Adversarial Representation Learning With Closed-Form Solvers

Bashir Sadeghi, Lan Wang, and Vishnu Naresh Boddeti (✉)

Michigan State University, East Lansing, MI 48823, USA
{sadeghib, wanglan3, vishnu}@msu.edu
http://hal.cse.msu.edu

Abstract. Adversarial representation learning aims to learn data representations for a target task while removing unwanted sensitive information at the same time. Existing methods learn model parameters iteratively through stochastic gradient descent-ascent, which is often unstable and unreliable in practice. To overcome this challenge, we adopt closed-form solvers for the adversary and target task. We model them as kernel ridge regressors and analytically determine an upper-bound on the optimal dimensionality of representation. Our solution, dubbed OptNet-ARL, reduces to a stable one one-shot optimization problem that can be solved reliably and efficiently. OptNet-ARL can be easily generalized to the case of multiple target tasks and sensitive attributes. Numerical experiments, on both small and large scale datasets, show that, from an optimization perspective, OptNet-ARL is stable and exhibits three to five times faster convergence. Performance wise, when the target and sensitive attributes are dependent, OptNet-ARL learns representations that offer a better trade-off front between (a) utility and bias for fair classification and (b) utility and privacy by mitigating leakage of private information than existing solutions.
Code is available at https://github.com/human-analysis.

Keywords: Fair machine learning · Adversarial representation learning · Closed-form solver · Kernel ridge regression.

1 Introduction

Adversarial Representation Learning (ARL) is a promising framework that affords explicit control over unwanted information in learned data representations. This concept has practically been employed in various applications, such as, learning unbiased and fair representations [7, 28, 29, 37], learning controllable representations that are invariant to sensitive attributes [31, 40], mitigating leakage of sensitive information [10, 33–35], unsupervised domain adaption [13], learning flexibly fair representations [7, 37], and many more.

The goal of ARL is to learn a data encoder $E : \boldsymbol{x} \mapsto \boldsymbol{z}$ that retains sufficient information about a desired *target* attribute, while removing information about a known *sensitive* attribute. The basic idea of ARL is to learn such a mapping under an adversarial setting. The learning problem is setup as a three-player minimax game between three entities (see Fig.1a, an encoder E, a predictor T, and a proxy adversary A. Target

predictor T seeks to extract target information and make correct predictions on the target task. The proxy adversary A mimics an unknown real adversary and seeks to extract sensitive information from learned representation. As such, the proxy adversary serves only to aid the learning process and is not an end goal by itself. Encoder E seeks to simultaneously aid the target predictor and limit the ability of the proxy adversary to extract sensitive information from the representation $\boldsymbol{z}$. By doing so, the encoder learns to remove sensitive information from the representation. In most ARL settings, while the encoder is a deep neural network, the target predictor and adversary are typically shallow neural networks.

The vanilla algorithm for learning the parameters of the encoder, target and adversary networks is gradient descent-ascent (GDA) [33, 40], where the players take a gradient step simultaneously. However, applying GDA, including its stochastic version, is not an optimal strategy for ARL and is known to suffer from many drawbacks. Firstly, GDA has undesirable convergence properties; it fails to converge to a local minimax and can converge to fixed points that are not local minimax, while being very unstable and slow in practice [8, 19]. Secondly, GDA exhibits strong rotation around fixed points, which requires using very small learning rates [3, 30] to converge. Numerous solutions [14, 30, 32] 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.

In this paper, we take a different perspective and propose an alternative solution for adversarial representation learning. Our key insight is to replace the shallow neural networks with other analytically tractable models with similar capacity. We propose to adopt simple learning algorithms that admit closed-form solutions, such as linear or kernel ridge regressors for the target and adversary, while modeling the encoder as a deep neural network. 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 obtain upper bound on optimal dimensionality of the embedding $\boldsymbol{z}$, (3) the simplicity and differentiability allows us to backpropagate through the closed-form solution, (4) practically it resolves the notorious rotational behaviour of iterative minimax gradient dynamics, resulting in a simple optimization that is empirically stable, reliable, converges faster to a local optima, and ultimately results in a more effective encoder E.

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.

Notation: Scalars are denoted by regular lower case or Greek letters, e.g., n, λ. Vectors are boldface lowercase letters, e.g., $\boldsymbol{x}$, $\boldsymbol{y}$; Matrices are uppercase boldface letters, e.g., $\boldsymbol{X}$. A $n \times n$ identity matrix is denoted by $\boldsymbol{I}$, sometimes with a subscript indicating its size, e.g., $\boldsymbol{I}_n$. Centered (mean subtracted w.r.t columns) data matrix is indicated by "~", e.g.,

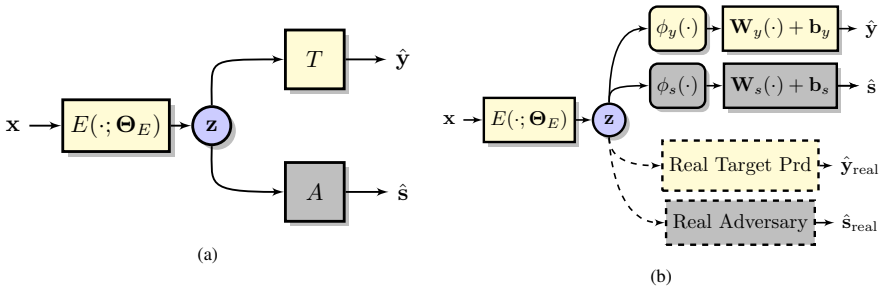


Fig. 1: **Adversarial Representation Learning:** (a) Consists of three players, an encoder E that obtains a compact representation $\boldsymbol{z}$ of input data $\boldsymbol{x}$, predictors T and S that seek to extract a desired target $\boldsymbol{y}$ and sensitive $\boldsymbol{s}$ attribute, respectively from the embedding. (b) OptNet-ARL adopts kernel regressors as proxy target predictor and adversary for learning the encoder. The learned encoder is evaluated against a real target predictor and adversary, which potentially can be neural networks.

$\tilde{\boldsymbol{X}}$. Assume that $\boldsymbol{X}$ contains n columns, then $\tilde{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{D}$ where $\boldsymbol{D} = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ and $\mathbf{1}$ denotes a vector of ones with length of n. Given matrix $\boldsymbol{M} \in \mathbb{R}^{m\times m}$, we use $\mathrm{Tr}[\boldsymbol{M}]$ to denote its trace (i.e., the sum of its diagonal elements); its Frobenius norm is denoted by $\|\boldsymbol{M}\|_F$, which is related to the trace as $\|\boldsymbol{M}\|_F^2 = \mathrm{Tr}[\boldsymbol{M}\boldsymbol{M}^T]$. The pseudo-inverse of $\boldsymbol{M}$ is denoted by $\boldsymbol{M}^\dagger$. The subspace spanned by the columns of $\boldsymbol{M}$ is denoted by $\mathcal{R}(M)$ or simply $\mathcal{M}$ (in calligraphy); the orthogonal complement of $\mathcal{M}$ is denoted by $\mathcal{M}^\perp$. The orthogonal projector onto $\mathcal{M}$ is denoted by $P_{\mathcal{M}}$.

2 Prior Work

Adversarial Representation Learning: The basic idea of learning data representations with controllable semantic information has been effective across multiple topics. Domain adaptation [12, 13, 38], where the goal is to learn representations that are invariant to the domain, is one of the earliest applications of ARL. More recently, adversarial learning has been extensively used [5–7, 10, 11, 33, 37, 40, 42] and advocated [29] for the task of learning fair, invariant or privacy preserving representations of data. All of the aforementioned approaches represent each entity in ARL by neural networks and optimize their parameters through stochastic gradient descent-ascent (SGDA). As we show in this paper, SGDA is unstable and sub-optimal for learning. Therefore, we trade-off model expressively for ease of learning through a hybrid approach of modeling the encoder by a deep neural network and target and adversary with closed-form regressors. Such a solution reduces alternating optimization into a simple optimization problem which is much more stable, reliable and effective. Table 1 shows a comparative summary of ARL approaches.

Optimization in Minmax Games: A growing class of learning algorithms, including ARL, GANs etc., involve more than one objective and are trained via games played by cooperating or dueling neural networks. An overview of the challenges presented

Table 1: Comparison between different ARL methods (n: sample size, b: batch size).

Method	Encoder / Target & Adversary	Optimization	Scalability	Enc Soln	Input Data
SGDA-ARL [29, 40]	deep NN / shallow NN	alternating SGD	$\geq \mathcal{O}(b^3)$	unknown	raw data
Kernel-SARL [35]	kernel regressor / linear	closed-form	$\mathcal{O}(n^3)$	global optima	features
OptNet-ARL (ours)	deep NN / kernel regressor	SGD	$\mathcal{O}(b^3)$	local optima	raw data

by such algorithms and a plausible solution in general n-player games can be found in [26]. In the context of two-player minimax games such as GANs, a number of solutions [3, 8, 14, 19, 30, 32] have been proposed to improve the optimization dynamics, many of them relying on the idea of taking an extrapolation step [23]. For example, [30] deploys some regularizations to encourage agreement between different players and improve the convergence properties. In another example, [32] uses double gradient to stabilize the optimization procedure. In contrast to all of these approaches, that work with the given fixed models for each player, we seek to change the model of the players in the ARL setup for ease of optimization. In the context of ARL, [35] considers the setting where all the players, including the encoder, are linear regressors. While they obtained a globally optimum solution, the limited model capacity hinders the flexibility (cannot directly use raw data), scalability and performance (limited by pre-trained features) of their solution. In this paper we advocate the use of ridge regressors (linear or kernel) for the target and adversary, while modeling the encoder as a deep neural network. This leads to a problem that obviates the need for gradient descent-ascent and can instead be easily optimized with standard SGD. Not only does this approach lead to stable optimization, it also scales to larger datasets and exhibits better empirical performance.

Differentiable Solvers: 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 [17]. Similarly [39] proposed an asymmetric architecture which incorporates a Correlation Filter as a differentiable layer. Differential optimization as a layer in neural networks was introduced by [1, 2]. More recently, differentiable solvers have also been adopted for meta-learning [4, 25] as well. The primary motivation for all the aforementioned approaches is to endow deep neural networks with differential optimization and ultimately achieve faster convergence of the end-to-end system. 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.

3 Problem Setting

Let the data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$ be n realizations of d-dimensional data, $\boldsymbol{x} \in \mathbb{R}^d$. Similarly, we denote n realizations of sensitive attribute vector $\boldsymbol{s} \in \mathbb{R}^q$ and target attribute vector $\boldsymbol{y} \in \mathbb{R}^p$ by matrices $\boldsymbol{S} = [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n]$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n]$, respectively. Treating the attributes as vectors enables us to consider both multi-class classification and regression under the same formulation. Each data sample $\boldsymbol{x}_k$ is associated with the sensitive attribute $\boldsymbol{s}_k$ and the target attribute $\boldsymbol{y}_k$, respectively.

The ARL problem is formulated with the goal of learning parameters of an embedding function $E(\cdot;\boldsymbol{\Theta}_E)$ that maps a data sample $\boldsymbol{x}$ to $\boldsymbol{z} \in \mathbb{R}^r$ with two objectives: (i) aiding a target predictor $T(\cdot;\boldsymbol{\Theta}_y)$ to accurately infer the target attribute $\boldsymbol{y}$ from $\boldsymbol{z}$, and (ii) preventing an adversary $A(\cdot;\boldsymbol{\Theta}_s)$ from inferring the sensitive attribute $\boldsymbol{s}$ from $\boldsymbol{z}$. The ARL problem can be formulated as a bi-level optimization,

$$\min_{\boldsymbol{\Theta}_E} \min_{\boldsymbol{\Theta}_y} \mathcal{L}_y\left(T(E(\boldsymbol{x};\boldsymbol{\Theta}_E);\boldsymbol{\Theta}_y),\boldsymbol{y}\right) \quad \text{s.t.} \quad \min_{\boldsymbol{\Theta}_s} \mathcal{L}_s\left(A(E(\boldsymbol{x};\boldsymbol{\Theta}_E);\boldsymbol{\Theta}_s),\boldsymbol{s}\right) \geq \alpha \tag{1}$$

where $\mathcal{L}_y$ and $\mathcal{L}_s$ are the loss functions (averaged over the training dataset) for the target predictor and the adversary, respectively; $\alpha \in (0,\infty)$ is a user defined value that determines the minimum tolerable loss α for the adversary on the sensitive attribute; and the minimization in the constraint is equivalent to the encoder operating against an optimal adversary. Denote the global minimums of the adversary and target estimators as

$$\begin{aligned} J_y(\boldsymbol{\Theta}_E) &:= \min_{\boldsymbol{\Theta}_y} \mathcal{L}_y\left(T(E(\boldsymbol{x};\boldsymbol{\Theta}_E);\boldsymbol{\Theta}_y),\boldsymbol{y}\right) \\ J_s(\boldsymbol{\Theta}_E) &:= \min_{\boldsymbol{\Theta}_s} \mathcal{L}_s\left(A(E(\boldsymbol{x};\boldsymbol{\Theta}_E);\boldsymbol{\Theta}_s),\boldsymbol{s}\right). \end{aligned} \tag{2}$$

The constrained optimization problem in (1) can be alternately solved through its Lagrangian version:

$$\min_{\boldsymbol{\Theta}_E} \Big\{(1-\lambda)J_y(\boldsymbol{\Theta}_E) - \lambda J_s(\boldsymbol{\Theta}_E)\Big\},\ 0 \leq \lambda \leq 1. \tag{3}$$

3.1 Motivating Exact Solvers

Most state-of-the-art ARL algorithms cannot solve the optimization problems in (2) optimally (e.g., SGDA). For any given $\boldsymbol{\Theta}_E$, denote the non-optimal adversary and target predictors loss functions as $J_y^{\text{approx}}(\boldsymbol{\Theta}_E)$ and $J_s^{\text{approx}}(\boldsymbol{\Theta}_E)$, respectively. It is obvious that for any given $\boldsymbol{\Theta}_E$, it holds

$$J_y^{\text{approx}}(\boldsymbol{\Theta}_E) \geq J_y(\boldsymbol{\Theta}_E) \quad \text{and} \quad J_s^{\text{approx}}(\boldsymbol{\Theta}_E) \geq J_s(\boldsymbol{\Theta}_E).$$

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

$$\min_{\boldsymbol{\Theta}_E} \Big\{(1-\lambda)J_y^{\text{approx}}(\boldsymbol{\Theta}_E) - \lambda J_s^{\text{approx}}(\boldsymbol{\Theta}_E)\Big\}, 0 \leq \lambda \leq 1. \tag{4}$$

Intuitively, solution(s) of (4) do not outperform that of (3). We now formulate this intuition more concretely.

Definition 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) if and only if $b_1 > a_1$ and $b_2 < a_2$ hold simultaneously.*

Theorem 2. *For any $\lambda_1,\ \lambda_2 \in [0,1]$, consider the following optimization problems*

$$\boldsymbol{\Theta}_E^{exact} = \arg\min_{\boldsymbol{\Theta}_E} \Big\{(1-\lambda_1)J_y(\boldsymbol{\Theta}_E) - \lambda_1 J_s(\boldsymbol{\Theta}_E)\Big\} \tag{5}$$

and

$$\boldsymbol{\Theta}_E^{approx} = \arg\min_{\boldsymbol{\Theta}_E} \Big\{(1-\lambda_2)J_y^{approx}(\boldsymbol{\Theta}_E) - \lambda_2 J_s^{approx}(\boldsymbol{\Theta}_E)\Big\}$$

Then, any adversary-target objective trade-off generated by $\big(J_s(\boldsymbol{\Theta}_E^{exact}), J_y(\boldsymbol{\Theta}_E^{exact})\big)$ *cannot be dominated by the trade-off generated by* $\big(J_s(\boldsymbol{\Theta}_E^{approx}), J_y(\boldsymbol{\Theta}_E^{approx})\big)$.

See supplementary material for the proof of all Lemmas and Theorems.

4 Approach

Existing instances of ARL adopt deep neural networks to represent E, T and A and learn their respective parameters $\{\boldsymbol{\Theta}_E, \boldsymbol{\Theta}_y, \boldsymbol{\Theta}_s\}$ through stochastic gradient descent-ascent (SGDA). Consequently, the target and adversary in Eq. 2 are not solved to optimality, thereby resulting in a sub-optimal encoder.

4.1 Closed-Form Adversary and Target Predictor

The machine learning literature offers a wealth of methods with exact solutions that are appropriate for modeling both the adversary and target predictors. In this paper, we argue for and adopt simple, fast and differentiable methods such as kernel ridge regressors as shown in Fig. 1b. On one hand, such modeling allows us to obtain the optimal estimators globally for any given encoder $E(\cdot; \boldsymbol{\Theta}_E)$.

On the other hand, kernelized ridge regressors can be stronger than the shallow neural networks that are used in many ARL-based solutions (e.g., [11, 29, 33, 40]). Although it is not the focus of this paper, it is worth noting that even deep neural networks in the infinite-width limit reduce to linear models with a kernel called the neural tangent kernel [18], and as such can be adopted to increase the capacity of our regressors.

Consider two reproducing kernel Hilbert spaces (RKHS) of functions $\mathcal{H}_s$ and $\mathcal{H}_y$ for adversary and target regressors, respectively. 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 kernels for $\mathcal{H}_s$ and $\mathcal{H}_y$ can be represented as $k_s(\boldsymbol{z}_1, \boldsymbol{z}_2) = \langle \phi_s(\boldsymbol{z}_1), \phi_s(\boldsymbol{z}_2) \rangle_{\mathcal{H}_s}$ and $k_y(\boldsymbol{z}_1, \boldsymbol{z}_2) = \langle \phi_y(\boldsymbol{z}_1), \phi_y(\boldsymbol{z}_2) \rangle_{\mathcal{H}_y}$. Under this setting, we can relate the target and sensitive attributes to any given embedding $\boldsymbol{z}$ as,

$$\hat{\boldsymbol{y}} = \boldsymbol{W}_y \phi_y(\boldsymbol{z}) + \boldsymbol{b}_y, \qquad \hat{\boldsymbol{s}} = \boldsymbol{W}_s \phi_s(\boldsymbol{z}) + \boldsymbol{b}_s \tag{6}$$

where $\boldsymbol{\Theta}_y = \{\boldsymbol{W}_y, \boldsymbol{b}_y\}$ and $\boldsymbol{\Theta}_s = \{\boldsymbol{W}_s, \boldsymbol{b}_s\}$ are the regression parameters, $\boldsymbol{W}_y \in \mathbb{R}^{p \times r_y}$ and $\boldsymbol{W}_s \in \mathbb{R}^{q \times r_s}$, $\boldsymbol{b}_y \in \mathbb{R}^p$ and $\boldsymbol{b}_s \in \mathbb{R}^q$ respectively.

Let the entire embedding of input data be denoted as $\boldsymbol{Z} := [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n]$ and the corresponding features maps as $\boldsymbol{\Phi}_y := [\phi_y(\boldsymbol{z}_1), \cdots, \phi_y(\boldsymbol{z}_n)]$ and $\boldsymbol{\Phi}_s := [\phi_s(\boldsymbol{z}_1), \cdots, \phi_s(\boldsymbol{z}_n)]$, respectively. Furthermore, we denote the associated Gram matrices by $\boldsymbol{K}_y = {\boldsymbol{\Phi}_y}^T \boldsymbol{\Phi}_y$ and $\boldsymbol{K}_s = {\boldsymbol{\Phi}_s}^T \boldsymbol{\Phi}_s$. A centered Gram matrix $\tilde{\boldsymbol{K}}$ corresponding to the Gram matrix $\boldsymbol{K}$ can be obtained [16] as,

$$\tilde{\boldsymbol{K}} = \tilde{\boldsymbol{\Phi}}^T \tilde{\boldsymbol{\Phi}} = (\boldsymbol{\Phi}\boldsymbol{D})^T (\boldsymbol{\Phi}\boldsymbol{D}) = \boldsymbol{D}^T \boldsymbol{K} \boldsymbol{D}. \tag{7}$$

Invoking the representer theorem [36], the regression parameters can be represented as $\boldsymbol{W}_y = \boldsymbol{\Lambda}_y \tilde{\boldsymbol{\Phi}}_y^T$ and $\boldsymbol{W}_s = \boldsymbol{\Lambda}_s \tilde{\boldsymbol{\Phi}}_s^T$ for target and adversary respectively, where $\boldsymbol{\Lambda}_y \in$

$\mathbb{R}^{p\times n}$ and $\boldsymbol{\Lambda}_s \in \mathbb{R}^{n\times q}$ are new parameter matrices. As a result, the kernelized regressors in (6) can be equivalently expressed as

$$\hat{y} = \boldsymbol{\Lambda}_y \tilde{\boldsymbol{\Phi}}_y^T \phi_y(\boldsymbol{z}) + \boldsymbol{b}_y, \quad \hat{s} = \boldsymbol{\Lambda}_s \tilde{\boldsymbol{\Phi}}_s^T \phi_s(\boldsymbol{z}) + \boldsymbol{b}_s. \tag{8}$$

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 kernelized adversary and target predictors for any given embedding $\boldsymbol{Z}$.

Lemma 3. *Let $J_y(\boldsymbol{Z})$ and $J_s(\boldsymbol{Z})$ be regularized minimum MSEs for adversary and target:*

$$J_y(\boldsymbol{Z}) = \min_{\boldsymbol{\Lambda}_y, \boldsymbol{b}_y} \Big\{ \mathbb{E}\big\{ \big\| \hat{\boldsymbol{y}} - \boldsymbol{y} \big\|^2 \big\} + \gamma_y \big\| \boldsymbol{\Lambda}_y \big\|_F^2 \Big\},$$
$$J_s(\boldsymbol{Z}) = \min_{\boldsymbol{\Lambda}_s, \boldsymbol{b}_s} \Big\{ \mathbb{E}\big\{ \big\| \hat{\boldsymbol{s}} - \boldsymbol{s} \big\|^2 \big\} + \gamma_s \big\| \boldsymbol{\Lambda}_s \big\|_F^2 \Big\}$$

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 kernelized adversary and target can be obtained as

$$J_y(\boldsymbol{Z}) = \frac{1}{n} \big\| \tilde{\boldsymbol{Y}} \big\|_F^2 - \frac{1}{n} \left\| P_{\mathcal{M}_y} \begin{bmatrix} \tilde{\boldsymbol{Y}}^T \\ \boldsymbol{0}_n \end{bmatrix} \right\|_F^2,$$
$$J_s(\boldsymbol{Z}) = \frac{1}{n} \big\| \tilde{\boldsymbol{S}} \big\|_F^2 - \frac{1}{n} \left\| P_{\mathcal{M}_s} \begin{bmatrix} \tilde{\boldsymbol{S}}^T \\ \boldsymbol{0}_n \end{bmatrix} \right\|_F^2 \tag{9}$$

where

$$\boldsymbol{M}_y = \begin{bmatrix} \tilde{\boldsymbol{K}}_y \\ \sqrt{n\gamma_y} \boldsymbol{I}_n \end{bmatrix}, \qquad \boldsymbol{M}_s = \begin{bmatrix} \tilde{\boldsymbol{K}}_s \\ \sqrt{n\gamma_s} \boldsymbol{I}_n \end{bmatrix}$$

are both full column rank matrices and a projection matrix for any full column rank matrix $\boldsymbol{M}$ is

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

It is straightforward to generalize this method to the case of multiple target and adversary predictors through equation (3). In this case we will have multiple λ's to trade-off between fairness and utility.

4.2 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 neural networks, 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. *Let $\boldsymbol{z}$ in Figure 1b 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. Then, for any $0 \leq \lambda \leq 1$ the optimal dimensionality of embedding vector, r is the number of negative eigenvalues of*

$$\boldsymbol{B} = \lambda \tilde{\boldsymbol{S}}^T \tilde{\boldsymbol{S}} - (1-\lambda)\tilde{\boldsymbol{Y}}^T \tilde{\boldsymbol{Y}}. \tag{10}$$

Given a dataset with the target and sensitive labels, $\boldsymbol{Y}$ and $\boldsymbol{S}$ respectively, the matrix $\boldsymbol{B}$ 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. One large datasets where $\boldsymbol{B} \in \mathbb{R}^{n \times n}$, the Nyström method with data sampling [24] can be adopted.

4.3 Gradient of Closed-Form Solution

In order to find the gradient of the encoder loss function in (3) with J_y and J_s given in (9), we can ignore the constant terms, $\|\tilde{\boldsymbol{Y}}\|_F$ and $\|\tilde{\boldsymbol{S}}\|_F$. Then, the optimization problem in (3) would be equivalent to

$$\begin{aligned} &\min_{\boldsymbol{\Theta}_E} \left\{ (1-\lambda) \left\| P_{\mathcal{M}_s} \begin{bmatrix} \tilde{\boldsymbol{S}}^T \\ \boldsymbol{0}_n \end{bmatrix} \right\|_F^2 - \lambda \left\| P_{\mathcal{M}_y} \begin{bmatrix} \tilde{\boldsymbol{Y}}^T \\ \boldsymbol{0}_n \end{bmatrix} \right\|_F^2 \right\} \\ &= \min_{\boldsymbol{\Theta}_E} \left\{ (1-\lambda) \sum_{k=1}^{p} \|P_{\mathcal{M}_s} \boldsymbol{u}_s^k\|^2 - \lambda \sum_{m=1}^{q} \|P_{\mathcal{M}_y} \boldsymbol{u}_y^m\|^2 \right\} \end{aligned} \tag{11}$$

where the vectors $\boldsymbol{u}_s^k$ and $\boldsymbol{u}_y^m$ are the k-th and m-th columns of $\begin{bmatrix} \tilde{\boldsymbol{S}}^T \\ \boldsymbol{0}_n \end{bmatrix}$ and $\begin{bmatrix} \tilde{\boldsymbol{Y}}^T \\ \boldsymbol{0}_n \end{bmatrix}$, respectively. Let $\boldsymbol{M}$ be an arbitrary matrix function of $\boldsymbol{\Theta}_E$, and θ be arbitrary scalar element of $\boldsymbol{\Theta}_E$. Then, from [15] we have

$$\frac{1}{2} \frac{\partial \|P_{\mathcal{M}} \boldsymbol{u}\|^2}{\partial \theta} = \boldsymbol{u}^T P_{\mathcal{M}^\perp} \frac{\partial \boldsymbol{M}}{\partial \theta} \boldsymbol{M}^\dagger \boldsymbol{u} \tag{12}$$

where

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

Equation (12) can be directly used to obtain the gradient of objective function in (11).

Directly computing the gradient in Eq (12) requires a pseudoinverse 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 [24] 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 neural networks are of the order of 10s to 1000s, 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 [45].

5 Experiments

In this section we will evaluate the efficacy of our proposed approach, OptNet-ARL, on three different tasks; Fair Classification on UCI [9] datatset, 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 [11, 20, 29, 33, 40] etc., (2) **ExtraSGDA-ARL:** a state-of-the-art variant of stochastic gradient descent-ascent that uses an extra gradient step [23] for optimizing minimax games. Specifically, we use the ExtraAdam algorithm from [14], and (3) **SARL:** a global optimum solution for a kernelized regressor encoder and linear target and adversary [35]. Specifically, **hypervolume** (HV) [43], 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 the stability.

In the training stage, the encoder, a deep neural network, is optimized **end-to-end** against kernel ridge regressors (RBF Gaussian kernel[1]) in the case of OptNet-ARL and multi-layer perceptrons (MLPs) for the baselines. Table 2 summarizes the network architecture of all experiments. We note that the optimal embedding dimensionality, r for binary target is equal to one which is consistent with Fisher's linear discriminant analysis [44]. The embedding is instance normalized (unit norm). So we adopted a fixed value of $\sigma = 1$ for 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 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 encoder) and for different trade-off parameters $\lambda \in [0, 1]$. We report the median and standard deviation across the five repetitions.

[1] $k(\boldsymbol{z}, \boldsymbol{z}') = \exp\left(-\frac{\|\boldsymbol{z}-\boldsymbol{z}'\|^2)}{2\sigma^2}\right.$

Table 2: Network Architectures in Experiments.

Method (ARL)	Encoder	Embd Dim	Target (Train)	Adversary (Train)	Target (Test)	Adversary (Test)
			Adult			
SGDA [29, 40]	MLP-4-2	1	MLP-4	MLP-4	MLP-4-2	MLP-4-2
ExtraSGDA [23]	MLP-4-2	1	MLP-4	MLP-4	MLP-4-2	MLP-4-2
SARL [35]	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 [29, 40]	MLP-4	1	MLP-2	MLP-2	logistic	logistic
ExtraSGDA [23]	MLP-4	1	MLP-2	MLP-2	logistic	logistic
SARL [35]	RBF krnl	1	linear	linear	logistic	logistic
OptNet-ARL (ours)	MLP-4	1	RBF krnl	RBF krnl	logistic	logistic
			CelebA			
SGDA [29, 40]	ResNet-18	128	MLP-64	MLP-64	MLP-32-16	MLP-32-16
ExtraSGDA [23]	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 [29, 40]	MLP-8-4	2	MLP-8-4	MLP-8-4	MLP-4-4	MLP-4-4
ExtraSGDA [23]	MLP-8-4	2	MLP-8-4	MLP-8-4	MLP-4-4	MLP-4-4
SARL [35]	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

5.1 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 target and sensitive attributes are 0.03 and 0.02 for the Adult and German dataset, 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 loss in accuracy for the target task.

Stability: Since there is no trade-off between the two attributes, we compare stability by reporting the median and standard deviation of the target and adversary performance in Table 3. Our results indicate that OptNet-ARL achieves a higher accuracy for target task and lower leakage of sensitive attribute and with less variance. For instance, in Adult dataset, OptNet-ARL method achieves 83.86% and 83.81% target accuracy with almost zero sensitive leakage. For OptNet-ARL the standard deviation of 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%).

Fair Classification Performance: We compare our proposed approach with many baseline results on these datasets. The optimal dimensionality for OptNet-ARL is $r = 1$ as determined by Theorem 4 and $r = 50$ for the baselines (common choice in previous work). Diff value in Table 3 shows the difference between adversary accuracy and ran-

Table 3: Fair Classification On UCI Dataset (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 [41]	82.3	67.0	0.4	72.3	80.5	0.5
AEVB [21]	81.9	66.0	1.4	72.5	79.5	1.5
VFAE [28]	81.3	67.0	0.4	72.7	79.7	1.3
SARL [35]	84.1	67.4	0.0	76.3	80.9	0.1
SGDA-ARL [40]	83.61 ± 0.38	67.08 ± 0.48	0.40	76.53 ± 1.07	87.13 ± 5.70	6.13
ExtraSGDA-ARL [14]	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

dom guessing. On both datasets, both Linear-SARL and OptNet-ARL can achieve high performance on target task with a tiny sensitive attribute leakage for the German dataset.

5.2 Mitigating Sensitive Information Leakage

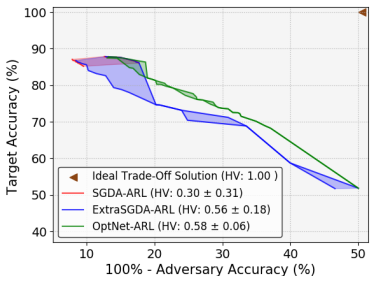
The CelebA dataset [27] contains $202,599$ face images of $10,177$ celebrities. Each image contains 40 different binary attributes (e.g., gender, emotion, age, etc.). Images are pre-processed and aligned to a fixed size of 112×96 and we use the official train-test splits. The target task is defined as 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 [35] 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 but 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 2.

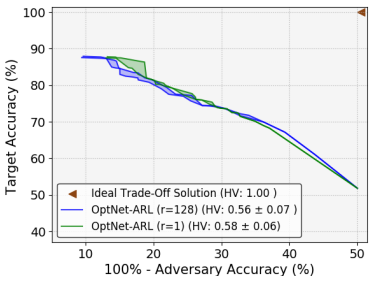
Stability and Trade-off: Figure 2(a) shows the attainment surface [22] and hypervolume [43] (median and standard deviation) for all methods. SGDA spans only a small part of the trade-off and at the same time exhibits large variance around the median curves. Overall both baselines are unstable and unreliable when the two attributes are dependent on each other. On the other hand, OptNet-ARL solutions are very stable and while also achieving a better trade-off between target and adversary accuracy.

Optimal Embedding Dimensionality: Figure 2(b) compares the utility-bias trade-off 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 2(c). ExtraSGDA-ARL takes an extra step to update the weights and therefore, it is slightly slower than SGDA-ARL. OptNet-ARL



(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)

Fig. 2: **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).

on the other hand is significantly faster to obtain 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 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 target task. The results are presented in Figure 2 (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 loss of target task. 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.

5.3 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 $\boldsymbol{\Sigma} = 0.2^2\boldsymbol{I}_2$. Denote by $f(\boldsymbol{x})$ the distribution of input data. Then,

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

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 testing set independently. For visualization, the testing set is shown in Figure 3(a). In this illustrative dataset , the correlation between input data and color is 0.61 and therefore there is no encoder which results in full target performance at no leakage of sensitive attribute. Network architecture details are shown in Table 2.

Stability and Trade-off: Figure 3(b) illustrates the five-run attainment surfaces and median hypervolumes for all methods. Since the the dimensionality of both input and output is 2, the optimal embedding dimensionality is equal to 2 which we set it 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-sized 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. Addition to Gaussian kernel, we also used inverse multi quadratic (IMQ) kernel [46][2] for OptNet to examine the effect kernel of function. As we observe from Figure 3(b), IMQ-OptNet-ARL performs almost similar to OptNet-ARR with Gaussian kernel in terms of both trade-off and stability.

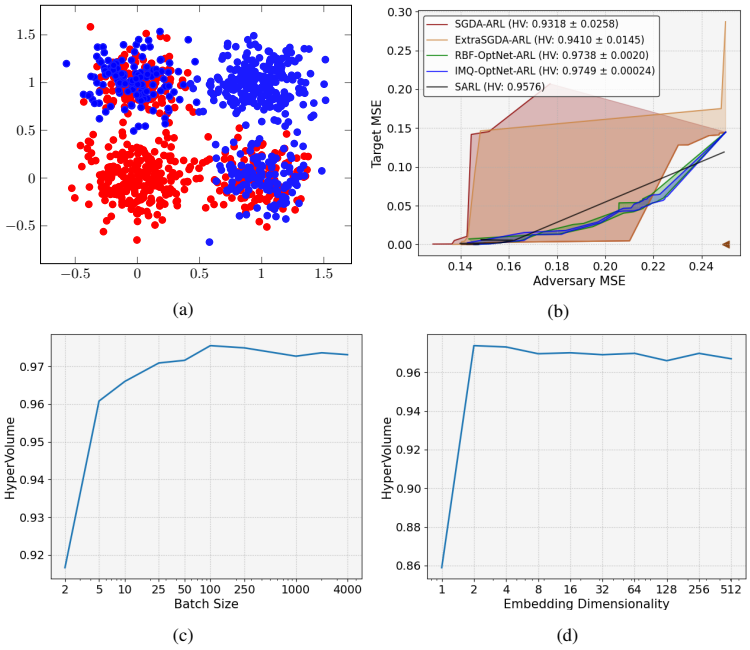
Batch Size: In order to examine the effect of batch size on OptNet-ARL (with Gaussian kernel), we train the encoder with different values of batch size between 2 and 4000 (entire training data). The results are illustrated in Figure 3(c). We observer that the trade-off HV is quite insensitive to batch sizes greater than 25 which implies that the gradient of min-batch is an accurately enough estimator of the gradient of entire data.

Embedding Dimensionality: We also study the effect of embedding dimensionality (r) by examining different values for r in $[1, 512]$ using RBF-OptNet-ARL. The results are illustrated in Figure 3(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, $r = 2$.

6 Concluding Remarks

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 stochastic gradient descent-ascent (SGDA) and their variants

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



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

Fig. 3: **Mixture of Gaussians:** (a) Input data. The target task is to learn a representation which is informative enough to reconstruct the input data and at the same time hide the color information (● vs ●). (b) Trade-off between the MSEs of adversary and target task for different ARL methods. (c) The HVs of OptNet-ARL (Gaussian kernel) vs different batch size values in $[2, 4000]$. (d) The HV values of OptNet-ARL (Gaussian kernel) vs different values of r in $[1, 512]$.

are sub-optimal, unstable and unreliable in practice. In this paper, 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 out-performs existing ARL approaches on both small and large scale datasets.

Acknowledgements: This work was performed under the following financial assistance award 60NANB18D210 from U.S. Department of Commerce, National Institute of Standards and Technology.

References

1. Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., Kolter, Z. Differentiable convex optimization layers. In: Advances in Neural Information Processing Systems (2019)
2. Amos, B., Kolter, J. Z. Optnet: Differentiable optimization as a layer in neural networks. In: International Conference on Machine Learning (2017)

3. Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., Graepel, T. The mechanics of n-player differentiable games. In: International Conference on Machine Learning (2018)
4. Bertinetto, L., Henriques, J. F., Torr, P. H. Meta-learning with differentiable closed-form solvers. In: International Conference on Learning Representations (2018)
5. Bertran, M., Martinez, N., Papadaki, A., Qiu, Q., Rodrigues, M., Reeves, G., Sapiro, G. Adversarially Learned Representations for Information Obfuscation and Inference.In: International Conference on Machine Learning (2019)
6. Beutel, A., Chen, J., Zhao, Z., Chi, E. H. Data decisions and theoretical implications when adversarially learning fair representations. Fairness, Accountability, and Transparency in Machine Learning (2017)
7. Creager, E., Madras, D., Jacobsen, J. H., Weis, M., Swersky, K., Pitassi, T., Zemel, R. Flexibly fair representation learning by disentanglement. In: International Conference on Machine Learning, pp. 1436-1445 (2019)
8. Daskalakis, C., Panageas, I. The limit points of (optimistic) gradient descent in min-max optimization. In: Advances in Neural Information Processing Systems (2018)
9. UCI machine learning repository, `http://archive.ics.uci.edu/ml`.
10. Edwards, H., Storkey, A. Censoring representations with an adversary. In: International Conference on Learning Representations (2015)
11. Elazar, Y., Goldberg, Y. Adversarial removal of demographic attributes from text data. In: Empirical Methods in Natural Language Processing (2018)
12. Ganin, Y., Lempitsky, V. Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp. 1180-1189 (2015)
13. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F.,Marchand, M., Lempitsky, V. Domain-adversarial training of neural networks.Journal of Machine Learning Research, **17** (1), pp. 2096-2030 (2016)
14. Gidel, G., Berard, H., Vignoud, G., Vincent, P., Lacoste-Julien, S. A variational inequality perspective on generative adversarial networks. In: International Conference on Learning Representations (2019)
15. Golub, G. H., Pereyra, V. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. SIAM Journal on Numerical Analysis, **10** (2), pp 413-432 (1973)
16. Gretton, A., Herbrich, R., Smola, A., Bousquet, O., Schölkopf, B. Kernel methods for measuring. Journal of Machine Learning Research independence, **6** , pp. 2075-2129 (2005)
17. Ionescu, C., Vantzos, O., Sminchisescu, C. Training deep networks with structured layers by matrix backpropagation. In: IEEE International Conference on Computer Vision (2015)
18. Jacot, A., Gabriel, F., Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In: Advances in Neural Information Processing Systems (2018)
19. Jin, C., Netrapalli, P., Jordan, M. What is Local Optimality in Nonconvex-Nonconcave Minimax Optimization ? arXiv preprint arXiv:1902.00618 (2019)
20. Kim, B., Kim, H., Kim, K., Kim, S., Kim, J. Learning not to learn: Training deep neural networks with biased data. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
21. Kingma, D. P., Welling, M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
22. Knowles, J. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In: International Conference on Intelligent Systems Design and Applications (2015)
23. Korpelevich, G. M. The extragradient method for finding saddle points and other problems. Matecon, **12**, pp. 747-756 (1976)
24. Kumar, S., Mohri, M., Talwalkar, A. Sampling Methods for the Nyström Method. Journal of Machine Learning Research, **13** (4), pp.981-1006 (2012)

25. Lee, K., Maji, S., Ravichandran, A., Soatto, S. Meta-learning with differentiable convex optimization. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
26. Letcher, A., Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., Graepel, T. Differentiable Game Mechanics. Journal of Machine Learning Research, **20** (84), pp. 1-40 (2019)
27. Liu, Z., Luo, P., Wang, X., Tang, X. Deep learning face attributes in the wild. In: IEEE International Conference on Computer Vision (2015)
28. Louizos, C., Swersky, K., Li, Y., Welling, M., Zemel, R. The variational fair autoencoder. arXiv preprint arXiv:1511.00830 (2015)
29. Madras, D., Creager, E., Pitassi, T., Zemel, R. Learning adversarially fair and transferable representations. In: International Conference on Machine Learning (2018)
30. Mescheder, L., Nowozin, S., Geiger, A. The numerics of gans. In: Advances in Neural Information Processing Systems (2017)
31. Moyer, D., Gao, S., Brekelmans, R., Steeg, G. V., Galstyan, A. Invariant Representations without Adversarial Training, In: Advances in Neural Information Processing Systems (2018)
32. Nagarajan, V., Kolter, J. Z. Gradient descent GAN optimization is locally stable. In: Advances in Neural Information Processing Systems (2017)
33. Roy, P. C., Boddeti, V. N. Mitigating Information Leakage in Image Representations: A Maximum Entropy Approach. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
34. Sadeghi, B., Boddeti, V. N. Imparting fairness to pre-trained biased representations. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (2020)
35. Sadeghi, B., Yu, R., Boddeti, V. On the Global Optima of Kernelized Adversarial Representation Learning. In: IEEE International Conference on Computer Vision (2019)
36. Shawe-Taylor, J., Cristianini, N. Kernel methods for pattern analysis. Cambridge University Press (2014)
37. Song, J., Kalluri, P., Grover, A., Zhao, S., Ermon, S. Learning controllable fair representations. In: International Conference on Artificial Intelligence and Statistics (2019)
38. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T. Adversarial discriminative domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
39. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P. H. End-to-end representation learning for correlation filter based tracking. In: IEEE conference on computer vision and pattern recognition (2017)
40. Xie, Q., Dai, Z., Du, Y., Hovy, E., Neubig, G. Controllable invariance through adversarial feature learning. In: Advances in Neural Information Processing Systems (2017)
41. Zemel, R., Wu, Y., Swersky, K., Pitassi, T., Dwork, C. Learning fair representations. In: International Conference on Machine Learning (2013)
42. Zhang, B. H., Lemoine, B., Mitchell, M. Mitigating unwanted biases with adversarial learning. In: AAAI/ACM Conference on AI, Ethics, and Society (2018)
43. Zitzler, E., Thiele, L. Multiobjective optimization using evolutionary algorithms—a comparative case study. In: International conference on parallel problem solving from nature (1998)
44. Fisher, Ronald A. The use of multiple measurements in taxonomic problems. In: Annals of human eugenics.Wiley Online Library (1926)
45. Hardt, M., Recht, B., Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In: International conference on machine learning (2016)
46. Souza, César R. Kernel functions for machine learning applications. In: Creative commons attribution-noncommercial-share alike (2016)