Kernelized Concept Erasure

Shauli Ravfogel[1,2] **Francisco Vargas**[3] **Yoav Goldberg**[1,2] **Ryan Cotterell**[4]
[1]Bar-Ilan University [2]Allen Institute for Artificial Intelligence
[3]University of Cambridge [4]ETH Zürich
{shauli.ravfogel, yoav.goldberg}@gmail.com
fav25@cam.ac.uk ryan.cotterell@inf.ethz.ch

Abstract

The representation space of neural models for textual data emerges in an unsupervised manner during training. Understanding how those representations encode human-interpretable concepts is a fundamental problem. One prominent approach for the identification of concepts in neural representations is searching for a linear subspace whose erasure prevents the prediction of the concept from the representations. However, while many linear erasure algorithms are tractable and interpretable, neural networks do not necessarily represent concepts in a linear manner. To identify non-linearly encoded concepts, we propose a kernelization of a linear minimax game for concept erasure. We demonstrate that it is possible to prevent specific nonlinear adversaries from predicting the concept. However, the protection does not transfer to different nonlinear adversaries. Therefore, exhaustively erasing a non-linearly encoded concept remains an open problem.

https://github.com/shauli-ravfogel/adv-kernel-removal

1 Introduction

Large neural networks in NLP produce real-valued representations that encode the bit of human language that they were trained on, e.g., words, sentences, or text grounded in images. For instance, GloVe (Pennington et al., 2014) produces real-valued representations of isolated words, BERT (Devlin et al., 2019) produces real-valued representations of sentences, and VilBERT produces real-valued representations of visually grounded language (Lu et al., 2019; Bugliarello et al., 2021). These real-valued representations naturally encode various properties of the objects they represent. For instance, a good representation of the first author's laptop computer ought to encode its manufacturer, size, and color somewhere among its real values.

We now describe the premise of our paper in more detail. We adopt the notion of a **concept** due to Gärdenfors (2000). For Gärdenfors, objects can be thought of as having a geometric representation. Different dimensions in the representation space that objects inhabit might correspond to their color, size, and shape. Gärdenfors then goes further and defines a **concept** as a convex region of the representation space.[1] Building on Gärdenfors' notion of a concept, this paper studies a task that we refer to as **concept erasure**.

We now motivate the task more formally by extending the example. Imagine we have N different laptops, whose real-valued representations are denoted as $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Now, consider concept labels $y_1, \ldots, y_N$ that encode each laptop's color and are taken from the set $\{\text{GREY}, \text{SILVER}, \text{BLACK}, \text{WHITE}\}$. In the concept erasure paradigm, we seek an **erasure function** $r(\cdot)$ such that $r(\mathbf{x}_1), \ldots, r(\mathbf{x}_N)$ are *no longer* predictive of the colors of the laptops, $y_1, \ldots, y_N$, but *retain* all the other information encoded in the original representations, i.e., they remain predictive with respect to other laptop-related concepts. Our hope is that the geometry of the erasure function $r(\cdot)$ then tells us the structure of the laptop color concept.

Concept erasure is tightly related to concept identification. Once we have successfully removed a given concept, e.g., color, from a representation $\mathbf{x}_n$, it is reasonable to argue that the erasure function r has meaningfully identified the concept within the representation space. In the rest of the paper, we will say that $r(\cdot)$ **neutralizes** the concept in the representation space. For instance, we say that the r in our example neutralizes the concept of a laptop's color. It follows that concept identification is related to bias mitigation (Bolukbasi et al., 2016; Gonen and Goldberg, 2019; Maudslay et al., 2019), e.g., one may want to identify and remove the gender encoded in learned word representations produced by an embedding method such as word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). Indeed, the empirical portion of this paper will focus on removing gender

[1] In this work, we exclusively focus on linear subspaces, which are convex, rather than more general convex regions.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6034–6055
December 7-11, 2022

from word representations in order to mitigate bias.

Previous work on concept erasure (Ravfogel et al., 2021) focuses on the *linear* case, i.e., where r is a linear function. While linear concept erasure methods have certainly found success (Bolukbasi et al., 2016), there is no *a-priori* reason to suspect that neural networks encode concepts in a linear manner. In this work, we take the first step toward the goal of identifying a *non-linear* function $r(\cdot)$ and a corresponding non-linearly encoded concept subspace. We directly build on Ravfogel et al. (2022), who cast linear concept erasure as a minimax game. Under their formulation, the function $r(\cdot)$ learns to remove the concept, while an adversary tries to predict the concept. We extend their work by deriving a class of general minimax games based on kernelization that largely maintains the tractability of the linear approach. Our kernelized method performs concept erasure in a reproducing kernel Hilbert space, which may have a much higher dimensionality (Schölkopf and Smola, 2002) and correspond to a non-linear subspace of the original representation space.

Empirically, we experiment with gender erasure from GloVe and BERT representations. We show that a kernelized adversary can classify the gender of the representations with over 99% accuracy if $r(\cdot)$ is taken to be a linear function. This gives us concrete evidence that gender is indeed encoded non-linearly in the representations. We further find that solving our kernelized minimax game yields an erasure function $r(\cdot)$ that protects against an adversary that shares the same kernel. However, we also find that it is difficult to protect against *all* kernelized adversaries at once: Information removed by one kernel type can be recovered by adversaries using other kernel types. That is, the gender concept is not exclusively encoded in a space that corresponds to any one kernel. This suggests that non-linear concept erasure is very much an open problem.

2 Linear Concept Erasure

We provide an overview of the linear minimax formulation before we introduce its kernelization.

2.1 Notation

Let $\mathcal{D} = \{(y_n, \mathbf{x}_n)\}_{n=1}^{N}$ be a dataset of N concept–representation pairs, where the labels y_n represent the concept to be neutralized. The goal of linear concept erasure is to learn a *linear* erasure function $r(\cdot)$ from $\mathcal{D}$ such that it is impossible to predict y_n from the modified representations $r(\mathbf{x}_n)$. We focus on classification, where the concept labels y_n are derived from a finite set $\{1, \ldots, V\}$ of V discrete values, and the representations $\mathbf{x}_n \in \mathbb{R}^D$ are D-dimensional real column vectors. To predict the concepts labels y_n from the representations $\mathbf{x}_n$, we make use of (and later kernelize) classifiers that are **linear models**, i.e., classifiers of the form $\widehat{y}_n = \boldsymbol{\theta}^\top \mathbf{x}_n$ where $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^D$ is a column vector of parameters that lives in a space Θ. We also consider an arbitrary **loss function** $\ell(\cdot, \cdot) \geq 0$ where $\ell(y_n, \widehat{y}_n)$ tells us how close the prediction $\widehat{y}_n$ is to y_n.

Using this notation, linear concept erasure is realized by identifying a linear concept subspace whose neutralization, achieved through an orthogonal projection matrix, prevents the classifier from predicting the concept. We define $\mathcal{P}_k$ as the set of all $D \times D$ orthogonal projection matrices that neutralize a rank k subspace. More formally, we have that $P \in \mathcal{P}_k \leftrightarrow P = I_D - W^\top W, W \in \mathbb{R}^{k \times D}, WW^\top = I_k$, where I_k denotes the $k \times k$ identity matrix and I_D denotes the $D \times D$ identity matrix. We say that the matrix P neutralizes the k-dimensional rowspace of W.

2.2 A Linear Minimax Game

Following this formalization, it is natural to define a minimax game (Neumann and Morgenstern, 1944) between a projection matrix $P \in \mathcal{P}_k$ that aims to remove the concept subspace, and a linear model parameterized by $\boldsymbol{\theta}$ that aims to recover it:

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{P \in \mathcal{P}_k} \sum_{n=1}^{N} \ell\Big(y_n, \boldsymbol{\theta}^\top P \mathbf{x}_n\Big) \tag{1}$$

This is a special case of the general adversarial framework (Goodfellow et al., 2014). However, in our case, the predictor $\boldsymbol{\theta}$ does *not* interact with the original input $\mathbf{x}_n$. Instead, the classifier attempts to predict the concept label y_n from $P\mathbf{x}_n$. We now give a concrete example of the linear concept: When we have a binary logistic loss ($V = 2$) and a k-dimensional neutralized subspace, the game takes the following form:

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{P \in \mathcal{P}_k} \sum_{n=1}^{N} y_n \log \frac{\exp \boldsymbol{\theta}^\top P \mathbf{x}_n}{1 + \exp \boldsymbol{\theta}^\top P \mathbf{x}_n} \tag{2}$$

In this paper, we focus on $k = 1$, i.e., we aim to identify a 1-dimensional concept subspace.[2] The optimization over $\mathcal{P}_k$ in Eq. (1) renders the game non-convex. To address this issue, Ravfogel et al. (2022) propose a concave relaxation of the objective:

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{P\in\mathcal{F}_k} \sum_{n=1}^{N} \ell\Big(y_n, \boldsymbol{\theta}^\top P \mathbf{x}_n\Big) \tag{3}$$

where the relaxation is shown in gray. In words, instead of optimizing over rank-k projection matrices, a non-convex set, we optimize over its convex hull, the Fantope (Boyd and Vandenberghe, 2014):

$$\mathcal{F}_k = \{A \in \mathcal{S}^D \mid I_D \succcurlyeq A \succcurlyeq 0, \mathrm{tr}(A) = k\} \tag{4}$$

3 Non-linear Concept Erasure

It has often been shown that human-interpretable concepts, and in particular gender, are encoded non-linearly in the representation space (Gonen and Goldberg, 2019; Ravfogel et al., 2020). However, prior work on concept erasure, e.g., the method discussed in §2, assumes that concepts are encoded linearly. Our goal is to extend the game defined in Eq. (3) to be able to neutralize non-linearly encoded concepts while preserving the relative tractability and interpretability of the linear methods. A natural manner through which we can achieve these goals is by kernelization (Shawe-Taylor and Cristianini, 2004; Hofmann et al., 2008).

The underlying assumption motivating kernel methods is that the features needed for the task live in a reproducing kernel Hilbert space (RKHS; Aronszajn, 1950). At an intuitive level, an RKHS allows us to extend some of results from linear algebra to potentially infinite-dimensional vector spaces (Canu and Smola, 2006). The main technical contribution of this paper is the derivation of a kernelized version of the linear adversarial game presented in Eq. (3). We perform this derivation in this section after providing some background on kernel methods. We show that the resulting kernelized game is both non-convex and too computationally heavy to solve directly. Thus, we introduce a Nyström approximation that results in an efficient and light-weight formulation, given in Eq. (10), that is still able to isolate concepts encoded non-linearly.

[2]This construction can be extended to subspaces of dimension $k > 1$ using a version of the Gram–Schmidt process in the RKHS.

3.1 Background: Kernel Methods

Kernel methods are based on reproducing kernel Hilbert spaces (RKHS). Without going into the technical details, an RKHS is a space of "nice" functions equipped with a kernel (Yosida, 2012). A **kernel** $\kappa(\cdot,\cdot) \geq 0$ is a similarity measure that generalizes a positive definite matrix. When we have a kernel over an RKHS, then the kernel corresponds to the dot product of a feature map, i.e., $\kappa(\mathbf{x},\mathbf{y}) = \mathbf{\Phi}(\mathbf{x})^\top\mathbf{\Phi}(\mathbf{y})$. This insight gives us a natural manner to construct kernels. For instance, the linear kernel $\kappa(\mathbf{x},\mathbf{y}) = \mathbf{x}^\top\mathbf{y}$ corresponds to the standard dot product in Euclidean space. The degree-2 polynomial kernel $\kappa(\mathbf{x},\mathbf{y}) = (\gamma\mathbf{x}^\top\mathbf{y} + \alpha)^2$ corresponds to a dot product in a six-dimensional feature space if $\mathbf{x},\mathbf{y} \in \mathbb{R}^2$.

Kernels are more general than positive definite matrices in that they can exist in infinite-dimensional spaces. For example, the Gaussian kernel $\exp\left(-\gamma||\mathbf{x}-\mathbf{y}||_2^2\right)$ is infinite-dimensional.[3] However, for any finite set of points $\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$, we can construct a **Gram matrix** $K \in \mathbb{R}^{N\times N}$ where $K_{nm} = \kappa(\mathbf{x}_n,\mathbf{x}_m)$ encodes the similarity between $\mathbf{x}_n$ and $\mathbf{x}_m$. The matrix K is guaranteed to be positive definite. Kernels are useful because they allow us to implicitly learn functions in a potentially infinite-dimensional RKHS without materializing that space.

3.2 A Kernelized Minimax Game

Inspection of the linear adversarial game in Eq. (1) reveals that both the adversary and the predictor interact with the input only via an inner product. Thus, the game can be kernelized by replacing the inner product with a kernel operation. We establish this kernelization by first proving the following representer theorem-like lemma, which shows that $\boldsymbol{w},\boldsymbol{\theta}$ can be written in terms of spans of the projected training set.

Lemma 1. *(Minimax Game Representer Theorem) Let $\mathcal{H}$ be a reproducing kernel Hilbert space with canonical feature map $\mathbf{\Phi}: \mathbb{R}^D \to \mathcal{H}$, i.e., $\mathbf{\Phi}(\mathbf{x}) = \kappa(\mathbf{x},\cdot)$. Consider the game:*

$$\max_{\boldsymbol{w}\in\mathcal{H}} \min_{\boldsymbol{\theta}\in\mathcal{H}} \sum_{n=1}^{N} \ell\left(y_n, \langle\boldsymbol{\theta}, \mathrm{P}_{\boldsymbol{w}}^{\perp}\mathbf{\Phi}(\mathbf{x}_n)\rangle\right) \tag{5}$$

where $\mathrm{P}_{\boldsymbol{w}}^{\perp}$ is the operator that projects onto the orthogonal complement of $\boldsymbol{w}$. For every attained

[3]One way to see this is to show a Gram matrix of any size N will always have full rank.

local optimum $\boldsymbol{\theta}^*$, $\boldsymbol{w}^*$ of Eq. (5), there is another local optimum $\boldsymbol{\theta}^*_U$, $\boldsymbol{w}^*_U$ with the same value as $\boldsymbol{\theta}^*$, $\boldsymbol{w}^*$ in $U \overset{\text{def}}{=} \mathsf{span}\left\{\boldsymbol{\Phi}(\mathbf{x}_1), \ldots, \boldsymbol{\Phi}(\mathbf{x}_N)\right\}$, the span of the training data.[4]

Proof. See App. A.1 for the proof. ∎

Having expressed $\boldsymbol{w}, \boldsymbol{\theta}$ as a function of the training data, we can proceed to proving the kernelization of the adversarial game.

Lemma 2. *Let $\mathcal{H}$ be a reproducing kernel Hilbert space with canonical feature map $\boldsymbol{\Phi}$, and let $\boldsymbol{\Phi}(\mathbf{z})$ be a point in $\mathcal{H}$. Next, let $\boldsymbol{w} = \sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_n)$ and $\boldsymbol{\theta} = \sum_{n=1}^{N} \beta_n \boldsymbol{\Phi}(\mathbf{x}_n)$ be points in the reproducing kernel Hilbert space. Now, let $\boldsymbol{\Phi}_{proj}(\mathbf{z})$ be the orthogonal projection of $\boldsymbol{\Phi}(\mathbf{z})$ onto the orthogonal complement of the subspace spanned by $\boldsymbol{w}$. Then, we have:*

$$\langle \boldsymbol{\theta}, \boldsymbol{\Phi}_{proj}(\mathbf{z}) \rangle = \sum_{m=1}^{N} \beta_m \left(\kappa(\mathbf{x}_m, \mathbf{z}) - \frac{\boldsymbol{\alpha}^\top K^{(m)}(\mathbf{z}) \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}} \right) \tag{6}$$

where $K^{(m)}_{ij}(\mathbf{z}) \overset{\text{def}}{=} \kappa(\mathbf{x}_i, \mathbf{z})\kappa(\mathbf{x}_m, \mathbf{x}_j)$.

Proof. See App. A.2 for the proof. ∎

Lemma 2 suggests the following form of the kernelized game given in Eq. (5):

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^N} \max_{\boldsymbol{\alpha} \in \mathbb{R}^N} \sum_{n=1}^{N} \ell\left(y_n, \sum_{m=1}^{N} \beta_m \left(\kappa(\mathbf{x}_m, \mathbf{z}_n) - \frac{\boldsymbol{\alpha}^\top K^{(m,n)} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}} \right) \right) \tag{7}$$

where we define $K^{(m,n)}_{ij}(\mathbf{z}) \overset{\text{def}}{=} \kappa(\mathbf{x}_i, \mathbf{z}_n)\kappa(\mathbf{x}_m, \mathbf{x}_j)$. In contrast to Eq. (5), all computations in Eq. (7) are in Euclidean space.

Theorem 1. *The reproducing kernel Hilbert space game Eq. (5) attains the same local optima as Eq. (7).*

Proof. First, plug the result of Lemma 2 into Eq. (5). Optimality follows by Lemma 1, the representer lemma. ∎

Now, we turn to the runtime of the game.

[4] Traditionally, a representer theorem has an extra regularization term on the function estimator that ensures the optimum is attained and that it is global (Kimeldorf and Wahba, 1970). We do not employ such a regularizer and, thus, have a weaker result.

Proposition 1. *The objective in Eq. (7) can be computed in $\mathcal{O}(N^4)$ time.*

Proof. Assuming that $\kappa(\cdot, \cdot)$ may be computed in $\mathcal{O}(1)$, computing $K^{(m,n)}$ takes $\mathcal{O}(N^2)$ time. We have to do $\mathcal{O}(N^2)$ such computations, which results in $\mathcal{O}(N^4)$ time. Note that K may be pre-computed once in $\mathcal{O}(N^2)$ time. Thus, $K^{(m,n)}$ is the bottleneck, so the whole algorithm takes $\mathcal{O}(N^4)$ time. ∎

There are two problems with naïvely using the formulation in Eq. (7). First, and similarly to Eq. (1), the problem is not convex–concave due to the optimization over $\boldsymbol{\alpha}$, which implicitly defines an orthogonal projection matrix of rank 1. Second, the evaluation time of Eq. (7) is $\mathcal{O}\left(N^4\right)$, as argued in Proposition 1, which makes using a training set larger than a few hundred examples infeasible. We solve both of these computational issues with the Nyström approximation.

3.3 The Nyström Approximation

The general idea behind the Nyström method (Nyström, 1930) is to calculate a low-rank approximation of the kernel matrix. It is a commonly used technique for improving the runtime of kernel methods (Williams and Seeger, 2000).

3.3.1 Convexifying the Objective

Consider the Gram matrix $K \in \mathbb{R}^{N \times N}$ where $K_{nm} = \kappa(\mathbf{x}_n, \mathbf{x}_m)$, $\mathbf{x}_n$ is the n^{th} training representation, and $\mathbf{x}_m$ is the m^{th} training representation. We start with the eigendecomposition of $K = U\Sigma U^\top = U\sqrt{\Sigma}\sqrt{\Sigma}U^\top$, which can be computed in $\mathcal{O}\left(N^3\right)$ time (Golub and Van Loan, 2013). We are justified in taking the square root of Σ because K is necessarily positive definite. Now, we define an approximate feature map for observations $\mathbf{x}_n$ in the training data using the eigenvalues and vectors:

$$\widetilde{\boldsymbol{\Phi}}(\mathbf{x}_n) \overset{\text{def}}{=} \left(U\sqrt{\Sigma}\right)_n \tag{8}$$

To compute the features for representations $\mathbf{x}$ *not* in the training data, we use the following:

$$\widetilde{\boldsymbol{\Phi}}(\mathbf{x}) \overset{\text{def}}{=} \sum_{n=1}^{N} \kappa(\mathbf{x}, \mathbf{x}_n)\widetilde{\boldsymbol{\Phi}}(\mathbf{x}_n) \tag{9}$$

which is an average, weighted by the kernel $\kappa(\cdot, \cdot)$, of the features obtained during training. Plugging Eq. (8) into Eq. (3) yields:

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{P \in \mathcal{F}_k} \sum_{n=1}^{N} \ell\left(y_n, \langle \boldsymbol{\theta}, P\widetilde{\boldsymbol{\Phi}}(\mathbf{x}_n) \rangle\right) \tag{10}$$

which is identical to the linear game in Eq. (3), except that it uses the transformed features $\widetilde{\mathbf{\Phi}}(\mathbf{x}_n)$ to approximate the true feature map $\mathbf{\Phi}$.

Importantly, the game in Eq. (3) is convex–concave, so the Nyström approximation allows us to derive a kernelized convex–concave game in Eq. (10). The runtime of a gradient-based optimization procedure for this game over T epochs is now $\mathcal{O}\left(TN^2 + N^3\right)$.[5] While this is an improvement over $\mathcal{O}\left(TN^4\right)$, it is still not fast enough to be of practical use.

3.3.2 Improving the Runtime

The runtime bottleneck of the game in Eq. (10) is the $\mathcal{O}\left(N^3\right)$ time it takes to compute the eigendecomposition of the Gram matrix K. Under the assumption that $\mathrm{rank}\left(K\right) = L$, we can improve this bound to $\mathcal{O}\left(L^3 + L^2 N\right)$ (Drineas and Mahoney, 2005). For the case that $L \ll N$, this is a substantial improvement. Moreover, it implies that the approximation feature map $\widetilde{\mathbf{\Phi}}(\mathbf{x}_n) \in \mathbb{R}^L$. After T steps of optimization, the runtime is now $\mathcal{O}\left(TL^2 + L^3 + L^2 N\right)$, which is fast enough to be useful in practice. A natural question to ask is what happens if we apply the Nyström approximation, thereby assuming $\mathrm{rank}\left(K\right) = L$, but in practice $\mathrm{rank}\left(K\right) > L$? In this case, we are effectively computing a low-rank approximation of the kernel matrix. Several bounds on the accuracy of this approximation have been proven in the literature (Drineas and Mahoney, 2005; Jin et al., 2013; Nemtsov et al., 2016); we refer the reader to these works for more details on the approximation error.

3.4 Pre-image Mapping

After solving the game in Eq. (10), we obtain a projection matrix P that neutralizes the concept *inside* an (approximated) RKHS. In other words, we have a function $r(\cdot)$ that prevents a classifier from predicting a concept label from the representation $\widetilde{\mathbf{\Phi}}(\mathbf{x}_n)$. However, for many applications, we want a version of the input $\mathbf{x}_n$ *in the original space* with the concept neutralized, i.e., we want $r(\mathbf{x}_n)$. Neutralization in the original space requires solving the **pre-image problem** (Mika et al., 1998). In the case of Nyström features, we seek a mapping $P\widetilde{\mathbf{\Phi}}(\mathbf{x}) \mapsto \mathbf{x}$, which is a mapping from $\mathbb{R}^L$ to $\mathbb{R}^D$, that projects the neutralized features back into the input space.

In practice, this task can also be performed via a mapping $\mathbf{x} \mapsto \mathbf{x}$ from $\mathbb{R}^D \mapsto \mathbb{R}^D$, which learns to reproduce in the input space the transformation that P performs in the RKHS. We choose the latter approach, and train a multilayer perceptron (MLP) $f_\lambda(\cdot) : \mathbb{R}^D \to \mathbb{R}^D$.[6] To estimate the parameters λ of the MLP $f_\lambda(\cdot)$, we optimize a two-termed objective over all points $\mathbf{x}_n$:

$$\underset{\lambda \in \Lambda}{\mathrm{argmin}} \left|\left| P\widetilde{\mathbf{\Phi}}(\mathbf{x}_n) - \widetilde{\mathbf{\Phi}}(f_\lambda(\mathbf{x}_n)) \right|\right|_2^2 + \left|\left| (I - P)\widetilde{\mathbf{\Phi}}(f_\lambda(\mathbf{x}_n)) \right|\right|_2^2 \tag{11}$$

where Λ is the parameter space. The first term encourages $f_\lambda(\cdot)$ to perform the same transformation in the input space as P does in the RKHS. The second term ensures that P has no effect on the RKHS features computed on the neutralized $f_\lambda(\mathbf{x}_n)$.

4 Experimental Setup

In §3, we established an algorithm that allows us to attempt kernelized concept erasure of non-linearly encoded concepts. To summarize, this method requires first solving the game in Eq. (10) for a chosen neutralizing kernel, then training a pre-image network according to Eq. (11) to obtain neutralized representations in the input space.

We hypothesize that a non-linearly encoded concept can be exhaustively removed after mapping into the right RKHS. In order for this to hold, the neutralized representations must satisfy two conditions: Adversaries using non-linear classifiers should not be able to predict the erased concept from these representations, and these representations should preserve all other information encoded in them prior to erasure. With binary gender as our non-linearly encoded concept, we conduct several experiments testing both conditions. Before presenting our results, we lay out our experimental setup (see App. A.3 for more details).

Data. We run our main experiments on the identification and erasure of binary gender in static GloVe representations (Pennington et al., 2014). We focus on Ravfogel et al.'s (2020) dataset, where word representations are coupled with binary labels indicating whether they are male-biased or female-biased. As a preprocessing step, we normalize the GloVe representations to have unit norm. For an

[5] Calculating the Nyström features entails computing the eigendecomposition of the Gram matrix, which is $\mathcal{O}\left(N^3\right)$. Then, we calculate the objective over all N examples for T epochs, which is $\mathcal{O}\left(TN^2\right)$.

[6] This mapping may not always be a function because there may be more than one point in the RKHS that corresponds to a point in the original space.

extrinsic evaluation of our method on a main task (profession prediction), we use the Bias-in-Bios dataset of De-Arteaga et al. (2019), which consists of a large set of short biographies annotated for both gender and race. Following Ravfogel et al. (2022), we embed each biography using the [CLS] representation of pre-trained BERT.

Kernels. We consider the following kernels:

- Poly : $\kappa(\mathbf{x}, \mathbf{y}) = (\gamma \mathbf{x}^\top \mathbf{y} + \alpha)^{\mathsf{d}}$
- RBF : $\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma ||\mathbf{x} - \mathbf{y}||_2^2\right)$
- Laplace : $\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma ||\mathbf{x} - \mathbf{y}||_1\right)$
- Linear : $\kappa(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$
- Sigmoid : $\kappa(\mathbf{x}, \mathbf{y}) = \tanh(\gamma \mathbf{x}^\top \mathbf{y} + \alpha)$
- Multiple : a convex combination of the above kernels. We consider the following two methods for combining kernels:
  - EasyMKL : a convex combination learned with the EasyMKL algorithm (Aiolli and Donini, 2015) targeted for gender prediction.[7]
  - UniformMK : a uniform combination of all kernels.

We experiment with different values for the hyperparameters $\gamma > 0$, $\alpha > 0$ and $\mathsf{d} > 0$ (see App. A.3 for details). We use $L = 1024$-dimensional vectors for the Nyström approximation.

Reported metrics. Each result is reported as the mean $\pm$ standard deviation, computed across four runs of the experiment with random restarts.

Solving the minimax game. We solve the relaxed adversarial game given in Eq. (10) by alternate gradient-based optimization (Goodfellow et al., 2016). Concretely, we alternate between updating the predictor's parameters $\boldsymbol{\theta}$ and the projection matrix P. Updates to $\boldsymbol{\theta}$ are performed with gradient descent, and updates to P are performed with gradient *ascent*, including a projection onto the Fantope to ensure that the constraint is met. For the Fantope projection step, we use Vu et al.'s (2013) algorithm, the details of which are restated in Ravfogel et al. (2022).

[7]We use the implementation of MKLpy with the default parameters.

Pre-image calculation. As our pre-image network $f_\lambda(\cdot)$, we use an MLP with two hidden layers of sizes 512 and 300, respectively. We use layer normalization after each hidden layer and ReLU activations. See App. A.3.1 for basic empirical validation of the pre-image calculation procedure.

5 Effect on Concept Encoding

In this section, we pose the **exhaustive RKHS hypothesis**: The hypothesis that binary gender can be *exhaustively* removed when the representations are mapped into the right RKHS. That is, there exists a unique kernel such that, for any choice of non-linear predictor, the adversary cannot recover gender information from the pre-image representations obtained via our special kernel. As a baseline, we note that gender prediction accuracy on the original representations, prior to any intervention, is above 99% with *every* kernel, including the linear kernel. This means that the gender concept is linearly separable in the original input space. In this context, we conduct the following experiments on gender neutralization.

Same adversary. We start by calculating the neutralized pre-images for each kernel type, and then apply the same kernel adversary to recover gender information. This experiment tests whether we can protect against the same kernel adversary.

Transfer between kernels. To directly test the exhaustive RKHS hypothesis, we calculate the neutralized pre-image representations with respect to a neutralizing kernel, and then use a different adversarial kernel to recover gender. For instance, we calculate neutralized pre-image representations with respect to a polynomial kernel, and then predict gender using a Laplace kernel, or a polynomial kernel with different hyperparameters.

Biased associations. Gender can manifest itself in the pre-image representations via biased associations, even when gender is neutralized according to our adversarial test. To assess the impact of our intervention on this notion of gender encoding, we run the WEAT test (Islam et al., 2017) on the neutralized pre-image representations.

5.1 Pre-image Gender Recovery: Same Adversary

In Table 1, we report average gender prediction accuracy on the neutralized pre-images for the case where the neutralizing and adversarial

Type	Accuracy
Poly	0.59 ± 0.15
RBF	0.69 ± 0.16
Laplace	0.75 ± 0.11
Linear	0.54 ± 0.02
Sigmoid	0.49 ± 0.00
EasyMKL	0.69 ± 0.01
UniformMK	0.49 ± 0.00

Table 1: Gender prediction accuracy from the neutralized pre-image representations when using the same kernel for neutralization and recovery. Numbers are averages over hyperparameters of each kernel and over four randomized runs of the experiment.

kernels are of the same type and share the same hyperparameters.[8] Numbers are averages over the results of all hyperparameter values of each kernel. See App. A.4 for the full results. As can be seen, for most—but not all—kernels, we effectively hinder the ability of the non-linear classifier to predict gender.

5.2 Pre-image Gender Recovery: Transfer Between Kernels

In Table 2, we report average gender prediction accuracy on the neutralized pre-images for the case where the neutralizing and adversarial kernels are different. Numbers are averages over several different hyperparameter settings for the neutralizing kernel, while the adversarial kernel hyperparameters are fixed as detailed in App. A.7. Also, see the appendix for a full breakdown by neutralizing kernel hyperparameters. Table 2 allows us to test the exhaustive RKHS hypothesis. Under this hypothesis, we would expect to see that for at least one neutralizing kernel, no non-linear classifers are able to accurately recover gender. For a thorough test of the hypothesis, we introduce an MLP with a single hidden layer as an additional adversary.

Remarkably, we observe a complete lack of generalization of our concept erasure intervention to other types of non-linear predictors. In particular, no neutralizing kernel significantly hinders the ability of an MLP with a single hidden layer to predict gender: the MLP always recovers the gender labels with an accuracy of 97%. Furthermore, concept erasure does not transfer between different kernel types, and even between kernels of the same family with different hyperparameter settings. For instance, when using a polynomial neutralizing kernel, we protect against a polynomial adversarial kernel with the same parameters (mean accuracy of 59% in Table 1), but not against a polynomial adversarial kernel with different hyperparameters (98% mean accuracy for Poly in Table 2).

We do see transfer to sigmoid, and—to a lesser degree—linear kernel adversaries, with a classification accuracy of 54–65% for the linear adversary, and 49% for the sigmoid adversary. Surprisingly, the sigmoid kernel seems weaker than the linear kernel, achieving a near-random accuracy of 49%. The results do not show evidence of a proper hierarchy in the expressiveness of the different kernels, and convex combinations of the different kernels—either learned (EasyMKL) or uniform (UniformMK)—do not provide better protection than individual kernels.

In short, while we are able to effectively protect against the same kernel, transfer to different kernels is non-existent. This result does not support the exhaustive RKHS hypothesis: We do not find a single RKHS that exhaustively encodes the binary gender concept.

5.3 Effect on Gendered Word Associations

In the case where the concept of gender cannot be recovered by an adversary, binary gender could still manifest itself in more subtle ways. For instance, it may be the case that the names of gender-biased professions, such as STEM fields, are closer in representation space to male-associated words than to female-associated words. We aim to measure the extent to which our neutralized pre-image representations exhibit this measure of gender bias.

Evaluation. To quantify bias associations, Islam et al. (2017) propose the WEAT word association test. This test measures WEAT's d, a statistic that quantifies the difference in similarity between two sets of gendered words (e.g., male first names and female first names) and two sets of potentially biased words (e.g., stereotypically male and stereotypically female professions).[9] We repeat the experiments of Gonen and Goldberg (2019) and Ravfogel et al. (2020). Following Gonen and

[8] We use the Nyström approximation for neutralization, but predict gender from the pre-image representations using SVM classifiers that use the true kernel.

[9] It asks, for example, whether the degree to which male first names are closer to scientific terms versus artistic terms is significantly different from the same quantity calculated for female first names.

	Poly	RBF	Laplace	Linear	Sigmoid	EasyMKL	UniformMK	MLP
Poly	0.98 ± 0.00	0.96 ± 0.00	0.93 ± 0.00	0.55 ± 0.01	0.49 ± 0.00	0.98 ± 0.00	0.49 ± 0.00	0.97 ± 0.00
RBF	0.98 ± 0.00	0.96 ± 0.00	0.93 ± 0.00	0.59 ± 0.02	0.49 ± 0.00	0.98 ± 0.00	0.49 ± 0.00	0.97 ± 0.00
Laplace	0.98 ± 0.00	0.96 ± 0.00	0.94 ± 0.00	0.61 ± 0.01	0.49 ± 0.00	0.98 ± 0.00	0.59 ± 0.02	0.97 ± 0.00
Linear	0.98 ± 0.00	0.96 ± 0.00	0.93 ± 0.00	0.54 ± 0.02	0.49 ± 0.00	0.98 ± 0.00	0.49 ± 0.00	0.97 ± 0.00
Sigmoid	0.98 ± 0.00	0.93 ± 0.01	0.89 ± 0.01	0.65 ± 0.03	0.49 ± 0.00	0.97 ± 0.00	0.64 ± 0.03	0.97 ± 0.00
EasyMKL	0.98 ± 0.00	0.96 ± 0.00	0.94 ± 0.00	0.57 ± 0.03	0.49 ± 0.00	0.98 ± 0.00	0.49 ± 0.00	0.97 ± 0.00
UniformMK	0.98 ± 0.00	0.96 ± 0.00	0.94 ± 0.01	0.58 ± 0.08	0.49 ± 0.00	0.98 ± 0.00	0.49 ± 0.00	0.97 ± 0.00

Table 2: Evaluation of the neutralized pre-image representations when using a different kernel for neutralization and recovery. The neutralizing kernel is presented in the rows, and the kernel adversary used for recovery in the columns. Note that on the diagonal, the *hyperparameters* of the same kernel family differ between the neutralizing kernel and the adversary that recovers gender form the pre-image representations. See App. A.7 for a breakdown by neutralizing kernel hyperparameters and for details on the hyperparameters of the kernel adversaries.

Kernel	WEAT's d	WEAT's p-value
Poly	0.74 ± 0.01	0.08 ± 0.00
RBF	0.74 ± 0.00	0.08 ± 0.00
Laplace	0.71 ± 0.03	0.09 ± 0.01
Linear	0.74 ± 0.00	0.08 ± 0.00
Sigmoid	0.75 ± 0.02	0.08 ± 0.01
EasyMKL	0.73 ± 0.00	0.08 ± 0.00
UniformMK	0.73 ± 0.00	0.08 ± 0.00
Original	1.56	0.000

Table 3: WEAT results on pre-image representations. Numbers are averages over hyperparameters of each kernel.

Goldberg (2019), we represent the male and female groups with names commonly associated with males and females, rather than with explicitly gendered words (e.g., pronouns). Three tests evaluate the association between name groups and i) career and family-related words; ii) art and mathematics-related words; and iii) names of artistic and scientific fields. Successful neutralization of gender would imply that these word groups are less closely associated in the pre-image representations.

Results. In Table 3, we report the test statistic and the p value for the third test using the names of scientific and artistic fields to represent gender-biased words.[10] For all kernels, we observe a significant drop in the test statistic, from the original value of 1.56 to around 0.74. This suggests that the intervention significantly decreases the association between female and male names and stereotypically biased words. Notably, the reduction is similar for *all* kernels, including the linear one. While non-linear erasure is more effective in neutralizing gender against adversarial recovery, linear and non-linear methods perform equally well according to this bias association test. This finding highlights the importance of measuring different manifestations of a concept when using concept neutralization as a bias mitigation method.

6 Negative Impact on the Representations

Our method has shown a satisfactory ability to prevent the same kernel from recovering the concept. However, does erasure remove *too much* information?[11] As previously stated, our intervention should erase a concept without altering any of the other information encoded in the original representations. In this section, we evaluate whether the non-gender related semantic content of the original representations is preserved in our neutralized pre-images. We do so via the following tests: i) an **intrinsic evaluation** of the semantic content of the neutralized pre-image word representation space, and ii) an **extrinsic evaluation** of our method when applied to contextualized word representations for a profession prediction task, measuring the extent to which we hinder a model's ability to perform the main task.

6.1 Intrinsic Evaluation of Damage to Semantic Content

To measure the influence of our method on the semantics encoded in the representation space, we use SimLex-999 (Hill et al., 2015), an annotated dataset of word pairs with human similarity scores for each pair. First, we calculate the cosine similarity between the representations of each pair of words using the original representations. Then, we repeat this calculation for each type

[10]See App. A.6 for results of the other two biased word association tests.

[11]It is trivial to neutralize a concept by completely zeroing out the representations.

of kernel using the pre-image representations. Finally, we measure the correlation between the similarity of words in representation space and human similarity scores, before and after intervention. The original correlation is 0.400, and it is left nearly unchanged by *any* of the kernel interventions, yielding values between 0.387 and 0.396. To qualitatively demonstrate the absence of negative impact, we show in App. A.5 that the nearest neighbors of randomly sampled words do not change significantly after gender erasure.

6.2 Extrinsic Evaluation on Contextualized Representations

The previous experiments focused on the influence of concept neutralization on uncontextualized representations. Here, we apply our concept neutralization method on contextualized BERT representations and assess its effect on profession prediction. We embed each biography in the dataset of De-Arteaga et al. (2019) using the [CLS] representation of pre-trained BERT, and apply our method using only the RBF kernel. After collecting the pre-image representations, we train a linear classifier on the main task of profession prediction.

Results. Averaged over different hyperparameter settings of the RBF kernel, we achieve a profession prediction accuracy after neutralization of $74.19 \pm 0.056\%$. For reference, prediction accuracy using the original BERT representations is 76.93%. This suggests that the pre-images still encode most of the profession information, which is largely orthogonal to the neutralized gender information.

7 Discussion

We have demonstrated that in the case where the neutralizing kernel and the adversarial kernel are the same, we are able to neutralize a non-linearly encoded concept reasonably well. We have also shown that our method neutralizes gender in a comprehensive manner, without damaging the representation. However, this neutralization does not transfer to different non-linear adversaries, which are still able to recover gender.

While the lack of transfer to other non-linear predictors may seem surprising, one should keep in mind that changing the kernel type, or changing kernel hyperparameters, results in a different implicit feature mapping. Even the features defined by a linear kernel are not a proper subset of the features defined by a polynomial kernel of degree 2.[12] As such, removing the features which make the concept of interest linearly separable in one RKHS does not necessarily prevent a classifier parameterized by another kernel or an MLP from predicting the concept. In the context of gender erasure, these results suggest that protection against a diverse set of non-linear adversaries remains an open problem.

8 Conclusion

We propose a novel method for the identification and erasure of non-linearly encoded concepts in neural representations. We first map the representations to an RKHS, before identifying and neutralizing the concept in that space. We use our method to empirically assess the exhaustive RKHS hypothesis: We hypothesize that there exists a unique kernel that exhaustively identifies the concept of interest. We find that while we are able to protect against a kernel adversary of the same type, this protection does not transfer to different nonlinear classifiers, thereby contradicting to the RKHS hypothesis. Exhaustive concept erasure and protection against a diverse set of non-linear adversaries remains an open problem.

Limitations

The empirical experiments in this work involve the removal of binary gender information from pre-trained representations. We note the fact that gender is a non-binary concept as a major limitation of our work. This task may have real-world applications, in particular relating to fairness. We would encourage readers to be careful when attempting to deploy methods such as the one discussed in this paper. Regardless of any proofs, one should carefully measure the effectiveness of the approach in the context in which it is to be deployed. Please consider, among other things, the exact data to be used, the fairness metrics under consideration, and the overall application.

We urge practitioners not to regard this method as a solution to the problem of bias in neural models, but rather as a preliminary research effort toward mitigating certain aspects of the problem. Unavoidably, the datasets we use do not reflect all the subtle and implicit ways in which gender bias is manifested. As such, it is likely that different forms

[12] Features defined by a linear kernel can be a proper subset of the features defined by a polynomial kernel if there is a fixed coordinate in all examples prior to the kernel mapping.

of bias still exist in the representations following the application of our method.

Ethical Concerns

We do not foresee any ethical concerns with this work.

Acknowledgements

The authors sincerely thank Clément Guerner for his thoughtful and comprehensive comments and revisions to the final version of this work. This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program, grant agreement No. 802774 (iEXTRACT). Ryan Cotterell acknowledges Google for support from the Research Scholar Program.

References

Fabio Aiolli and Michele Donini. 2015. EasyMKL: a scalable multiple kernel learning algorithm. *Neurocomputing*, 169:215–224.

M. A. Aizerman, E. A. Braverman, and L. Rozonoer. 1964. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25(6):917–936.

Nachman Aronszajn. 1950. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29.

Stephen P. Boyd and Lieven Vandenberghe. 2014. *Convex Optimization*. Cambridge University Press.

Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. 2021. Multimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language BERTs. *Transactions of the Association for Computational Linguistics*, 9:978–994.

Stéphane Canu and Alex Smola. 2006. Kernel methods and the exponential family. *Neurocomputing*, 69(7-9):714–720.

Hande Celikkanat, Sami Virpioja, Jörg Tiedemann, and Marianna Apidianaki. 2020. Controlling the imprint of passivization and negation in contextualized representations. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 136–148, Online. Association for Computational Linguistics.

Maria De-Arteaga, Alexey Romanov, Hanna M. Wallach, Jennifer T. Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Cem Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. *CoRR*, abs/1901.09451.

Sunipa Dev and Jeff M Phillips. 2019. Attenuating bias in word vectors. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 879–887. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Petros Drineas and Michael W. Mahoney. 2005. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Ky Fan. 1953. Minimax theorems. *Proceedings of the National Academy of Sciences of the United States of America*, 39(1):42.

Peter Gärdenfors. 2000. *Conceptual Spaces: The Geometry of Thought*. The MIT Press.

Gene H. Golub and Charles F. Van Loan. 2013. *Matrix Computations*. Johns Hopkins Press.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614, Minneapolis, Minnesota. Association for Computational Linguistics.

Hila Gonen, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. 2020. It's not Greek to mBERT: Inducing word-level translations from multilingual BERT. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 45–56, Online. Association for Computational Linguistics.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27.

Evan Hernandez and Jacob Andreas. 2021. The low-dimensional linear geometry of contextualized word representations. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 82–93, Online. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. 2008. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220.

Aylin Caliskan Islam, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora necessarily contain human biases. *Science*, 356(6334):183–186.

Rong Jin, Tianbao Yang, Mehrdad Mahdavi, Yu-Feng Li, and Zhi-Hua Zhou. 2013. Improved bounds for the Nyström method with application to kernel classification. *IEEE Transactions on Information Theory*, 59(10):6939–6949.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2668–2677. PMLR.

George S. Kimeldorf and Grace Wahba. 1970. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. 2019. It's all in the name: Mitigating gender bias with name-based counterfactual data substitution. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5267–5275, Hong Kong, China. Association for Computational Linguistics.

Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. 1998. Kernel PCA and de-noising in feature spaces. In *Advances in Neural Information Processing Systems*, volume 11, pages 536–542.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations*.

Arik Nemtsov, Amir Averbuch, and Alon Schclar. 2016. Matrix compression using the Nyström method. *Intelligent Data Analysis*, 20(5):997–1019.

John von Neumann and Oskar Morgenstern. 1944. *Theory of Games and Economic Behavior*. Princeton University Press.

Evert J. Nyström. 1930. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*, 54:185–204.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.

Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. 2021. Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 194–209, Online. Association for Computational Linguistics.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan Cotterell. 2022. Linear adversarial concept erasure. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18400–18421. PMLR.

Bashir Sadeghi, Runyi Yu, and Vishnu Boddeti. 2019. On the global optima of kernelized adversarial representation learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7970–7978. IEEE.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. 1997. Kernel principal component analysis. In *Artificial Neural Networks — ICANN'97*, pages 583–588, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bernhard Schölkopf and Alexander J. Smola. 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Francisco Vargas and Ryan Cotterell. 2020. Exploring the linear subspace hypothesis in gender bias mitigation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2902–2913, Online. Association for Computational Linguistics.

Vincent Q. Vu, Juhee Cho, Jing Lei, and Karl Rohe. 2013. Fantope projection and selection: A near-optimal convex relaxation of sparse PCA. In *Advances in Neural Information Processing Systems*, volume 26.

Jennifer C. White, Tiago Pimentel, Naomi Saphra, and Ryan Cotterell. 2021. A non-linear structural probe. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 132–138, Online. Association for Computational Linguistics.

Christopher Williams and Matthias Seeger. 2000. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, volume 13.

Kosaku Yosida. 2012. *Functional Analysis*. Springer.

A Related Work

Mitigation of Gender Bias. The identification of linear subspaces that encode binary gender has attracted considerable research interest (Bolukbasi et al., 2016; Gonen and Goldberg, 2019; Dev and Phillips, 2019; Ravfogel et al., 2020). While bias mitigation is a central use case of concept erasure, concept subspaces have been applied to a number of tasks. Concept subspaces have been used to analyze the content of neural representations, e.g., for causal analysis (Elazar et al., 2021; Ravfogel et al., 2021), for analyzing the geometry of the representation space (Celikkanat et al., 2020; Gonen et al., 2020; Hernandez and Andreas, 2021), and for concept-based interpretability (Kim et al., 2018).

Kernelization of Linear Methods. The kernelization of linear machine learning algorithms is a common practice, and has many use cases, such as the kernelized perceptron (Aizerman et al., 1964) and kernel PCA (Schölkopf et al., 1997). White et al. (2021) proposed a kernelization of a structural probe that extracts syntactic structure from neural representations. Vargas and Cotterell (2020) proposed a kernelization of the PCA-based bias mitigation method of Bolukbasi et al. (2016), and found that it does not improve on the linear mitigation procedure. Since the effectiveness of this method has been questioned (Gonen and Goldberg, 2019), we consider a more principled and well-motivated approach for the identification and neutralization of the concept subspace. Sadeghi et al. (2019) proposed a kernelization of an alternative, regression-based linear adversarial objective, which is not limited to *orthogonal projections*. Our formulation is different in that it considers any linear model, and is restricted to the neutralization of linear subspaces via projection. This makes our method potentially less expressive, but more interpretable.

A.1 A Representer Lemma for Kernelized Minimax Games

Lemma 1. *(Minimax Game Representer Theorem) Let $\mathcal{H}$ be a reproducing kernel Hilbert space with canonical feature map $\boldsymbol{\Phi} : \mathbb{R}^D \to \mathcal{H}$, i.e., $\boldsymbol{\Phi}(\mathbf{x}) = \kappa(\mathbf{x}, \cdot)$. Consider the game:*

$$\max_{\boldsymbol{w}\in\mathcal{H}} \min_{\boldsymbol{\theta}\in\mathcal{H}} \sum_{n=1}^{N} \ell\left(y_n, \langle \boldsymbol{\theta}, \mathrm{P}_{\boldsymbol{w}}^{\perp}\, \boldsymbol{\Phi}(\mathbf{x}_n)\rangle\right) \tag{5}$$

where $\mathrm{P}_{\boldsymbol{w}}^{\perp}$ is the operator that projects onto the orthogonal complement of $\boldsymbol{w}$. For every attained local optimum $\boldsymbol{\theta}^*$, $\boldsymbol{w}^*$ of Eq. (5), there is another local optimum $\boldsymbol{\theta}_U^*$, $\boldsymbol{w}_U^*$ with the same value as $\boldsymbol{\theta}^*$, $\boldsymbol{w}^*$ in $U \stackrel{\text{def}}{=} \mathsf{span}\Big\{\boldsymbol{\Phi}(\mathbf{x}_1), \ldots, \boldsymbol{\Phi}(\mathbf{x}_N)\Big\}$, the span of the training data.[13]

Proof. For brevity, first notice we can re-express the objective as

$$\max_{\boldsymbol{w},\in\mathcal{H}} \min_{\boldsymbol{\theta}\in\mathcal{H}} \sum_{n=1}^{N} \ell\left(y_n, \langle \boldsymbol{\theta}, \left(\mathrm{I} - \frac{\boldsymbol{w}\boldsymbol{w}^\top}{\boldsymbol{w}^\top\boldsymbol{w}}\right) \boldsymbol{\Phi}(\mathbf{x}_n)\rangle\right) \tag{12}$$

We will show that both $\boldsymbol{w}$ and $\boldsymbol{\theta}$ can be expressed as a linear combination of terms from the training data without losing expressive power Now, decompose $\boldsymbol{w}$ as follows: $\boldsymbol{w} = \boldsymbol{w}_U + \boldsymbol{w}_{\perp U}$, where we represent $\boldsymbol{w}$ as the sum of $\boldsymbol{w}$ projected onto U and onto its orthogonal complement $U_\perp$. Now, note that for any element of the training data $\boldsymbol{\Phi}(\mathbf{x}_n)$, we have

[13]Traditionally, a representer theorem has an extra regularization term on the function estimator that ensures the optimum is attained and that it is global (Kimeldorf and Wahba, 1970). We do not employ such a regularizer and, thus, have a weaker result.

$$\left(\mathrm{I} - \frac{\boldsymbol{w}\boldsymbol{w}^\top}{\boldsymbol{w}^\top\boldsymbol{w}}\right)\boldsymbol{\Phi}(\mathbf{x}_n) = \left(\mathrm{I} - \frac{(\boldsymbol{w}_U + \boldsymbol{w}_{\perp U})(\boldsymbol{w}_U + \boldsymbol{w}_{\perp U})^\top}{\boldsymbol{w}^\top\boldsymbol{w}}\right)\boldsymbol{\Phi}(\mathbf{x}_n) \tag{13a}$$

$$= \left(\mathrm{I} - \frac{\boldsymbol{w}_U\boldsymbol{w}_U^\top}{\boldsymbol{w}^\top\boldsymbol{w}} - \underbrace{\frac{\boldsymbol{w}_U\boldsymbol{w}_{\perp U}^\top}{\boldsymbol{w}^\top\boldsymbol{w}}}_{=0} - \underbrace{\frac{\boldsymbol{w}_{\perp U}\boldsymbol{w}_U^\top}{\boldsymbol{w}^\top\boldsymbol{w}}}_{=0} - \frac{\boldsymbol{w}_{\perp U}\boldsymbol{w}_{\perp U}^\top}{\boldsymbol{w}^\top\boldsymbol{w}}\right)\boldsymbol{\Phi}(\mathbf{x}_n) \tag{13b}$$

$$= \left(\mathrm{I} - \frac{\boldsymbol{w}_U\boldsymbol{w}_U^\top}{\boldsymbol{w}^\top\boldsymbol{w}} - \frac{\boldsymbol{w}_{\perp U}\boldsymbol{w}_{\perp U}^\top}{\boldsymbol{w}^\top\boldsymbol{w}}\right)\boldsymbol{\Phi}(\mathbf{x}_n) \tag{13c}$$

$$= \boldsymbol{\Phi}(\mathbf{x}_n) - \frac{\boldsymbol{w}_U\boldsymbol{w}_U^\top}{\boldsymbol{w}^\top\boldsymbol{w}}\boldsymbol{\Phi}(\mathbf{x}_n) - \frac{\boldsymbol{w}_{\perp U}\boldsymbol{w}_{\perp U}^\top}{\boldsymbol{w}^\top\boldsymbol{w}}\boldsymbol{\Phi}(\mathbf{x}_n) \tag{13d}$$

$$= \boldsymbol{\Phi}(\mathbf{x}_n) - \boldsymbol{w}_U\frac{\boldsymbol{w}_U^\top\boldsymbol{\Phi}(\mathbf{x}_n)}{\boldsymbol{w}^\top\boldsymbol{w}} \tag{13e}$$

However, we have that $\boldsymbol{\Phi}(\mathbf{x}_n) - \boldsymbol{w}_U\frac{\boldsymbol{w}_U^\top\boldsymbol{\Phi}(\mathbf{x}_n)}{\boldsymbol{w}^\top\boldsymbol{w}}$ is in U. Likewise, we can decompose $\boldsymbol{\theta} = \boldsymbol{\theta}_U + \boldsymbol{\theta}_{\perp U}$. Further manipulation reveals

$$\left\langle\boldsymbol{\theta}, \left(\mathrm{I} - \boldsymbol{w}\boldsymbol{w}^\top\right)\boldsymbol{\Phi}(\mathbf{x}_n)\right\rangle = \boldsymbol{\theta}_U^\top\boldsymbol{\Phi}(\mathbf{x}_n) - \boldsymbol{\theta}_U^\top\boldsymbol{w}_U\frac{\boldsymbol{w}_U^\top\boldsymbol{\Phi}(\mathbf{x}_n)}{\boldsymbol{w}^\top\boldsymbol{w}} \tag{14}$$

Thus, for any $\boldsymbol{\theta}, \boldsymbol{w}, \in \mathcal{H}$ there exists a $\boldsymbol{\theta}_U, \boldsymbol{w}_U \in U$ that yields the same value of the objective as $\boldsymbol{\theta}, \boldsymbol{w}$. Now, we can parameterize $\boldsymbol{\theta}_U$ and $\boldsymbol{w}_U$ as

$$\boldsymbol{w}_U = \sum_{n=1}^{N}\alpha_n\boldsymbol{\Phi}(\mathbf{x}_n) \tag{15}$$

$$\boldsymbol{\theta}_U = \sum_{n=1}^{N}\beta_n\boldsymbol{\Phi}(\mathbf{x}_n) \tag{16}$$

for real coefficients $\boldsymbol{\alpha} \in \mathbb{R}^N$ and $\boldsymbol{\beta} \in \mathbb{R}^N$. We conclude that for any local optimum $\boldsymbol{\theta}^*, \boldsymbol{w}^* \in \mathcal{H}$, the projection of $\boldsymbol{\theta}^*$ and $\boldsymbol{w}^*$ onto U yields a local optimum with the same value. ■

We note that under regularity conditions, i.e., certain compactness and convexity restrictions over the feasible sets and the loss function, the min and the max can be swapped as per the celebrated Von Neumann–Fan minimax theorem (Fan, 1953). For the aforementioned reasons, we believe Lemma 1 justifies the parameterizations used for $\boldsymbol{\theta}, \boldsymbol{w}$, e.g., Eq. (19).

A.2 Kernelization of the Minimax Game

We show that the game Eq. (1) can be kernelized for the case $k = 1$, i.e., a setting where the matrix $\mathcal{P}_k$ removes a one-dimensional subspace. Specifically, we will show that the product $\langle\boldsymbol{\theta}, \mathrm{P}\boldsymbol{\Phi}(\mathbf{x}_n)\rangle$ in Eq. (1) can be expressed as a function of the kernel $\kappa(\cdot,\cdot)$.

Lemma 2. *Let $\mathcal{H}$ be a reproducing kernel Hilbert space with canonical feature map $\boldsymbol{\Phi}$, and let $\boldsymbol{\Phi}(\mathbf{z})$ be a point in $\mathcal{H}$. Next, let $\boldsymbol{w} = \sum_{n=1}^N \alpha_n\boldsymbol{\Phi}(\mathbf{x}_n)$ and $\boldsymbol{\theta} = \sum_{n=1}^N \beta_n\boldsymbol{\Phi}(\mathbf{x}_n)$ be points in the reproducing kernel Hilbert space. Now, let $\boldsymbol{\Phi}_{proj}(\mathbf{z})$ be the orthogonal projection of $\boldsymbol{\Phi}(\mathbf{z})$ onto the orthogonal complement of the subspace spanned by $\boldsymbol{w}$. Then, we have:*

$$\langle\boldsymbol{\theta}, \boldsymbol{\Phi}_{proj}(\mathbf{z})\rangle = \sum_{m=1}^{N}\beta_m\Bigg(\kappa(\mathbf{x}_m, \mathbf{z}) - \frac{\boldsymbol{\alpha}^\top K^{(m)}(\mathbf{z})\boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top K\boldsymbol{\alpha}}\Bigg) \tag{6}$$

where $K_{ij}^{(m)}(\mathbf{z}) \stackrel{\text{def}}{=} \kappa(\mathbf{x}_i, \mathbf{z})\kappa(\mathbf{x}_m, \mathbf{x}_j)$.

Proof. The projection onto the orthogonal complement of $\boldsymbol{w} = \sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_n)$ is defined as the following

$$\mathrm{P}_{\boldsymbol{w}}^{\perp} \stackrel{\text{def}}{=} \mathrm{I} - \frac{\left(\sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_i)\right)\left(\sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_n)^{\top}\right)}{\left(\sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_n)^{\top}\right)\left(\sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_n)\right)} \tag{17}$$

where I is the identity operator. Algebraic manipulation reveals

$$\begin{aligned}
\mathrm{P}_{\boldsymbol{w}}^{\perp}\boldsymbol{\Phi}(\mathbf{z}) &\stackrel{\text{def}}{=} \mathrm{I}\boldsymbol{\Phi}(\mathbf{z}) - \frac{\left(\sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_n)\right)\left(\sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_{n'})^{\top}\right)}{\left(\sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_n)^{\top}\right)\left(\sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_n)\right)} \boldsymbol{\Phi}(\mathbf{z}) && \text{(18a)}\\
&= \boldsymbol{\Phi}(\mathbf{z}) - \frac{\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m \boldsymbol{\Phi}(\mathbf{x}_n)\boldsymbol{\Phi}(\mathbf{x}_m)^{\top}\boldsymbol{\Phi}(\mathbf{z})}{\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m \boldsymbol{\Phi}(\mathbf{x}_n)^{\top}\boldsymbol{\Phi}(\mathbf{x}_m)} && \text{(18b)}\\
&= \boldsymbol{\Phi}(\mathbf{z}) - \frac{\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m \boldsymbol{\Phi}(\mathbf{x}_n)\kappa(\mathbf{x}_m, \mathbf{z})}{\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m \kappa(\mathbf{x}_n, \mathbf{x}_m)} && \text{(18c)}\\
&= \boldsymbol{\Phi}(\mathbf{z}) - \underbrace{\left(\frac{\sum_{m=1}^{N} \alpha_m \kappa(\mathbf{x}_m, \mathbf{z})}{\boldsymbol{\alpha}^{\top}K\boldsymbol{\alpha}}\right)}_{\in\mathbb{R}} \sum_{n=1}^{N} \alpha_n \boldsymbol{\Phi}(\mathbf{x}_n) && \text{(18d)}\\
&= \boldsymbol{\Phi}(\mathbf{z}) - \left(\frac{\sum_{m=1}^{N} \alpha_m \kappa(\mathbf{x}_n, \mathbf{z})}{\boldsymbol{\alpha}^{\top}K\boldsymbol{\alpha}}\right) \boldsymbol{w} && \text{(18e)}\\
&\stackrel{\text{def}}{=} \boldsymbol{\Phi}_{\text{proj}}(\mathbf{z}) && \text{(18f)}
\end{aligned}$$

Now, consider an element of the reproducing kernel Hilbert space

$$\boldsymbol{\theta} = \sum_{n=1}^{N} \beta_n \boldsymbol{\Phi}(\mathbf{x}_n) \tag{19}$$

Further algebraic manipulation reveals

$$\begin{aligned}
\left\langle \boldsymbol{\theta}, \boldsymbol{\Phi}_{\text{proj}}(\mathbf{z}) \right\rangle &= \left\langle \boldsymbol{\theta}, \boldsymbol{\Phi}(\mathbf{z}) - \left(\frac{\sum_{n=1}^{N} \alpha_n \kappa(\mathbf{x}_n, \mathbf{z})}{\boldsymbol{\alpha}^{\top}K\boldsymbol{\alpha}}\right) \boldsymbol{w} \right\rangle && \text{(20a)}\\
&= \left\langle \sum_{m=1}^{N} \beta_m \boldsymbol{\Phi}(\mathbf{x}_m), \boldsymbol{\Phi}(\mathbf{z}) - \left(\frac{\sum_{n=1}^{N} \alpha_n \kappa(\mathbf{x}_n, \mathbf{z})}{\boldsymbol{\alpha}^{\top}K\boldsymbol{\alpha}}\right) \boldsymbol{w} \right\rangle && \text{(20b)}\\
&= \sum_{m=1}^{N} \beta_m \kappa(\mathbf{x}_m, \mathbf{z}) - \sum_{m=1}^{N} \beta_m \left(\frac{\sum_{n=1}^{N} \alpha_n \kappa(\mathbf{x}_n, \mathbf{z})}{\boldsymbol{\alpha}^{\top}K\boldsymbol{\alpha}}\right) \kappa(\mathbf{x}_m, \boldsymbol{w}) && \text{(20c)}\\
&= \sum_{m=1}^{N} \beta_m \kappa(\mathbf{x}_m, \mathbf{z}) - \frac{1}{\boldsymbol{\alpha}^{\top}K\boldsymbol{\alpha}} \sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\beta_m \kappa(\mathbf{x}_n, \mathbf{z})\kappa(\mathbf{x}_m, \boldsymbol{w}) && \text{(20d)}\\
&= \sum_{m=1}^{N} \beta_m \kappa(\mathbf{x}_m, \mathbf{z}) - \sum_{m=1}^{N} \beta_m \frac{\left(\sum_{n=1}^{N} \alpha_n \kappa(\mathbf{x}_n, \mathbf{z})\kappa(\mathbf{x}_m, \boldsymbol{w})\right)}{\boldsymbol{\alpha}^{\top}K\boldsymbol{\alpha}} && \text{(20e)}\\
&= \sum_{m=1}^{N} \beta_m \kappa(\mathbf{x}_m, \mathbf{z}) - \sum_{m=1}^{N} \beta_m \frac{\left(\sum_{n=1}^{N} \alpha_n \kappa(\mathbf{x}_n, \mathbf{z})\boldsymbol{\Phi}(\mathbf{x}_m)^{\top}\left(\sum_{n'=1}^{N} \alpha_{n'} \boldsymbol{\Phi}(\mathbf{x}_{n'})\right)\right)}{\boldsymbol{\alpha}^{\top}K\boldsymbol{\alpha}} && \text{(20f)}\\
&= \sum_{m=1}^{N} \beta_m \kappa(\mathbf{x}_m, \mathbf{z}) - \sum_{m=1}^{N} \beta_m \frac{\left(\sum_{n=1}^{N}\sum_{n'=1}^{N} \alpha_{n'}\alpha_n \kappa(\mathbf{x}_n, \mathbf{z})\boldsymbol{\Phi}(\mathbf{x}_m)^{\top}\boldsymbol{\Phi}(\mathbf{x}_{n'})\right)}{\boldsymbol{\alpha}^{\top}K\boldsymbol{\alpha}} && \text{(20g)}
\end{aligned}$$

$$= \sum_{m=1}^{N} \beta_m \kappa(\mathbf{x}_m, \mathbf{z}) - \sum_{m=1}^{N} \beta_m \frac{\left(\sum_{n=1}^{N} \sum_{n'=1}^{N} \alpha_n \alpha_{n'} \kappa(\mathbf{x}_n, \mathbf{z}) \kappa(\mathbf{x}_m, \mathbf{x}_{n'})\right)}{\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}} \tag{20h}$$

$$= \sum_{m=1}^{N} \beta_m \left(\kappa(\mathbf{x}_m, \mathbf{z}) - \frac{\boldsymbol{\alpha}^\top K^{(m)}(\mathbf{z}) \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}} \right) \tag{20i}$$

where we define the following matrix component-wise: $K_{ij}^{(m)}(\mathbf{z}) \stackrel{\text{def}}{=} \kappa(\mathbf{x}_i, \mathbf{z})\kappa(\mathbf{x}_m, \mathbf{x}_j)$. Eq. (20i) can be evaluated without explicitly applying the kernel transformation $\boldsymbol{\Phi}$. In terms of notation, when we have $\left\langle \boldsymbol{\theta}, \boldsymbol{\Phi}_{\text{proj}}(\mathbf{z}_n) \right\rangle$, we write

$$\sum_{m=1}^{N} \beta_m \left(\kappa(\mathbf{x}_m, \mathbf{z}_n) - \frac{\boldsymbol{\alpha}^\top K^{(m,n)} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}} \right) \tag{21}$$

where we define $K_{ij}^{(m,n)}(\mathbf{z}) \stackrel{\text{def}}{=} \kappa(\mathbf{x}_i, \mathbf{z}_n)\kappa(\mathbf{x}_m, \mathbf{x}_j)$. This proves the result. ■

A.3 Experimental setting

Data. We conduct experiments on the uncased version of the GloVe representations, which are 300-dimensional, licensed under Apache License, Version 2.0. Following Ravfogel et al. (2020), to approximate the gender labels for the vocabulary, we project all representations on the $\overrightarrow{\text{he}} - \overrightarrow{\text{she}}$ direction, and take the $7,500$ most male-biased and female-biased words. Note that unlike (Bolukbasi et al., 2016), we use the $\overrightarrow{\text{he}} - \overrightarrow{\text{she}}$ direction only to induce approximate gender labels, but then proceed to measure the bias in various ways that go beyond neutralizing just the $\overrightarrow{\text{he}} - \overrightarrow{\text{she}}$ direction. We use the same train–dev–test split of Ravfogel et al. (2020), but discard the gender-neutral words (i.e., we cast the problem as a binary classification). We obtain training, evaluation, and test sets of sizes 7,350, 3,150 and 4,500, respectively. We perform four independent runs of the entire method for all kernel types, with different random seeds.

The kernelized minimax game. For each kernel, we experiment with the following combinations of hyperparameter values:

- Poly : d $\in \{2, 3\}$; $\gamma \in \{0.05, 0.1, 0.15\}$; $\alpha \in \{0.8, 1, 1.2\}$.
- RBF : $\gamma \in \{0.1, 0.15, 0.2\}$
- Laplace : $\gamma \in \{0.1, 0.15, 0.2\}$
- Sigmoid : $\alpha \in \{0, 0.01\}$; $\gamma \in \{0.005, 0.003\}$.

We approximate the kernel space using $L = 1024$ Nyström landmarks. We run the adversarial game Eq. (10) for each of the kernel mappings we consider, by performing alternate minimization and maximization over $\boldsymbol{\theta}$ and P, respectively. As our optimization procedure, we use stochastic gradient descent with a learning rate of 0.08 and minibatches of size 256. We run for 35,000 batches, and choose the projection matrix P which leads to the biggest *decrease* in the linear classification accuracy on the evaluation set. In all cases, we identify a matrix which decreases classification accuracy to near-random accuracy. All training is done on a single NVIDIA GeForce GTX 1080 Ti GPU.

Pre-image mapping. We train an MLP with 2 hidden layers of sizes 512 and 300 to map the original inputs $\mathbf{x}_n$ to inputs which, after being mapped to kernel space, are close to the neutralized features. We use dropout of 0.1, ReLU activation and layer normalization after each hidden layer. We use a skip connection between the input and the output layer, i.e., we set the final output of the MLP to be the sum of its inputs and outputs.[14] We train for 15,000 batches of size 128 and choose the model that yields the lowest loss on the evaluation set.

[14]This is useful because we assume the gender information is not very salient in the original representations, since these encode much more information. With the skip connection, the MLP only needs to learn what to remove, rather than learning to generate plausible representations.

Non-linear gender prediction. We consider the following non-linear predictors: SVMs with different kernels, as well as an MLP with 128 hidden units and ReLU activations. We use the sklearn implementation of predictors. They are trained on the reconstructed pre-image of the training set, and tested on the reconstructed pre-image of the test set. Note that while in training we used an approximation of the kernel function, we predict gender from the pre-images using SVM classifiers that rely on the actual, exact kernel.

A.3.1 Pipeline Evaluation

In this appendix, we include sanity check experiments that aim to assess whether the minimax game Eq. (10) effectively removes *linearly*-present concepts from the *non-linear* kernel features, and whether the training of the pre-image network succeeds.

Concept erasure in kernel space. Do we effectively neutralize the concept in the approximate kernel space? For each kernel, we solve the game Eq. (10) and use the final projection matrix P to create neutralized features. We neutralize the features in RKHS by mapping $\widetilde{\mathbf{\Phi}}(\mathbf{x}_n) \mapsto P\widetilde{\mathbf{\Phi}}(\mathbf{x}_n)$, and train a *linear* classifier to recover the gender labels from the neutralized representations. We get a classification accuracy of 50.59±0.04, very close to majority accuracy of 50.58. This suggests that the process is effective in protecting against the kernel which was applied in training (training a linear classifier on the kernel-transformed representations is equivalent to training a kernel classifier on the original representations). Notice, however, that we cannot test *other* non-linear kernel classifiers on these representations in a similar way: If the approximate kernel mapping $\widetilde{\mathbf{\Phi}}(\cdot)$ corresponds, for example, to a polynomial kernel, we cannot measure the success of an RBF kernel in recovering the bias information after the intervention without performing the pre-image mapping.

Pre-image mapping. Our neutralization algorithm relies on calculating the pre-image of the kernel features after the intervention. To evaluate the quality of the pre-image mapping, we measure the relative reconstruction error $\left|\left|\frac{P\widetilde{\mathbf{\Phi}}(\mathbf{x}_n)-\widetilde{\mathbf{\Phi}}(f(\mathbf{x}_n))}{P\widetilde{\mathbf{\Phi}}(\mathbf{x}_n)}\right|\right|_2^2$ over all points $\mathbf{x}_n$ on the evaluation set. When averaged over all 4 seeds and the different kernels we experimented with, we get a reconstruction error of 1.81±1.61% (range 0.45-7.94).

A.4 Gender Prediction from the Pre-image

In Table 4 we report the full evaluation results on the pre-image neutralized representations, where we have used the same kernel (and the same hyperparameters) for neutralization and gender recovery from the pre-image.

Kernel	γ	α	d	WEAT's d	WEAT's p-value	Gender Acc.
Poly	0.05	0.8	2	0.73 ± 0.01	0.084 ± 0.002	0.49 ± 0.00
Poly	0.05	1	2	0.74 ± 0.01	0.080 ± 0.002	0.49 ± 0.00
Poly	0.05	1.2	2	0.74 ± 0.01	0.081 ± 0.002	0.49 ± 0.00
Poly	0.05	0.8	3	0.75 ± 0.01	0.077 ± 0.003	0.49 ± 0.00
Poly	0.05	1	3	0.74 ± 0.01	0.080 ± 0.003	0.49 ± 0.00
Poly	0.05	1.2	3	0.74 ± 0.01	0.081 ± 0.003	0.49 ± 0.00
Poly	0.1	0.8	2	0.74 ± 0.01	0.080 ± 0.003	0.49 ± 0.00
Poly	0.1	1	2	0.74 ± 0.00	0.080 ± 0.001	0.49 ± 0.00
Poly	0.1	1.2	2	0.74 ± 0.01	0.081 ± 0.002	0.49 ± 0.00
Poly	0.1	0.8	3	0.74 ± 0.00	0.081 ± 0.002	0.53 ± 0.01
Poly	0.1	1	3	0.73 ± 0.01	0.082 ± 0.004	0.65 ± 0.03
Poly	0.1	1.2	3	0.73 ± 0.01	0.084 ± 0.004	0.72 ± 0.01
Poly	0.15	0.8	2	0.73 ± 0.01	0.082 ± 0.003	0.56 ± 0.03
Poly	0.15	1	2	0.74 ± 0.02	0.081 ± 0.006	0.54 ± 0.02
Poly	0.15	1.2	2	0.73 ± 0.01	0.084 ± 0.004	0.55 ± 0.02
Poly	0.15	0.8	3	0.73 ± 0.01	0.082 ± 0.003	0.88 ± 0.02
Poly	0.15	1	3	0.73 ± 0.01	0.082 ± 0.003	0.92 ± 0.00
Poly	0.15	1.2	3	0.74 ± 0.01	0.079 ± 0.003	0.93 ± 0.00
RBF	0.1	-	-	0.75 ± 0.01	0.078 ± 0.003	0.49 ± 0.01
RBF	0.15	-	-	0.74 ± 0.01	0.079 ± 0.003	0.68 ± 0.03
RBF	0.2	-	-	0.74 ± 0.01	0.081 ± 0.003	0.89 ± 0.01
Laplace	0.1	-	-	0.72 ± 0.03	0.086 ± 0.008	0.62 ± 0.04
Laplace	0.15	-	-	0.74 ± 0.05	0.080 ± 0.015	0.77 ± 0.05
Laplace	0.2	-	-	0.67 ± 0.05	0.107 ± 0.020	0.88 ± 0.04
Linear	-	-	-	0.74 ± 0.01	0.079 ± 0.004	0.54 ± 0.02
Sigmoid	0.005	0	-	0.78 ± 0.05	0.069 ± 0.014	0.49 ± 0.00
Sigmoid	0.005	0.01	-	0.73 ± 0.03	0.082 ± 0.010	0.49 ± 0.00
Sigmoid	0.003	0	-	0.73 ± 0.05	0.083 ± 0.017	0.49 ± 0.00
Sigmoid	0.003	0.01	-	0.76 ± 0.03	0.074 ± 0.008	0.49 ± 0.00
EasyMKL	-	-	-	0.73 ± 0.01	0.084 ± 0.005	0.69 ± 0.01
UniformMK	-	-	-	0.73 ± 0.01	0.084 ± 0.002	0.49 ± 0.00
Original	-	-	-	1.56	0.000	≥ 0.99

Table 4: Evaluation of the neutralized pre-image representations. We show the WEAT test's statistics and p-value, as well as the gender prediction accuracy of a kernel classifier of the same type as the one applied during neutralization.

A.5 Closest Neighbors

In Table 5, we show the closest neighbors to randomly-sampled word representations before and after gender erasure under the polynomial kernel. The results for other kernels are qualitatively similar.

Word	Neighbors before	Neighbors after
spiritual	faith, religious, healing	emotional, religious, healing
lesson	learn, teach, lessons	teaching, teach, lessons
faces	faced, facing, face	faced, facing, face
forget	know, let, remember	know, let, remember
converter	ipod, conversion, convert	ipod, conversion, convert
clean	keep, wash, cleaning	keep, wash, cleaning
formal	elegant, dress, appropriate	elegant, appropriate, dress
identity	identify, context, identification	context, identify, identification
other	these, those, many	these, those, many
licensed	registered, certified, license	registered, certified, license
ratings	reviews, rated, rating	reviews, rated, rating
properly	proper, effectively, correctly	effectively, proper, correctly
build	create, built, building	built, create, building
solutions	systems, technologies, solution	services, technologies, solution
afghanistan	troops, pakistan, iraq	troops, pakistan, iraq
wallpaper	desktop, pictures, picture	desktop, pictures, picture
sound	audio, noise, sounds	audio, noise, sounds
gender	sexual, male, age	male, differences, age
boat	cruise, ship, fishing	cruise, ship, fishing
downtown	portland, city, neighborhood	portland, neighborhood, city
lawyers	attorney, lawyer, attorneys	attorney, lawyer, attorneys
smart	how, easy, intelligent	wise, easy, intelligent
spending	budget, spent, spend	budget, spent, spend
contest	winners, winner, competition	winners, winner, competition
want	n't, know, need	n't, know, need
advice	guidance, suggestions, tips	guidance, suggestions, tips
professionals	managers, professional, experts	managers, professional, experts
g	d, b, f	d, b, f
australian	zealand, british, australia	zealand, british, australia
na	mo, o, da	mo, o, da

Table 5: Closest neighbors to randomly-sampled words from GloVe vocabulary, for the original representations, and for the pre-images after our intervention.

A.6 WEAT Results

Here we report the results of the WEAT test for the career and family-related words (Table 6) and art and mathematics-related words (Table 7).

Kernel	γ	α	d	WEAT-d	p-value
Poly	0.05	0.8	2	0.72 ± 0.00	0.093 ± 0.002
Poly	0.05	1	2	0.72 ± 0.01	0.091 ± 0.003
Poly	0.05	1.2	2	0.72 ± 0.01	0.093 ± 0.004
Poly	0.05	0.8	3	0.73 ± 0.00	0.089 ± 0.001
Poly	0.05	1	3	0.74 ± 0.01	0.086 ± 0.004
Poly	0.05	1.2	3	0.73 ± 0.01	0.089 ± 0.002
Poly	0.1	0.8	2	0.72 ± 0.01	0.091 ± 0.005
Poly	0.1	1	2	0.73 ± 0.00	0.089 ± 0.001
Poly	0.1	1.2	2	0.72 ± 0.01	0.090 ± 0.003
Poly	0.1	0.8	3	0.72 ± 0.01	0.090 ± 0.003
Poly	0.1	1	3	0.72 ± 0.01	0.090 ± 0.003
Poly	0.1	1.2	3	0.72 ± 0.01	0.091 ± 0.002
Poly	0.15	0.8	2	0.71 ± 0.01	0.095 ± 0.004
Poly	0.15	1	2	0.74 ± 0.00	0.087 ± 0.001
Poly	0.15	1.2	2	0.72 ± 0.01	0.091 ± 0.002
Poly	0.15	0.8	3	0.72 ± 0.00	0.093 ± 0.001
Poly	0.15	1	3	0.72 ± 0.01	0.092 ± 0.002
Poly	0.15	1.2	3	0.73 ± 0.01	0.090 ± 0.005
RBF	0.1	-	-	0.72 ± 0.01	0.090 ± 0.003
RBF	0.15	-	-	0.73 ± 0.01	0.090 ± 0.005
RBF	0.2	-	-	0.72 ± 0.01	0.091 ± 0.003
Laplace	0.1	-	-	0.75 ± 0.05	0.083 ± 0.017
Laplace	0.15	-	-	0.77 ± 0.02	0.076 ± 0.007
Laplace	0.2	-	-	0.70 ± 0.02	0.098 ± 0.008
Linear	-	-	-	0.72 ± 0.02	0.090 ± 0.006
Sigmoid	0.005	0	-	0.73 ± 0.03	0.087 ± 0.010
Sigmoid	0.005	0.01	-	0.73 ± 0.02	0.087 ± 0.006
Sigmoid	0.003	0	-	0.76 ± 0.06	0.079 ± 0.019
Sigmoid	0.003	0.01	-	0.76 ± 0.07	0.080 ± 0.020
EasyMKL	-	-	-	0.72 ± 0.02	0.091 ± 0.005
UniformMK	-	-	-	0.72 ± 0.00	0.092 ± 0.002
Original	-	-	-	1.69	0.000

Table 6: Word association bias test (WEAT) for career and family-related terms

Kernel	γ	α	d	WEAT-d	p-value
Poly	0.05	0.8	2	0.78 ± 0.00	0.068 ± 0.001
Poly	0.05	1	2	0.78 ± 0.01	0.067 ± 0.002
Poly	0.05	1.2	2	0.78 ± 0.00	0.067 ± 0.001
Poly	0.05	0.8	3	0.78 ± 0.01	0.066 ± 0.002
Poly	0.05	1	3	0.78 ± 0.01	0.066 ± 0.002
Poly	0.05	1.2	3	0.78 ± 0.00	0.066 ± 0.001
Poly	0.1	0.8	2	0.78 ± 0.00	0.068 ± 0.001
Poly	0.1	1	2	0.78 ± 0.01	0.066 ± 0.002
Poly	0.1	1.2	2	0.77 ± 0.01	0.069 ± 0.004
Poly	0.1	0.8	3	0.78 ± 0.00	0.067 ± 0.001
Poly	0.1	1	3	0.77 ± 0.01	0.070 ± 0.001
Poly	0.1	1.2	3	0.78 ± 0.01	0.068 ± 0.002
Poly	0.15	0.8	2	0.78 ± 0.01	0.068 ± 0.003
Poly	0.15	1	2	0.78 ± 0.01	0.068 ± 0.003
Poly	0.15	1.2	2	0.78 ± 0.00	0.068 ± 0.001
Poly	0.15	0.8	3	0.78 ± 0.01	0.067 ± 0.002
Poly	0.15	1	3	0.78 ± 0.01	0.067 ± 0.002
Poly	0.15	1.2	3	0.77 ± 0.01	0.069 ± 0.002
RBF	0.1	-	-	0.79 ± 0.01	0.066 ± 0.003
RBF	0.15	-	-	0.78 ± 0.01	0.067 ± 0.002
RBF	0.2	-	-	0.78 ± 0.01	0.067 ± 0.002
Laplace	0.1	-	-	0.80 ± 0.04	0.064 ± 0.012
Laplace	0.15	-	-	0.81 ± 0.03	0.061 ± 0.009
Laplace	0.2	-	-	0.77 ± 0.04	0.070 ± 0.013
linear	-	-	-	0.79 ± 0.01	0.066 ± 0.002
Sigmoid	0.005	0	-	0.82 ± 0.04	0.057 ± 0.010
Sigmoid	0.005	0.01	-	0.77 ± 0.05	0.070 ± 0.012
Sigmoid	0.003	0	-	0.76 ± 0.03	0.073 ± 0.009
Sigmoid	0.003	0.01	-	0.79 ± 0.03	0.066 ± 0.009
EasyMKL	-	-	-	0.78 ± 0.00	0.066 ± 0.001
UniformMK	-	-	-	0.78 ± 0.01	0.069 ± 0.002
Original	-	-	-	1.56	0.000

Table 7: Word association bias test (WEAT) for art and mathematics-related terms

A.7 Transfer Results

	UniformMK	EasyMKL	RBF	Poly	Laplace	Sigmoid	Linear	MLP
poly, γ =0.05, d= 2, α =0.8	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.01	0.49 $\pm$ 0.00	0.61 $\pm$ 0.06	0.97 $\pm$ 0.00
poly, γ =0.05, d= 2, α =1	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.01	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.52 $\pm$ 0.01	0.97 $\pm$ 0.00
poly, γ =0.05, d= 2, α =1.2	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.53 $\pm$ 0.02	0.97 $\pm$ 0.00
poly, γ =0.05, d= 3, α =0.8	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.01	0.98 $\pm$ 0.00	0.93 $\pm$ 0.01	0.49 $\pm$ 0.00	0.54 $\pm$ 0.03	0.97 $\pm$ 0.00
poly, γ =0.05, d= 3, α =1	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.01	0.49 $\pm$ 0.00	0.59 $\pm$ 0.05	0.97 $\pm$ 0.00
poly, γ =0.05, d= 3, α =1.2	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.53 $\pm$ 0.01	0.97 $\pm$ 0.00
poly, γ =0.1, d= 2, α =0.8	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.01	0.49 $\pm$ 0.00	0.54 $\pm$ 0.04	0.97 $\pm$ 0.00
poly, γ =0.1, d= 2, α =1	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.01	0.49 $\pm$ 0.00	0.53 $\pm$ 0.02	0.97 $\pm$ 0.00
poly, γ =0.1, d= 2, α =1.2	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.54 $\pm$ 0.01	0.97 $\pm$ 0.00
poly, γ =0.1, d= 3, α =0.8	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.54 $\pm$ 0.00	0.97 $\pm$ 0.00
poly, γ =0.1, d= 3, α =1	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.00	0.49 $\pm$ 0.00	0.56 $\pm$ 0.02	0.97 $\pm$ 0.00
poly, γ =0.1, d= 3, α =1.2	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.00	0.49 $\pm$ 0.00	0.54 $\pm$ 0.03	0.97 $\pm$ 0.00
poly, γ =0.15, d= 2, α =0.8	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.00	0.49 $\pm$ 0.00	0.57 $\pm$ 0.02	0.97 $\pm$ 0.00
poly, γ =0.15, d= 2, α =1	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.54 $\pm$ 0.01	0.97 $\pm$ 0.00
poly, γ =0.15, d= 2, α =1.2	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.00	0.49 $\pm$ 0.00	0.54 $\pm$ 0.01	0.97 $\pm$ 0.00
poly, γ =0.15, d= 3, α =0.8	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.01	0.49 $\pm$ 0.00	0.58 $\pm$ 0.03	0.97 $\pm$ 0.00
poly, γ =0.15, d= 3, α =1	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.00	0.49 $\pm$ 0.00	0.57 $\pm$ 0.01	0.97 $\pm$ 0.00
poly, γ =0.15, d= 3, α =1.2	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.00	0.49 $\pm$ 0.00	0.56 $\pm$ 0.01	0.97 $\pm$ 0.00
rbf, γ =0.1	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.58 $\pm$ 0.04	0.97 $\pm$ 0.00
rbf, γ =0.15	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.60 $\pm$ 0.03	0.97 $\pm$ 0.00
rbf, γ =0.2	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.01	0.49 $\pm$ 0.00	0.60 $\pm$ 0.04	0.97 $\pm$ 0.00
laplace, γ =0.1	0.60 $\pm$ 0.06	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.01	0.49 $\pm$ 0.00	0.61 $\pm$ 0.03	0.97 $\pm$ 0.00
laplace, γ =0.15	0.56 $\pm$ 0.04	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.61 $\pm$ 0.01	0.97 $\pm$ 0.00
laplace, γ =0.2	0.59 $\pm$ 0.06	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.62 $\pm$ 0.02	0.97 $\pm$ 0.00
linear	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.93 $\pm$ 0.00	0.49 $\pm$ 0.00	0.54 $\pm$ 0.02	0.97 $\pm$ 0.00
sigmoid, γ =0.005, α =0	0.64 $\pm$ 0.05	0.97 $\pm$ 0.00	0.93 $\pm$ 0.01	0.98 $\pm$ 0.00	0.90 $\pm$ 0.01	0.49 $\pm$ 0.00	0.65 $\pm$ 0.05	0.97 $\pm$ 0.00
sigmoid, γ =0.005, α =0.01	0.63 $\pm$ 0.02	0.97 $\pm$ 0.00	0.94 $\pm$ 0.00	0.98 $\pm$ 0.00	0.90 $\pm$ 0.01	0.49 $\pm$ 0.00	0.65 $\pm$ 0.04	0.97 $\pm$ 0.00
sigmoid, γ =0.003, α =0	0.63 $\pm$ 0.07	0.97 $\pm$ 0.00	0.92 $\pm$ 0.01	0.98 $\pm$ 0.00	0.89 $\pm$ 0.01	0.49 $\pm$ 0.00	0.66 $\pm$ 0.07	0.97 $\pm$ 0.00
sigmoid, γ =0.003, α =0.01	0.65 $\pm$ 0.03	0.97 $\pm$ 0.00	0.92 $\pm$ 0.01	0.98 $\pm$ 0.00	0.89 $\pm$ 0.01	0.49 $\pm$ 0.00	0.65 $\pm$ 0.05	0.97 $\pm$ 0.00
EasyMKL	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.00	0.49 $\pm$ 0.00	0.57 $\pm$ 0.03	0.97 $\pm$ 0.00
UniformMK	0.49 $\pm$ 0.00	0.98 $\pm$ 0.00	0.96 $\pm$ 0.00	0.98 $\pm$ 0.00	0.94 $\pm$ 0.01	0.49 $\pm$ 0.00	0.58 $\pm$ 0.08	0.97 $\pm$ 0.00

Table 8: Gender prediction from the neutralized pre-image representations using non-linear adversaries that differ from the neutralizing kernel.

In this appendix, we provide gender prediction accuracy, on the neutralized pre-image representations, with predictors that are *different* from those used in training (Experiment §5.2).

Setup. After projecting out the gender concept in kernel space, and computing the pre-image of the neutralized representations, we apply different non-linear kernels as well as an MLP to predict gender. We use the following parameters:

- RBF : $\gamma = 0.3$.
- Poly : d $= 3, \gamma = 0.5, \alpha = 0.3$.
- Laplace : $\gamma = 0.3$.
- Sigmoid : $\alpha = 0, \gamma = 0.01$.
- MLP : A network with a single 128-dimensional hidden layer with ReLU activations.

All classifiers were trained using sklearn.

Results. The results are shown in Table 8. Rows denote the kernel that was applied for neutralization in Eq. (10), while columns denote the type of adversarial classifier applied on the final pre-image representations. Numbers denote accuracy in gender prediction.