cdots, 512]$	IMQ krnl	IMQ krnl	MLP-4-4	MLP-4-4

stability by reporting the median and standard deviation of the target and adversary performance in Table 4.2. Our results indicate that OptNet-ARL achieves a higher accuracy for target tasks and lower leakage of the sensitive attribute with less variance. For instance, in the Adult dataset, our OptNet-ARL method achieves 83.81% target accuracy with almost zero sensitive leakage. For OptNet-ARL, the standard deviation of the sensitive attribute is exactly zero, which demonstrates its effectiveness and stability in comparison to the baselines. Similarly, for the German dataset, OptNet-ARL achieves 80.13% for sensitive accuracy, which is close to random chance (around 81%) while having the largest target accuracy compared to the baselines.

Table 4.2: Fair classification on UCI Adult and German datasets (in %)

	Adult Dataset			German Dataset		
Method	Target (income)	Sensitive (gender)	Diff 67.83	Target (credit)	Sensitive (age)	Diff 81
Raw Data	85.0	85.0	17.6	80.0	87.0	6.0
LFR [36]	82.3	67.0	0.4	72.3	80.5	0.5
AEVB [94]	81.9	66.0	1.4	72.5	79.5	1.5
VFAE [37]	81.3	67.0	0.4	72.7	79.7	1.3
SARL [49]	84.1	67.4	0.0	76.3	80.9	0.1
SGDA-ARL [24]	83.61 ± 0.38	67.08 ± 0.48	0.40	76.53 ± 1.07	87.13 ± 5.70	6.13
ExtraSGDA-ARL [69]	83.66 ± 0.26	66.98 ± 0.49	0.4	75.60 ± 1.68	86.80 ± 4.05	5.80
OptNet-ARL	83.81 ± 0.23	67.38 ± 0.00	0.00	76.67 ± 2.21	80.13 ± 1.48	0.87

4.4.2 Mitigating Sensitive Information Leakage

The CelebA dataset [111] contains $202,599$ face images of $10,177$ celebrities. Each image contains 40 different binary attributes (e.g., gender, emotion, age, etc.). Images are pre-processed and aligned to a fixed size of 112×96, and we use the official train-test splits. The target task is defined as predicting the presence or absence of high cheekbones (binary), with the sensitive attribute being smiling/not smiling (binary). The choice of this attribute pair is motivated by the presence of a trade-off between them. We observe that the correlation between this attribute pair is equal to 0.45, indicating that there is no encoder that can maintain target performance without leaking the sensitive attribute.

For this experiment, we note that SARL [49] cannot be employed since (1) it does not scale to large datasets ($\mathcal{O}(n^3)$) like CelebA, and (2) it cannot be applied directly on raw images and needs features extracted from a pre-trained network. Most other attribute pairs in this dataset either suffer from severe class imbalance or small correlation, indicating the lack of a trade-off. Network architecture details are shown in Table 4.1.

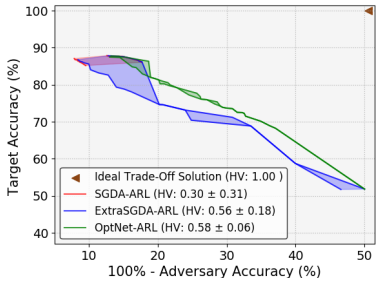
Stability and Trade-off: Figure 4.3 (a) shows the attainment surface [112] and hypervolume [109] (median and standard deviation) for all methods. SGDA-ARL spans only a small part

of the trade-off and, at the same time, exhibits large variance around the median curves. Overall, both baselines are unstable and unreliable when the two attributes are dependent on each other. On the other hand, OptNet-ARL solutions are very stable while also achieving a better trade-off between target and adversary accuracy.

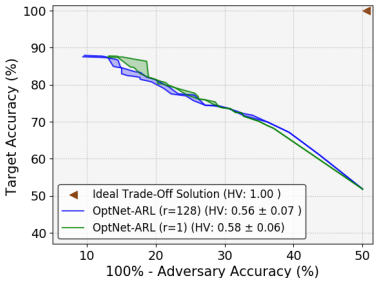
Optimal Embedding Dimensionality: Figure 4.3 (b) compares the utility-bias trade-off of the sub-optimal embedding dimensionality ($r = 128$) with that of the optimal dimensionality ($r = 1$). We can observe that optimal embedding dimensionality ($r = 1$) is producing a more stable trade-off between adversary and target accuracies.

Training Time: It takes five runs for SGDA-ARL and ExtraSGDA and two runs for OptNet-ARL to train a reliable encoder for overall 11 different values of $\lambda \in [0, 1)$. The summary of training time is given in Figure 4.3 (c). ExtraSGDA-ARL takes an extra step to update the weights, and therefore, it is slightly slower than SGDA-ARL. OptNet-ARL, on the other hand, is significantly faster in obtaining reliable results. Even for a single run, OptNet-ARL is faster than the baselines. This is because OptNet-ARL uses closed-form solvers for adversary and target and therefore does not need to train any additional networks downstream to the encoder.

Independent Features: We consider the target task to be the binary classification of smiling/not smiling, with the sensitive attribute being gender. In this case, the correlation between gender and target feature is 0.02, indicating that the two attributes are almost independent, and hence it should be feasible for an encoder to remove the sensitive information without affecting the target task. The results are presented in Figure 4.3 (d). In contrast to the scenario where the two attributes are dependent, we observe that all ARL methods can perfectly hide the sensitive information (gender) from representation without sacrificing on the target task performance. Therefore, OptNet-ARL is especially effective in a more practical setting where the target and sensitive attributes are correlated and hence can only attain a trade-off.



(a)



(b)

Method	One Run	Training Time
SGDA-ARL [39]	79.8	398.9
ExtraSGDA-ARL [14]	84.0	419.7
OptNet-ARL (ours)	76.8	153.6

(c)

Method (ARL)	Target (smiling) MSE	Sensitive (gender) Δ %
No Privacy	93.0	-
SGDA [39]	93.0	0.0
ExtraSGDA [14]	93.0	0.0
OptNet-ARL (ours)	93.0	0.0

(d)

Figure 4.3: **CelebA:** (a) Trade-off between adversary and target accuracy for dependent pair (smiling/not-smiling, high cheekbones). (b) Comparison between the trade-offs of optimal embedding dimensionality $r = 1$ and that of $r = 128$. (c) Overall and single run training time for different ARL methods. (d) Trade-off between adversary and target for independent pair (smiling/not-smiling, gender).

4.4.3 Ablation Study on Mixture of Four Gaussians

In this experiment, we consider a simple example where the data is generated by a mixture of four different Gaussian distributions. Let $\{f_i\}_{i=1}^4$ be all Gaussian distributions with means at $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$, respectively and covariance matrices all equal to $\boldsymbol{C} = 0.2^2\boldsymbol{I}_2$. Denote by $f(X)$ the distribution of the input data. Then, it follows that

$$f(X|\,\bullet) = f_1(X) + \frac{1}{2}f_2(X) + +\frac{1}{2}f_3(X), \qquad P\{\bullet\} = \frac{1}{2}$$
$$f(X|\,\bullet) = f_4(X) + \frac{1}{2}f_2(X) + +\frac{1}{2}f_3(X), \qquad P\{\bullet\} = \frac{1}{2}$$

The sensitive attribute is assumed to be the color (0 for red and 1 for blue), and the target task is reconstructing the input data. We sample 4000 points for training and 1000 points for the testing

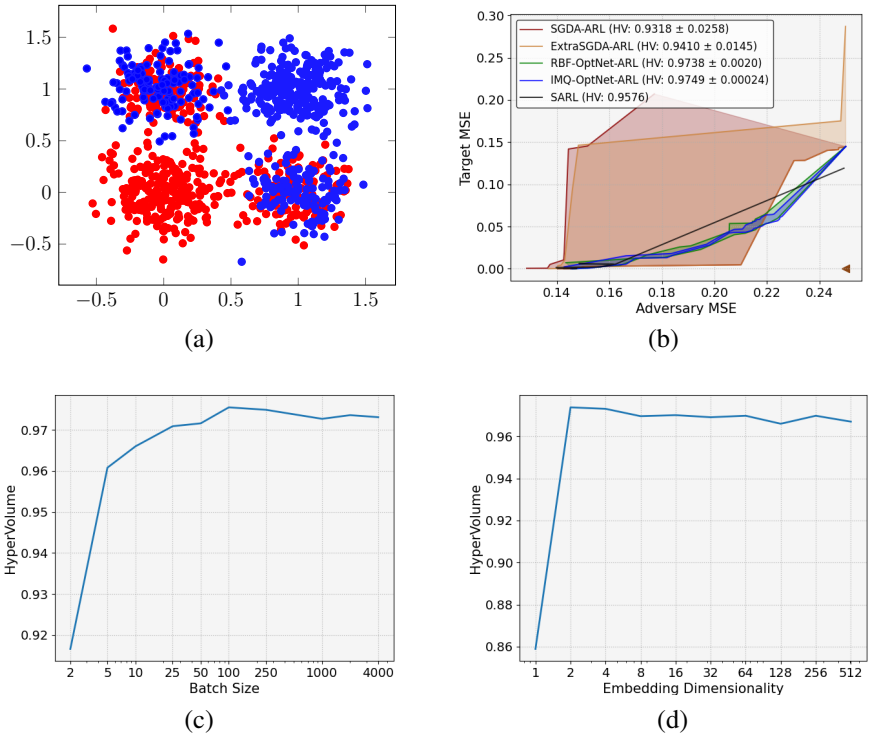
set independently. For visualization, the testing set is shown in Figure 4.4 (a). In this illustrative dataset, the correlation between input data and color is 0.61, and therefore there is no encoder that results in full target performance at no leakage of the sensitive attribute. Network architecture details are shown in Table 4.1.

Stability and Trade-off: Figure 4.4 (b) illustrates the five-run attainment surfaces and median hypervolumes for all methods. Since the dimensionality of both input and output is 2, the optimal embedding dimensionality is equal to 2, which we set in this experiment. We note that SARL achieves hypervolume better than SGDA and ExtraSGDA ARLs, which is not surprising due to the strong performance of SARL on small-size datasets. However, SARL is not applicable to large datasets. Among other baselines, ExtraSGDA-ARL appears to be slightly better. In contrast, the solutions obtained by RBF-OptNet-ARL (Gaussian kernel) outperform all baselines and are highly stable across different runs, which can be observed from both attainment surfaces and hypervolumes. In addition to Gaussian kernel, we also used inverse multi quadratic (IMQ) kernel [113][1] for OptNet-ARL to examine the effect kernel function. As we observe from Figure 4.4 (b), IMQ-OptNet-ARL performs almost similar to OptNet-ARR with Gaussian kernel in terms of both trade-off and stability.

Batch Size: In order to examine the effect of batch size on OptNet-ARL (with Gaussian kernel), we train the encoder with different values of batch size between 2 and 4000 (entire training data). The results are illustrated in Figure 4.4 (c). We observe that the HV is quite insensitive to batch sizes greater than 25, which implies that the gradient of the mini-batch is an accurate enough estimator of the gradient of the entire data.

Embedding Dimensionality: We also study the effect of embedding dimensionality (r) by

[1] $k(\boldsymbol{z}, \boldsymbol{z}') = \frac{1}{\sqrt{\|\boldsymbol{z}-\boldsymbol{z}'\|^2+c^2}}$



(a) (b) (c) (d)

Figure 4.4: **Mixture of Gaussians:** (a) Input data. The target task is to learn a representation that is informative enough to reconstruct the input data and, at the same time, hide the color information (● versus ●). (b) The trade-off between the MSEs of adversary and target task for different ARL methods. (c) The HVs of OptNet-ARL (Gaussian kernel) versus different batch size values in $[2, 4000]$. (d) The HV values of OptNet-ARL (Gaussian kernel) versus different values of r in $[1, 512]$.

examining different values for r in $[1, 512]$ using RBF-OptNet-ARL. The results are illustrated in Figure 4.4 (d). It is evident that the optimal embedding dimensionality ($r = 2$) outperforms other values of r. Additionally, HV of $r = 1$ suffers severely due to the information loss in embedding, while for $2 < r \leq 512$, the trade-off performance is comparable to that of optimal embedding dimensionality, i.e., $r = 2$.

4.5 Summary

Adversarial representation learning is a minimax theoretic game formulation that affords explicit control over unwanted information in learned data representations. Optimization algorithms for

ARL, such as SGDA and their variants, are sub-optimal, unstable, and unreliable in practice. In this chapter, we introduced OptNet-ARL to address this challenge by employing differentiable closed-form solvers, such as kernelized ridge regressors, to model the ARL players that are downstream from the representation. OptNet-ARL reduces iterative SGDA to a simple optimization, leading to a fast, stable, and reliable algorithm that outperforms existing ARL approaches on both small and large-scale datasets.

Chapter 5

Universal Invariant Representation Learning

5.1 Introduction

Ideally, the utility-invariance trade-off is defined as a bi-objective optimization problem:

$$\inf_{f \in \mathcal{H}_X,\, g_Y \in \mathcal{H}_Y} \mathbb{E}_{XY}\left[L_Y\left(g_Y\left(f(X)\right), Y\right)\right] \quad \text{such that} \quad \text{Dep}\left(f(X), S\right) \leq \epsilon, \tag{5.1}$$

where f is the encoder that extracts the representation $Z = f(X)$ from X, g_Y predicts $\widehat{Y}$ from the representation Z, $\mathcal{H}_X$ and $\mathcal{H}_Y$ are the corresponding hypothesis classes, and L_Y is the loss function for predicting the target attribute Y. The function $\text{Dep}(\cdot, \cdot) \geq 0$ is a parametric or non-parametric measure of statistical dependence, i.e., $\text{Dep}(Q, U) = 0$ implies Q and U are independent, and $\text{Dep}(Q, U) > 0$ implies Q and U are dependent with larger values indicating greater degrees of dependence. The scalar $\epsilon \geq 0$ is a user-defined parameter that controls the trade-off between the two objectives, with $\epsilon \to \infty$ being the standard scenario that has no invariance constraints with respect to (w.r.t.) S while $\epsilon \to 0$ enforces $Z \perp\!\!\!\perp S$ (i.e., total invariance). Involving all Borel functions in $\mathcal{H}_X$ and $\mathcal{H}_Y$ ensures that the best possible trade-off is included within the feasible solution space. For example, when $\epsilon \to \infty$ and L_Y is MSE loss, the optimal Bayes estimation,

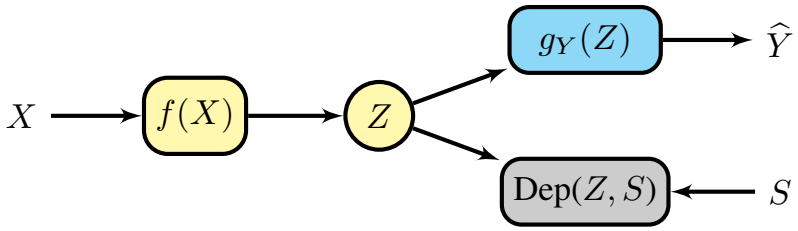


Figure 5.1: Invariant representation learning seeks a representation $Z = f(X)$ that contains as much information necessary for the downstream target predictor g_Y while being independent of the semantic attribute S.

$g_Y(f(X)) = \mathbb{E}[Y \mid X]$ is attainable.

In this chapter, we consider the linear combination of utility and invariance in (5.1) and define the optimal utility-invariance trade-off (denoted by $\mathcal{T}_{\text{Opt}}$) as a single objective optimization problem:

Definition 5.1.

$$\mathcal{T}_{\text{Opt}} := \tag{5.2}$$
$$\inf_{f \in \mathcal{H}_X} \left\{ (1-\lambda) \inf_{g_Y \in \mathcal{H}_Y} \mathbb{E}_{X,Y}\left[L_Y\left(g_Y\left(f(X)\right), Y\right)\right] + \lambda \, \text{Dep}\left(f(X), S\right) \right\}, \quad 0 \leq \lambda < 1,$$

where λ controls the trade-off between utility and invariance (e.g., $\lambda = 0$ corresponds to ignoring the invariance and only optimizing the utility, while $\lambda \to 1$ corresponds to $Z \perp\!\!\!\perp S$).

The motivation behind deploying this single-objective IRepL is that any solution to this simplified problem is a solution to the bi-objective problem in (5.1) and even (5.2) is challenging to solve, and it has not been fully investigated by existing works. An illustration of the utility-invariance trade-off is illustrated in Figure 5.2. In this chapter, we restrict $\mathcal{H}_X$ to be some RKHSs and $\text{Dep}(Z, S)$ to be a simplified version of the Hilbert-Schmidt Independence Criterion (HSIC) [81]. Further, we replace the target loss function in (5.2) by $\text{Dep}(Z, Y)$ as presented and justified in Sections 5.3.1 and 5.5.2.

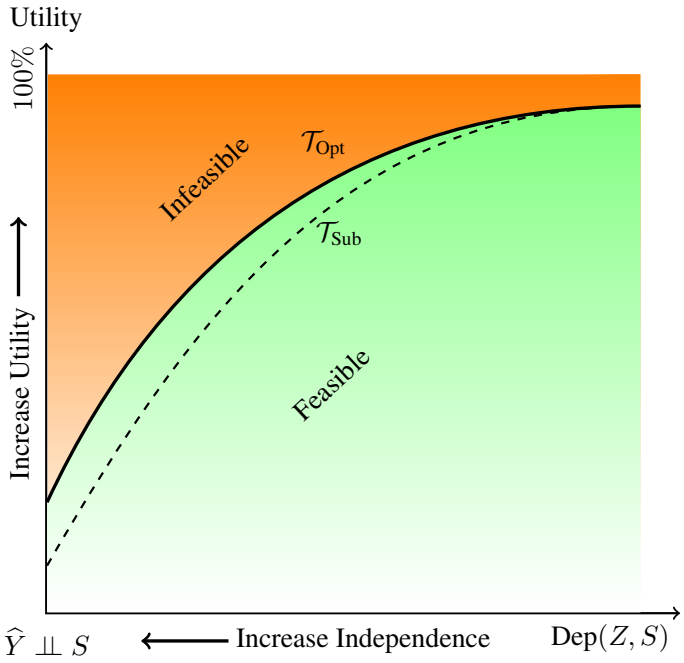


Figure 5.2: (The trade-off (denoted by $\mathcal{T}_{\text{Opt}}$) between utility (target task performance) and invariance (measured by the dependence metric $\text{Dep}(Z, S)$) is induced by a controlled representation learner in the hypothesis class of all Borel functions.

The basic idea of representation learning that discards unwanted semantic information has been explored under many contexts like invariant, fair, or privacy-preserving learning. In domain adaptation [11, 12, 39], the goal is to learn features that are independent of the data domain. In fair learning [41, 42, 43, 40, 36, 21, 23, 24, 22, 44, 2, 26, 45, 46, 47, 48, 49], the goal is to discard the demographic information that leads to unfair outcomes. Similarly, there is growing interest in mitigating unintended leakage of private information from representations [51, 52, 1, 53, 54].

A vast majority of this body of work is empirical in nature. They implicitly look for single or multiple points on the trade-off between utility and semantic information and do not explicitly seek to characterize the whole trade-off front. Overall, these approaches are not concerned with or aware of the inherent utility-invariance trade-off. In contrast, with the cost of restricting encoders to lie in some RKHSs, we *exactly* characterize the trade-off and propose a practical learning algorithm that achieves this trade-off.

5.1.1 Adversarial Representation Learning

Most practical approaches for learning fair, invariant, domain adaptive, or privacy-preserving representations discussed above are based on adversarial representation learning (ARL). ARL is typically formulated as

$$\inf_{f\in\mathcal{H}_X}\left\{(1-\lambda)\inf_{g_Y\in\mathcal{H}_Y}\mathbb{E}_{X,Y}\left[L_Y\left(g_Y\left(f(X)\right),Y\right)\right]-\lambda\inf_{g_S\in\mathcal{H}_S}\mathbb{E}_{X,S}\left[L_S\left(g_S\left(f(X)\right),S\right)\right]\right\} \quad (5.3)$$

where L_S is the loss function of a hypothetical adversary g_S who intends to extract the semantic attribute S through the best estimator within the hypothesis class $\mathcal{H}_S$ and $0 \leq \lambda < 1$ is the utility-invariance trade-off parameter. ARL is a special case of (5.2) where the negative loss of the adversary, $-\inf_{g_S\in\mathcal{H}_S}\mathbb{E}_{X,S}\left[L_S\left(g_S\left(f(X)\right),S\right)\right]$ plays the role of $\text{Dep}(f(X),S)$. However, this form of adversarial learning suffers from a drawback. The induced independence measure is not guaranteed to account for all modes of non-linear dependence between S and Z if the adversary loss function L_S is not bounded like MSE or cross-entropy [56, 114]. In the case of MSE loss, even if the loss is maximized at a bounded value, where the corresponding representation $Z = f(X)$ is also bounded, still, it is not guaranteed that $Z \perp\!\!\!\perp S$ is attainable (see Section 5.2 for more details). This implies that designing the adversary loss in ARL that accounts for all modes of dependence is challenging, and it can be infeasible for some loss functions.

5.1.2 Trade-Offs in Invariant Representation Learning:

Prior work has established the existence of trade-offs in IRepL, both empirically and theoretically. In the following, we categorize them based on properties of interest.

Restricted Class of Attributes: A majority of existing work considers IRepL trade-offs under restricted settings, i.e., binary and/or categorical attributes Y and S. For instance, [60] uses

information-theoretic tools and characterizes the utility-fairness trade-off in terms of lower bounds when both Y and S are binary labels. Later [55] provided both upper and lower bounds for binary labels. By leveraging Chernoff bound, [61] proposed a construction method to generate an ideal representation beyond the input data to achieve perfect fairness while maintaining the best performance on the target task. In the case of categorical features, a lower bound on utility-fairness trade-off has been provided by [6] for the total invariance scenario (i.e., $Z \perp\!\!\!\perp S$). In contrast to this body of work, our trade-off analysis applies to multidimensional continuous/discrete attributes. To the best of our knowledge, the only prior works with a general setting are [49] and [9]. However, in [9], both S and Y are restricted to be continuous/discrete or binary at the same time (e.g., it is not possible to have Y binary while S is continuous).

Characterizing Exact versus Bounds on Trade-Off: To the best of our knowledge, all existing approaches except [49], which obtains the trade-off for the linear dependence only, characterize the trade-off in terms of upper and/or lower bounds. In contrast, we *exactly* characterize a near-optimal trade-off with closed-form expressions for encoders belonging to some RKHSs.

Optimal Encoder and Representation: Another property of practical interest is the optimal encoder that achieves the desired point on the utility-invariance trade-off and the corresponding representation(s). Existing works which only study bounds on the trade-off do not obtain the encoder that achieves those bounds. [49] do develop a learning algorithm that obtains a globally optimal encoder, but only under a linear dependence measure between Z and S. HSIC, a universal measure of dependence, has been adopted by prior work (e.g., [62]) to quantify all types of dependencies between Z and S. However, these methods adopt stochastic gradient descent for optimizing the underlying non-convex optimization problem. As such, they fail to provide guarantees that the representation learning problem converges to a global optima. In contrast, we obtain a closed-form solution for the optimal encoder and its corresponding representation while detecting all modes of

dependence between Z and S.

Summary of Contributions: i) We design a dependence measure that accounts for all modes of dependence between Z and S (under a mild assumption) while allowing for analytical tractability. ii) We employ functions in RKHSs and obtain closed-form solutions for the IRepL optimization problem. Consequently, we exactly characterize a near-optimal approximation of $\mathcal{T}_{\text{Opt}}$ via encoders restricted to RKHSs. iii) We obtain a closed-form estimator for the encoder that achieves the near-optimal trade-off, and we establish its numerical convergence. iv) Using random Fourier features (RFF) [115], we provide a scalable version (in terms of both memory and computation) of our IRepL algorithm. v) We numerically quantify our $\mathcal{T}_{\text{Opt}}$ (denoted by K-$\mathcal{T}_{\text{Opt}}$) on an illustrative problem as well as large-scale real-world datasets, Folktables [116] and CelebA [111], where we compare K-$\mathcal{T}_{\text{Opt}}$ to those obtained by existing works.

5.2 Deficiency of Mean-Squared Error as A Measure of Dependence

Theorem 5.2. *Let $\mathcal{H}_S$ contain all Borel functions, S be a d_S-dimensional RV, and $L_S(\cdot,\cdot)$ be MSE loss. Then,*

$$Z \in \arg\sup \left\{ \inf_{g_S \in \mathcal{H}_S} \mathbb{E}_{X,S} \left[L_S \left(g_S \left(Z \right), S \right) \right] \right\} \Leftrightarrow \mathbb{E}[S \mid Z] = \mathbb{E}[S].$$

Proof. Let S_i, $(g_S(Z))_i$, and $(\mathbb{E}[S \mid Z])_i$ denote the i-th entries of S, $g_S(Z)$, and $\mathbb{E}[S \mid Z]$, respec-

tively. Then, it follows that

$$\begin{aligned}
\inf_{g_S \in \mathcal{H}_S} \mathbb{E}_{X,S}\left[L_S\left(g_S\left(Z\right), S\right)\right] &= \inf_{g_S \in \mathcal{H}_S} \sum_{i=1}^{d_S} \mathbb{E}_{X,S}\left[\left(\left(g_S(Z)\right)_i - S_i\right)^2\right] \\
&= \sum_{i=1}^{d_S} \mathbb{E}_{X,S}\left[\left(\left(\mathbb{E}[S \mid Z]\right)_i - S_i\right)^2\right] \\
&\leq \sum_{i=1}^{d_S} \mathbb{E}_{S}\left[\left(\left(\mathbb{E}[S]\right)_i - S_i\right)^2\right] = \sum_{i=1}^{d_S} \mathbb{V}\text{ar}[S_i],
\end{aligned}$$

where the second step is due to the optimality of conditional mean (i.e., Bayes estimation) for MSE [117] and the last step is because independence between Z and S leads to an upper bound on MSE. Therefore, if $Z \in \arg\sup\left\{\inf_{g_S \in \mathcal{H}_S} \mathbb{E}_{X,S}\left[L_S\left(g_S\left(Z\right), S\right)\right]\right\}$, then $\mathbb{E}[S \mid Z] = \mathbb{E}[S]$. On the other hand, if $\mathbb{E}[S \mid Z] = \mathbb{E}[S]$, then it follows immediately that

$$Z \in \arg\sup\left\{\inf_{g_S \in \mathcal{H}_S} \mathbb{E}_{X,S}\left[L_S\left(g_S\left(Z\right), S\right)\right]\right\}.$$

□

This theorem implies that an optimal adversary does not necessarily lead to a representation Z that is statistically independent of S but instead leads to S being mean independent of the representation Z.

5.3 Problem Setting

5.3.1 Problem Setup

The representation RV Z can be expressed as

$$\begin{aligned} Z = \boldsymbol{f}(X) &:= [Z_1, \cdots, Z_r]^T \in \mathbb{R}^r, \quad Z_j = f_j(X), f_j \in \mathcal{H}_X \; \forall j = 1, \ldots, r, \\ &= \boldsymbol{\Theta} \left[k_X(\boldsymbol{x}_1, X), \cdots, k_X(\boldsymbol{x}_n, X)\right] \end{aligned} \tag{5.4}$$

where r is the dimensionality of the representation and $\boldsymbol{\Theta} \in \mathbb{R}^{r \times n}$. As we will discuss in Corollary 5.1, unlike common practice where r is chosen on an ad-hoc basis, it is an object of interest for optimization. We consider a general scenario where both Y and S can be continuous/discrete or categorical, or one of Y or S is continuous/discrete while the other is categorical. To accomplish this, we replace the target loss, $\inf_{g_Y \in \mathcal{H}_Y} \mathbb{E}_{X,Y}\left[L_Y\left(g_Y(Z), Y\right)\right]$ in (5.2) by the negative of a non-parametric measure of dependence, i.e., $-\text{Dep}\left(Z, Y\right)$. The main reason for this replacement is that maximizing statistical dependency between the representation Z and the target attribute Y can flexibly learn a representation that is applicable for different downstream target tasks, including regression, classification, clustering, etc [118]. Particularly, Theorem 5.8 in Section 5.5.2 indicates that with an appropriate choice of involved RKHS for $\text{Dep}\left(Z, Y\right)$, we can learn a representation that lends itself to an estimator that performs as optimally as the Bayes estimation, i.e., $\mathbb{E}_X[Y|X]$. Furthermore, in an unsupervised setting, where there is no target attribute Y, the target loss can be replaced with $\text{Dep}\left(Z, X\right)$, which implicitly forces the representation Z to be as dependent on the input data X. This scenario is of practical interest when a data producer aims to provide an invariant representation for an unknown downstream target task.

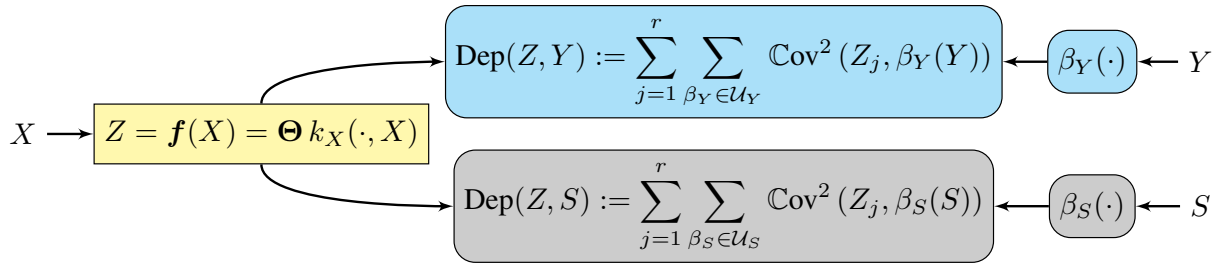


Figure 5.3: Our IRepL model consists of three components: i) An r-dimensional encoder $\boldsymbol{f}$ belonging to the universal RKHS $\mathcal{H}_X$. ii) A measure of dependence that accounts for all dependence modes between data representation Z and semantic attribute S induced by the covariance between $Z = \boldsymbol{f}(X)$ and $\beta_S(S)$ where β_S belongs to a universal RKHS $\mathcal{H}_S$. iii) A measure of dependency between Z and the target attribute Y defined similarly to that for S.

5.4 Choice of Dependence Measure

We only discuss for $\text{Dep}\,(Z,S)$ since $\text{Dep}\,(Z,Y)$ follows similarly. Accounting for all possible non-linear relations between RVs is a key desideratum of dependence measures. A well-known example of such measures is MI (e.g., MINE [119]). However, calculating MI for multidimensional continuous representation is analytically challenging and computationally intractable. Kernel-based measures are an alternative solution with the attractive properties of being computationally feasible/efficient and analytically tractable [83].

Principally, $Z \perp\!\!\!\perp S$ iff $\mathbb{C}\text{ov}(\alpha(Z), \beta_S(S)) = 0$ for all Borel functions $\alpha : \mathbb{R}^r \to \mathbb{R}$ and $\beta_s : \mathbb{R}^{d_S} \to \mathbb{R}$ belonging to the universal RKHSs $\mathcal{H}_Z$ and $\mathcal{H}_S$, respectively. Alternatively, $Z \perp\!\!\!\perp S$ iff $\text{HSIC}(Z,S) = 0$ for HSIC [81] being defined as

$$\text{HSIC}(Z,S) := \sum_{\alpha \in \mathcal{U}_Z} \sum_{\beta_S \in \mathcal{U}_S} \mathbb{C}\text{ov}^2\left(\alpha(Z), \beta_S(S)\right), \tag{5.5}$$

where $\mathcal{U}_Z$ and $\mathcal{U}_S$ are countable orthonormal basis sets for the separable universal RKHSs $\mathcal{H}_Z$ and $\mathcal{H}_S$, respectively. However, since $Z = \boldsymbol{f}(X)$, calculating $\mathbb{C}\text{ov}(\alpha(Z), \beta_S(S))$ necessitates the application of a cascade of kernels, which limits the analytical tractability of our solution.

Therefore, we adopt a simplified version of HSIC that considers transformation on S only but affords analytical tractability for solving the IRepL optimization problem. We define this measure as

$$\text{Dep}(Z, S) := \sum_{j=1}^{r} \sum_{\beta_S \in \mathcal{U}_S} \mathbb{C}\text{ov}^2 \left(Z_j, \beta_S(S) \right), \tag{5.6}$$

where $Z_j = f_j(X)$ for f_js defined in (5.4). We note that $\text{Dep}(\cdot, \cdot)$, unlike HSIC and other kernelization-based dependence measures, is not symmetric. However, symmetry is not necessary for measuring statistical dependence. The measure $\text{Dep}(Z, S)$ in (5.6) captures all modes of non-linear dependence under the assumption that the distribution of a low-dimensional projection of high-dimensional data is approximately normal [120], [121]. To see why this reasoning is relevant, we note from (5.4) that Z can be expressed as $Z = \boldsymbol{\Theta} V$, where $V \in \mathbb{R}^n$ and $\boldsymbol{\Theta} \in \mathbb{R}^{r \times n}$. This indicates that for large n and small r (which is the case for most real-world datasets), Z is indeed a low-dimensional projection of high-dimensional data. In other words, $(Z, \beta_S(S))$ is approximately a jointly Gaussian RV. In our numerical experiments in Section 5.6 we empirically observe that $\text{Dep}(Z, S)$ enjoys a monotonic relation with the underlying invariance measure and captures all modes of dependency in practice, especially as $Z \perp\!\!\!\perp S$. Nevertheless, if the normality assumption on the distribution of $(Z, \beta_S(S))$ fails, $\text{Dep}(Z, S)$ reduces to measuring the linear dependency between Z and $\beta_S(S)$ for all Borel functions β_S. This corresponds to measuring the mean independency of Z from S, i.e., how much information a predictor (linear and non-linear) can infer (in the sense of MSE) about Z from S. See Section 5.2 for more technical details on mean independency.

Lemma 5.3. Let $\boldsymbol{K}_X, \boldsymbol{K}_S \in \mathbb{R}^{n \times n}$ be the Gram matrices corresponding to $\mathcal{H}_X$ and $\mathcal{H}_S$, respectively, i.e., $(\boldsymbol{K}_X)_{ij} = k_X(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $(\boldsymbol{K}_S)_{ij} = k_S(\boldsymbol{s}_i, \boldsymbol{s}_j)$, where covariance is empirically

estimated as

$$\mathbb{C}\text{ov}\left(f_j(X), \beta_S(S)\right) \approx \frac{1}{n}\sum_{i=1}^{n} f_j(\boldsymbol{x}_i)\beta_S(\boldsymbol{s}_i) - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{k=1}^{n} f_j(\boldsymbol{x}_i)\beta_S(\boldsymbol{s}_k).$$

It follows that, the corresponding empirical estimation for $\text{Dep}\,(Z, S)$ is

$$\text{Dep}^{\text{emp}}\,(Z, S) \;=\; \frac{1}{n^2}\left\|\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{H}\boldsymbol{L}_S\right\|_F^2, \tag{5.7}$$

where $\boldsymbol{H} = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ is the centering matrix, and $\boldsymbol{L}_S$ is a full column-rank matrix in which $\boldsymbol{L}_S\boldsymbol{L}_S^T = \boldsymbol{K}_S$ (Cholesky factorization). Furthermore, the empirical estimator in (5.7) has a bias of $\mathcal{O}(n^{-1})$ and a convergence rate of $\mathcal{O}(n^{-1/2})$.

Proof. See Appendix B.2 □

Notice that the dependence measure between Z and Y can be defined similarly.

5.5 Exact Kernelized Trade-Off

Consider the optimization problem corresponding to $\mathcal{T}_{\text{Opt}}$ in (5.2). Recall that $Z = \boldsymbol{f}(X)$ is an r-dimensional RV, where the embedding dimensionality r is also a variable to be optimized. A common desideratum of learned representations is that of compactness [122], to avoid learning representations with redundant information where different dimensions are highly correlated to each other. Therefore, going beyond the assumption that each component of $\boldsymbol{f}$ (i.e., f_js) belongs to the universal RKHS $\mathcal{H}_X$, we impose additional constraints on the representation. Specifically, we constrain the search space of the encoder $\boldsymbol{f}(\cdot)$ to learn a disentangled representation [122] as

follows

$$\mathcal{A}_r := \left\{ (f_1, \cdots, f_r) \mid f_i, f_j \in \mathcal{H}_X, \mathbb{C}\text{ov}\left(f_i(X), f_j(X)\right) + \gamma \langle f_i, f_j \rangle_{\mathcal{H}_X} = \delta_{i,j} \right\}. \tag{5.8}$$

In the above set, the $\mathbb{C}\text{ov}\left(f_i(X), f_j(X)\right)$ part enforces the covariance matrix of $Z = \boldsymbol{f}(X)$ to be an identity matrix. This kind of disentanglement is used in the principal component analysis (PCA) and encourages the variance of each entry of Z to be one and different entries of Z to be uncorrelated with each other. The regularization part, $\gamma \langle f_i, f_j \rangle_{\mathcal{H}_X}$ encourages the encoder components to be as orthogonal as possible to each other and to be of the unit norm, which aids with numerical stability during empirical estimation [85]. As the following theorem states formally, such disentanglement is an invertible transformation, and therefore it does not nullify any information.

Theorem 5.4. *Let $Z = \boldsymbol{f}(X)$ be an arbitrary representation of the input data, where $\boldsymbol{f} \in \mathcal{H}_X$. Then, there exists an invertible Borel function $\boldsymbol{h}$, such that $\boldsymbol{h} \circ \boldsymbol{f}$ belongs to $\mathcal{A}_r$.*

Proof. See Appendix B.3 □

This Theorem implies that the disentanglement preserves the performance of the downstream task since any target network can revert the disentanglement $\boldsymbol{h}$ and access to the original representation Z. In addition, any deterministic measurable transformation of Z will not add any information about S that does not already exist in Z.

We define our K$-\mathcal{T}_{\text{Opt}}$ as

$$\sup_{\boldsymbol{f} \in \mathcal{A}_r} \left\{ J\left(\boldsymbol{f}, \lambda\right) := (1-\lambda)\, \text{Dep}\left(\boldsymbol{f}(X), Y\right) - \lambda\, \text{Dep}\left(\boldsymbol{f}(X), S\right) \right\}, \quad 0 \leq \lambda < 1, \tag{5.9}$$

where λ is the utility-invariance trade-off parameter. Fortunately, the above optimization problem lends itself to a closed-form solution as follows.

Theorem 5.5. *Consider the operator Σ_{SX} to be induced by the bi-linear functional*

$$\mathbb{C}ov(\alpha(X), \beta_S(S)) = \langle \beta_S, \Sigma_{SX}\alpha \rangle_{\mathcal{H}_S}$$

and define Σ_{YX} and Σ_{XX}, similarly. Then, a global optima for the optimization problem in (5.9) is the eigenfunctions corresponding to the r largest eigenvalues of the following generalized eigenvalue problem

$$\left((1-\lambda)\,\Sigma_{YX}^{*}\Sigma_{YX} - \lambda\,\Sigma_{SX}^{*}\Sigma_{SX}\right)\boldsymbol{f} = \tau\,\left(\Sigma_{XX} + \gamma\, I_X\right)\boldsymbol{f}, \tag{5.10}$$

where γ is the disentanglement regularization parameter defined in (5.8)*, I_X is the identity operator in $\mathcal{H}_X$, and Σ^* is the adjoint of Σ.*

Proof. Consider $\text{Dep}(Z, S)$ in (5.6):

$$\begin{aligned}
\text{Dep}(Z,S) &= \sum_{\beta_S \in \mathcal{U}_S} \sum_{j=1}^{r} \mathbb{C}\text{ov}^2\left(f_j(X), \beta_S(S)\right) \\
&= \sum_{j=1}^{r} \sum_{\beta_S \in \mathcal{U}_S} \left\langle \beta_S, \Sigma_{SX} f_j \right\rangle_{\mathcal{H}_S}^2 \\
&= \sum_{j=1}^{r} \|\Sigma_{SX} f_j\|_{\mathcal{H}_S}^2,
\end{aligned}$$

where the last step is due to Parseval's identity for the orthonormal basis set. Similarly, we have $\text{dep}(Z, Y) = \sum_{j=1}^{r} \|\Sigma_{YX} f_j\|_{\mathcal{H}_Y}^2$. Recall that $Z = \boldsymbol{f}(X) = [(f_1(X), \cdots, f_r(X)]$, then it follows

that

$$\begin{aligned} J(\boldsymbol{f}(X)) &= (1-\lambda)\sum_{j=1}^{r}\|\Sigma_{YX}f_j\|^2_{\mathcal{H}_Y} - \lambda\sum_{j=1}^{r}\|\Sigma_{SX}f_j\|^2_{\mathcal{H}_S} \\ &= (1-\lambda)\sum_{j=1}^{r}\langle\Sigma_{YX}f_j, \Sigma_{YX}f_j\rangle_{\mathcal{H}_Y} - \lambda\sum_{j=1}^{r}\langle\Sigma_{SX}f_j, \Sigma_{SX}f_j\rangle_{\mathcal{H}_S} \\ &= \sum_{j=1}^{r}\left\langle f_j, \left((1-\lambda)\,\Sigma^*_{YX}\Sigma_{YX} - \lambda\,\Sigma^*_{SX}\Sigma_{SX}\right)f_j\right\rangle_{\mathcal{H}_X}, \end{aligned}$$

where Σ^* is the adjoint operator of Σ. Further, note that $\mathbb{C}\text{ov}(f_i(X), f_j(X)) = \langle f_i, \Sigma_{XX}f_j\rangle_{\mathcal{H}_X}$. As a result, the optimization problem in (5.10) can be restated as

$$\sup_{\langle f_i,(\Sigma_{XX}+\gamma I_X)f_k\rangle_{\mathcal{H}_X}=\delta_{i,k}} \sum_{j=1}^{r}\left\langle f_j, \left((1-\lambda)\Sigma^*_{YX}\Sigma_{YX} - \lambda\,\Sigma^*_{SX}\Sigma_{SX}\right)f_j\right\rangle_{\mathcal{H}_X}, \quad 1\le i,k\le r$$

where I_X denotes identity operator from $\mathcal{H}_X$ to $\mathcal{H}_X$. This optimization problem is known as generalized Rayleigh quotient [123] and a possible solution to it is given by the eigenfunctions corresponding to the r largest eigenvalues of the following generalized problem

$$\left((1-\lambda)\,\Sigma_{XY}\Sigma_{YX} - \lambda\,\Sigma_{XS}\Sigma_{SX}\right)f = \lambda\,\left(\Sigma_{XX}+\gamma I_X\right)f.$$

□

Remark. If the trade-off parameter $\lambda = 0$ (i.e., no semantic independence constraint is imposed) and $\gamma \to 0$, the solution in Theorem 5.5 is equivalent to a supervised kernel-PCA. On the other hand, if $\lambda \to 1$ (i.e., utility is ignored and only semantic independence is considered), the solution in Theorem 5.5 is the eigenfunctions corresponding to the r smallest eigenvalues of $\Sigma^*_{SX}\Sigma_{SX}$, which are the directions that are the least explanatory of the semantic attribute S.

Now, consider the empirical counterpart of the optimization problem (5.9),

$$\sup_{\boldsymbol{f}\in\mathcal{A}_r} \left\{ J^{\text{emp}}(\boldsymbol{f},\lambda) := (1-\lambda)\,\text{Dep}^{\text{emp}}\left(\boldsymbol{f}(X),Y\right) - \lambda\,\text{Dep}^{\text{emp}}\left(\boldsymbol{f}(X),S\right)\right\}, \quad 0 \le \lambda < 1 \tag{5.11}$$

where $\text{Dep}^{\text{emp}}\left(\boldsymbol{f}(X),S\right)$ is given in (5.7) and $\text{Dep}^{\text{emp}}\left(\boldsymbol{f}(X),Y\right)$ is defined similarly.

Theorem 5.6. *Let the Cholesky factorization of $\boldsymbol{K}_X$ be $\boldsymbol{K}_X = \boldsymbol{L}_X\boldsymbol{L}_X^T$, where $\boldsymbol{L}_X \in \mathbb{R}^{n\times d}$ ($d \le n$) is a full column-rank matrix. Let $r \le d$, then a solution to (5.11) is*

$$\boldsymbol{f}^{Opt}(X) = \boldsymbol{\Theta}^{Opt}\left[k_X(\boldsymbol{x}_1,X),\cdots,k_X(\boldsymbol{x}_n,X)\right]^T,$$

where $\boldsymbol{\Theta}^{opt} = \boldsymbol{U}^T\boldsymbol{L}_X^{\dagger}$ and the columns of $\boldsymbol{U}$ are eigenvectors corresponding to the r largest eigenvalues of the following generalized eigenvalue problem.

$$\boldsymbol{L}_X^T\left((1-\lambda)\boldsymbol{H}\boldsymbol{K}_Y\boldsymbol{H} - \lambda\boldsymbol{H}\boldsymbol{K}_S\boldsymbol{H}\right)\boldsymbol{L}_X\boldsymbol{u} = \tau\left(\frac{1}{n}\,\boldsymbol{L}_X^T\boldsymbol{H}\boldsymbol{L}_X + \gamma\boldsymbol{I}\right)\boldsymbol{u}. \tag{5.12}$$

Further, the objective value of (5.11) is equal to $\sum_{j=1}^{r}\beta_j$, where $\{\beta_1,\cdots,\beta_r\}$ are the r largest eigenvalues of (5.12)*.*

Proof. Consider the Cholesky factorization, $\boldsymbol{K}_x = \boldsymbol{L}_x\boldsymbol{L}_x^T$ where $\boldsymbol{L}_x$ is a full column-rank matrix.

Using the representer theorem, the disentanglement property in (5.8) can be expressed as

$$
\begin{aligned}
& \mathbb{C}\mathrm{ov}\left(f_i(X),\ f_j(X)\right) + \gamma \left\langle f_i, f_j \right\rangle_{\mathcal{H}_X} \\
= \ & \frac{1}{n}\sum_{k=1}^{n} f_i(\boldsymbol{x}_k) f_j(\boldsymbol{x}_k) - \frac{1}{n^2}\sum_{k=1}^{n} f_i(\boldsymbol{x}_k) \sum_{m=1}^{n} f_j(\boldsymbol{x}_m) + \gamma \left\langle f_i, f_j \right\rangle_{\mathcal{H}_X} \\
= \ & \frac{1}{n}\sum_{k=1}^{n}\sum_{t=1}^{n} \boldsymbol{K}_X(\boldsymbol{x}_k, \boldsymbol{x}_t)\theta_{it} \sum_{m=1}^{n} \boldsymbol{K}_X(\boldsymbol{x}_k, \boldsymbol{x}_m)\theta_{jm} - \frac{1}{n^2}\boldsymbol{\theta}_i^T \boldsymbol{K}_X \mathbf{1}_n \mathbf{1}_n^T \boldsymbol{K}_X \boldsymbol{\theta}_j + \gamma \left\langle f_i, f_j \right\rangle_{\mathcal{H}_X} \\
= \ & \frac{1}{n}\left(\boldsymbol{K}_X \boldsymbol{\theta}_i\right)^T \left(\boldsymbol{K}_x \boldsymbol{\theta}_j\right) - \frac{1}{n^2}\boldsymbol{\theta}_i^T \boldsymbol{K}_X \mathbf{1}_n \mathbf{1}_n^T \boldsymbol{K}_X \boldsymbol{\theta}_j \\
& + \gamma \left\langle \sum_{k=1}^{n} \theta_{ik} k_X(\cdot, \boldsymbol{x}_k), \sum_{t=1}^{n} \theta_{it} k_X(\cdot, \boldsymbol{x}_t) \right\rangle_{\mathcal{H}_X} \\
= \ & \frac{1}{n}\boldsymbol{\theta}_i^T \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{K}_X \boldsymbol{\theta}_j + \gamma\, \boldsymbol{\theta}_i^T \boldsymbol{K}_X \boldsymbol{\theta}_j \\
= \ & \frac{1}{n}\boldsymbol{\theta}_i^T \boldsymbol{L}_X \left(\boldsymbol{L}_X^T \boldsymbol{H} \boldsymbol{L}_X + n\gamma\, \boldsymbol{I}\right) \boldsymbol{L}_X^T \boldsymbol{\theta}_j \\
= \ & \delta_{i,j}.
\end{aligned}
$$

As a result, $\boldsymbol{f} \in \mathcal{A}_r$ is equivalent to

$$
\boldsymbol{\Theta} \boldsymbol{L}_X \underbrace{\left(\frac{1}{n}\boldsymbol{L}_X^T \boldsymbol{H} \boldsymbol{L}_X + \gamma \boldsymbol{I}\right)}_{:=\boldsymbol{C}} \boldsymbol{L}_X^T \boldsymbol{\Theta}^T = \boldsymbol{I}_r,
$$

where $\boldsymbol{\Theta} := [\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_r]^T \in \mathbb{R}^{r \times n}$.

Let $\boldsymbol{V} = \boldsymbol{L}_X^T \boldsymbol{\Theta}^T$ and consider the optimization problem in (13):

$$
\begin{aligned}
& \sup_{\boldsymbol{f} \in \mathcal{A}_r} \left\{ (1-\lambda)\, \mathrm{Dep}^{\mathrm{emp}}(\boldsymbol{f}(X), Y) - \lambda\, \mathrm{Dep}^{\mathrm{emp}}(\boldsymbol{f}(X), S) \right\} \\
= & \sup_{\boldsymbol{f} \in \mathcal{A}_r} \frac{1}{n^2} \left\{ (1-\lambda)\, \|\boldsymbol{\Theta} \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{L}_Y\|_F^2 - \lambda\, \|\boldsymbol{\Theta} \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{L}_S\|_F^2 \right\} \\
= & \sup_{\boldsymbol{f} \in \mathcal{A}_r} \frac{1}{n^2} \left\{ (1-\lambda)\, \mathrm{Tr} \left\{ \boldsymbol{\Theta} \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{K}_Y \boldsymbol{H} \boldsymbol{K}_X \boldsymbol{\Theta}^T \right\} - \lambda\, \mathrm{Tr} \left\{ \boldsymbol{\Theta} \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{K}_S \boldsymbol{H} \boldsymbol{K}_X \boldsymbol{\Theta}^T \right\} \right\} \\
= & \max_{\boldsymbol{V}^T \boldsymbol{C} \boldsymbol{V} = \boldsymbol{I}_r} \frac{1}{n^2} \mathrm{Tr} \left\{ \boldsymbol{\Theta} \boldsymbol{L}_X \boldsymbol{B} \boldsymbol{L}_X^T \boldsymbol{\Theta}^T \right\} \\
= & \max_{\boldsymbol{V}^T \boldsymbol{C} \boldsymbol{V} = \boldsymbol{I}_r} \frac{1}{n^2} \mathrm{Tr} \left\{ \boldsymbol{V}^T \boldsymbol{B} \boldsymbol{V} \right\} \qquad (5.13)
\end{aligned}
$$

where the second step is due to (5.7) and

$$
\boldsymbol{B} \;:=\; \boldsymbol{L}_X^T \left((1-\lambda) \boldsymbol{H} \boldsymbol{K}_Y \boldsymbol{H} - \lambda \boldsymbol{H} \boldsymbol{K}_S \boldsymbol{H} \right) \boldsymbol{L}_X
$$

It is shown in [90] that an[1] optimizer of (5.13) is any matrix $\boldsymbol{U}$ whose columns are eigenvectors corresponding to r largest eigenvalues of generalized problem

$$
\boldsymbol{B}\boldsymbol{u} = \tau\, \boldsymbol{C}\boldsymbol{u} \qquad (5.14)
$$

and the maximum value is the summation of r largest eigenvalues. Once $\boldsymbol{U}$ is determined, then, any $\boldsymbol{\Theta}$ in which $\boldsymbol{L}_X^T \boldsymbol{\Theta}^T = \boldsymbol{U}$ is optimal $\boldsymbol{\Theta}$ (denoted by $\boldsymbol{\Theta}^{\mathrm{opt}}$). Note that $\boldsymbol{\Theta}^{\mathrm{opt}}$ is not unique and has a general form of

$$
\boldsymbol{\Theta}^T = \left(\boldsymbol{L}_X^T \right)^{\dagger} \boldsymbol{U} + \boldsymbol{\Lambda}_0, \quad \mathcal{R}(\boldsymbol{\Lambda}_0) \subseteq \mathcal{N} \left(\boldsymbol{L}_X^T \right).
$$

[1] Optimal $\boldsymbol{V}$ is not unique.

However, setting $\boldsymbol{\Lambda}_0$ to zero would lead to a minimum norm for $\boldsymbol{\Theta}$. Therefore, we opt $\boldsymbol{\Theta}^{\text{opt}} = \boldsymbol{U}^T \boldsymbol{L}_X^{\dagger}$. □

Corollary 5.1. *Embedding Dimensionality*: A useful corollary of Theorem 5.6 is characterizing optimal embedding dimensionality as a function of the trade-off parameter, λ:

$$r^{\text{Opt}}(\lambda) := \arg \sup_{0 \le r \le d} \left\{ \sup_{\boldsymbol{f} \in \mathcal{A}_r} \{ J^{\text{emp}}(\boldsymbol{f}, \lambda) \} \right\} = \text{ number of non-negative eigenvalues of (5.12).}$$

Proof. From proof of Theorem 5.6, we know that

$$\sup_{\boldsymbol{f} \in \mathcal{A}_r} \{ (1-\lambda) \, \text{Dep}^{\text{emp}}(\boldsymbol{f}(X), Y) - \lambda \, \text{Dep}^{\text{emp}}(\boldsymbol{f}(X), S) \} = \sum_{j=1}^{r} \tau_j,$$

where $\{\tau_1, \cdots, \tau_n\}$ are eigenvalues of the generalized problem in (5.12) in decreasing order. It follows immediately that

$$\arg \sup_r \left\{ \sum_{j=1}^{r} \tau_j \right\} = \text{number of non-negative elements of } \{\tau_1, \cdots, \tau_l\}.$$

□

To examine these results, consider two extreme cases: i) If there is no semantic independence constraint (i.e., $\lambda = 0$), all eigenvalues of (5.12) are non-negative since $\boldsymbol{H} \boldsymbol{K}_Y \boldsymbol{H}$ is a non-negative definite matrix and $\frac{1}{n} \boldsymbol{L}_X^T \boldsymbol{H} \boldsymbol{L}_X + \gamma \boldsymbol{I}$ is a positive definite matrix. This indicates that r^{Opt} is equal to the maximum possible value (that is equal to d), and therefore it is not required for Z to nullify any information in X. ii) If we only concern about the semantic independence and ignore the target task utility (i.e., $\lambda \to 1$), all eigenvalues of (5.12) are non-positive and therefore r^{Opt} would be the number of zero eigenvalues of (5.12). This indicates that $\text{Dep}^{\text{emp}}(Z, S)$ in (5.7) is equal to

zero, since $\boldsymbol{\Theta}^{\text{opt}}\boldsymbol{K}_X$ is zero for zero eigenvalues of (5.12) when $\lambda \to 1$. In this case, adding more dimension to Z will necessarily increase $\text{Dep}^{\text{emp}}(Z, S)$.

The following Theorem characterizes the convergence behavior of empirical K$-\mathcal{T}_{\text{Opt}}$ to its population counterpart.

Theorem 5.7. *Assume that k_S and k_Y are bounded by one and $f_j^2(\boldsymbol{x}_i) \leq M$ for any $j = 1, \ldots, r$ and $i = 1, \ldots, n$ for which $\boldsymbol{f} = (f_1, \ldots, f_r) \in \mathcal{A}_r$. Then, for any $n > 1$ and $0 < \delta < 1$, with probability at least $1 - \delta$, we have*

$$\left| \sup_{\boldsymbol{f} \in \mathcal{A}_r} J(\boldsymbol{f}, \lambda) - \sup_{\boldsymbol{f} \in \mathcal{A}_r} J^{emp}(\boldsymbol{f}, \lambda) \right| \leq rM \sqrt{\frac{\log(6/\delta)}{0.22^2\, n}} + \mathcal{O}\left(\frac{1}{n}\right).$$

Proof. See Appendix B.4 □

Note that, for any $\boldsymbol{x}$ in the training set, $f_j(\boldsymbol{x})$ can be calculated as $f_j(\boldsymbol{x}) = \sum_{i=1}^{n} \theta_{ji} k_X(\boldsymbol{x}_i, \boldsymbol{x})$. We can assume that $k_X(\cdot, \cdot)$ is bounded. For example, in RBF Gaussian and Laplacian RKHSs (that are universal), $k_X(\cdot, \cdot) \leq 1$. This implies that $f_j^2(\boldsymbol{x}) \leq \sqrt{n}\|\boldsymbol{\theta}_j\|$, where $\boldsymbol{\theta}_j$ is j-th row of $\boldsymbol{\Theta}$ in equation (5.4). One always can normalize $f_j(\boldsymbol{x})$ by dividing it by the maximum of $\sqrt{n}\|\boldsymbol{\theta}_j\|$ over js, or by dividing by the maximum of $|f_j(\boldsymbol{x}_i)|$ over is and js. Notice that, this normalization is only a scalar multiplication and has no effect on the invariance of $Z = \boldsymbol{f}(X)$ to S and the utility of any downstream target task predictor $g_Y(Z)$.

5.5.1 Numerical Complexity

Computational Complexity: If $\boldsymbol{L}_X$ in (5.12) is provided in the training dataset, then, the computational complexity of obtaining the optimal encoder is $\mathcal{O}(l^3)$, where $l \leq n$ is the numerical rank of the Gram matrix $\boldsymbol{K}_X$. However, the dominating part of the computational complexity is

due to the Cholesky factorization, $\boldsymbol{K}_X = \boldsymbol{L}_X \boldsymbol{L}_X^T$ which is $\mathcal{O}(n^3)$. Using random Fourier features (RFF) [115], $k_X(\boldsymbol{x}, \boldsymbol{x}')$ can be approximated by $\boldsymbol{r}_X(\boldsymbol{x})^T \boldsymbol{r}_X(\boldsymbol{x}')$, where $\boldsymbol{r}_X(\boldsymbol{x}) \in \mathbb{R}^d$. In this situation, the Cholesky factorization can be directly calculated as

$$\boldsymbol{L}_X = \begin{bmatrix} \boldsymbol{r}_X(\boldsymbol{x}_1)^T \\ \vdots \\ \boldsymbol{r}_X(\boldsymbol{x}_n)^T \end{bmatrix} \in \mathbb{R}^{n \times d}. \tag{5.15}$$

As a result, the computational complexity of obtaining the optimal encoder becomes $\mathcal{O}(d^3)$, where the RFF dimension, d can be significantly less than the sample size n with negligible sacrifice on $k_X(\boldsymbol{x}, \boldsymbol{x}') \approx \boldsymbol{r}_X(\boldsymbol{x})^T \boldsymbol{r}_X(\boldsymbol{x}')$ approximation.

Memory Complexity: The memory complexity of (5.12), if calculated naively, is $\mathcal{O}(n^2)$ since $\boldsymbol{K}_Y$ and $\boldsymbol{K}_S$ are n by n matrices. However, using RFF together with Cholesky factorization $\boldsymbol{K}_Y = \boldsymbol{L}_Y \boldsymbol{L}_Y^T$, $\boldsymbol{K}_S = \boldsymbol{L}_S \boldsymbol{L}_S^T$, the left-hand side of (5.12) can be re-arranged as

$$(1-\lambda) \left(\boldsymbol{L}_X^T \tilde{\boldsymbol{L}}_Y \right) \left(\tilde{\boldsymbol{L}}_Y^T \boldsymbol{L}_X \right) - \lambda \left(\boldsymbol{L}_X^T \tilde{\boldsymbol{L}}_S \right) \left(\tilde{\boldsymbol{L}}_S^T \boldsymbol{L}_X \right), \tag{5.16}$$

where $\tilde{\boldsymbol{L}}_Y^T = \boldsymbol{H} \boldsymbol{L}_Y = \boldsymbol{L}_Y - \frac{1}{n} \mathbf{1}_n (\mathbf{1}_n^T \boldsymbol{L}_Y)$ and therefore, the required memory complexity is $\mathcal{O}(nd)$. Note that, $\tilde{\boldsymbol{L}}_S^T$ and $\boldsymbol{H} \boldsymbol{L}_X$ can be calculated similarly.

5.5.2 Target Task Performance in K$-\mathcal{T}_{\text{Opt}}$

Assume that the desired target loss function is MSE. In the following Theorem, we show that maximizing Dep $(\boldsymbol{f}(X), Y)$ over $\boldsymbol{f} \in \mathcal{A}_r$ can learn a representation Z that is informative enough for a target predictor on Z to achieve the most optimal estimation, i.e., the Bayes estimation ($\mathbb{E}[Y \,|\, X]$).

Theorem 5.8. *Let $\boldsymbol{f}^*$ be the optimal encoder by maximizing Dep$(\boldsymbol{f}(X), Y)$, where $\gamma \to 0$ and $\mathcal{H}_Y$ is a linear RKHS. Then, there exist $\boldsymbol{W} \in \mathbb{R}^{d_Y \times r}$ and $\boldsymbol{b} \in \mathbb{R}^{d_Y}$ such that $\boldsymbol{W}\boldsymbol{f}^*(X) + \boldsymbol{b}$ is the Bayes estimator, i.e.,*

$$\begin{aligned}\left[\|\boldsymbol{W}Z^* + \boldsymbol{b} - Y\|^2\right] &= \inf_{h \text{ is Borel}} \mathbb{E}_{X,Y}\left[\|h(X) - Y\|^2\right] \\ &= \mathbb{E}_{X,Y}\left[\|\mathbb{E}[Y|\,X] - Y\|^2\right].\end{aligned}$$

Proof. We only prove this theorem for the empirical version due to its convergence to the population counterpart. The optimal Bayes estimator can be the composition of the kernelized encoder $Z = \boldsymbol{f}(X)$ and a linear regressor on top of it. More specifically, $\widehat{Y} = \boldsymbol{W}\boldsymbol{f}(X) + \boldsymbol{b}$ can approach to $\mathbb{E}[Y|\,X]$ if we optimize $\boldsymbol{f}$, $\boldsymbol{W}$, and $\boldsymbol{b}$ all together. This is because $\boldsymbol{f} \in \mathcal{H}_X$ can approximate any Borel function (due to the universality of $\mathcal{H}_X$) and, since $r \geq d_y$, $\boldsymbol{W}$ can be surjective. Let $\boldsymbol{Z} := [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n] \in \mathbb{R}^{r \times n}$ and $\boldsymbol{Y} := [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n] \in \mathbb{R}^{d_y \times n}$. Further, let $\tilde{\boldsymbol{Z}}$ and $\tilde{\boldsymbol{y}}$ be the centered (i.e., mean subtracted) version of $\boldsymbol{Z}$ and $\boldsymbol{Y}$, respectively. We firstly optimize $\boldsymbol{b}$ for any given $\boldsymbol{f}$, r, and $\boldsymbol{W}$:

$$\begin{aligned}\boldsymbol{b}_{\text{opt}} &:= \arg\min_{\boldsymbol{b}} \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{b} - \boldsymbol{y}_i\|^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{y}_i - \boldsymbol{W}\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i.\end{aligned}$$

Then, optimizing over $\boldsymbol{W}$ would lead to

$$\begin{aligned}\min_{\boldsymbol{W}} \frac{1}{n}\left\|\boldsymbol{W}\tilde{\boldsymbol{Z}} - \tilde{\boldsymbol{Y}}\right\|_F^2 &= \frac{1}{n}\min_{\boldsymbol{W}}\left\|\tilde{\boldsymbol{Z}}^T\boldsymbol{W}^T - \tilde{\boldsymbol{Y}}^T\right\|_F^2 \\ &= \min_{\boldsymbol{W}} \frac{1}{n}\left\|\tilde{\boldsymbol{Z}}^T\boldsymbol{W}^T - P_{\tilde{Z}}\tilde{\boldsymbol{Y}}^T\right\|_F^2 + \frac{1}{n}\left\|P_{\tilde{\boldsymbol{Z}}^\perp}\tilde{\boldsymbol{Y}}^T\right\|_F^2 \\ &= \frac{1}{n}\left\|P_{\tilde{\boldsymbol{Z}}^\perp}\tilde{\boldsymbol{Y}}^T\right\|_F^2 = \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}\right\|_F^2 - \frac{1}{n}\left\|P_{\tilde{\boldsymbol{Z}}}\tilde{\boldsymbol{Y}}^T\right\|_F^2,\end{aligned}$$

where $P_{\tilde{\boldsymbol{Z}}}$ denotes the orthogonal projector onto the column space of $\tilde{\boldsymbol{Z}}^T$ and a possible minimizer is $\boldsymbol{W}_{\text{opt}}^T = (\tilde{\boldsymbol{Z}}^T)^\dagger \tilde{\boldsymbol{Y}}^T$ or equivalently $\boldsymbol{W}_{\text{opt}} = \tilde{\boldsymbol{Y}}(\tilde{\boldsymbol{Z}})^\dagger$. Since the MSE loss is a function of the range (column space) of $\tilde{\boldsymbol{Z}}^T$, we can consider only $\tilde{\boldsymbol{Z}}^T$ with orthonormal columns or equivalently $\frac{1}{n}\tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^T = \boldsymbol{I}_r$. In this setting, it holds $P_{\tilde{\boldsymbol{Z}}} = \frac{1}{n}\tilde{\boldsymbol{Z}}^T\tilde{\boldsymbol{Z}}$. Now, consider optimizing $\boldsymbol{f}(X) = \boldsymbol{\Theta}\left[k_X(\boldsymbol{x}_1, X), \cdots, k_X(\boldsymbol{x}_n, X)\right]^T$. We have, $\tilde{\boldsymbol{Z}} = \boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{H}$ where $\boldsymbol{H}$ is the centering matrix. Let $\boldsymbol{V} = \boldsymbol{L}_x^T\boldsymbol{\Theta}^T$ and $\boldsymbol{C} = \frac{1}{n}\boldsymbol{L}_X^T\boldsymbol{H}\boldsymbol{L}_X$, then it follows that

$$
\begin{aligned}
& \min_{\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{H}\boldsymbol{K}_X\boldsymbol{\Theta}^T=n\boldsymbol{I}_r} \frac{1}{n}\left\{\left\|\tilde{\boldsymbol{Y}}\right\|_F^2 - \left\|P_{\tilde{\boldsymbol{Z}}}\tilde{\boldsymbol{Y}}^T\right\|_F^2\right\} \\
= \;& \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}\right\|_F^2 - \max_{\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{H}\boldsymbol{K}_X\boldsymbol{\Theta}^T=n\boldsymbol{I}_r} \frac{1}{n}\left\|P_{\tilde{\boldsymbol{Z}}}\tilde{\boldsymbol{Y}}^T\right\|_F^2 \\
= \;& \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}\right\|_F^2 - \max_{\boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}=\boldsymbol{I}_r} \frac{1}{n^2}\text{Tr}\left[\tilde{\boldsymbol{Y}}\boldsymbol{H}\boldsymbol{K}_X\boldsymbol{\Theta}^T\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{H}\tilde{\boldsymbol{Y}}^T\right] \\
= \;& \frac{1}{n^2}\left\|\tilde{\boldsymbol{Y}}\right\|_F^2 - \max_{\boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}=\boldsymbol{I}_r} \frac{1}{n^2}\text{Tr}\left[\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{H}\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}\boldsymbol{H}\boldsymbol{K}_X\boldsymbol{\Theta}^T\right] \\
= \;& \left\|\tilde{\boldsymbol{Y}}\right\|_F^2 - \max_{\boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}=\boldsymbol{I}_r} \frac{1}{n^2}\text{Tr}\left[\boldsymbol{V}^T\boldsymbol{L}_X^T\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\boldsymbol{V}\right] \\
= \;& \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}\right\|_F^2 - \frac{1}{n^2}\sum_{j=1}^{r}\lambda_j,
\end{aligned}
$$

where $\lambda_1, \cdots, \lambda_r$ are r largest eigenvalues of the following generalized problem

$$\boldsymbol{B}_0\boldsymbol{u} = \lambda\,\boldsymbol{C}\boldsymbol{u}$$

and $\boldsymbol{B}_0 := \boldsymbol{L}_X^T\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_X$. This resembles the eigenvalue problem in Section , equation (5.14) where $\lambda = 0$, $\mathcal{H}_Y$ is a linear RKHS and $\gamma \to 0$. □

This Theorem implies that not only Dep $(\boldsymbol{f}(X), Y)$ can preserve all the necessary information in Z to optimally predict Y, also, the learned representation is simple enough for a linear regressor to achieve the optimal performance.

5.6 Experiments

In this section, we numerically quantify our K$-\mathcal{T}_{\text{Opt}}$ through the closed-form solution for the encoder obtained in Section 5.5 on an illustrative toy example and two real-world datasets, Folktables and CelebA.

5.6.1 Baselines

We consider two types of baselines: (1) ARL (the main framework for IRepL) with MSE or Cross-Entropy as the adversarial loss. Such methods are expected to fail to learn a fully invariant representation [56, 114]. These include [24, 22, 2], and SARL [49]. (2) HSIC-based adversarial loss that accounts for all modes of dependence, and as such is theoretically expected to learn a fully invariant representation [62]. Among these baselines, except for SARL, all the others are optimized via iterative minimax optimization which is often unstable and not guaranteed to converge. On the other hand, SARL obtains a closed-form solution for the global-optima of the minimax optimization under a linear dependence measure between Z and, S which may fail to capture all modes of dependence between Z and S.

5.6.2 Datasets

Gaussian Toy Example: We design an illustrative toy example where X and S are mean independent in some dimensions but not fully independent in those dimensions. Specifically, X and S are 4-dimensional continuous RVs and generated as following

$$
\begin{aligned}
&U = [U_1, U_2, U_3, U_4] \sim \mathcal{N}\left(\mathbf{0}_4, \boldsymbol{I}_4\right), \quad N \sim \mathcal{N}\left(\mathbf{0}_4, \boldsymbol{I}_4\right), \quad U \perp\!\!\!\perp N \\
&X = \cos\left(\frac{\pi}{6} U\right) + 0.005N, \;\; S = \left[\sin\left(\frac{\pi}{6}\left[U_1, U_2\right]\right), \cos\left(\frac{\pi}{6}\left[U_3, U_4\right]\right)\right],
\end{aligned} \tag{5.17}
$$

where $\sin(\cdot)$ and $\cos(\cdot)$ are applied point-wise. To generate the target attribute, we define four binary RVs as follows.

$$Y_i = \mathbf{1}_{\{|U_i|>T\}}(U_i), \;\; i = 1, 2, 3, 4,$$

where $\mathbf{1}_{\mathcal{B}}(\cdot)$ is the indicator function, and we set $T = 0.6744$, so it holds that $\mathbb{P}[Y_i = 0] = \mathbb{P}[Y_i = 1] = 0.5$ for $i = 1, 2, 3, 4$. Finally, we define Y as a 16-class categorical RV concatenated by Y_is. Since S is dependent on X through all the dimensions of X, then, a wholly invariant Z (i.e., $Z \perp\!\!\!\perp S$) should not contain any information about X. However, since $[S_1, S_2]$ is only mean independent of $[X_1, X_2]$ (i.e., $\mathbb{E}[S_1, S_2 | X_1, X_2] = \mathbb{E}[S_1, S_2]$), ARL baselines with MSE as the adversary loss, i.e., [24, 22, 2] and SARL cannot capture the dependency of Z to $[S_1, S_2]$ and result in a representation that is always dependent on $[S_1, S_2]$ (see Section 5.2 for theoretical details). We sample $18,000$ instances from $\boldsymbol{p}_{X,Y,S}$, independently, and split these samples equally into training, validation, and testing partitions.

Folktables: We consider a fair representation learning task on Folktables [116] dataset (a derivation of the US census data). Particularly, we use 2018-WA (Washington) and 2018-NY (New York) census data where the target attribute Y is the employment status (binary for WA and 4 categories for NY) and the semantic attribute S is age (discrete value between 0 and 95 years). We seek to learn a representation that predicts employment status while being fair in demographic parity (DP) w.r.t. age. DP requires that the prediction $\widehat{Y}$ be independent of S which can be achieved by enforcing $Z \perp\!\!\!\perp S$. The WA and NY datasets contain $76,225$ and $196,967$ samples, respectively, each constructed from 16 different features. We randomly split the data into training (70%), validation (15%), and testing (15%) partitions. Further, we adopt embeddings for categorical features (learned in a supervised fashion by Y) and normalization for continuous/discrete features (by dividing to the maximum value).

CelebA: CelebA dataset [111] contains $202,599$ face images of $10,177$ different celebrities with standard training, validation, and testing splits. Each image is annotated with 40 different attributes. We choose the target attribute Y as the high cheekbone attribute (binary) and the semantic attributes S to be the concatenation of gender and age (a 4-class categorical RV). The objective of this experiment is similar to that of Folktables. Since raw image data is not appropriate for kernel methods, we pre-train a ResNet-18 [124] (supervised by Y) on CelebA images and extract features of dimension 256. These features are used as the input data for all methods.

5.6.3 Evaluation Metrics

We use the accuracy of the classification tasks (16-class classification for Gaussian toy example, employment prediction for Folktables, and high cheekbone prediction for CelebA) as a utility. For Folktables and CelebA datasets, we define DP violation as

$$\text{DPV}(\widehat{Y}, S) := \mathbb{E}_{\widehat{Y}}\left[\mathbb{V}\text{ar}_S\left(\mathbb{P}[\widehat{Y}|\, S]\right)\right] \tag{5.18}$$

and use it as a metric to measure the variance (unfairness) of the prediction $\widehat{Y}$ w.r.t. the semantic attribute S. For the Gaussian toy example, the above metric is challenging to compute because S is a continuous RV. To circumvent this difficulty, we deploy KCC [79]

$$\text{KCC}(Z, S) := \sup_{\alpha\in\mathcal{H}_Z, \beta\in\mathcal{H}_S} \frac{\mathbb{C}\text{ov}(\alpha(Z), \beta(S))}{\sqrt{\mathbb{V}\text{ar}(\alpha(Z))\,\mathbb{V}\text{ar}(\beta(S))}}, \tag{5.19}$$

as a measure of invariance of Z to S, where $\mathcal{H}_Z$ and $\mathcal{H}_S$ are RBF-Gaussian RKHS. The reason for using KCC instead of HSIC is that, unlike HSIC, KCC is normalized, and therefore it is a more readily interpretable measure for comparing the invariance of representations between different

methods.

5.6.4 Choice of (Y, S) Pair

The existence of a utility-invariance trade-off ultimately depends on the statistical dependency between target and semantic attributes. If Dep(Z, S) is negligible, then there does not exist a trade-off. Keeping this in mind, we first chose the semantic attribute to be a sensitive attribute for Folktables (i.e., age) and CelebA (i.e., concatenation of age and gender) datasets. Then, we calculated the data imbalance (i.e., $|\mathbb{P}[Y = 0] - 0.5|$) and KCC$(Y, S)$ for all possible Ys. Finally, we chose Y with a small data imbalance and a moderate KCC(Y, S). For Folktables dataset, $|\mathbb{P}[\text{employment} = 0] - 0.5| = 0.04$ and KCC$(\text{employment}, \text{age}) = 0.4$. For CelebA dataset, $|\mathbb{P}[\text{high cheekbone} = 0] - 0.5| = 0.05$ and KCC$(\text{high cheekbone}, [\text{age}, \text{gender}]) = 0.1$.

5.6.5 Implementation Details

For all methods, we pick different values of λ (100 λs for the Gaussian toy example and 70 λs for Folktables and CelebA datasets) between zero and one for obtaining the utility-invariance trade-off. We train the baselines that use a neural network for encoder five times with different random seeds. We let the random seed also change the training-validation-testing split for the Folktables dataset (CelebA and Gaussian datasets have fixed splits).

Embedding Dimensionality: None of the baseline methods have any strategy to find the optimum embedding dimensionality (r) and they all set r to be constant w.r.t. λ. Therefore, for baseline methods, we set $r = 15$ (i.e., the minimum dimensionality required to linearly classify 16 different categories) for the Gaussian toy example and $r = 3$ (i.e., the minimum dimensionality required to linearly classify 4 different categories) for Folktables-NY dataset, that is also equal to r^{Opt} when

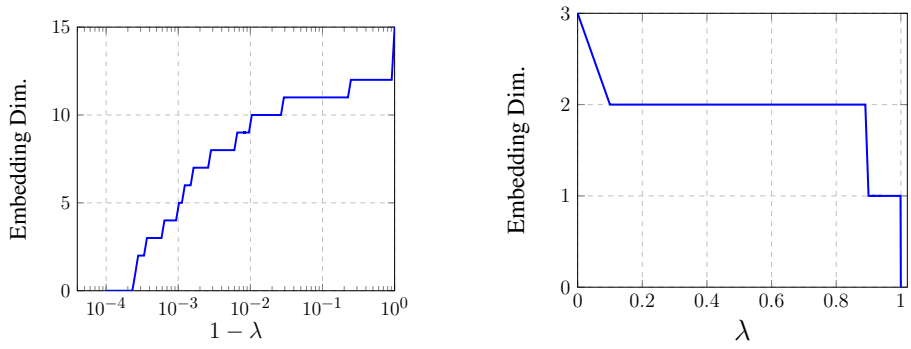


Figure 5.4: Plots of $r^{\text{Opt}}(\lambda)$ versus the dependence trade-off parameter $1-\lambda$ for (a) the Gaussian toy dataset and (b) Folktables-NY dataset. There is a non-decreasing relation between $r^{\text{Opt}}(\lambda)$ and $1-\lambda$.

$\lambda = 0$. For K-$\mathcal{T}_{\text{Opt}}$, we use $r^{\text{Opt}}(\lambda)$ in Corollary 5.1. See Figure 5.4 for the plot of r^{Opt} versus λ for the toy Gaussian and Folktables-NY datasets. For Folktables-WA and CelebA datasets, $r^{\text{Opt}}(\lambda = 0)$ is equal to one, and therefore we let $r = 1$ for all methods and all $0 \leq \lambda < 1$.

K$-\mathcal{T}_{\text{Opt}}$: We let $\mathcal{H}_X$, $\mathcal{H}_S$, and $\mathcal{H}_Y$ be RBF Gaussian RKHS, where we compute the corresponding band-widths (i.e., σs) using the median strategy introduced by [125]. We optimize the regularization parameter γ in the disentanglement set (5.8) by minimizing the corresponding target losses over γs in $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ on validation sets. RFF (as discussed in Section 5.5.1) is deployed for all datasets. For RFF dimensionality, we started with a small value and gradually increased it until reaching the maximum possible performance for $\lambda = 0$ (i.e. the standard unconstrained representation learning) on the corresponding validation sets that results in 100 for the Gaussian dataset, 5000 for Folktables dataset, and 1000 for CelebA dataset.

SARL [49]: SARL method is similar to our K$-\mathcal{T}_{\text{Opt}}$ except that $\mathcal{H}_Y$ and $\mathcal{H}_S$ are linear RKHSs, and therefore we set σ_X and γ similar to that of K$-\mathcal{T}_{\text{Opt}}$.

ARL [24, 22, 2]: The representation $Z = \boldsymbol{f}(X)$ is extracted via the encoder $\boldsymbol{f}$, which is an MLP (4 hidden layers and 15, 15 neurons for Gaussian data; 3 hidden layers and 128, 64 neurons for Folktables and CelebA datasets). These architecture choices were based on starting with a single linear

layer and gradually increasing the number of layers and neurons until over-fitting was observed. This results in the number of encoder parameters for the Gaussian toy example to be 735, while that is $100 = 100 * r^{\text{Opt}}(\lambda \to 1) \leq 100 * r^{\text{Opt}}(\lambda) \leq 100 * r^{\text{Opt}}(\lambda = 0) = 1500$ for K$-\mathcal{T}_{\text{Opt}}$. For Folktables and CelebA, those are $41,024$ and $15,616$, respectively, for ARL and 5000 and 1000 for K$-\mathcal{T}_{\text{Opt}}$. The representation Z is fed to a target task predictor g_Y and a proxy adversary g_S networks where both are MLP with (2 hidden layers, and 16 neurons for Gaussian data, 2 hidden layers, and 128 neurons for Folktables and CelebA datasets). All involved networks ($\boldsymbol{f}, g_Y, g_S$) are trained end-to-end. We use stochastic gradient descent-ascent (SGDA) [24] with AdamW [126] as an optimizer to alternately train the encoder, target predictor, and proxy adversary networks. We choose batch size as 500 for Gaussian data; and 128 for Folktables and CelebA datasets. Then, the corresponding learning rates are optimized over $\{10^{-2}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}, 10^{-5}\}$ by minimizing the target loss on the corresponding validation sets.

HSIC-IRepL [62]: This method can be formulated as (5.2) where $\text{Dep}(Z, S)$ is replaced by $\text{HSIC}(Z, S)$. The encoder and target predictor networks have the same architecture as the ARL. Therefore, we follow the same optimization procedure as ARL to train the involved neural networks.

5.6.6 Results

Utility-Invariance Trade-offs: Figures 5.5 and 5.6 show the utility-invariance and $\text{Dep}(Z, Y) - \text{Dep}(Z, S)$ trade-offs for the toy Gaussian, Folktables-WA, Folktables-NY, and CelebA datasets, respectively. The invariance measure for the Gaussian toy example is KCC (5.19), and the invariance measure for Folktables and CelebA datasets is the fairness measure, DPV (5.18). We make the following observations: 1) K$-\mathcal{T}_{\text{Opt}}$ is highly stable and almost spans the entire trade-off front for all datasets except Folktables-NY which can be due to the inability of scalarized

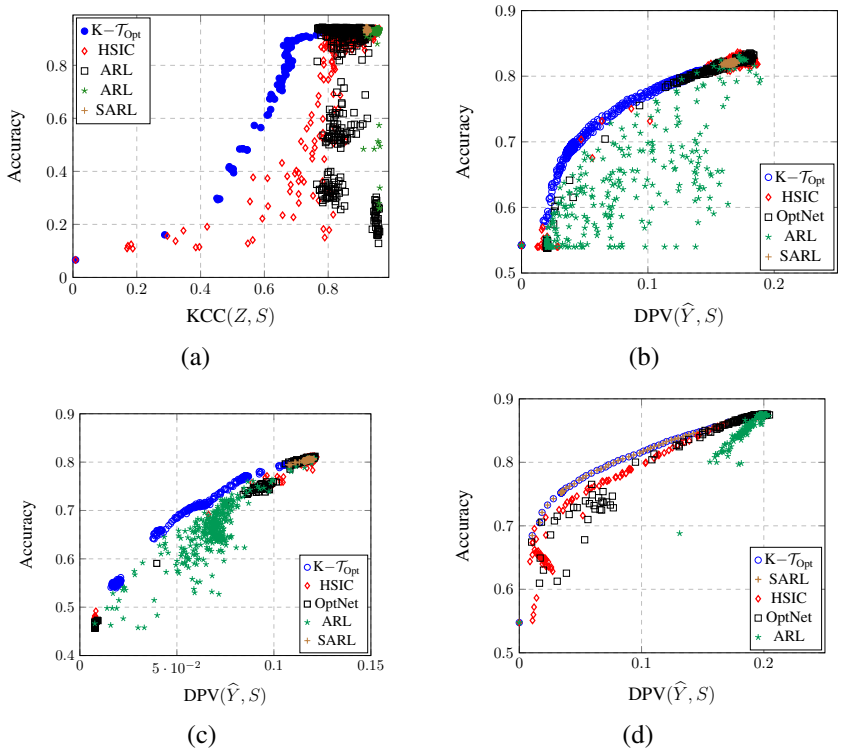


Figure 5.5: Utility versus invariance trade-offs obtained by K$-\mathcal{T}_{\text{Opt}}$ and other baselines for (a) Gaussian, (b) Folktables-WA, (c) Folktables-NY, and (d) CelebA datasets. K-$\mathcal{T}_{\text{Opt}}$ stably spans the entire trade-off front and considerably dominates other methods for all datasets. (a) ARL and SARL span a small portion of the trade-off front since S is mean independent (but not fully independent) of X in some dimensions for the Gaussian toy example. HSIC-IRepL, despite using a universal dependence measure, performs sub-optimally due to the lack of convergence guarantees to the global optima.

single-objective formulation in (5.2), in contrast to the constrained optimization in (5.1), to find all Pareto-optimal points. 2) There is almost the same trend in the trade-off between Dep(Z, Y) and Dep(Z, S) (Figure 5.6) as the utility-invariance trade-off (Figure 5.5). This is a desired observation since Dep(Z, Y)-Dep(Z, S) trade-off is what we actually optimized in (5.9) as a surrogate to utility-invariance trade-off. 3) The baseline method HSIC-IRepL, despite using a universal dependence measure, leads to a sub-optimal trade-off front due to the lack of convergence guarantees to the global optima. 4) The baselines, ARL and SARL span only a small portion of the trade-off front in the Gaussian toy example, since some dimensions of the semantic attribute S in (5.17) are mean independent (but not fully independent) to some dimensions of X and therefore the adversary

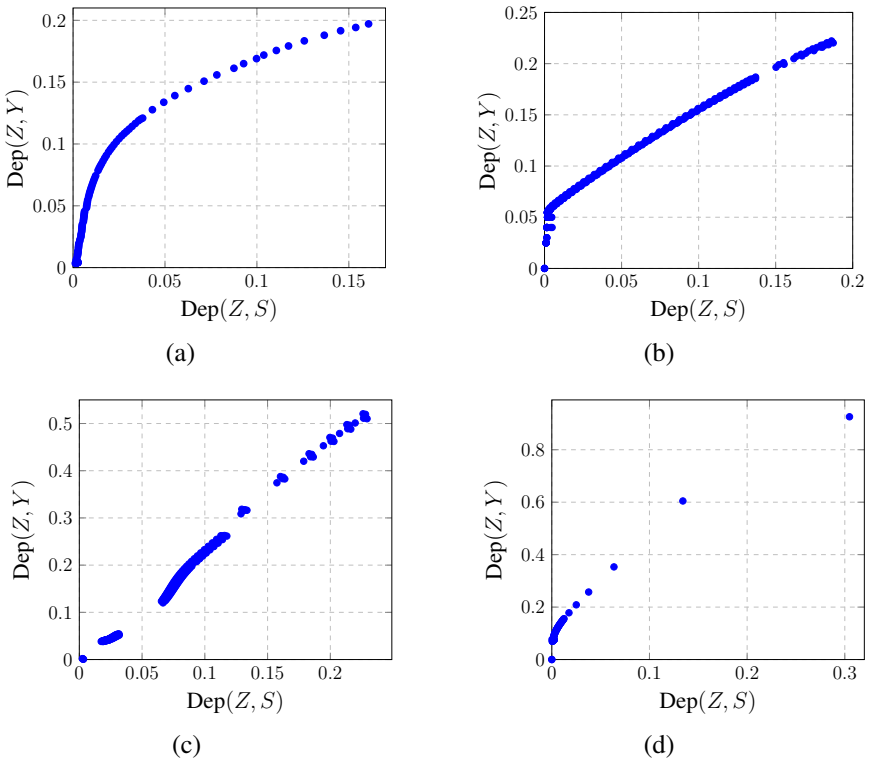


Figure 5.6: Dep(Z, Y) versus Dep(Z, S) in K$-\mathcal{T}_{\text{Opt}}$ for (a) Gaussian, (b) Folktables-WA, (c) Folktables-NY, and (d) CelebA datasets. We can observe that there is the same trend in Dep(Z, Y)-Dep(Z, S) trade-off as utility-invariance-trade-off in Figure 5.5.

does not provide any information to the encoder to discard $[S_1, S_2]$ from the representation. In this dataset, ARL and SARL baselines do not approach $Z \perp\!\!\!\perp S$, i.e., KCC$(Z, S) = 0$ cannot be attained for any value of the trade-off parameter λ. 5) ARL shows high deviation on Folktables dataset due to the unstable nature of the minimax optimization. 6) SARL performs as good as K$-\mathcal{T}_{\text{Opt}}$ for CelebA dataset. This is because both S and Y are categorical for CelebA dataset, and therefore linear RKHS on one-hot encoded attribute performs just as well as universal RKHSs [127].

Universality of Dep(Z, S): We empirically examine the practical validity of our assumption in Section 5.4 and verify if our dependence measure Dep(Z, S), defined in (5.6), can capture all modes of dependency between Z and S. Figure 5.7 (a) shows the plot of the universal dependence measure KCC(Z, S) versus Dep(Z, S) for the Gaussian dataset and Figures 5.7 (b, c) illustrate the relation between DPV$(\widehat{Y}, S)$ and Dep(Z, S) for Folktables and CelebA datasets, respectively. We

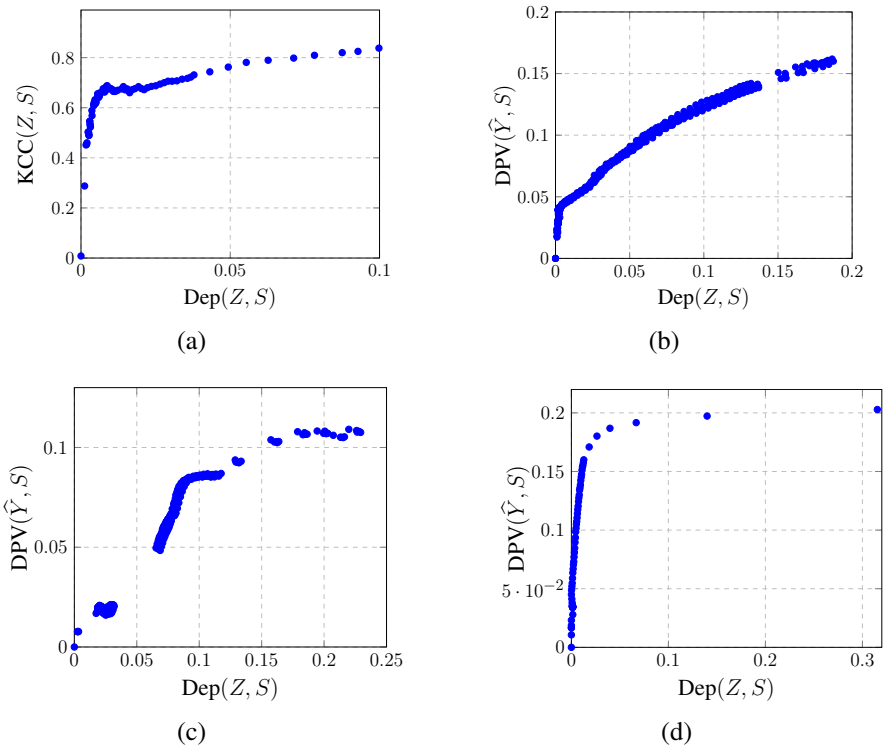
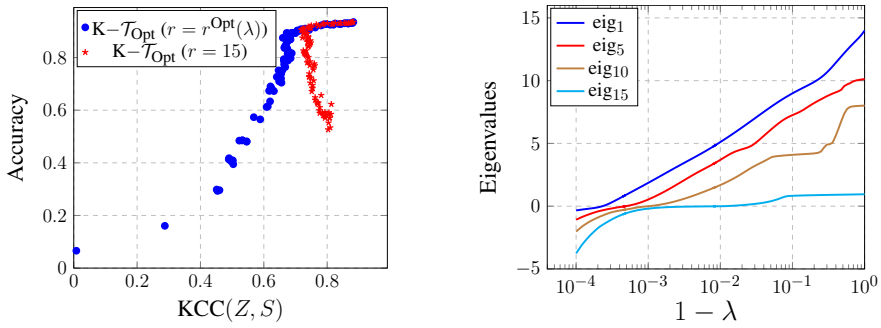


Figure 5.7: Invariance versus Dep(Z,S) of K$-\mathcal{T}_{\text{Opt}}$ for (a) Gaussian, (b) Folktables-WA, (c) Folktables-NY, and (d) CelebA datasets. Dep(Z,S) enjoys a monotonic relation with the underlying invariance measures.

observe that there is a non-decreasing relation between the corresponding invariance measures and Dep(Z,S). More importantly, as KCC$(Z,S)\to 0$ (or DPV$(\widehat{Y},S)\to 0$) so does dep(Z,S). These observations verify that Dep(Z,S) accounts for all modes of dependence between Z and S.

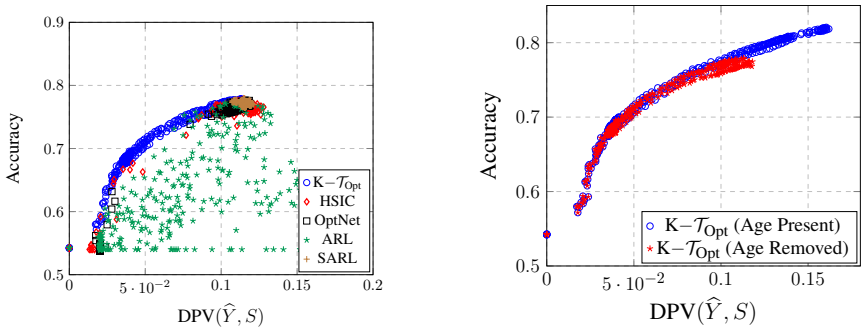
5.6.7 Ablation Study

Effect of Embedding Dimensionality: In this experiment, we examine the significance of the embedding dimensionality, $r^{\text{Opt}}(\lambda)$, discussed in Corollary 5.1. We obtain the utility-invariance trade-off when the embedding dimensionality is fixed to be $r = r^{\text{Opt}}(\lambda = 0) = 15$. A comparison plot between the utility-invariance trade-off induced by $r^{\text{Opt}}(\lambda)$ and the fixed $r = 15$ is illustrated in Figure 5.8 (a). We can observe that not only the utility-invariance trade-off for fixed r is dominated by that of $r^{\text{Opt}}(\lambda)$, but also, using fixed r is unable to achieve the total invariance



(a) (b)

Figure 5.8: (a) Comparison between the utility-invariance trade-offs induced by the optimal embedding dimensionality $r^{\text{Opt}}(\lambda)$ and that of fixed $r = 15$. Fixed $r = 15$ is significantly dominated by that of $r^{\text{Opt}}(\lambda)$ and fails to attain $Z \perp\!\!\!\perp S$. (b) The first, fifth, tenth, and fifteenth largest eigenvalues in (5.12) versus $1 - \lambda$. Given λ, r^{Opt} is equal to the number of non-negative eigenvalues. As $1 - \lambda$ decreases, the largest eigenvalues approach to negative numbers.



(a) (b)

Figure 5.9: (a) Utility versus invariance trade-offs for all methods when age (i.e., the sensitive attribute) is discarded from the input data. (b) A comparison between trade-offs of K−$\mathcal{T}_{\text{Opt}}$ when age is present versus age is discarded from the input data. Removing the age attribute slightly degrades the trade-off due to information discarding.

representation, i.e., $Z \perp\!\!\!\perp S$. Further, $r^{\text{Opt}}(\lambda)$ and some of the largest eigenvalues of (5.12) vs the invariance trade-off parameter λ are plotted in Figures 5.8 (b, c), respectively. We recall from Corollary 5.1 that, for any given λ, r^{Opt} is the number of non-negative eigenvalues of (5.12).

5.7 Summary

Invariant representation learning (IRepL) often involves a trade-off between utility and invariance. While the existence of such trade-off and its bounds have been studied, its *exact* characterization has not been investigated. This chapter takes some steps to address this problem by, i) establishing the *exact* kernelized trade-off (denoted by K$-\mathcal{T}_{\text{Opt}}$), ii) determining the optimal dimensionality of the data representation necessary to achieve a desired optimal trade-off point, and iii) developing a scalable learning algorithm for encoders in some RKHSs to achieve K$-\mathcal{T}_{\text{Opt}}$. Numerical results on both an illustrative example and two real-world datasets show that commonly used adversarial representation learning-based techniques are unable to attain the optimal trade-off estimated by our solution.

Chapter 6

Conclusion and Future Work

Invariant representation learning often introduces a trade-off between utility and invariance. However, a learning algorithm that can optimally achieve any point on the trade-off front is challenging. To circumvent this difficulty, in this dissertation, we propose to model IRepL players via functions in some RKHSs. Consequently, we found a closed-form solution for the underlying IRepL problem. Additionally, the optimal embedding dimensionality is also determined. We also considered the case where the encoder is an NN. In this situation, we model the target network and invariance measure via kernelized ridge regressors that yield, in turn, closed-form solutions for the target network and invariance measure. As a result, the unstable SGDA optimization scheme turns to SGD, which is a stable optimization scheme. Numerical experiments on real-world datasets like the US Census and CelebA confirm the optimality and efficiency of our proposed methods compared to baseline IRepL algorithms. Finally, our theoretical results and empirical solutions shed light on the utility-invariance trade-offs in various settings, such as algorithmic fairness and privacy-preserving learning.

6.1 Limitations

This dissertation restricts the target predictor to be modeled via functions in some RKHSs to afford analytical tractability and theoretical guarantees in Chapters 3, 5. Our proposed IRepL formulation is under the scalarization of the bi-objective trade-off formulation. Even though any solution to the

scalarized version is a solution to the original bi-objective IRepL problem, some regions on the utility-invariance trade-off may not be spotted by the scalarized counterpart. In general, IRepL is a function of the involved dependence measure that quantifies the dependence of learned representations on the sensitive attribute. As such, the trade-off obtained in this dissertation is optimal for HSIC-like dependence measures and may not be optimum for other invariance measures like KCC or COCO.

6.2 Future Work

As a result of the limitations mentioned above, studying the bi-objective trade-off (rather than the scalarization) and other universal measures are interesting directions for future work. Moreover, the invariance criterion deployed in this dissertation is analogous to demographic parity in fairness which can be an unsuitable fairness notion in some practical scenarios [128, 129]. Our method in this dissertation can be extended to other notions of fairness, such as equalized odds and equality of opportunity [128] by modifying the invariance measure to capture the conditional dependency of the representation on the sensitive attribute given the target label.

6.3 Broader Impact

IRepL can enable many machine learning systems to prevent the leakage of private (sensitive) information while being effective for the desired prediction task(s). In particular, IRepL has an immediate application in fairness which is a significant societal problem. Our approach in this dissertation proposes some algorithms that can be used to learn fair representations of data. More generally, these approaches can enable machine learning systems to discard specific data before

making predictions. Of course, as with any technological or algorithmic solutions, one can also employ them for harmful purposes.

BIBLIOGRAPHY

[1] P. Roy and V. N. Boddeti, "Mitigating information leakage in image representations: A maximum entropy approach," *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[2] D. Madras, E. Creager, T. Pitassi, and R. Zemel, "Learning adversarially fair and transferable representations," *arXiv preprint arXiv:1802.06309*, 2018.

[3] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[4] V. Grari, O. E. Hajouji, S. Lamprier, and M. Detyniecki, "Learning unbiased representations via rényi minimization," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 749–764, Springer, 2021.

[5] A. K. Menon and R. C. Williamson, "The cost of fairness in binary classification," *Conference on Fairness, Accountability and Transparency*, pp. 107–118, 2018.

[6] H. Zhao and G. J. Gordon, "Inherent tradeoffs in learning fair representations," *arXiv preprint arXiv:1906.08386*, 2019.

[7] T. L. Gouic, J.-M. Loubes, and P. Rigollet, "Projection to fairness in statistical learning," *arXiv preprint arXiv:2005.11720*, 2020.

[8] H. Zhao, "Costs and benefits of wasserstein fair regression," *arXiv preprint arXiv:2106.08812*, 2021.

[9] H. Zhao, C. Dan, B. Aragam, T. S. Jaakkola, G. J. Gordon, and P. Ravikumar, "Fundamental limits and tradeoffs in invariant representation learning," *arXiv preprint arXiv:2012.10713*, 2020.

[10] H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon, "On learning invariant representations for domain adaptation," *International Conference on Machine Learning*, pp. 7523–7532, 2019.

[11] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *International Conference on Machine Learning*, 2015.

[12] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[14] L. Zhang and X. Gao, "Transfer adaptation learning: A decade survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[15] T. Isobe, D. Li, L. Tian, W. Chen, Y. Shan, and S. Wang, "Towards discriminative representation learning for unsupervised person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8526–8536, 2021.

[16] H. Xia, H. Zhao, and Z. Ding, "Adaptive adversarial network for source-free domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9010–9019, 2021.

[17] X. Liu, Z. Guo, S. Li, F. Xing, J. You, C.-C. J. Kuo, G. El Fakhri, and J. Woo, "Adversarial unsupervised domain adaptation with conditional and label shift: Infer, align and iterate," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10367–10376, 2021.

[18] W. Zhang, X. Li, H. Ma, Z. Luo, and X. Li, "Universal domain adaptation in fault diagnostics with hybrid weighted deep adversarial learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 7957–7967, 2021.

[19] X. Fan, Q. Wang, J. Ke, F. Yang, B. Gong, and M. Zhou, "Adversarially adaptive normalization for single domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8208–8217, 2021.

[20] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui, "Towards out-of-distribution generalization: A survey," *arXiv preprint arXiv:2108.13624*, 2021.

[21] H. Edwards and A. Storkey, "Censoring representations with an adversary," *arXiv preprint arXiv:1511.05897*, 2015.

[22] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," *AAAI/ACM Conference on AI, Ethics, and Society*, 2018.

[23] A. Beutel, J. Chen, Z. Zhao, and E. H. Chi, "Data decisions and theoretical implications when adversarially learning fair representations," *arXiv preprint arXiv:1707.00075*, 2017.

[24] Q. Xie, Z. Dai, Y. Du, E. Hovy, and G. Neubig, "Controllable invariance through adversarial feature learning," *Advances in Neural Information Processing Systems*, pp. 585–596, 2017.

[25] V. Mirjalili, S. Raschka, A. Namboodiri, and A. Ross, "Semi-adversarial networks: Convolutional autoencoders for imparting privacy to face images," in *International Conference on Biometrics*, 2018.

[26] M. Bertran, N. Martinez, A. Papadaki, Q. Qiu, M. Rodrigues, G. Reeves, and G. Sapiro, "Adversarially learned representations for information obfuscation and inference," *International Conference on Machine Learning*, 2019.

[27] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information fusion*, vol. 58, pp. 82–115, 2020.

[28] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[29] S. L. Blodgett, S. Barocas, H. Daumé III, and H. Wallach, "Language (technology) is power: A critical survey of" bias" in nlp," *arXiv preprint arXiv:2005.14050*, 2020.

[30] S. Qian, V. H. Pham, T. Lutellier, Z. Hu, J. Kim, L. Tan, Y. Yu, J. Chen, and S. Shah, "Are my deep learning systems fair? an empirical study of fixed-seed training," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30211–30227, 2021.

[31] S. Ravfogel, M. Twiton, Y. Goldberg, and R. D. Cotterell, "Linear adversarial concept erasure," in *International Conference on Machine Learning*, pp. 18400–18421, PMLR, 2022.

[32] K. Karkkainen and J. Joo, "Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1548–1558, 2021.

[33] M. Du, S. Mukherjee, G. Wang, R. Tang, A. Awadallah, and X. Hu, "Fairness via representation neutralization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12091–12103, 2021.

[34] A. Wang, A. Liu, R. Zhang, A. Kleiman, L. Kim, D. Zhao, I. Shirai, A. Narayanan, and O. Russakovsky, "Revise: A tool for measuring and mitigating bias in visual datasets," *International Journal of Computer Vision*, pp. 1–21, 2022.

[35] L. Wu, L. Chen, P. Shao, R. Hong, X. Wang, and M. Wang, "Learning fair representations for recommendation: A graph-based perspective," in *Proceedings of the Web Conference 2021*, pp. 2198–2208, 2021.

[36] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," *International Conference on Machine Learning*, pp. 325–333, 2013.

[37] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The variational fair autoencoder," *arXiv preprint arXiv:1511.00830*, 2015.

[38] D. Moyer, S. Gao, R. Brekelmans, A. Galstyan, and G. Ver Steeg, "Invariant representations without adversarial training," in *Advances in Neural Information Processing Systems*, pp. 9102–9111, 2018.

[39] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," *Advances in Neural Information Processing Systems*, vol. 31, pp. 8559–8570, 2018.

[40] F. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney, "Optimized pre-processing for discrimination prevention," *Advances in Neural Information Processing Systems*, pp. 3992–4001, 2017.

[41] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," *Innovations in Theoretical Computer Science Conference*, pp. 214–226, 2012.

[42] S. Ruggieri, "Using t-closeness anonymity to control for non-discrimination.," *Transactions on Data Privacy*, vol. 7, no. 2, pp. 99–129, 2014.

[43] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 259–268, 2015.

[44] J. Song, P. Kalluri, A. Grover, S. Zhao, and S. Ermon, "Learning controllable fair representations," *International Conference on Artificial Intelligence and Statistics*, 2019.

[45] E. Creager, D. Madras, J.-H. Jacobsen, M. A. Weis, K. Swersky, T. Pitassi, and R. Zemel, "Flexibly fair representation learning by disentanglement," *International Conference on Machine Learning*, 2019.

[46] F. Locatello, G. Abbati, T. Rainforth, S. Bauer, B. Schölkopf, and O. Bachem, "On the fairness of disentangled representations," *Advances in Neural Information Processing Systems*, pp. 14611–14624, 2019.

[47] J. Mary, C. Calauzenes, and N. El Karoui, "Fairness-aware learning for continuous attributes and treatments," *International Conference on Machine Learning*, pp. 4382–4391, 2019.

[48] N. Martinez, M. Bertran, and G. Sapiro, "Minimax pareto fairness: A multi objective perspective," *International Conference on Machine Learning*, pp. 6755–6764, 2020.

[49] B. Sadeghi, R. Yu, and V. Boddeti, "On the global optima of kernelized adversarial representation learning," *IEEE International Conference on Computer Vision*, pp. 7971–7979, 2019.

[50] H. Zhang, Y.-F. Zhang, W. Liu, A. Weller, B. Schölkopf, and E. P. Xing, "Towards principled disentanglement for domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8024–8034, 2022.

[51] J. Hamm, "Minimax filter: Learning to preserve privacy from inference attacks," *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 4704–4734, 2017.

[52] M. Coavoux, S. Narayan, and S. B. Cohen, "Privacy-preserving neural representations of text," *arXiv preprint arXiv:1808.09408*, 2018.

[53] T. Xiao, Y.-H. Tsai, K. Sohn, M. Chandraker, and M.-H. Yang, "Adversarial learning of privacy-preserving and task-oriented representations," *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 12434–12441, 2020.

[54] M. Dusmanu, J. L. Schönberger, S. N. Sinha, and M. Pollefeys, "Privacy-preserving visual feature descriptors through adversarial affine subspace embedding," *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[55] D. McNamara, C. S. Ong, and R. C. Williamson, "Costs and benefits of fair representation learning," *AAAI/ACM Conference on AI, Ethics, and Society*, pp. 263–270, 2019.

[56] E. Adeli, Q. Zhao, A. Pfefferbaum, E. V. Sullivan, L. Fei-Fei, J. C. Niebles, and K. M. Pohl, "Representation learning with statistical independence to mitigate bias," *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2513–2523, 2021.

[57] C. Agarwal, H. Lakkaraju, and M. Zitnik, "Towards a unified framework for fair and stable graph representation learning," in *Uncertainty in Artificial Intelligence*, pp. 2114–2124, PMLR, 2021.

[58] K. Yang, J. H. Yau, L. Fei-Fei, J. Deng, and O. Russakovsky, "A study of face obfuscation in imagenet," in *International Conference on Machine Learning*, pp. 25313–25330, PMLR, 2022.

[59] B. Rodríguez-Gálvez, R. Thobaben, and M. Skoglund, "A variational approach to privacy and fairness," in *2021 IEEE Information Theory Workshop (ITW)*, pp. 1–6, IEEE, 2021.

[60] H. Zhao, J. Chi, Y. Tian, and G. J. Gordon, "Trade-offs and guarantees of adversarial representation learning for information obfuscation," *arXiv preprint arXiv:1906.07902*, 2019.

[61] S. Dutta, D. Wei, H. Yueksel, P.-Y. Chen, S. Liu, and K. Varshney, "Is there a trade-off between fairness and accuracy? A perspective using mismatched hypothesis testing," *International Conference on Machine Learning*, pp. 2803–2813, 2020.

[62] N. Quadrianto, V. Sharmanska, and O. Thomas, "Discovering fair representations in the data domain," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8227–8236, 2019.

[63] A. Letcher, D. Balduzzi, S. Racaniere, J. Martens, J. N. Foerster, K. Tuyls, and T. Graepel, "Differentiable game mechanics.," *Journal of Machine Learning Research*, vol. 20, no. 84, pp. 1–40, 2019.

[64] L. Mescheder, S. Nowozin, and A. Geiger, "The numerics of gans," in *Advances in Neural Information Processing Systems*, pp. 1825–1835, 2017.

[65] V. Nagarajan and J. Z. Kolter, "Gradient descent gan optimization is locally stable," in *Advances in Neural Information Processing Systems*, pp. 5585–5595, 2017.

[66] D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls, and T. Graepel, "The mechanics of n-player differentiable games," in *International Conference on Machine Learning*, 2018.

[67] C. Daskalakis and I. Panageas, "The limit points of (optimistic) gradient descent in min-max optimization," in *Advances in Neural Information Processing Systems*, 2018.

[68] C. Jin, P. Netrapalli, and M. I. Jordan, "What is local optimality in nonconvex-nonconcave minimax optimization?," *arXiv preprint arXiv:1902.00618*, 2019.

[69] G. Gidel, H. Berard, G. Vignoud, P. Vincent, and S. Lacoste-Julien, "A variational inequality perspective on generative adversarial networks," *International Conference on Learning Representations*, 2019.

[70] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," *SIAM Journal on Scientific Computing*, vol. 43, no. 5, pp. A3055–A3081, 2021.

[71] N. Loizou, H. Berard, G. Gidel, I. Mitliagkas, and S. Lacoste-Julien, "Stochastic gradient descent-ascent and consensus optimization for smooth games: Convergence analysis under expected co-coercivity," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19095–19108, 2021.

[72] G. Korpelevich, "The extragradient method for finding saddle points and other problems," *Matecon*, vol. 12, pp. 747–756, 1976.

[73] H. K. Khalil, "Nonlinear systems," *Printice-Hall Inc*, 1996.

[74] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[75] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[76] S. Baharlouei, M. Nouiehed, A. Beirami, and M. Razaviyayn, "R'enyi fair inference," *arXiv preprint arXiv:1906.12005*, 2019.

[77] A. Pérez-Suay, V. Laparra, G. Mateo-García, J. Muñoz-Marí, L. Gómez-Chova, and G. Camps-Valls, "Fair kernel learning," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 339–355, 2017.

[78] A. Rényi, "On measures of dependence," *Acta Mathematica Academiae Scientiarum Hungarica*, vol. 10, no. 3-4, pp. 441–451, 1959.

[79] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 1–48, 2002.

[80] A. Gretton, A. Smola, O. Bousquet, R. Herbrich, B. Schölkopf, and N. Logothetis, "Behaviour and convergence of the constrained covariance," *Max Planck Institute for Biological Cybernetics*, 2004.

[81] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," *International Conference on Algorithmic Learning Theory*, pp. 63–77, 2005.

[82] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[83] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf, "Kernel methods for measuring independence," *Journal of Machine Learning Research*, vol. 6, no. 12, pp. 2075–2129, 2005.

[84] B. K. Sriperumbudur, K. Fukumizu, and G. R. Lanckriet, "Universality, characteristic kernels and RKHS embedding of measures," *Journal of Machine Learning Research*, vol. 12, no. 7, 2011.

[85] K. Fukumizu, F. R. Bach, and A. Gretton, "Statistical consistency of kernel canonical correlation analysis.," *Journal of Machine Learning Research*, vol. 8, no. 2, 2007.

[86] E. Kreyszig, *Introductory functional analysis with applications*, vol. 1. wiley New York, 1978.

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

[88] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[89] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[90] E. Kokiopoulou, J. Chen, and Y. Saad, "Trace optimization and eigenproblems in dimension reduction methods," *Numerical Linear Algebra with Applications*, vol. 18, no. 3, pp. 565–602, 2011.

[91] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[92] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling methods for the Nyström method," *Journal of Machine Learning Research*, vol. 13, no. Apr, pp. 981–1006, 2012.

[93] D. Dua and C. Graff, "UCI machine learning repository," 2017.

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

[95] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 643–660, 2001.

[96] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," tech. rep., Citeseer, 2009.

[97] C. Ionescu, O. Vantzos, and C. Sminchisescu, "Training deep networks with structured layers by matrix backpropagation," *arXiv preprint arXiv:1509.07838*, 2015.

[98] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[99] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *International Conference on Machine Learning*, 2017.

[100] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," in *Advances in Neural Information Processing Systems*, 2019.

[101] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *International Conference on Learning Representations*, 2018.

[102] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[103] Y. Elazar and Y. Goldberg, "Adversarial removal of demographic attributes from text data," *Empirical Methods in Natural Language Processing*, 2018.

[104] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Advances in Neural Information Processing Systems*, 2018.

[105] G. H. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973.

[106] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," in *International Conference on Machine Learning*, pp. 1225–1234, PMLR, 2016.

[107] D. Dua and C. Graff, "Uci machine learning repository (2017)," *URL http://archive. ics. uci. edu/ml*, 2017.

[108] B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim, "Learning not to learn: Training deep neural networks with biased data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[109] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *International Conference on Parallel Problem Solving from Nature*, pp. 292–301, 1998.

[110] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[111] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," *IEEE International Conference on Computer Vision*, 2015.

[112] J. Knowles, "A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers," in *International Conference on Intelligent Systems Design and Applications (ISDA)*, 2005.

[113] C. R. Souza, "Kernel functions for machine learning applications," *Creative Commons Attribution-Noncommercial-Share Alike*, vol. 3, p. 29, 2010.

[114] V. Grari, O. E. Hajouji, S. Lamprier, and M. Detyniecki, "Learning unbiased representations via Rényi minimization," *arXiv preprint arXiv:2009.03183*, 2020.

[115] A. Rahimi, B. Recht, *et al.*, "Random features for large-scale kernel machines.," *Advances in Neural Information Processing Systems*, vol. 3, no. 4, p. 5, 2007.

[116] F. Ding, M. Hardt, J. Miller, and L. Schmidt, "Retiring adult: New datasets for fair machine learning," *arXiv preprint arXiv:2108.04884*, 2021.

[117] J. Jacod and P. Protter, *Probability essentials*. Springer Science & Business Media, 2012.

[118] E. Barshan, A. Ghodsi, Z. Azimifar, and M. Z. Jahromi, "Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds," *Pattern Recognition*, vol. 44, no. 7, pp. 1357–1371, 2011.

[119] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," *International Conference on Machine Learning*, pp. 531–540, 2018.

[120] P. Diaconis and D. Freedman, "Asymptotics of graphical projection pursuit," *The Annals of Statistics*, pp. 793–815, 1984.

[121] P. Hall and K.-C. Li, "On almost linearity of low dimensional projections from high dimensional data," *The Annals of Statistics*, pp. 867–889, 1993.

[122] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[123] R. L. Strawderman, "The symmetric eigenvalue problem (classics in applied mathematics, number 20)," *Journal of the American Statistical Association*, vol. 94, no. 446, p. 657, 1999.

[124] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[125] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola, "A kernel statistical test of independence," *Advances in Neural Information Processing Systems*, vol. 20, pp. 585–592, 2007.

[126] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[127] Y. Li, R. Pogodin, D. J. Sutherland, and A. Gretton, "Self-supervised learning with kernel dependence maximization," *arXiv preprint arXiv:2106.08320*, 2021.

[128] M. Hardt, E. Price, N. Srebro, *et al.*, "Equality of opportunity in supervised learning," in *Advances in Neural Information Processing Systems*, pp. 3315–3323, 2016.

[129] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. big data 5, 2 (2017), 153–163," *arXiv preprint arXiv:1610.07524*, 2017.

[130] J. Adebayo and L. Kagal, "Iterative orthogonal feature projection for diagnosing bias in black-box models," *Fairness, Accountability, and Transparency in Machine Learning*, 2016.

[131] P. Adler, C. Falk, S. A. Friedler, T. Nix, G. Rybeck, C. Scheidegger, B. Smith, and S. Venkatasubramanian, "Auditing black-box models for indirect influence," *Knowledge and Information Systems*, vol. 54, no. 1, pp. 95–122, 2018.

[132] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Advances in Neural Information Processing Systems*, pp. 4349–4357, 2016.

[133] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[134] B. Fish, J. Kun, and A. D. Lelkes, "Fair boosting: a case study," in *Workshop on Fairness, Accountability, and Transparency in Machine Learning*, Citeseer, 2015.

[135] G. Goh, A. Cotter, M. Gupta, and M. P. Friedlander, "Satisfying real-world goals with dataset constraints," in *Advances in Neural Information Processing Systems*, pp. 2415–2423, 2016.

[136] F. Kamiran and T. Calders, "Classification with no discrimination by preferential sampling," in *Proc. 19th Machine Learning Conf. Belgium and The Netherlands*, vol. 1, Citeseer, 2010.

[137] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 35–50, Springer, 2012.

[138] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Enhancement of the neutrality in recommendation.," in *Decisions@ RecSys*, pp. 8–14, Citeseer, 2012.

[139] T. Kamishima, S. Akaho, H. Asoh, and I. Sato, "Model-based approaches for independence-enhanced recommendation," in *International Conference on Data Mining Workshops*, pp. 860–867, IEEE, 2016.

[140] E. Kazemi, M. Zadimoghaddam, and A. Karbasi, "Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints," in *International Conference on Machine Learning*, pp. 2549–2558, 2018.

[141] D. H. Kim, T. Kong, and S. Jeong, "Finding solutions to generative adversarial privacy," *arXiv preprint arXiv:1810.02069*, 2018.

[142] J. Komiyama, A. Takeda, J. Honda, and H. Shimao, "Nonconvex optimization for regression with fairness constraints," in *International Conference on Machine Learning*, pp. 2742–2751, 2018.

[143] A. J. Laub, *Matrix analysis for scientists and engineers*, vol. 91. Siam, 2005.

[144] G. Meanti, L. Carratino, L. Rosasco, and A. Rudi, "Kernel methods through the roof: handling billions of points efficiently," *arXiv preprint arXiv:2006.10350*, 2020.

[145] G. Ristanoski, W. Liu, and J. Bailey, "Discrimination aware classification for imbalanced datasets," in *ACM International Conference on Information & Knowledge Management*, pp. 1529–1532, ACM, 2013.

[146] S. Samadi, U. Tantipongpipat, J. H. Morgenstern, M. Singh, and S. Vempala, "The price of fair pca: One extra dimension," in *Advances in Neural Information Processing Systems*, pp. 10999–11010, 2018.

[147] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *International Conference on World Wide Web*, pp. 1171–1180, International World Wide Web Conferences Steering Committee, 2017.

[148] I. Žliobaite, F. Kamiran, and T. Calders, "Handling conditional discrimination," in *International Conference on Data Mining*, pp. 992–1001, IEEE, 2011.

[149] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*, pp. 630–645, Springer, 2016.

[150] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[151] Y. Li, K. Swersky, and R. Zemel, "Learning unbiased features," *arXiv preprint arXiv:1412.5244*, 2014.

[152] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *International Conference on Computational Learning Theory*, 2001.

[153] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *Journal of Machine Learning Research*, vol. 7, no. Dec, pp. 2651–2667, 2006.

[154] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces," *Journal of Machine Learning Research*, vol. 5, no. Jan, pp. 73–99, 2004.

[155] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias," *ProPublica, May*, vol. 23, p. 2016, 2016.

[156] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on Fairness, Accountability and Transparency*, pp. 77–91, 2018.

[157] A. Datta, M. C. Tschantz, and A. Datta, "Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 1, pp. 92–112, 2015.

[158] L. Sweeney, "Discrimination in online ad delivery," *Queue*, vol. 11, no. 3, pp. 10–29, 2013.

[159] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[160] S. Verma and J. Rubin, "Fairness definitions explained," in *IEEE/ACM International Workshop on Software Fairness (FairWare)*, pp. 1–7, IEEE, 2018.

[161] T. Kamishima, S. Akaho, and J. Sakuma, "Fairness-aware learning through regularization approach," in *International Conference on Data Mining Workshops*, pp. 643–650, IEEE, 2011.

[162] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *International Conference on World Wide Web*, pp. 1171–1180, 2017.

[163] Y. Bechavod and K. Ligett, "Penalizing unfairness in binary classification," *arXiv preprint arXiv:1707.00044*, 2017.

[164] R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth, "A convex framework for fair regression," *arXiv preprint arXiv:1706.02409*, 2017.

[165] M. Kearns, S. Neel, A. Roth, and Z. S. Wu, "Preventing fairness gerrymandering: Auditing and learning for subgroup fairness," in *International Conference on Machine Learning*, pp. 2564–2572, 2018.

[166] B. Fish, J. Kun, and Á. D. Lelkes, "A confidence-based approach for balancing fairness and accuracy," in *International Conference on Data Mining*, pp. 144–152, SIAM, 2016.

[167] D. Alabi, N. Immorlica, and A. T. Kalai, "Unleashing linear optimizers for group-fair learning and optimization," *arXiv preprint arXiv:1804.04503*, 2018.

[168] L. E. Celis, L. Huang, V. Keswani, and N. K. Vishnoi, "Classification with fairness constraints: A meta-algorithm with provable guarantees," in *Conference on Fairness, Accountability, and Transparency*, pp. 319–328, 2019.

[169] T. Calders, F. Kamiran, and M. Pechenizkiy, "Building classifiers with independency constraints," in *IEEE International Conference on Data Mining Workshops*, pp. 13–18, 2009.

[170] F. Kamiran and T. Calders, "Classifying without discriminating," in *International Conference on Computer, Control and Communication*, pp. 1–6, 2009.

[171] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowledge and Information Systems*, vol. 33, no. 1, pp. 1–33, 2012.

[172] M. Wick, J.-B. Tristan, *et al.*, "Unlocking fairness: a trade-off revisited," *Advances in neural information processing systems*, vol. 32, 2019.

[173] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The Collected Works of Wassily Hoeffding*, pp. 409–426, Springer, 1994.

[174] J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf, "Distinguishing cause from effect using observational data: methods and benchmarks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1103–1204, 2016.

[175] D. Greenfeld and U. Shalit, "Robust learning with the hilbert-schmidt independence criterion," in *International Conference on Machine Learning*, pp. 3759–3768, PMLR, 2020.

[176] S. Yu, F. Alesiani, X. Yu, R. Jenssen, and J. C. Principe, "Measuring dependence with matrix-based entropy functional," *arXiv preprint arXiv:2101.10160*, 2021.

[177] B. Sadeghi, L. Wang, and V. N. Boddeti, "Adversarial representation learning with closed-form solvers," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 731–748, Springer, 2021.

[178] C. Frogner, C. Zhang, H. Mobahi, M. Araya-Polo, and T. Poggio, "Learning with a wasserstein loss," *arXiv preprint arXiv:1506.05439*, 2015.

[179] L. G. S. Giraldo, M. Rao, and J. C. Principe, "Measures of entropy from data using infinitely divisible kernels," *IEEE Transactions on Information Theory*, vol. 61, no. 1, pp. 535–548, 2014.

[180] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," *Advances in Neural Information Processing Systems*, vol. 19, 2006.

[181] J. Song, P. Kalluri, A. Grover, S. Zhao, and S. Ermon, "Learning controllable fair representations," *International Conference on Artificial Intelligence and Statistics*, 2019.

[182] B. Sadeghi and V. N. Boddeti, "Imparting fairness to pre-trained biased representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2020.

APPENDIX

A.1 Proof of Lemma 3.1

Lemma. Let X and U be two RVs with $\mathbb{E}[X] = 0$, $\mathbb{E}[U] = \boldsymbol{b}$, where $\boldsymbol{C}_X \succ 0$. Consider a linear regressor, $\widehat{U} = \boldsymbol{W}Z + \boldsymbol{b}$, where $\boldsymbol{W} \in \mathbb{R}^{d_U \times r}$ is the parameter matrix, and $Z \in \mathbb{R}^r$ is an encoded version of X for a given $\boldsymbol{\Theta}$: $X \mapsto Z = \boldsymbol{\Theta}X, \quad \boldsymbol{\Theta} \in \mathbb{R}^{r \times d_X}$. The minimum MSE that can be achieved by designing $\boldsymbol{W}$ is

$$\min_{\boldsymbol{W}} \mathbb{E}\left[\|U - \widehat{U}\|^2\right] = \text{Tr}\left[\boldsymbol{C}_U\right] - \left\|P_{\mathcal{M}}\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2,$$

where $\boldsymbol{M} = \boldsymbol{Q}_X\boldsymbol{\Theta}^T \in \mathbb{R}^{d_X \times r}$, and $\boldsymbol{C}_X = \boldsymbol{Q}_X^T\boldsymbol{Q}_X$ (Cholesky factorization).

Proof. Direct calculation yields:

$$\begin{aligned}
J_U &:= \mathbb{E}\left\{\left\|U - \widehat{U}\right\|^2\right\} \\
&= \text{Tr}\left[\mathbb{E}\left\{(U - \boldsymbol{b} - \boldsymbol{W}Z)(U - \boldsymbol{b} - \boldsymbol{W}Z)^T\right\}\right] \\
&= \text{Tr}\Big[\mathbb{E}\Big\{(U - \boldsymbol{b})(U - \boldsymbol{b})^T + (\boldsymbol{W}\boldsymbol{\Theta}X)(\boldsymbol{W}\boldsymbol{\Theta}X)^T \\
&\qquad\qquad - (U - \boldsymbol{b})(\boldsymbol{W}\boldsymbol{\Theta}X)^T - (\boldsymbol{W}\boldsymbol{\Theta}X)(U - \boldsymbol{b})^T\Big\}\Big] \\
&= \text{Tr}\left[\boldsymbol{C}_U + (\boldsymbol{W}\boldsymbol{\Theta})\boldsymbol{C}_X(\boldsymbol{W}\boldsymbol{\Theta})^T - \boldsymbol{C}_{UX}(\boldsymbol{W}\boldsymbol{\Theta})^T - (\boldsymbol{W}\boldsymbol{\Theta})\boldsymbol{C}_{UX}^T\right] \\
&= \text{Tr}\left[\boldsymbol{C}_U + (\boldsymbol{W}\boldsymbol{\Theta}\boldsymbol{Q}_X^T)(\boldsymbol{W}\boldsymbol{\Theta}\boldsymbol{Q}_X^T)^T - \boldsymbol{C}_{UX}(\boldsymbol{W}\boldsymbol{\Theta})^T - (\boldsymbol{W}\boldsymbol{\Theta})\boldsymbol{C}_{UX}^T\right] \\
&= \text{Tr}\Big[(\boldsymbol{W}\boldsymbol{\Theta}\boldsymbol{Q}_X^T - \boldsymbol{C}_{UX}\boldsymbol{Q}_X^{-1})(\boldsymbol{W}\boldsymbol{\Theta}\boldsymbol{Q}_X^T - \boldsymbol{C}_{UX}\boldsymbol{Q}_X^{-1})^T \\
&\qquad\qquad + \boldsymbol{C}_U - (\boldsymbol{C}_{UX}\boldsymbol{Q}_X^{-1})(\boldsymbol{C}_{UX}\boldsymbol{Q}_X^{-1})^T\Big] \\
&= \left\|\boldsymbol{Q}_X\boldsymbol{\Theta}\boldsymbol{W}^T - \boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XY}\right\|_F^2 - \left\|\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2 + \text{Tr}[\boldsymbol{C}_U]
\end{aligned}$$

Hence, the minimizer of J_U is obtained by minimizing the first term in the last equation, which is a standard least square error problem. Let $\boldsymbol{M} = \boldsymbol{Q}_X\boldsymbol{\Theta}$, then the minimizer is given by

$$\boldsymbol{W}^T = \boldsymbol{M}^\dagger\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}.$$

Finally, Using the orthogonal decompositions

$$\left\|\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2 = \left\|P_{\mathcal{M}}\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2 + \left\|P_{\mathcal{M}^\perp}\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2$$

and

$$\begin{aligned}\left\|\boldsymbol{Q}_X\boldsymbol{\Theta}\boldsymbol{W}^T - \boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2 &= \left\|\boldsymbol{M}\boldsymbol{W}^T - P_{\mathcal{M}}\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2 + \left\|P_{\mathcal{M}^\perp}\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2 \\ &= \left\|\underbrace{\boldsymbol{M}\boldsymbol{M}^\dagger}_{P_{\mathcal{M}}}\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU} - P_{\mathcal{M}}\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2 \\ &\qquad + \left\|P_{\mathcal{M}^\perp}\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2 \\ &= \left\|P_{\mathcal{M}^\perp}\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2,\end{aligned}$$

we obtain the minimum value of J_U as

$$\mathrm{Tr}\left[\boldsymbol{C}_U\right] - \left\|P_{\mathcal{M}}\boldsymbol{Q}_X^{-T}\boldsymbol{C}_{XU}\right\|_F^2.$$

□

A.2 Relation Between Constrained Optimization Problem in (3.7) and Its Scalarization in (3.8)

Consider the optimization problem in (3.7)

$$\boldsymbol{G}_\alpha = \arg\min_{\boldsymbol{G}} J_Y(\boldsymbol{G}), \quad \text{s.t.} \quad J_S(\boldsymbol{G}) \geq \alpha. \tag{1}$$

and the optimization problem in (3.8)

$$\boldsymbol{G}_\lambda = \arg\min_{\boldsymbol{G}} J_\lambda(\boldsymbol{G}) \tag{2}$$

where

$$J_{\lambda(\boldsymbol{G})} = (1-\lambda)\, J_Y(\boldsymbol{G}) - \lambda\, J_S(\boldsymbol{G}), \quad \lambda \in [0,1).$$

Claim For each $\lambda \in [0,1)$, solution $\boldsymbol{G}_\lambda$ of (2) is also a solution of (1) with

$$\alpha = J_S(\boldsymbol{G}_\lambda). \tag{3}$$

Proof. Let us consider (1) while assuming that (2) is satisfied. For each λ and corresponding solution $\boldsymbol{G}_\lambda$, let α be given as in (3). For an arbitrary $\boldsymbol{G}$ satisfying $J_S(\boldsymbol{G}) \geq \alpha$, we have

$$\begin{aligned}(1-\lambda)\, J_Y(\boldsymbol{G}_\lambda) - \lambda\,\alpha &= (1-\lambda)\, J_Y(\boldsymbol{G}_\lambda) - \lambda\, J_S(\boldsymbol{G}_\lambda) \\ &\leq (1-\lambda)\, J_Y(\boldsymbol{G}) - \lambda\, J_S(\boldsymbol{G}),\end{aligned} \tag{4}$$

where the second step is from the assumption that (2) is satisfied. Consequently, we have

$$(1-\lambda)\, \left[J_Y(\boldsymbol{G}) - J_Y(\boldsymbol{G}_\lambda)\right] \geq \lambda\, \left[J_S(\boldsymbol{G}) - \alpha\right] \geq 0. \tag{5}$$

Since $J_S(\boldsymbol{G}) \geq \alpha$, it follows that $J_Y(\boldsymbol{G}) \geq J_Y(\boldsymbol{G}_\lambda)$ and consequently $\boldsymbol{G}_\lambda$ is a possible minimizer of problem (1). □

A.3 Proof of Theorem 3.2

Theorem. As a function of $\boldsymbol{G} \in \mathbb{R}^{d_X \times r}$, the objective function in equation (3.8) is neither convex nor differentiable.

Proof. Recall that $P_{\mathcal{G}}$ is equal to $\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{G})^\dagger\boldsymbol{G}^T$. Therefore, due to the involvement of the pseudo inverse, (3.8) is not differentiable (see [105]).

For non-convexity consider the theorem that $f(\boldsymbol{G})$ is convex in $\boldsymbol{G} \in \mathbb{R}^{d_X \times r}$ if and only if $h(t) = f(t\,\boldsymbol{G}_1 + \boldsymbol{G}_2)$ is convex in $t \in \mathbb{R}$ for any constants $\boldsymbol{G}_1,\, \boldsymbol{G}_2 \in \mathbb{R}^{d_X \times r}$ (see [133]).

In order to use the above theorem, consider rank one matrices

$$\boldsymbol{G}_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{G}_2 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & 0 \end{bmatrix}.$$

Define $\boldsymbol{G} = (t\,\boldsymbol{G}_1 + \boldsymbol{G}_2)$. Then

$$P_{\mathcal{G}}(t) = \boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{G})^\dagger\boldsymbol{G}^T = \frac{1}{(t+1)^2+1} \begin{bmatrix} (t+1)^2 & (t+1) & 0 & \dots & 0 \\ (t+1) & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

Using basic properties of trace we get

$$(1-\lambda)\, J_Y(\boldsymbol{G}) - \lambda\, J_S(\boldsymbol{G}) = \text{Tr}\left[P_{\mathcal{G}}(t)\boldsymbol{B}\right],$$

where the matrix $\boldsymbol{B}$ is given in (3.11) and we used Lemma 3.1. Now, represent $\boldsymbol{B}$ as

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1d} \\ b_{12} & b_{22} & \dots & b_{2d} \\ \vdots & \vdots & \ddots & \\ b_{1d} & b_{2d} & \dots & b_{dd} \end{bmatrix}.$$

Thus,

$$\text{Tr}\left[P_{\mathcal{G}}(t)\boldsymbol{B}\right] = b_{11} + \frac{2b_{12}(t+1) + b_{22} - b_{11}}{(t+1)^2+1}.$$

It can be shown that the above function of t is convex only if $b_{12} = 0$ and $b_{11} = b_{22}$. On the other hand, if these two conditions hold, it can be similarly shown that $(1-\lambda)\, J_Y(\boldsymbol{G}) - \lambda\, J_S(\boldsymbol{G})$ is non-convex by considering a different pair of matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$. This implies that $(1-\lambda)\, J_Y(\boldsymbol{G}) - \lambda\, J_S(\boldsymbol{G})$ is not convex. □

A.4 Proof of Theorem 3.3

Theorem. Assume that the number of negative eigenvalues (β) of $\boldsymbol{B}$ in (3.11) is j. Denote $\gamma = \min\{r, j\}$. Then, the minimum value in (3.9) is given as

$$\beta_1 + \beta_2 + \cdots + \beta_\gamma$$

where $\beta_1 \leq \beta_2 \leq \ldots \leq \beta_\gamma < 0$ are the γ smallest eigenvalues of $\boldsymbol{B}$. And the minimum can be attained by $\boldsymbol{G} = \boldsymbol{V}$, where the columns of $\boldsymbol{V}$ are eigenvectors corresponding to all the γ negative eigenvalues of $\boldsymbol{B}$.

Proof. Consider the inner optimization problem of (3.10) in (3.9). Using the trace optimization problems and their solutions in [90], we get

$$\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}_i} J_\lambda(\boldsymbol{G}) = \min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}_i} \text{Tr}\Big[\boldsymbol{G}^T\boldsymbol{B}\boldsymbol{G}\Big] = \beta_1 + \beta_2 + \cdots + \beta_i,$$

where $\beta_1, \beta_2, \ldots, \beta_i$ are i smallest eigenvalues of $\boldsymbol{B}$ and minimum value can be achieved by the matrix $\boldsymbol{V}$ whose columns are corresponding eigenvectors. If the number of negative eigenvalues of $\boldsymbol{B}$ is less than r, then the optimum i in (3.9) is j, otherwise the optimum i is r. □

A.5 Non-Linear Extension Through Kernelization

We assume that X is non-linearly mapped to $\phi_X(X)$ as illustrated in Figure 3.5. Recall from (2.6) that

$$Z = \boldsymbol{\Theta}\left[k_X(\boldsymbol{x}_1, X), \cdots, k_X(\boldsymbol{x}_n, X)\right]^T.$$

For a given fixed $\boldsymbol{\Theta}$, we have

$$J_Y = \min_{\boldsymbol{W}_Y, \boldsymbol{b}_Y} \text{MSE}\,(\widehat{Y} - Y).$$

Note that the above optimization problem can be separated over $\boldsymbol{W}_Y$, $\boldsymbol{b}_Y$. Therefore, for a given $\boldsymbol{W}_Y$, we first minimize over $\boldsymbol{b}_Y$:

$$\begin{aligned}
&\min_{\boldsymbol{b}_Y} \mathbb{E}\left\{\|\boldsymbol{W}_Y Z + \boldsymbol{b}_Y - Y\|^2\right\} \\
&= \min_{\boldsymbol{b}_Y} \frac{1}{n}\sum_{k=1}^{n} \|\boldsymbol{W}_Y \boldsymbol{z}_k + \boldsymbol{b}_Y - \boldsymbol{y}_k\|^2 \\
&= \frac{1}{n}\sum_{k=1}^{n} \|\boldsymbol{W}_Y \boldsymbol{z}_k + \boldsymbol{c} - \boldsymbol{y}_k\|^2,
\end{aligned}$$

where the minimizer $\boldsymbol{c}$ is

$$\begin{aligned}
\boldsymbol{c} &= \frac{1}{n}\sum_{k=1}^{n} (\boldsymbol{y}_k - \boldsymbol{W}_Y \boldsymbol{z}_k) \\
&= \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{y}_k - \boldsymbol{W}_Y \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{z}_k \\
&= \mathbb{E}\{Y\} - \boldsymbol{W}_Y\, \mathbb{E}\{Z\}.
\end{aligned} \tag{6}$$

Let all the columns of $\boldsymbol{C}$ be equal to $\boldsymbol{c}$. Therefore we now have

$$\begin{aligned}
&\min_{\boldsymbol{W}_Y, \boldsymbol{b}_Y} \text{MSE}\,(\widehat{Y} - Y) \\
&= \min_{\boldsymbol{W}_Y} \frac{1}{n} \left\| \boldsymbol{W}_Y \boldsymbol{\Theta} \boldsymbol{K}_X + \boldsymbol{C} - \boldsymbol{Y} \right\|_F^2 \\
&= \min_{\boldsymbol{W}_Y} \frac{1}{n} \left\| \boldsymbol{W}_Y \boldsymbol{\Theta} \boldsymbol{K}_X \boldsymbol{H} - \tilde{\boldsymbol{Y}} \right\|_F^2 \\
&= \min_{\boldsymbol{W}_Y} \frac{1}{n} \left\| \boldsymbol{H} \boldsymbol{K}_X \boldsymbol{\Theta}^T \boldsymbol{W}_Y^T - \tilde{\boldsymbol{Y}}^T \right\|_F^2 \\
&= \min_{\boldsymbol{W}_Y} \frac{1}{n} \left\| \boldsymbol{M} \boldsymbol{W}_Y^T - P_{\mathcal{M}} \tilde{\boldsymbol{Y}}^T \right\|_F^2 + \frac{1}{n} \left\| P_{\mathcal{M}^\perp} \tilde{\boldsymbol{Y}}^T \right\|_F^2 \qquad (7) \\
&= \frac{1}{n} \left\| \underbrace{\boldsymbol{M} \boldsymbol{M}^\dagger}_{P_{\mathcal{M}}} P_{\mathcal{M}} \tilde{\boldsymbol{Y}}^T - P_{\mathcal{M}} \tilde{\boldsymbol{Y}}^T \right\|_F^2 + \frac{1}{n} \left\| P_{\mathcal{M}^\perp} \tilde{\boldsymbol{Y}}^T \right\|_F^2 \\
&= \frac{1}{n} \left\| P_{\mathcal{M}^\perp} \tilde{\boldsymbol{Y}}^T \right\|_F^2 \\
&= \frac{1}{n} \left\| \tilde{\boldsymbol{Y}}^T \right\|_F^2 - \frac{1}{n} \left\| P_{\mathcal{M}} \tilde{\boldsymbol{Y}}^T \right\|_F^2,
\end{aligned}$$

where the third step is due to (6), $\boldsymbol{M} = \boldsymbol{H} \boldsymbol{K}_X \boldsymbol{\Theta}^T$, and the fifth step is the orthogonal decomposition w.r.t. $\boldsymbol{M}$. Using the same approach, we get

$$J_S = \frac{1}{n} \left\| \tilde{\boldsymbol{S}}^T \right\|_F^2 - \frac{1}{n} \left\| P_{\mathcal{M}} \tilde{\boldsymbol{S}}^T \right\|_F^2. \qquad (8)$$

Assume that the columns of $\boldsymbol{L}_X$ are the orthonormal basis for the columns space of $\boldsymbol{H} \boldsymbol{K}_X$. For any $\boldsymbol{M}$, there exists $\boldsymbol{G}$ such that $\boldsymbol{L}_X \boldsymbol{G} = \boldsymbol{M}$. In general, there is no bijection between $\boldsymbol{\Theta}$ and $\boldsymbol{G}$ in the equality $\boldsymbol{H} \boldsymbol{K}_X \boldsymbol{\Theta}^T = \boldsymbol{L}_X \boldsymbol{G}$. But, there is a bijection between $\boldsymbol{G}$ and $\boldsymbol{\Theta}$ when constrained to $\boldsymbol{\Theta}$'s in which $\mathcal{R}(\boldsymbol{\Theta}^T) \subseteq \mathcal{N}(\boldsymbol{H} \boldsymbol{K}_X)^\perp$. This restricted bijection is sufficient since for any $\boldsymbol{\Theta}^T \in \mathcal{N}(\boldsymbol{H} \boldsymbol{K}_X)$ we have $\boldsymbol{M} = \boldsymbol{0}$. Once $\boldsymbol{G}$ is determined, $\boldsymbol{\Theta}^T$ can be obtained as

$$\boldsymbol{\Theta}^T = (\boldsymbol{H} \boldsymbol{K}_X)^\dagger \boldsymbol{L}_X \boldsymbol{G} + \boldsymbol{\Theta}_0, \;\; \boldsymbol{\Theta}_0 \subseteq \mathcal{N}(\boldsymbol{H} \boldsymbol{K}_X).$$

However, since

$$\|\boldsymbol{\Theta}\|_F^2 = \left\| \boldsymbol{\Theta}^T \right\|_F^2 = \left\| (\boldsymbol{H} \boldsymbol{K}_X)^\dagger \boldsymbol{L}_X \boldsymbol{G} \right\|_F^2 + \|\boldsymbol{\Theta}_0\|_F^2,$$

choosing $\boldsymbol{\Theta}_0 = \boldsymbol{0}$ results in minimum $\|\boldsymbol{\Theta}\|_F$, which is favorable in terms of robustness to the noise. Similar to (3.6), we have $P_{\mathcal{M}} = \boldsymbol{L}_X P_{\mathcal{G}} \boldsymbol{L}_X^T$. If we assume that the rank of $P_{\mathcal{G}}$ is i, $J_\lambda(\boldsymbol{G})$ in (3.10) can be expressed as

$$J_\lambda(\boldsymbol{G}) = \lambda \left\| \boldsymbol{L}_X \boldsymbol{G} \boldsymbol{G}^T \boldsymbol{L}_X^T \tilde{\boldsymbol{S}}^T \right\|_F^2 - (1 - \lambda) \left\| \boldsymbol{L}_X \boldsymbol{G} \boldsymbol{G}^T \boldsymbol{L}_X^T \tilde{\boldsymbol{Y}}^T \right\|_F^2,$$

where $P_{\mathcal{G}} = \boldsymbol{G} \boldsymbol{G}^T$ for some orthogonal matrix $\boldsymbol{G} \in \mathbb{R}^{d_X \times i}$. This resembles the optimization problem in (3.9) and therefore have the same solution as Theorem 3.3 with modified $\boldsymbol{B}$ as

$$\boldsymbol{B} = \boldsymbol{L}_X^T \left(\lambda \, \tilde{\boldsymbol{S}}^T \tilde{\boldsymbol{S}} - (1 - \lambda) \, \tilde{\boldsymbol{Y}}^T \tilde{\boldsymbol{Y}} \right) \boldsymbol{L}_X. \qquad (9)$$

Once $\boldsymbol{G}$ is determined, $\boldsymbol{\Theta}$ can be computed as $\boldsymbol{\Theta} = \boldsymbol{G}^T \boldsymbol{L}_X^T (\boldsymbol{H}\boldsymbol{K}_X)^\dagger$.

A.6 Proof of Theorem 3.4

Theorem. Let the columns of $\boldsymbol{L}_X$ be the orthonormal basis for $\boldsymbol{H}\boldsymbol{K}_X$. Further, assume that the columns of $\boldsymbol{V}_S$ are the singular vectors corresponding to zero singular values of $\tilde{\boldsymbol{S}}\boldsymbol{L}_X$ and the columns of $\boldsymbol{V}_Y$ are the singular vectors corresponding to non-zero singular values of $\tilde{\boldsymbol{Y}}\boldsymbol{L}_X$. Then, we have

$$
\begin{aligned}
\gamma_{\min} &:= \min_{\boldsymbol{\Theta}} J_Y(\boldsymbol{\Theta}) \;= \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\|\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\|_F^2 \\
\gamma_{\max} &:= \min_{\arg\max J_S(\boldsymbol{\Theta})} J_Y(\boldsymbol{\Theta}) = \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\boldsymbol{V}_S\right\|_F^2 \\
\alpha_{\min} &:= \max_{\arg\min J_Y(\boldsymbol{\Theta})} J_S(\boldsymbol{\Theta}) = \frac{1}{n}\left\|\tilde{\boldsymbol{S}}^T\right\|_F^2 - \frac{1}{n}\left\|\tilde{\boldsymbol{S}}\boldsymbol{L}_X\boldsymbol{V}_Y\right\|_F^2 \\
\alpha_{\max} &:= \max_{\boldsymbol{\Theta}} J_S(\boldsymbol{\Theta}) = \frac{1}{n}\left\|\tilde{\boldsymbol{S}}^T\right\|_F^2
\end{aligned}
$$

Proof. Firstly, we recall from Section that instead of $\boldsymbol{\Theta}$, we consider $\boldsymbol{G}$. These two matrices are related to each other as $\boldsymbol{H}\boldsymbol{K}_X\boldsymbol{\Theta}^T = \boldsymbol{L}_X\boldsymbol{G} = \boldsymbol{M}$, where the columns of $\boldsymbol{L}_X$ are the orthogonal basis for the column space of $\boldsymbol{H}\boldsymbol{K}_X$. Therefore we can now express the projection onto $\mathcal{M}$ in terms of projection onto $\mathcal{G}$, i.e.,$P_{\mathcal{M}} = \boldsymbol{L}_X P_{\mathcal{G}} \boldsymbol{L}_X$. Using (7), we get

$$
\begin{aligned}
\gamma_{\min} &= \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\max_{\boldsymbol{\Theta}}\left\|P_{\mathcal{M}}\tilde{\boldsymbol{Y}}^T\right\|_F^2 \\
&= \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\max_{\boldsymbol{G}}\left\|\boldsymbol{L}_X P_{\mathcal{G}} \boldsymbol{L}_X^T\tilde{\boldsymbol{Y}}^T\right\|_F^2 \\
&= \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\max_i\left\{\max_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}_i} \mathrm{Tr}\left[\boldsymbol{G}^T\boldsymbol{L}_X^T\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\boldsymbol{G}\right]\right\} \\
&= \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\mathrm{Tr}\left[\boldsymbol{V}_Y^T\boldsymbol{L}_X^T\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\boldsymbol{V}_Y\right] \qquad (10) \\
&= \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\sum_k \sigma_k^2 \\
&= \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\sum_{\sigma_k>0} \sigma_k^2 \\
&= \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\right\|_F^2,
\end{aligned}
$$

where the fourth step is borrowed from trace optimization problems studied in [90] and σ_k's are the singular values of $\tilde{\boldsymbol{Y}}\boldsymbol{L}_X$.

In order to interpret the bounds in more detail, we consider the one-dimensional case where

$X, Y, \in \mathbb{R}$. In this setting, the correlation coefficient (denoted by $\rho(\cdot,\cdot)$) between X and Y is

$$\begin{aligned}\rho(X,Y) &= \frac{\tilde{\boldsymbol{Y}}\tilde{\boldsymbol{X}}^T}{\sqrt{\tilde{\boldsymbol{Y}}\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T}} \\ &= \frac{\|\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\|_F}{\sigma_Y} \\ &= \sqrt{1-\frac{\gamma_{\min}}{\sigma_Y^2}},\end{aligned} \tag{11}$$

where $\sigma_Y^2 = \|\tilde{\boldsymbol{Y}}\|_F^2/n$. As a result, the normalized MSE can be expressed as

$$\frac{\gamma_{\min}}{\sigma_Y^2} = 1-\rho^2(X,Y). \tag{12}$$

Therefore, the lower bound of the target's MSE is independent of the encoder and is only related to the alignment between the subspaces spanned by the data and labels.

Next, we find an encoder that allows the target task to obtain its optimal loss, $\gamma_{\min}$, while seeking to minimize the leakage of sensitive attributes as much as possible. Thus, we constrain the domain of the encoder to $\{\arg\min J_Y(\boldsymbol{\Theta})\}$. Assume that the columns of the encoder $\boldsymbol{G}$ is the concatenation of the columns of $\boldsymbol{V}_Y$ together with at least one singular vector corresponding to a zero singular value of $\tilde{\boldsymbol{Y}}\boldsymbol{L}_X$. Therefore $\mathcal{V}_Y \subseteq \mathcal{G}$ and consequently $\|\boldsymbol{L}_X P_{\mathcal{V}_Y}\boldsymbol{L}_X^T\boldsymbol{U}\|_F^2 \leq \|\boldsymbol{L}_X P_{\mathcal{G}}\boldsymbol{L}_X^T\boldsymbol{U}\|_F^2$ for arbitrary matrix $\boldsymbol{U}$. As a result, $J_S(\boldsymbol{G}) \geq J_S(\boldsymbol{V}_Y)$ and at the same time $J_Y(\boldsymbol{G}) = J_Y(\boldsymbol{V}_Y)$. The latter can be observed from

$$\begin{aligned}\left\|\boldsymbol{L}_X P_{\mathcal{G}}\boldsymbol{L}_x^T\tilde{\boldsymbol{Y}}^T\right\|_F^2 &= \left\|\tilde{\boldsymbol{Y}}\boldsymbol{L}_X P_{\mathcal{G}}\boldsymbol{L}_X^T\right\|_F^2 \\ &= \left\|\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{L}_X^T\tilde{\boldsymbol{Y}}^T\right\|_F^2 \\ &= \left\|\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\boldsymbol{V}_Y\boldsymbol{V}_Y^T\boldsymbol{L}_X^T\right\|_F^2 \\ &= \left\|\boldsymbol{L}_X P_{\mathcal{V}_Y}\boldsymbol{L}_X^T\tilde{\boldsymbol{Y}}^T\right\|_F^2.\end{aligned} \tag{13}$$

We then have

$$\begin{aligned}\alpha_{\min} &= \frac{1}{n}\left\|\tilde{\boldsymbol{S}}^T\right\|_F^2 - \frac{1}{n}\left\|\boldsymbol{L}_X P_{\mathcal{V}_Y}\boldsymbol{L}_X^T\tilde{\boldsymbol{S}}^T\right\|_F^2 \\ &= \frac{1}{n}\left\|\tilde{\boldsymbol{S}}^T\right\|_F^2 - \frac{1}{n}\mathrm{Tr}\left[\boldsymbol{V}_Y^T\boldsymbol{L}_X^T\tilde{\boldsymbol{S}}^T\tilde{\boldsymbol{S}}\boldsymbol{L}_X\boldsymbol{V}_Y\right] \\ &= \frac{1}{n}\left\|\tilde{\boldsymbol{S}}^T\right\|_F^2 - \frac{1}{n}\left\|\tilde{\boldsymbol{S}}\boldsymbol{L}_X\boldsymbol{V}_Y\right\|_F^2.\end{aligned} \tag{14}$$

This bound can again be interpreted under the one-dimensional setting of $X, S \in \mathbb{R}$ as

$$\frac{\alpha_{\min}}{\sigma_S^2} = 1-\rho^2(X,S) \tag{15}$$

On the other hand, $\alpha_{\max}$ turns out to be,

$$\begin{aligned}\alpha_{\max} &= \frac{1}{n}\left\|\tilde{\boldsymbol{S}}^T\right\|_F^2 \\ &= \sigma_S^2,\end{aligned} \tag{16}$$

which can be achieved via a trivial choice of $\boldsymbol{G} = 0$. However, we let the columns of $\boldsymbol{G}$ be the singular vectors corresponding to all zero singular values of $\tilde{\boldsymbol{S}}\boldsymbol{L}_X$ in order to maximize $\left\|P_{\mathcal{M}}\tilde{\boldsymbol{Y}}^T\right\|_F$ and at the same time ensuring that $J_S(\boldsymbol{G})$ equal to $\alpha_{\max}$. As a result, we have

$$\gamma_{\max} = \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}^T\right\|_F^2 - \frac{1}{n}\left\|\tilde{\boldsymbol{Y}}\boldsymbol{L}_X\boldsymbol{V}_S\right\|_F^2.$$

For the one dimensional case i.e., $X, Y, S \in \mathbb{R}$, we get $V_S = 0$ and consequently,

$$\gamma_{\max} = \sigma_Y^2. \tag{17}$$

□

B.1 A Population Expression for Definition in (5.6)

A population expression for $\text{Dep}(Z, S)$ in (5.6) is given in the following.

$$\begin{aligned}\text{Dep}(Z,S) \;=\; \sum_{j=1}^{r}\Big\{&\mathbb{E}_{X,S,X',S'}\left[f_j(X)\, f_j(X')\, k_S(X,X')\right] \\ &+\mathbb{E}_X\left[f_j(X)\right]\mathbb{E}_{X'}\left[f_j(X')\right]\,\mathbb{E}_{S,S'}\left[k_S(X,S')\right] \\ &-2\,\mathbb{E}_{X,S}\left[f_j(X)\,\mathbb{E}_{X'}[f_j(X')]\,\mathbb{E}_{S'}[k_S(S,X')]\right]\Big\}\end{aligned}$$

where (X', S') is independent of (X, S) with the same distribution as $\boldsymbol{p}_{XS}$.

Proof. We first note that this population expression is inspired by that of HSIC [81].

Consider the operator Σ_{SX} induced by the linear functional $\mathbb{C}\text{ov}\left(\alpha(X), \beta_S(S)\right)$ $= \langle \beta_S, \Sigma_{SX}\alpha\rangle_{\mathcal{H}_S}$. Then, it follows that

$$\begin{aligned}\text{Dep}(Z,S) \;&=\; \sum_{j=1}^{r}\sum_{\beta_S\in\mathcal{U}_S}\mathbb{C}\text{ov}^2\left(f_j(X),\beta_S(S)\right) \\ &=\; \sum_{j=1}^{r}\sum_{\beta_S\in\mathcal{U}_S}\left\langle\beta_S,\Sigma_{SX}f_j\right\rangle_{\mathcal{H}_S}^2 \\ &=\; \sum_{j=1}^{r}\sum_{\beta_S\in\mathcal{U}_S}\left\langle\beta_S,\Sigma_{SX}f_j\right\rangle_{\mathcal{H}_S}^2 \\ &\overset{\text{(a)}}{=}\; \sum_{j=1}^{r}\|\Sigma_{SX}f_j\|_{\mathcal{H}_S}^2\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{r} \left\langle \Sigma_{SX} f_j, \Sigma_{SX} f_j \right\rangle_{\mathcal{H}_S} \\
&\overset{\text{(b)}}{=} \sum_{j=1}^{r} \mathbb{C}\text{ov}\left(f_j(X),\, (\Sigma_{SX} f_j)(S)\right) \\
&= \sum_{j=1}^{r} \mathbb{C}\text{ov}\left(f_j(X),\, \left\langle k_S(\cdot, S), \Sigma_{SX} f_j \right\rangle_{\mathcal{H}_S}\right) \\
&= \sum_{j=1}^{r} \mathbb{C}\text{ov}\left(f_j(X), \mathbb{C}\text{ov}(f_j(X'),\, k_S(S', S))\right) \\
&= \sum_{j=1}^{r} \mathbb{C}\text{ov}\left(\, f_j(X),\, \mathbb{E}_{X',S'}[f_j(X')\, k_S(S, S')] - \mathbb{E}_{X'}[f_j(X')]\, \mathbb{E}_{S'}[k_S(S, S')]\, \right) \\
&= \sum_{j=1}^{r} \Big\{ \mathbb{E}_{X,S,X',S'}\left[f_j(X)\, f_j(X')\, k_S(S, S')\right] \\
&\qquad\qquad + \mathbb{E}_X\left[f_j(X)\right] \mathbb{E}_{X'}\left[f_j(X')\right]\, \mathbb{E}_{S,S'}\left[k_S(S, S')\right] \\
&\qquad\qquad\qquad - 2\, \mathbb{E}_{X,S}\left[f_j(X)\, \mathbb{E}_{X'}[f_j(X')]\, \mathbb{E}_{S'}[k_S(S, S')]\right] \Big\}
\end{aligned}
$$

where (a) is due to Parseval relation for orthonormal basis and (b) is from the definition of Σ_{SX}. □

B.2 Proof of Lemma 5.3

Lemma. Let $\boldsymbol{K}_X, \boldsymbol{K}_S \in \mathbb{R}^{n \times n}$ be Gram matrices corresponding to $\mathcal{H}_X$ and $\mathcal{H}_S$, respectively, i.e., $(\boldsymbol{K}_X)_{ij} = k_X(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $(\boldsymbol{K}_S)_{ij} = k_S(\boldsymbol{s}_i, \boldsymbol{s}_j)$, where covariance is empirically estimated as

$$
\mathbb{C}\text{ov}\left(f_j(X), \beta_S(S)\right) \approx \frac{1}{n} \sum_{i=1}^{n} f_j(\boldsymbol{x}_i) \beta_S(\boldsymbol{s}_i) - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{k=1}^{n} f_j(\boldsymbol{x}_i) \beta_S(\boldsymbol{s}_k).
$$

It follows that, the corresponding empirical estimation for $\text{Dep}\,(Z, S)$ is

$$
\text{Dep}^{\text{emp}}\,(Z, S) \;:=\; \frac{1}{n^2} \left\| \boldsymbol{\Theta} \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{L}_S \right\|_F^2, \tag{18}
$$

where $\boldsymbol{H} = \boldsymbol{I}_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$ is the centering matrix, and $\boldsymbol{L}_S$ is a full column-rank matrix in which $\boldsymbol{L}_S \boldsymbol{L}_S^T = \boldsymbol{K}_S$ (Cholesky factorization). Furthermore, the empirical estimator in (5.7) has a bias of $\mathcal{O}(n^{-1})$ and a convergence rate of $\mathcal{O}(n^{-1/2})$.

Proof. Firstly, let us reconstruct the orthonormal set $\mathcal{U}_S$ when n i.i.d. observations $\{\boldsymbol{s}_j\}_{j=1}^n$ are

given. Invoking representer theorem, for two arbitrary elements β_i and β_m in $\mathcal{U}_S$, we have

$$\begin{aligned}
\langle \beta_i,\, \beta_m \rangle_{\mathcal{H}_S} &= \left\langle \sum_{j=1}^{n} \alpha_j k_S(\boldsymbol{s}_j, \cdot), \sum_{l=1}^{n} \eta_l k_S(\boldsymbol{s}_l, \cdot) \right\rangle_{\mathcal{H}_S} \\
&= \sum_{j=1}^{n} \sum_{l=1}^{n} \alpha_j \eta_l k_S(\boldsymbol{s}_j, \boldsymbol{s}_l) \\
&= \boldsymbol{\alpha}^T \boldsymbol{K}_S \boldsymbol{\eta} \\
&= \left\langle \boldsymbol{L}_S^T \boldsymbol{\alpha},\, \boldsymbol{L}_S^T \boldsymbol{\eta} \right\rangle_{\mathbb{R}^q}
\end{aligned}$$

where $\boldsymbol{L}_S \in \mathbb{R}^{n \times q}$ is a full column-rank matrix and $\boldsymbol{K}_S = \boldsymbol{L}_S \boldsymbol{L}_S^T$ is the Cholesky factorization of $\boldsymbol{K}_S$. As a result, searching for $\beta_i \in \mathcal{U}_S$ is equivalent to searching for $\boldsymbol{L}_S^T \boldsymbol{\alpha} \in \mathcal{U}_q$ where $\mathcal{U}_q$ is any complete orthonormal set for $\mathbb{R}^q$. Using empirical expression for covariance, we get

$$\begin{aligned}
\text{Dep}^{\text{emp}}(Z, S) &:= \sum_{\beta_S \in \mathcal{U}_S} \sum_{j=1}^{r} \left\{ \frac{1}{n} \sum_{i=1}^{n} f_j(\boldsymbol{x}_i) \beta_S(\boldsymbol{s}_i) - \frac{1}{n^2} \sum_{i=1}^{n} f_j(\boldsymbol{x}_i) \sum_{k=1}^{n} \beta_S(\boldsymbol{s}_k) \right\}^2 \\
&= \sum_{\boldsymbol{L}_S^T \boldsymbol{\alpha} \in \mathcal{U}_q} \sum_{j=1}^{r} \left\{ \frac{1}{n} \boldsymbol{\theta}_j^T \boldsymbol{K}_X \boldsymbol{K}_S \boldsymbol{\alpha} - \frac{1}{n^2} \boldsymbol{\theta}_j^T \boldsymbol{K}_X \mathbf{1}_n \mathbf{1}_n^T \boldsymbol{K}_S \boldsymbol{\alpha} \right\}^2 \\
&= \sum_{\boldsymbol{L}_S^T \boldsymbol{\alpha} \in \mathcal{U}_q} \sum_{j=1}^{r} \left\{ \frac{1}{n} \boldsymbol{\theta}_j^T \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{K}_S \boldsymbol{\alpha} \right\}^2 \\
&= \sum_{\boldsymbol{L}_S^T \boldsymbol{\alpha} \in \mathcal{U}_q} \sum_{j=1}^{r} \left\{ \frac{1}{n} \boldsymbol{\theta}_j^T \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{L}_S \boldsymbol{L}_S^T \boldsymbol{\alpha} \right\}^2 \\
&= \sum_{\boldsymbol{\zeta} \in \mathcal{U}_q} \sum_{j=1}^{r} \left\{ \frac{1}{n} \boldsymbol{\theta}_j^T \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{L}_S \boldsymbol{\zeta} \right\}^2 \\
&= \sum_{\boldsymbol{\zeta} \in \mathcal{U}_q} \frac{1}{n^2} \left\| \boldsymbol{\Theta} \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{L}_S \boldsymbol{\zeta} \right\|_2^2 \\
&= \frac{1}{n^2} \left\| \boldsymbol{\Theta} \boldsymbol{K}_X \boldsymbol{H} \boldsymbol{L}_S \right\|_F^2,
\end{aligned}$$

where $\boldsymbol{f}(X) = \boldsymbol{\Theta}[k_X(\boldsymbol{x}_1, X), \cdots, k_X(\boldsymbol{x}_n, X)]^T$ and $\boldsymbol{\Theta} := [\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_r]^T$.

We now show that the bias of $\text{Dep}^{\text{epm}}(Z, S)$ for estimating $\text{Dep}(Z, S)$ in (5.7) is $\mathcal{O}\left(\frac{1}{n}\right)$. To

achieve this, we split $\mathrm{Dep}^{\mathrm{epm}}(Z,S)$ into three terms as,

$$\begin{aligned}\frac{1}{n^2}\left\|\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{H}\boldsymbol{L}_S\right\|_F^2 &= \frac{1}{n^2}\mathrm{Tr}\left\{\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{H}\boldsymbol{K}_S\boldsymbol{H}\boldsymbol{K}_X\boldsymbol{\Theta}^T\right\}\\ &= \frac{1}{n^2}\mathrm{Tr}\left\{\boldsymbol{\Theta}\boldsymbol{K}_X\left(\boldsymbol{I}-\frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{K}_S\left(\boldsymbol{I}-\frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{K}_X\boldsymbol{\Theta}^T\right\}\\ &= \frac{1}{n^2}\underbrace{\mathrm{Tr}\left\{\boldsymbol{K}_X\boldsymbol{\Theta}^T\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{K}_S\right\}}_{\mathrm{I}}-\frac{2}{n^3}\underbrace{\mathrm{Tr}\left\{\mathbf{1}^T\boldsymbol{K}_X\boldsymbol{\Theta}^T\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{K}_S\mathbf{1}\right\}}_{\mathrm{II}}\\ &\quad+\frac{1}{n^4}\underbrace{\mathrm{Tr}\left\{\mathbf{1}^T\boldsymbol{K}_X\boldsymbol{\Theta}^T\boldsymbol{\Theta}\boldsymbol{K}_X\mathbf{1}\mathbf{1}^T\boldsymbol{K}_S\mathbf{1}\right\}}_{\mathrm{III}} \qquad (19)\end{aligned}$$

Let $\boldsymbol{c}_p^n$ denote the set of all p-tuples drawn without replacement from $\{1,\cdots,n\}$. Moreover, let $\boldsymbol{\Theta}=[\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_r]^T\in\mathbb{R}^{r\times n}$ and $(\boldsymbol{A})_{ij}$ denote the element of an arbitrary matrix $\boldsymbol{A}$ at i-th row and j-th column. Then, it follows that

(I):

$$\begin{aligned}&\mathbb{E}\left[\mathrm{Tr}\left\{\boldsymbol{K}_X\boldsymbol{\Theta}^T\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{K}_S\right\}\right]\\ =&\sum_{k=1}^r\mathbb{E}\left[\mathrm{Tr}\left\{\underbrace{\boldsymbol{K}_X\boldsymbol{\theta}_k}_{:=\boldsymbol{\alpha}_k}\boldsymbol{\theta}_k^T\boldsymbol{K}_X\boldsymbol{K}_S\right\}\right]\\ =&\sum_{k=1}^r\mathbb{E}\left[\mathrm{Tr}\left\{\boldsymbol{\alpha}_k\boldsymbol{\alpha}_k^T\boldsymbol{K}_S\right\}\right]\\ =&\sum_{k=1}^r\mathbb{E}\left[\sum_i(\boldsymbol{\alpha}_k\boldsymbol{\alpha}_k^T)_{ii}(\boldsymbol{K}_S)_{ii}+\sum_{(i,j)\in\boldsymbol{c}_2^n}(\boldsymbol{\alpha}_k\boldsymbol{\alpha}_k^T)_{ij}(\boldsymbol{K}_S)_{ji}\right]\\ =&\,n\sum_{k=1}^r\mathbb{E}_{X,S}\left[f_k^2(X)k_S(S,S)\right]\\ &+\frac{n!}{(n-2)!}\sum_{k=1}^r\mathbb{E}_{X,S,X',S'}\left[f_k(X)f_k(X')k_S(S,S')\right]\\ =&\,\mathcal{O}(n)+\frac{n!}{(n-2)!}\sum_{k=1}^r\mathbb{E}_{X,S,X',S'}\left[f_k(X)f_k(X')k_S(S,S')\right] \qquad (20)\end{aligned}$$

where (X,S) and (X',S') are independently drawn from the joint distribution $\boldsymbol{p}_{XS}$.

(II):

$$\begin{aligned}&\mathbb{E}\left[\mathbf{1}^T\boldsymbol{K}_X\boldsymbol{\Theta}^T\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{K}_S\mathbf{1}\right]\\ =&\sum_{k=1}^r\mathbb{E}\left[\mathbf{1}^T\underbrace{\boldsymbol{K}_X\boldsymbol{\theta}_k}_{\boldsymbol{\alpha}_k}\boldsymbol{\theta}_k^T\boldsymbol{K}_X\boldsymbol{K}_S\mathbf{1}\right]\end{aligned}$$

$$
\begin{aligned}
&= \sum_{k=1}^{r} \mathbb{E}\left[\mathbf{1}^T \boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T \boldsymbol{K}_S \mathbf{1}\right] \\
&= \sum_{k=1}^{r} \mathbb{E}\left[\sum_{m=1}^{n}\sum_{i=1}^{n}\sum_{j=1}^{n} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{mi} (\boldsymbol{K}_S)_{mj}\right] \\
&= \sum_{k=1}^{r} \mathbb{E}\left[\sum_{i} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{ii} (\boldsymbol{K}_S)_{ii} + \sum_{(m,j)\in \boldsymbol{c}_2^n} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{mm} (\boldsymbol{K}_S)_{mj}\right] \\
&+ \sum_{k=1}^{r} \mathbb{E}\left[\sum_{(m,i)\in \boldsymbol{c}_2^n} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{mi} (\boldsymbol{K}_S)_{mm} + \sum_{(m,j)\in \boldsymbol{c}_2^n} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{mj} (\boldsymbol{K}_S)_{mj}\right] \\
&+ \sum_{k=1}^{r} \mathbb{E}\left[\sum_{(m,i,j)\in \boldsymbol{c}_3^n} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{mi} (\boldsymbol{K}_S)_{mj}\right] \\
&= n \sum_{k=1}^{r} \mathbb{E}_{X,S}\left[f_k^2(X) k_S(S,S)\right] \\
&+ \frac{n!}{(n-2)!} \sum_{k=1}^{r} \mathbb{E}_{X,S,S'}\left[f_k^2(X) k_S(S,S')\right] \\
&+ \frac{n!}{(n-2)!} \sum_{k=1}^{r} \mathbb{E}_{X,S,X'}\left[f_k(X) f_k(X') k_S(S,S)\right] \\
&+ \frac{n!}{(n-2)!} \sum_{k=1}^{r} \mathbb{E}_{X,S,X',S'}\left[f_k(X) f_k(X') k_S(S,S')\right] \\
&+ \frac{n!}{(n-3)!} \sum_{k=1}^{r} \mathbb{E}_{X,S}\left[f_k(X) \mathbb{E}_{X'}[f_k(X')] \mathbb{E}_{S'}[k_S(S,S')]\right] \\
&= \frac{n!}{(n-3)!} \sum_{k=1}^{r} \mathbb{E}_{X,S}\left[f_k(X) \mathbb{E}_{X'}[f_k(X')] \mathbb{E}_{S'}[k_S(S,S')]\right] \\
&+ \mathcal{O}(n^2). \qquad (21)
\end{aligned}
$$

(III):

$$
\begin{aligned}
&\mathbb{E}\left[\mathbf{1}^T \boldsymbol{K}_X \boldsymbol{\Theta}^T \boldsymbol{\Theta} \boldsymbol{K}_X \mathbf{1}\mathbf{1}^T \boldsymbol{K}_S \mathbf{1}\right] \\
&= \sum_{k=1}^{r} \mathbb{E}\left[\mathbf{1}^T \underbrace{\boldsymbol{K}_X \boldsymbol{\theta}_k}_{\boldsymbol{\alpha}_k} \boldsymbol{\theta}_k^T \boldsymbol{K}_X \mathbf{1}\mathbf{1}^T \boldsymbol{K}_S \mathbf{1}\right] \\
&= \sum_{k=1}^{r} \mathbb{E}\left[\mathbf{1}^T \boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T \mathbf{1}\mathbf{1}^T \boldsymbol{K}_S \mathbf{1}\right]
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{k=1}^{r} \mathbb{E}\left[\sum_{i,j,m,l} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{ij} (\boldsymbol{K}_S)_{ml}\right] \\
&= \mathcal{O}(n^3) + \sum_{k=1}^{r} \mathbb{E}\left[\sum_{(i,j,m,l)\in \boldsymbol{c}_4^n} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{ij} (\boldsymbol{K}_S)_{ml}\right] \\
&= \frac{n!}{(n-4)!} \sum_{k=1}^{r} \mathbb{E}_X\left[f_k(X)\right] E_{X'}\left[f_k(X')\right] \mathbb{E}_{S,S'}\left[k_S(S,S')\right] \\
&+ \mathcal{O}(n^3). \qquad (22)
\end{aligned}
$$

Using above calculations together with Lemma 2 lead to

$$
\text{Dep}(Z,S) = \mathbb{E}\left[\text{Dep}^{\text{emp}}(Z,S)\right] + \mathcal{O}\left(\frac{1}{n}\right).
$$

We now obtain the convergence of $\text{dep}^{\text{emp}}(Z,S)$. Consider the decomposition in (19) together with (20), (21), and (22). Let $\boldsymbol{\alpha}_k := \boldsymbol{K}_X \boldsymbol{\theta}_k$, then it follows that

$$
\begin{aligned}
&\mathbb{P}\left\{\text{Dep}(Z,S) - \text{Dep}^{\text{emp}}(Z,S) \geq t\right\} \\
\leq\ & \mathbb{P}\Bigg\{ \sum_{k=1}^{r} \mathbb{E}_{X,S,X',S'}\left[f_k(X) f_k(X') k_S(S,S')\right] \\
& -\frac{(n-2)!}{n!} \sum_{k=1}^{r} \sum_{(i,j)\in \boldsymbol{c}_2^n} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{ij} (\boldsymbol{K}_S)_{ji} + \mathcal{O}\left(\frac{1}{n}\right) \geq at \Bigg\} \\
+\ & \mathbb{P}\Bigg\{ \sum_{k=1}^{r} \mathbb{E}_{X,S}\left[f_k(X) \mathbb{E}_{X'}[f_k(X')] \mathbb{E}_{S'}[k_S(S,S')]\right] \\
-\ & \frac{(n-3)!}{n!} \sum_{k=1}^{r} \sum_{(i,j,m)\in \boldsymbol{c}_3^n} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{mi} (\boldsymbol{K}_S)_{mj} + \mathcal{O}\left(\frac{1}{n}\right) \geq bt \Bigg\} \\
+\ & \mathbb{P}\Bigg\{ \sum_{k=1}^{r} E_X\left[f_k(X)\right] E_{X'}\left[f_k(X')\right] \mathbb{E}_{S,S'}\left[k_S(S,S')\right] \\
& -\frac{(n-4)!}{n!} \sum_{k=1}^{r} \sum_{(i,j,m,l)\in \boldsymbol{c}_4^n} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{ij} (\boldsymbol{K}_S)_{ml} + \mathcal{O}\left(\frac{1}{n}\right) \geq (1-a-b)t \Bigg\},
\end{aligned}
$$

where $a, b > 0$ and $a + b < 1$. For convenience, we omit the term $\mathcal{O}\left(\frac{1}{n}\right)$ and add it back in the last stage.

Define $\boldsymbol{\zeta} := (X, S)$ and consider the following U-statistics [173]

$$
u_1(\boldsymbol{\zeta}_i, \boldsymbol{\zeta}_j) \ = \ \frac{(n-2)!}{n!} \sum_{(i,j)\in \boldsymbol{c}_2^n} \sum_{k=1}^{r} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{ij} (\boldsymbol{K}_S)_{ij}
$$

$$u_2(\boldsymbol{\zeta}_i, \boldsymbol{\zeta}_j, \boldsymbol{\zeta}_m) = \frac{(n-3)!}{n!} \sum_{(i,j,m)\in \boldsymbol{c}_3^n} \sum_{k=1}^{r} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{mi} (\boldsymbol{K}_S)_{mj}$$

$$u_3(\boldsymbol{\zeta}_i, \boldsymbol{\zeta}_j, \boldsymbol{\zeta}_m, \boldsymbol{\zeta}_l) = \frac{(n-4)!}{n!} \sum_{(i,j,m,l)\in \boldsymbol{c}_4^n} \sum_{k=1}^{r} (\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T)_{ij} (\boldsymbol{K}_S)_{ml}$$

Then, from Hoeffding's inequality [173] it follows that

$$\mathbb{P}\left\{\text{Dep}(Z,S) - \text{Dep}^{\text{emp}}(Z,S) \geq t\right\} \leq e^{\frac{-2a^2t^2}{2r^2M^2}n} + e^{\frac{-2b^2t^2}{3r^2M^2}n} + e^{\frac{-2(1-a-b)^2t^2}{4r^2M^2}n},$$

where we assumed that $k_S(\cdot,\cdot)$ is bounded by one and $f_k^2(X_i)$ is bounded by M for any $k = 1, \cdots, r$ and $i = 1, \cdots, n$.

Further, if $0.22 \leq a < 1$, it holds that

$$e^{\frac{-2a^2t^2}{2r^2M^2}n} + e^{\frac{-2b^2t^2}{3r^2M^2}n} + e^{\frac{-2(1-a-b)^2t^2}{4r^2M^2}n} \leq 3e^{\frac{-a^2t^2}{r^2M^2}n}.$$

Consequently, we have

$$\mathbb{P}\left\{|\text{Dep}(Z,S) - \text{Dep}^{\text{emp}}(Z,S)| \geq t\right\} \leq 6e^{\frac{-a^2t^2}{r^2M^2}n}.$$

Therefore, with probability at least $1-\delta$, it holds

$$|\text{Dep}(Z,S) - \text{Dep}^{\text{emp}}(Z,S)| \leq \sqrt{\frac{r^2M^2\log(6/\sigma)}{\alpha^2 n}} + \mathcal{O}\left(\frac{1}{n}\right). \tag{23}$$

□

B.3 Proof of Theorem 5.4

Theorem. Let $Z = \boldsymbol{f}(X)$ be an arbitrary representation of the input data, where $\boldsymbol{f} \in \mathcal{H}_X$. Then, there exist an invertible Borel function $\boldsymbol{h}$, such that, $\boldsymbol{h} \circ \boldsymbol{f}$ belongs to $\mathcal{A}_r$.

Proof. Recall that the space of disentangled representation is

$$\mathcal{A}_r := \left\{(f_1, \cdots, f_r) \;\middle|\; f_i, f_j \in \mathcal{H}_X, \mathbb{C}\text{ov}\left(f_i(X), f_j(X)\right) + \gamma\langle f_i, f_j\rangle_{\mathcal{H}_X} = \delta_{i,j}\right\},$$

where $\gamma > 0$. Let I_X denote the identity operator from $\mathcal{H}_X$ to $\mathcal{H}_X$. We claim that $\boldsymbol{h} := [h_1, \cdots, h_r]$, where

$$\boldsymbol{G}_0 = \begin{bmatrix} \langle f_1, f_1\rangle_{\mathcal{H}_X} & \cdots & \langle f_1, f_r\rangle_{\mathcal{H}_X} \\ \vdots & \ddots & \vdots \\ \langle f_r, f_1\rangle_{\mathcal{H}_X} & \cdots & \langle f_r, f_r\rangle_{\mathcal{H}_X} \end{bmatrix}$$

$$\boldsymbol{G} = {\boldsymbol{G}_0}^{-1/2}$$

$$h_j \circ \boldsymbol{f} = \sum_{m=1}^{r} g_{jm} \left(\Sigma_{XX} + \gamma I_X\right)^{-1/2} f_j, \quad \forall j = 1, \cdots, r$$

is the desired invertible transformation. To see this, construct

$$
\begin{aligned}
& \mathbb{C}\mathrm{ov}\left(h_i(\boldsymbol{f}(X)), h_j(\boldsymbol{f}(X))\right) + \gamma\langle h_i \circ \boldsymbol{f}, h_j \circ \boldsymbol{f}\rangle_{\mathcal{H}_X} \\
= \; & \left\langle h_i \circ \boldsymbol{f}, (\Sigma_{XX} + \gamma I_X)\, h_j \circ \boldsymbol{f} \right\rangle_{\mathcal{H}_X} \\
= \; & \left\langle \sum_{m=1}^{r} g_{im} \left(\Sigma_{XX} + \gamma I_X\right)^{-1/2} f_i, \sum_{k=1}^{r} g_{jk} (\Sigma_{XX} + \gamma I_X) \left(\Sigma_{XX} + \gamma I_X\right)^{-1/2} f_j \right\rangle_{\mathcal{H}_X} \\
= \; & \sum_{m=1}^{r} \sum_{k=1}^{r} g_{im}\, g_{jk} \left\langle f_i, f_j \right\rangle_{\mathcal{H}_X} = (\boldsymbol{G}\, \boldsymbol{G}_0\, \boldsymbol{G})_{ij} = \delta_{i,j}
\end{aligned}
$$

The inverse of $\boldsymbol{h}$ is $\boldsymbol{h}' := [h'_1, \cdots, h'_r]$ where

$$
\begin{aligned}
& \boldsymbol{H} = \boldsymbol{G}_0^{1/2} \\
& h'_j \circ \boldsymbol{h} = \sum_{m=1}^{r} h_{jm} \left(\Sigma_{XX} + \gamma I_X\right)^{1/2} h_j, \quad \forall j = 1, \cdots, r.
\end{aligned}
$$

□

B.4 Proof of Theorem 5.7

Theorem. Assume that k_S and k_Y are bounded by one and $f_j^2(\boldsymbol{x}_i) \leq M$ for any $j = 1, \ldots, r$ and $i = 1, \ldots, n$ for which $\boldsymbol{f} = (f_1, \ldots, f_r) \in \mathcal{A}_r$. Then, for any $n > 1$ and $0 < \delta < 1$, with probability at least $1 - \delta$, we have

$$
\left| \sup_{\boldsymbol{f} \in \mathcal{A}_r} J(\boldsymbol{f}, \lambda) - \sup_{\boldsymbol{f} \in \mathcal{A}_r} J^{\text{emp}}(\boldsymbol{f}, \lambda) \right| \leq rM \sqrt{\frac{\log(6/\delta)}{0.22^2\, n}} + \mathcal{O}\left(\frac{1}{n}\right).
$$

Proof. Recall that in the proof of Lemma 5.3 we have shown that with probability at least $1 - \delta$, the following inequality holds

$$
|\text{Dep}(Z, S) - \text{Dep}^{\text{emp}}(Z, S)| \leq \sqrt{\frac{r^2 M^2 \log(6/\sigma)}{0.22^2\, n}} + \mathcal{O}\left(\frac{1}{n}\right).
$$

Using the same reasoning for $\text{dep}(Z, Y)$, with probability at least $1 - \delta$, we have

$$
|\text{Dep}(Z, Y) - \text{Dep}^{\text{emp}}(Z, Y)| \leq \sqrt{\frac{r^2 M^2 \log(6/\sigma)}{0.22^2\, n}} + \mathcal{O}\left(\frac{1}{n}\right).
$$

Since $J(\boldsymbol{f}(X)) = (1 - \lambda)\, \text{dep}(Z, Y) - \lambda\, \text{dep}(Z, S)$ and $J^{\text{emp}}(\boldsymbol{f}(X)) := (1 - \lambda)\, \text{dep}^{\text{emp}}(Z, Y) - \lambda\, \text{dep}^{\text{emp}}(Z, S)$, it follows that with probability at least $1 - \delta$,

$$
|J(\boldsymbol{f}, \lambda) - J^{\text{emp}}(\boldsymbol{f}, \lambda)| \leq rM \sqrt{\frac{\log(6/\sigma)}{0.22^2\, n}} + \mathcal{O}\left(\frac{1}{n}\right).
$$

We complete the proof by noting that, the following inequality holds for any bounded J and J^{emp}:

$$\left| \sup_{\boldsymbol{f} \in \mathcal{A}_r} J(\boldsymbol{f}, \lambda) - \sup_{\boldsymbol{f} \in \mathcal{A}_r} J^{\text{emp}}(\boldsymbol{f}, \lambda) \right| \leq \sup_{\boldsymbol{f} \in \mathcal{A}_r} \left| J(\boldsymbol{f}, \lambda) - J^{\text{emp}}(\boldsymbol{f}, \lambda) \right|.$$

□