Utility-Fairness Trade-Offs and How to Find Them (Supplementary Material)

Sepehr Dehdashtian Bashir Sadeghi Vishnu Naresh Boddeti
Michigan State University
{sepehr, sadeghib, vishnu}@msu.edu

In our main paper, we introduced two types of utility-fairness trade-offs and proposed a method, U-FaTE, to estimate them. Here, we provide some additional analysis to support our main results. The supplementary material is structured as follows:

1. Representation Disentanglement in (§1)
2. Training Process of U-FaTE in (§2)
3. Implementation Details in (§3)
4. Evaluation Metrics in (§4)
5. Weighted Normalized Euclidean Distance in (§5)
6. Proofs of closed-form solutions for different notions of fairness (§6)

1. Disentanglement of the Representation

A common objective of learned representations is compactness [1] to avoid learning representations with redundant information where different dimensions are highly correlated. Therefore, going beyond the assumption that each component of $\boldsymbol{f}(\cdot)$ (i.e., $f_j(\cdot)$) belongs to a universal RKHS $\mathcal{H}_X$, we impose additional constraints on the representation. Specifically, we constrain the search space of the encoder $\boldsymbol{f}(\cdot)$ to learn a disentangled representation [1] as

$$\mathcal{A}_r := \Big\{ (f_1, \cdots, f_r) \;\big|\; f_i, f_j \in \mathcal{H}_{\tilde{X}}, \mathbb{C}\text{ov}\left(f_i(\tilde{X}), f_j(\tilde{X})\right) + \gamma \langle f_i, f_j \rangle_{\mathcal{H}_{\tilde{X}}} = \delta_{i,j} \Big\}, \tag{1}$$

where the $\mathbb{C}\text{ov}\big(f_i(X), f_j(X)\big)$ part enforces the covariance of Z to be an identity matrix. This kind of disentanglement is used in PCA and encourages the variance of each entry of Z to be bounded and different entries of Z are uncorrelated to each other. The regularization part, $\gamma \langle f_i, f_j \rangle_{\mathcal{H}_X}$ encourages the encoder components to be as orthogonal as possible to each other and to be of the unit norm, and aids with numerical stability during empirical estimation [2].

2. Training Process of U-FaTE

Fig. 1 shows an overview of the training process of U-FaTE which includes two phases. In the first phase, the features of the training samples are extracted and used to find a closed-form solution for the encoder to maximize the objective function in (4) while the parameters of the feature extractor ($\boldsymbol{\Theta}_{FE}$) are frozen. In the second phase, the feature extractor is trained by updating its parameters using SGD in order to maximize (4) while the encoder is frozen. These two phases are repeated until convergence. These details are also mentioned in Algorithm 1.

3. Implementation Details

In training all of the methods, we pick different values of the fairness control parameter (λ) between zero and one to obtain the trade-offs. Moreover, each experiment is run for 5 different random seeds. For datasets that contain image data, we used the first two blocks of ResNet18 [4] and put a fully connected layer with 2048 neurons as the last layer of the feature extractor. We used an embedding layer for the dataset with tabular data to map the raw data into an embedding space. A 3-layer MLP is used as the target classifier network for all datasets and models. For both FolkTables and CelebA datasets, the number of dimensions of RFF is set to 1000. In the training phase, the cosine annealing scheduler [6] is used for scheduling the learning rate. The dimension of representations (r) is chosen $c-1$ where c is the number of target attribute's classes. To improve training stability, we normalize the feature extractor's output $\tilde{X}$. These implementation details are summarized in Tab. 1.

Algorithm 1: U-FaTE Training Process

Input: $\boldsymbol{X} \in \mathbb{R}^{n \times h \times w \times c}$, $\boldsymbol{Y} \in \mathbb{R}^{n \times |Y|}$, $\boldsymbol{S} \in \mathbb{R}^{n \times |S|}$, $m \in \mathbb{N}$
Output: $\boldsymbol{f}_{FE}$, $\boldsymbol{f}_{Enc}$
Initialize:
$i \leftarrow 0$;
$\boldsymbol{f}_{FE} \leftarrow$ Random Initialization;
; /* Train Feature Extractor and Encoder */
while $i < m$ **do**
  $\boldsymbol{f}_{Enc} \leftarrow \sup_{\boldsymbol{f}_{Enc} \in \mathcal{A}_r} \{J^{\text{emp}}(\boldsymbol{f}(X; \boldsymbol{\Theta}_{Enc}))\}$; /* solve (6) */
  $\boldsymbol{f}_{FE} \leftarrow \text{SGD}\Big\{ \sup_{\boldsymbol{f}_{FE}} \{J^{\text{emp}}(\boldsymbol{f}(X; \boldsymbol{\Theta}_{FE}))\} \Big\}$; /* solve (2) */
  $i \leftarrow i + 1$
end
$\boldsymbol{f}_{CLF} \leftarrow \text{SGD}\Big\{ \sup_{\boldsymbol{f}_{CLF}} \{J(\boldsymbol{f}(X; \boldsymbol{\Theta}_{CLF}), \boldsymbol{Y})\} \Big\}$; /* Train Classifier */

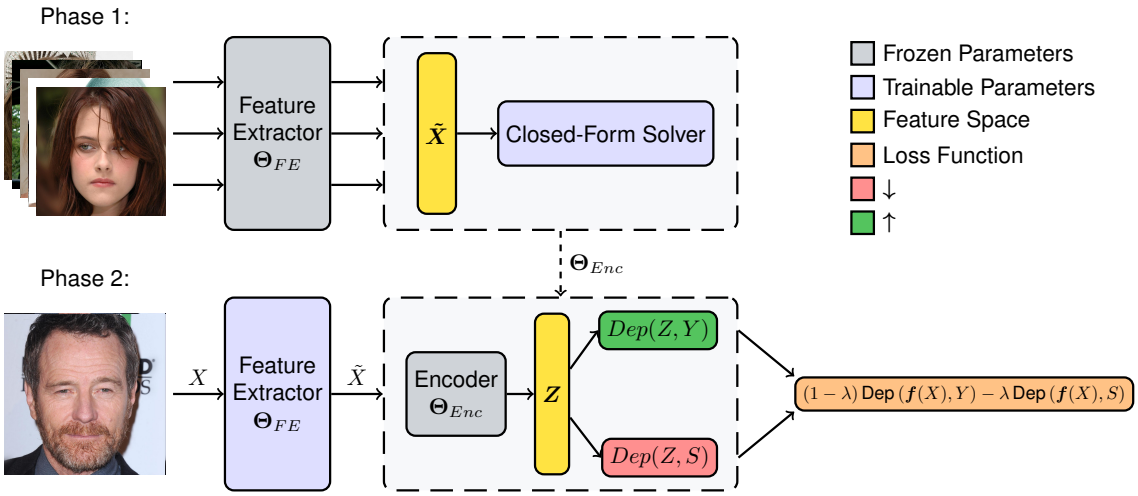

Figure 1. Training process of U-FaTE contains two phases. **Phase 1:** The closed-form solution for the encoder is calculated using the features generated by the feature extractor while its parameters are frozen. **Phase 2:** The feature extractor is trained using the loss provided by the calculated encoder parameters from Phase 1.

Dataset	RFF Dim.	r	Training samples
CelebA	1000	1	182,637
FairFace	1000	1	86,744
FolkTables	1000	1	75,745

Table 1. Implementation details of U-FaTE for each dataset.

4. Evaluation Metrics

For measuring the utility of the target prediction, we use the accuracy of the classification task. Furthermore, we use equality of odds difference (EOOD) or equal opportunity difference (EOD). EOOD[3] is defined as

$$\text{EOOD} := \left| P\left(\hat{Y} = 1 | S = 0, Y = y\right) - P\left(\hat{Y} = 1 | S = 1, Y = y\right) \right|, \tag{2}$$

where $y \in \{0, 1, \cdots, |Y| - 1\}$, $\hat{Y}$ is the predicted label, and S is the sensitive attribute. According to this criterion, the model should exhibit similar prediction error rates for different groups, irrespective of their sensitive attributes. EOD[3] can also be defined as

$$\text{EOD} := \left| P\left(\hat{Y} = 1 | S = 0, Y = 1\right) - P\left(\hat{Y} = 1 | S = 1, Y = 1\right) \right| \tag{3}$$

This is a relaxation of EOOD for the case of binary target tasks. This metric indicates that the model should provide equal opportunities for positive predictions for individuals with the same true outcome, regardless of their sensitive attributes.

5. Weighted Normalized Euclidean Distance

In the main paper, to compare methods based on their point-to-point distance to LST and DST, we use two *weighted normalized Euclidean distance* defined as:

$$\text{Dist}_{\text{LST}}(x^i) = \sqrt{w \cdot \left(\frac{\text{LST}_f - x^i_f}{\max_f}\right)^2 + (1-w) \cdot \left(\frac{\text{LST}_{\text{Acc}} - x^i_{\text{Acc}}}{\max_{\text{Acc}}}\right)^2} \tag{4}$$

$$\text{Dist}_{\text{DST}}(x^i) = \sqrt{w \cdot \left(\frac{\text{DST}_f - x^i_f}{\max_f}\right)^2 + (1-w) \cdot \left(\frac{\text{DST}_{\text{Acc}} - x^i_{\text{Acc}}}{\max_{\text{Acc}}}\right)^2} \tag{5}$$

where $f \in \mathcal{F}$ is the fairness metric and $\mathcal{F} = \{\text{EOD}, \text{EOOD}, \text{DPV}\}$, w is the control parameter that adjusts the weights of each term —fairness distance and accuracy distance—in the overall distance. For calculating distances in Table 1 of the main paper, we choose $w = 0.5$ which means that the distances in the fairness axis and distances in the accuracy axis are equally important to us.

6. Solutions for Different Notions of Fairness

6.1. Proof of Theorem 1 for EO

Theorem 1. *Let the Cholesky factorization of $\boldsymbol{K}_X$ be $\boldsymbol{K}_X = \boldsymbol{L}_X \boldsymbol{L}_X^T$, where $\boldsymbol{L}_X \in \mathbb{R}^{n \times d}$ $(d \leq n)$ is a full column-rank matrix. Let $r \leq d$, then a solution to (4) is*

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

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

$$\left((1-\lambda)\frac{1}{n^2}\boldsymbol{L}_X^T \boldsymbol{H}\boldsymbol{K}_Y \boldsymbol{H}\boldsymbol{L}_X - \lambda \frac{1}{n_0^2}\boldsymbol{L}_X[Y = y_0]^T \boldsymbol{H}\boldsymbol{K}_S[Y = y_0]\boldsymbol{H}\boldsymbol{L}_X[Y = y_0]\right)\boldsymbol{u} = \tau\left(\frac{1}{n}\boldsymbol{L}_X^T \boldsymbol{H}\boldsymbol{L}_X + \gamma \boldsymbol{I}\right)\boldsymbol{u}. \tag{6}$$

Proof. Consider the Cholesky factorization, $\boldsymbol{K}_X = \boldsymbol{L}_X \boldsymbol{L}_X^T$ where $\boldsymbol{L}_X$ is a full column-rank matrix. Using the representer theorem, the disentanglement property in (1) can be expressed as

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

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

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

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

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

$$
\begin{aligned}
& \sup_{\boldsymbol{f}\in\mathcal{A}_r} \left\{(1-\lambda)\,\mathrm{Dep}^{\mathrm{emp}}(\boldsymbol{f}(X), Y) - \lambda\,\mathrm{Dep}^{\mathrm{emp}}(\boldsymbol{f}(X), S|Y=1)\right\} \\
= & \sup_{\boldsymbol{f}\in\mathcal{A}_r} \left\{(1-\lambda)\frac{1}{n^2}\left\|\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{H}\boldsymbol{L}_Y\right\|_F^2 - \lambda\,\frac{1}{n_0^2}\left\|\boldsymbol{\Theta}\boldsymbol{K}_X[Y=y_0]\boldsymbol{H}\boldsymbol{L}_{S_0}\right\|_F^2\right\} \\
= & \sup_{\boldsymbol{f}\in\mathcal{A}_r} \left\{(1-\lambda)\frac{1}{n^2}\,\mathrm{Tr}\left\{\boldsymbol{\Theta}\boldsymbol{K}_X\boldsymbol{H}\boldsymbol{K}_Y\boldsymbol{H}\boldsymbol{K}_X\boldsymbol{\Theta}^T\right\} - \lambda\,\frac{1}{n_0^2}\mathrm{Tr}\left\{\boldsymbol{\Theta}\boldsymbol{K}_X[Y=y_0]\boldsymbol{H}\boldsymbol{K}_{S_0}\boldsymbol{H}\boldsymbol{K}_X[Y=y_0]^T\boldsymbol{\Theta}^T\right\}\right\} \\
= & \max_{\boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}=\boldsymbol{I}_r} \mathrm{Tr}\left\{\boldsymbol{\Theta}\boldsymbol{L}_X\boldsymbol{B}\boldsymbol{L}_X^T\boldsymbol{\Theta}^T\right\} \\
= & \max_{\boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}=\boldsymbol{I}_r} \mathrm{Tr}\left\{\boldsymbol{V}^T\boldsymbol{B}\boldsymbol{V}\right\} \qquad (7)
\end{aligned}
$$

where the second step is due to (3) and

$$
\boldsymbol{B} \;:=\; \left((1-\lambda)\frac{1}{n^2}\boldsymbol{L}_X^T\boldsymbol{H}\boldsymbol{K}_Y\boldsymbol{H}\boldsymbol{L}_X - \lambda\frac{1}{n_0^2}\boldsymbol{L}_X[Y=y_0]^T\boldsymbol{H}\boldsymbol{K}_S[Y=y_0]\boldsymbol{H}\boldsymbol{L}_X[Y=y_0]\right)
$$

It is shown in [5] that an[1] optimizer of (7) is any matrix $\boldsymbol{U}$ whose columns are eigenvectors corresponding to r largest eigenvalues of generalized problem

$$
\boldsymbol{B}\boldsymbol{u} = \tau\,\boldsymbol{C}\boldsymbol{u} \qquad (8)
$$

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

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

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

6.2. Proof of Theorem 1 for EOO

Theorem 2. *Let the Cholesky factorization of $\boldsymbol{K}_X$ be $\boldsymbol{K}_X = \boldsymbol{L}_X\boldsymbol{L}_X^T$, where $\boldsymbol{L}_X \in \mathbb{R}^{n\times d}$ ($d \le n$) is a full column-rank matrix. Let $r \le d$, then a solution to (4) is*

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

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

$$
\left((1-\lambda)\frac{1}{n^2}\boldsymbol{L}_X^T\boldsymbol{H}\boldsymbol{K}_Y\boldsymbol{H}\boldsymbol{L}_X - \lambda\sum_{y=0}^{c_y-1}\frac{1}{n_y^2}\boldsymbol{L}_X[Y=y]^T\boldsymbol{H}\boldsymbol{K}_S[Y=y]\boldsymbol{H}\boldsymbol{L}_X[Y=y]\right)\boldsymbol{u} = \tau\left(\frac{1}{n}\,\boldsymbol{L}_X^T\boldsymbol{H}\boldsymbol{L}_X + \gamma\boldsymbol{I}\right)\boldsymbol{u}. \qquad (9)
$$

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

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

theorem, the disentanglement property in (1) can be expressed as

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

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

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

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

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

$$
\begin{aligned}
& \sup_{\boldsymbol{f}\in\mathcal{A}_r} \left\{ (1-\lambda)\, \mathrm{Dep}^{\mathrm{emp}}(\boldsymbol{f}(X), Y) - \lambda \sum_{y=0}^{c_y-1} \mathrm{Dep}^{\mathrm{emp}}(\boldsymbol{f}(X), S|Y=y) \right\} \\
= \;& \sup_{\boldsymbol{f}\in\mathcal{A}_r} \left\{ (1-\lambda)\frac{1}{n^2} \left\|\boldsymbol{\Theta}\boldsymbol{K}_X \boldsymbol{H} \boldsymbol{L}_Y\right\|_F^2 - \lambda \sum_{y=0}^{c_y-1} \frac{1}{n_y^2} \left\|\boldsymbol{\Theta}\boldsymbol{K}_X[Y=y] \boldsymbol{H} \boldsymbol{L}_{S_y}\right\|_F^2 \right\} \\
= \;& \sup_{\boldsymbol{f}\in\mathcal{A}_r} \left\{ (1-\lambda)\frac{1}{n^2} \mathrm{Tr}\left\{\boldsymbol{\Theta}\boldsymbol{K}_X \boldsymbol{H} \boldsymbol{K}_Y \boldsymbol{H} \boldsymbol{K}_X \boldsymbol{\Theta}^T\right\} - \lambda \sum_{y=0}^{c_y-1} \frac{1}{n_y^2} \mathrm{Tr}\left\{\boldsymbol{\Theta}\boldsymbol{K}_X[Y=y] \boldsymbol{H} \boldsymbol{K}_{S_y} \boldsymbol{H} \boldsymbol{K}_X[Y=y]^T \boldsymbol{\Theta}^T\right\} \right\} \\
= \;& \max_{\boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}=\boldsymbol{I}_r} \mathrm{Tr}\left\{\boldsymbol{\Theta}\boldsymbol{L}_X \boldsymbol{B} \boldsymbol{L}_X^T \boldsymbol{\Theta}^T\right\} \\
= \;& \max_{\boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}=\boldsymbol{I}_r} \mathrm{Tr}\left\{\boldsymbol{V}^T \boldsymbol{B} \boldsymbol{V}\right\} \qquad (10)
\end{aligned}
$$

where the second step is due to (3) and

$$
\boldsymbol{B} \;:=\; \left((1-\lambda)\frac{1}{n^2}\boldsymbol{L}_X^T \boldsymbol{H} \boldsymbol{K}_Y \boldsymbol{H} \boldsymbol{L}_X - \lambda \sum_{y=0}^{c_y-1} \frac{1}{n_y^2} \boldsymbol{L}_X[Y=y]^T \boldsymbol{H} \boldsymbol{K}_S[Y=y] \boldsymbol{H} \boldsymbol{L}_X[Y=y] \right)
$$

It is shown in [5] that an[2] optimizer of (10) is any matrix $\boldsymbol{U}$ whose columns are eigenvectors corresponding to r largest eigenvalues of generalized problem

$$
\boldsymbol{B}\boldsymbol{u} = \tau\, \boldsymbol{C}\boldsymbol{u} \qquad (11)
$$

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

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

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

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

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. 1

[2] Kenji Fukumizu, Francis R Bach, and Arthur Gretton. Statistical consistency of kernel canonical correlation analysis. *Journal of Machine Learning Research*, 8(2), 2007. 1

[3] Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, 2016. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[5] Effrosini Kokiopoulou, Jie Chen, and Yousef Saad. Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications*, 18(3):565–602, 2011. 4, 5

[6] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1