Fair Streaming Principal Component Analysis: Statistical and Algorithmic Viewpoint

Junghyun Lee* **Hanseul Cho*** **Se-Young Yun** **Chulhee Yun**
Kim Jaechul Graduate School of AI
KAIST
Seoul, Republic of Korea
{jh_lee00, jhs4015, yunseyoung, chulhee.yun}@kaist.ac.kr

Abstract

Fair Principal Component Analysis (PCA) is a problem setting where we aim to perform PCA while making the resulting representation fair in that the projected distributions, conditional on the sensitive attributes, match one another. However, existing approaches to fair PCA have two main problems: theoretically, there has been no statistical foundation of fair PCA in terms of learnability; practically, limited memory prevents us from using existing approaches, as they explicitly rely on full access to the entire data. On the theoretical side, we rigorously formulate fair PCA using a new notion called *probably approximately fair and optimal* (PAFO) learnability. On the practical side, motivated by recent advances in streaming algorithms for addressing memory limitation, we propose a new setting called *fair streaming PCA* along with a memory-efficient algorithm, fair noisy power method (FNPM). We then provide its *statistical* guarantee in terms of PAFO-learnability, which is the first of its kind in fair PCA literature. Lastly, we verify the efficacy and memory efficiency of our algorithm on real-world datasets.

1 Introduction

Algorithmic fairness ensures that machine learning algorithms do not propagate nor exacerbate bias, which may lead to discriminatory decision-making (Barocas and Selbst, 2016) and thus has been a very active area of research. This has direct implications in our everyday life, including but not limited to criminal justice (Kirchner et al., 2016), education (Kizilcec and Lee, 2021), and more. See Mehrabi et al. (2021) for a comprehensive survey of bias and fairness in machine learning.

Often, one needs to consider fairness for a large number of high-dimensional data points. One of the standard tools for dealing with such high-dimensional data is PCA (Hotelling, 1933; Pearson, 1901), a classical yet still one of the most popular algorithms for performing interpretable dimensionality reduction. It has been adapted as a baseline and/or standard tool in exploratory data analysis, whose application ranges from natural sciences, engineering (Abdi and Williams, 2010; Jolliffe and Cadima, 2016), and even explainable AI (Li et al., 2023; Tjoa and Guan, 2021). Due to its ubiquity and wide applicability, several works have studied defining fairness in PCA. and developing a fair variant of it. A recent line of research (Kleindessner et al., 2023; Lee et al., 2022; Olfat and Aswani, 2019) defines PCA fairness in the context of fair representation (Zemel et al., 2013) in that the projected group conditional distributions should match.

However, existing fair PCA approaches suffer from two problems. Theoretically, they provide no statistical foundation of fair PCA or guarantees for their algorithms. By statistical, we mean how to measure the quality of the estimated solution compared to the true solution obtained when the entire distribution is known, e.g., PAC-learnability (Shalev-Schwartz and Ben-David, 2014).

[*] Equal contributions

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

On top of that, the second problem arises from a practical viewpoint: memory limitation. All the aforementioned fair PCA algorithms assume that the learner can store the entire data points and incurs memory complexity of order at least $\mathcal{O}(d \max(N, d))$, where d is the dimensionality of the data and N is the number of data points. As memory limitation is often a critical bottleneck in deploying machine learning algorithms (Mitliagkas et al., 2013), as much as fairness is important, it is also paramount that imposing fairness to PCA does not incur too much memory overhead. A popular approach to mitigate such memory limitation for PCA is to consider the one-pass, streaming setting, where each data point is revealed to the learner sequentially, each point is irretrievably gone unless she explicitly stores it, and she can use only $\mathcal{O}(dk)$ memory, with k being the target dimension of projection. Indeed, without the fairness constraint, streaming PCA has been studied extensively; see Balzano et al. (2018) and references therein.

In this work, we address both problems in a principled manner. Our contributions are as follows:

- We provide an alternative formulation of fair PCA based on the "Null It Out" approach (Section 3). Based on the new formulation, we introduce the concept of *probably approximately fair and optimal* (PAFO)-learnability to formalize the problem of fair PCA (Section 4).
- To address the memory limitation, we propose a new problem setting, called *fair streaming PCA*, and propose a simple yet memory-efficient algorithm based on the noisy power method (Section 5). We note that our algorithm incurs a much lower memory complexity even compared to the most efficient variant of fair PCA proposed by Kleindessner et al. (2023).
- We then prove that our algorithm achieves the PAFO-learnability for fair streaming PCA (Section 6). Such statistical guarantee is the first of its kind in fair PCA literature.
- Lastly, we empirically validate our algorithm on several real-world datasets. Notably, we run FNPM on the original *full-resolution* CelebA dataset on which existing fair PCA algorithms fail due to high memory requirements. It shows turning such a non-streaming setting into a streaming setting and applying our algorithm allows one to bypass the memory limitation (Section 7).

2 Preliminaries

Notations. For $\ell \geq 1$, let $\boldsymbol{I}_\ell$ be the identity matrix of size $\ell \times \ell$. For $k < d$, we bring the *Stiefel manifold* $St(d,k) = \{\boldsymbol{A} \in \mathbb{R}^{d \times k} : \boldsymbol{A}^\intercal \boldsymbol{A} = \boldsymbol{I}_k\}$, which is basically the collection of all rank-k orthogonal matrices. We denote an orthonormal column basis of a (full-rank) matrix $\boldsymbol{M} \in \mathbb{R}^{d \times k}$ obtained by QR decomposition as $\texttt{QR}(\boldsymbol{M}) \in St(d,k)$ and denote its column space by $\mathrm{col}(\boldsymbol{A})$. Also, for $\boldsymbol{A} \in St(d,k)$, we denote the orthogonal projection matrix to $\mathrm{col}(\boldsymbol{A})^\perp$ as $\boldsymbol{\Pi}_{\boldsymbol{A}} = \boldsymbol{I}_d - \boldsymbol{A}\boldsymbol{A}^\intercal$, where $\perp$ is the orthogonal complement operator. Moreover, we denote the collection of all possible d-dimensional probability distributions as $\mathcal{P}_d$. Lastly, we use the usual $\mathcal{O}$, Ω, and Θ notations for asymptotic analyses, where tildes ($\tilde{\mathcal{O}}$, $\tilde{\Omega}$, and $\tilde{\Theta}$, resp.) are used for hiding logarithmic factors.

Setup. Assume that the sensitive attribute variable w.r.t., which we will be imposing fairness, is binary[2], denoted by $s \in \{0,1\}$. For each $s \in \{0,1\}$, let $\mathcal{D}_s$ be a d-dimensional distribution of mean $\boldsymbol{\mu}_s$ and covariance $\boldsymbol{\Sigma}_s$, both of which are assumed to be well-defined. We often call them group-conditional mean and covariance, respectively. With a fixed, *unknown* mixture parameter $p \in (0,1)$, let us denote the total data distribution as $\mathcal{D} := p\mathcal{D}_0 + (1-p)\mathcal{D}_1$. Equivalently, the sensitive attribute follows $s \sim \mathrm{Bernoulli}(p)$, and the conditional random variable $\boldsymbol{x}|s$ is sampled from $\mathcal{D}_s$. In that case, $\boldsymbol{\mu}_s = \mathbb{E}[\boldsymbol{x}|s]$ and $\boldsymbol{\Sigma}_s = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal|s] - \boldsymbol{\mu}_s\boldsymbol{\mu}_s^\intercal$. We often write $p_0 = 1-p$ and $p_1 = p$ for brevity. We also define the mean difference $\boldsymbol{f} := \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$ and the *mean-augmented* covariance difference (or, simply, covariance difference) $\boldsymbol{Q} := \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal|s=1] - \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal|s=0] = \boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_0 + \boldsymbol{\mu}_1\boldsymbol{\mu}_1^\intercal - \boldsymbol{\mu}_0\boldsymbol{\mu}_0^\intercal$. Accordingly, denote the true mean and covariance of $\mathcal{D}$ as $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively. For simplicity, let us assume that $\mathcal{D}$ is centered, i.e., $\boldsymbol{\mu} = \boldsymbol{0}$; note that this does *not* mean that the group conditional distributions $\mathcal{D}_s$'s are centered.

PCA. In the *offline* setting, the full covariance matrix $\boldsymbol{\Sigma}$ is given which is often a sample covariance matrix $\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i^\top$ for n data points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$. The goal of vanilla (offline) PCA (Hotelling, 1933; Pearson, 1901) is to compute the loading matrix $\boldsymbol{V} \in \mathbb{R}^{d \times k}$ that preserves as much variance as possible after projecting $\boldsymbol{\Sigma}$ via $\boldsymbol{V}$, i.e., maximize $\mathrm{tr}(\boldsymbol{V}^\intercal\boldsymbol{\Sigma}\boldsymbol{V})$. Here, $k < d$ is the target dimension

[2] The discussion here can be easily extended to non-binary sensitive attributes and/or multiple groups; see Section 3.4 and 3.5 of Kleindessner et al. (2023).

to which the data's dimensionality d is to be reduced and is chosen by the learner. We additionally consider the constraint of $\boldsymbol{V}^\intercal \boldsymbol{V} = \boldsymbol{I}_k$ (*i.e.*, $\boldsymbol{V} \in St(d,k)$) to ensure that the resulting coordinate after the transformation is orthogonal and thus amenable to various statistical interpretations (Johnson and Wichern, 2008). Without any fairness constraint, Eckart-Young theorem (Eckart and Young, 1936) implies that the solution is characterized as a matrix whose columns are the top-k eigenvectors of $\boldsymbol{\Sigma}$.

Fair PCA. Recently, it has been suggested that performing vanilla PCA on real-world datasets may exhibit bias, making the final outputted projection "unfair". As is often the case, there can be multiple definitions of fairness in PCA, but the following two are the most popular: equalizing reconstruction losses (Kamani et al., 2022; Samadi et al., 2018; Tantipongpipat et al., 2019; Vu et al., 2022), or equalizing the projected distributions (Kleindessner et al., 2023; Lee et al., 2022; Olfat and Aswani, 2019) from the perspective of fair representation (Zemel et al., 2013); we focus on the latter one.

3 An Alternative Approach to Fair PCA

3.1 "Null It Out" Formulation of Fair PCA

In this work, we consider fair PCA as learning fair representation (Zemel et al., 2013). The goal is to preserve as much variance as possible while obfuscating any information regarding the sensitive attribute. Slightly different from previous fair PCA approaches (Kleindessner et al., 2023; Lee et al., 2022; Olfat and Aswani, 2019), we take the "Null It Out" approach as proposed in Ravfogel et al. (2020). Intuitively, we want to nullify the "directions" in which the sensitive attribute s can be inferred, and in this work, we consider two such "directions": mean difference $\boldsymbol{f}$ and *eigenvectors* of covariance difference $\boldsymbol{Q}$. To give the learner flexibility in choosing the trade-off between fairness and performance (measured in explained variance), let $m \geq 1$ be the number of top eigenvectors of $\boldsymbol{Q}$ to nullify. Thus the learner is nullifying $(m+1)$-dimensional subspace that is "unfair" w.r.t. s, which we refer to as the ***unfair subspace***. Precisely, we formulate our fair PCA as follows:

$$\max_{\boldsymbol{V}^\intercal \boldsymbol{V} = \boldsymbol{I}_k} \operatorname{tr}(\boldsymbol{V}^\intercal \boldsymbol{\Sigma} \boldsymbol{V}), \quad \text{subject to } \operatorname{col}(\boldsymbol{V}) \subset \operatorname{col}(\boldsymbol{f})^\perp \cap \operatorname{col}(\boldsymbol{P}_m)^\perp, \tag{1}$$

where d is the data dimensionality, k is the target dimension, and $\boldsymbol{P}_m$ is top-m eigenvectors of $\boldsymbol{Q}$. Here, $\boldsymbol{V} \in St(d,k)$ is called the loading matrix.

3.2 Closed Form Solution of Fair PCA

To first construct the unfair subspace that is spanned by $\boldsymbol{f}$ as well as $\boldsymbol{P}_m$, let us define $\boldsymbol{N} \in St(d, m')$ to be the orthogonal matrix whose columns form a basis of $\operatorname{col}([\boldsymbol{P}_m \mid \boldsymbol{f}])$. Then, $\boldsymbol{N}$ has a closed form as follows: $m' = m$ if $\boldsymbol{f} \in \operatorname{col}(\boldsymbol{P}_m)$ and $m' = m+1$ otherwise, and

$$\boldsymbol{N} = \begin{cases} \boldsymbol{P}_m, & \text{if } \boldsymbol{f} \in \operatorname{col}(\boldsymbol{P}_m), \\ \mathtt{QR}([\boldsymbol{P}_m | \boldsymbol{f}]) = \left[\boldsymbol{P}_m \,\middle|\, \widetilde{\boldsymbol{f}}\right], & \text{otherwise, where } \widetilde{\boldsymbol{f}} = \frac{\boldsymbol{\Pi}_{\boldsymbol{P}_m} \boldsymbol{f}}{\|\boldsymbol{\Pi}_{\boldsymbol{P}_m} \boldsymbol{f}\|_2} \in \operatorname{col}(\boldsymbol{P}_m)^\perp. \end{cases} \tag{2}$$

Note that $\widetilde{\boldsymbol{f}}$ is a unit vector in a direction that $\boldsymbol{f}$ is projected onto $\operatorname{col}(\boldsymbol{P}_m)^\perp = \operatorname{null}(\boldsymbol{P}_m^\intercal)$. For this $\boldsymbol{N}$, our constraint in (1) can be interpreted as an equivalent nullity constraint $\boldsymbol{N}\boldsymbol{N}^\intercal \boldsymbol{V} = \boldsymbol{0}$:

$$\max_{\boldsymbol{V}^\intercal \boldsymbol{V} = \boldsymbol{I}_k} \operatorname{tr}(\boldsymbol{V}^\intercal \boldsymbol{\Sigma} \boldsymbol{V}), \quad \text{subject to } \boldsymbol{N}\boldsymbol{N}^\intercal \boldsymbol{V} = \boldsymbol{0}. \tag{3}$$

The above is equivalent to the following problem without any constraint other than orthgonality:

$$\max_{\boldsymbol{V}^\intercal \boldsymbol{V} = \boldsymbol{I}_k} \operatorname{tr}\left(\boldsymbol{V}^\intercal \boldsymbol{\Pi}_{\boldsymbol{N}} \boldsymbol{\Sigma} \boldsymbol{\Pi}_{\boldsymbol{N}} \boldsymbol{V}\right), \tag{4}$$

which is basically the vanilla k-PCA problem of a matrix $\boldsymbol{\Pi}_{\boldsymbol{N}} \boldsymbol{\Sigma} \boldsymbol{\Pi}_{\boldsymbol{N}} = (\boldsymbol{I} - \boldsymbol{N}\boldsymbol{N}^\intercal)\boldsymbol{\Sigma}(\boldsymbol{I} - \boldsymbol{N}\boldsymbol{N}^\intercal)$. Therefore, if $\boldsymbol{U}_k$ is the top-k orthonormal column basis of that matrix, $\boldsymbol{U}_k$ is indeed the closed-form solution of our problem (4).

3.3 Comparison to Covariance Matching Constraint

Previous works on fair PCA (Kleindessner et al., 2023; Olfat and Aswani, 2019) consider an exact covariance-matching constraint (namely, $\boldsymbol{V}^\intercal(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_0)\boldsymbol{V} = \boldsymbol{0}$). In fact, this is equivalent to the condition $\boldsymbol{V}^\intercal \boldsymbol{Q} \boldsymbol{V} = \boldsymbol{0}$ under the mean constraint $\boldsymbol{f}^\intercal \boldsymbol{V} = \boldsymbol{0}$, which can be derived as follows:

$$\boldsymbol{V}^\intercal(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_0)\boldsymbol{V} = \boldsymbol{V}^\intercal\left(\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal | s=1] - \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal | s=0]\right)\boldsymbol{V} - \boldsymbol{V}^\intercal\left(\boldsymbol{\mu}_1\boldsymbol{\mu}_1^\intercal - \boldsymbol{\mu}_0\boldsymbol{\mu}_0^\intercal\right)\boldsymbol{V}$$

$$= \boldsymbol{V}^\intercal \boldsymbol{Q} \boldsymbol{V} - \boldsymbol{V}^\intercal \left(\boldsymbol{f} \boldsymbol{\mu}_1^\intercal + \boldsymbol{\mu}_0 \boldsymbol{f}^\intercal \right) \boldsymbol{V} = \boldsymbol{V}^\intercal \boldsymbol{Q} \boldsymbol{V} = \boldsymbol{0}.$$

One immediate problem with this is that the constraint may be infeasible depending on the choice of $\boldsymbol{\Sigma}_0$, $\boldsymbol{\Sigma}_1$, or $\boldsymbol{Q}$; e.g., when $\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_0$ is positive definite. Although Kleindessner et al. (2023); Olfat and Aswani (2019) proposed some ways of relaxing the constraint, they had no discussion of the impact of the relaxation in a rigorous sense. On the contrary, our constraint is always feasible regardless of the choice of $\boldsymbol{Q}$, and due to that fact, we are able to consider a much more rigorous definition of fair PCA (Definition 4.2), which we present next.

4 Statistical Viewpoint: PAFO-Learnability of PCA

As all the distribution statistics $(\boldsymbol{\Sigma}, p, \cdots)$ are unknown, the learner, given some finite number of samples, must learn all of them *and* to solve fair PCA. In supervised learning, such a problem is often formalized in a PAC-learnability framework (Shalev-Schwartz and Ben-David, 2014). In the context of PAC-learnability for unsupervised learning settings, TV-learning, which is the task of learning distribution, has been mainly considered so far (Ananthakrishnan et al., 2021; Hopkins et al., 2023). However, unlike TV-learning, learning the whole distribution is unnecessary in fair PCA; moreover, fair PCA has the fairness constraint $\boldsymbol{N}\boldsymbol{N}^\intercal \boldsymbol{V} = \boldsymbol{0}$ to be satisfied. Inspired by the unsupervised PAC-learnability as well as constrained PAC-learnability (Chamon and Ribeiro, 2020), we propose a new notion of learnability for fair PCA, called *PAFO-learnability (probably approximately fair and optimal)*, as follows:

Definition 4.1 (Projection Learner)**.** *A* **projection learner** *is a function that takes $k \geq 1$ and d-dimensional samples as input and outputs a loading matrix $\boldsymbol{V} \in St(d, k)$.*

Definition 4.2 (PAFO-Learnability of PCA)**.** *Let d, k, m be integers such that $1 \leq k \leq d$ and $m < d$. We say that $\mathcal{F}_d \subset \mathcal{P}_d \times \mathcal{P}_d \times (0,1)$ is* **PAFO-learnable for PCA** *if there exists a function $N_{\mathcal{F}_d} : (0,1)^3 \rightarrow \mathbb{N}$ and a projection learner $\mathcal{A}$ satisfying the following: For every $(\varepsilon_1, \varepsilon_2, \delta) \in (0,1)^3$, and $(\mathcal{D}_0, \mathcal{D}_1, p) \in \mathcal{F}_d$, when running $\mathcal{A}$ on $N \geq N_{\mathcal{F}_d}(\varepsilon_1, \varepsilon_2, \delta)$ i.i.d. samples from $\mathcal{D} := p\mathcal{D}_1 + (1-p)\mathcal{D}_0$ of the form $(s, \boldsymbol{x})$, $\mathcal{A}$ returns $\boldsymbol{V}$ s.t., with probability at least $1 - \delta$ (over the draws of the N samples),*

$$\operatorname{tr}\left(\boldsymbol{V}^\intercal \boldsymbol{\Sigma} \boldsymbol{V}\right) \geq \operatorname{tr}\left(\boldsymbol{V}^{\star\intercal} \boldsymbol{\Sigma} \boldsymbol{V}^\star\right) - \varepsilon_1, \quad \|\boldsymbol{N}\boldsymbol{N}^\intercal \boldsymbol{V}\|_2 \leq \varepsilon_2,$$

where $\boldsymbol{N}$ is as defined in Eqn. (2) and $\boldsymbol{V}^\star$ is any solution to Eqn. (4) (with prescribed k and m).

Like in the usual PAC-learnability, $N_{\mathcal{F}_d}$ is referred to as the *sample complexity* of fair PCA. Observe how the optimality is measured w.r.t. the optimal solution of *fair* PCA, not the vanilla PCA.

5 Algorithmic Viewpoint: Fair Streaming PCA

Algorithm 1: `UnfairSubspace`

```
Input: Block size $b$, Number of iterations $U$;
Output: A orthogonal matrix $\widehat{\boldsymbol{N}}$;
$\boldsymbol{W}_0 = \mathtt{QR}(\mathcal{N}(0,1)^{d \times m})$;
$(\overline{\boldsymbol{m}}^{(0)}, \overline{\boldsymbol{m}}^{(1)}, B^{(0)}, B^{(1)}) = (\boldsymbol{0}_d, \boldsymbol{0}_d, 0, 0)$;
for $u \in [U]$ do
    Sample $b$ data points $\{(s_i, \boldsymbol{x}_i)\}_{i=1}^b$;
    foreach $s \in \{0, 1\}$ do
        Compute $b^{(s)}, \boldsymbol{m}^{(s)}, \boldsymbol{C}^{(s)}$ as Eqn. (5);
        $\overline{\boldsymbol{m}}^{(s)} \leftarrow \frac{B^{(s)}\overline{\boldsymbol{m}}^{(s)} + b^{(s)}\boldsymbol{m}^{(s)}}{B^{(s)} + b^{(s)}}$;
        $B^{(s)} \leftarrow B^{(s)} + b^{(s)}$;
    $\boldsymbol{W}_u \leftarrow \mathtt{QR}\left(\boldsymbol{C}^{(1)} - \boldsymbol{C}^{(0)}\right)$;
$\boldsymbol{g} \leftarrow \overline{\boldsymbol{m}}^{(1)} - \overline{\boldsymbol{m}}^{(0)}$;
$\boldsymbol{g} \leftarrow \boldsymbol{g} - \boldsymbol{W}_U \boldsymbol{W}_U^\intercal \boldsymbol{g}$;
if $\|\boldsymbol{g}\|_2$ is too close to 0 then $\widehat{\boldsymbol{N}} = \boldsymbol{W}_U$
  else $\widehat{\boldsymbol{N}} = \left[\boldsymbol{W}_U \,\middle|\, \widetilde{\boldsymbol{g}}\right]$, with $\widetilde{\boldsymbol{g}} = \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2}$ ;
return $\widehat{\boldsymbol{N}}$
```

Algorithm 2: Fair NPM

```
Input: Block sizes $B$, $b^{\mathtt{US}}$, Numbers of iterations $T$, $U^{\mathtt{US}}$;
Output: $\boldsymbol{V} \in St(d, k)$;
$\widehat{\boldsymbol{N}} \leftarrow \mathtt{UnfairSubspace}(b^{\mathtt{US}}, U^{\mathtt{US}})$;
$\boldsymbol{V}_0 \leftarrow \mathtt{QR}(\mathcal{N}(0,1)^{d \times k})$;
for $t \in [T]$ do
    $\boldsymbol{V}_t \leftarrow \boldsymbol{V}_{t-1} - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal \boldsymbol{V}$;
    $\boldsymbol{C} \leftarrow \boldsymbol{0}_{d \times k}$;
    for $i \in [B]$ do
        Receive $(*, \boldsymbol{x})$;
        $\boldsymbol{C} \leftarrow \boldsymbol{C} + \frac{1}{B}\boldsymbol{x}\boldsymbol{x}^\intercal \boldsymbol{V}_t$;
    $\boldsymbol{V}_t \leftarrow \boldsymbol{V}_t - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal \boldsymbol{V}_t$;
    $\boldsymbol{V}_t \leftarrow \mathtt{QR}\left(\boldsymbol{V}_t\right)$;
return $\boldsymbol{V}$
```

We now introduce a new problem setting, *fair streaming PCA*. In this setting, the learner receives a stream of pairs $(s_t, \boldsymbol{x}_t) \in \{0,1\} \times \mathbb{R}^d$ sequentially. Note that the sensitive attribute information s_t is also available at each time-step; this is commonly assumed when considering fairness in streaming setting (Bera et al., 2022; El Halabi et al., 2020). Precisely, we assume the following model of the data generation process: at each time-step t, a sensitive attribute is chosen as $s_t \sim \text{Bernoulli}(p)$, then the data is sampled from the corresponding sensitive group's conditional distribution $\boldsymbol{x}_t | s_t \sim \mathcal{D}_{s_t}$. Importantly, as done in previous streaming PCA literature (Mitliagkas et al., 2013), we assume that the learner has only $\mathcal{O}(dk)$ memory, where d is the data dimension and k is the target dimension. We can formally define the PAFO-learnability in this streaming setting:

Definition 5.1. *We say that $\mathcal{F}_d \subseteq \mathcal{P}_d \times \mathcal{P}_d \times (0,1)$ is* **PAFO-learnable for streaming PCA** *if the projection learner $\mathcal{A}$ for which Definition 4.2 holds uses only $\mathcal{O}(dk)$ memory for streaming data.*

5.1 Our Algorithm: Fair Noisy Power Method (FNPM)

One only needs to estimate $\boldsymbol{N}$ to use the off-the-shelf streaming PCA algorithm. As $\boldsymbol{N}$ is of size $d \times m$, storing its estimate is no problem for the memory constraint as long as $m = \mathcal{O}(k)$. Naturally, we proceed via a two-stage approach; first, estimate $\boldsymbol{N}$ sufficiently well, then with the fixed estimate of $\boldsymbol{N}$, apply the noisy power method (Hardt and Price, 2014; Mitliagkas et al., 2013) for $\boldsymbol{V}$.

For estimating $\boldsymbol{N}$, one needs to estimate $\boldsymbol{f}$ and $\boldsymbol{P}_m$. Estimating $\boldsymbol{f}$ can be done using the usual cumulative averaging. As for $\boldsymbol{P}_m$, we can consider the two main approaches for streaming PCA: Oja's method (Huang et al., 2021; Oja, 1982; Oja and Karhunen, 1985) and noisy power method (NPM) (Hardt and Price, 2014; Mitliagkas et al., 2013). We first show that Oja's method is *inapplicable* for our purpose, as it may ignore some eigenvectors corresponding to negative (but large in magnitude) eigenvalues of $\boldsymbol{Q}$. For instance, if $\boldsymbol{Q} = -2\boldsymbol{e}_1\boldsymbol{e}_1^\intercal + \boldsymbol{e}_2\boldsymbol{e}_2^\intercal + 4\boldsymbol{e}_3\boldsymbol{e}_3^\intercal$ with $\boldsymbol{e}_i$ being the standard basis vectors, then Oja's method with $m = 2$ would yield $[\boldsymbol{e}_2|\boldsymbol{e}_3]$ when we actually want $[\boldsymbol{e}_1|\boldsymbol{e}_3]$. For the same reason, simply shifting the eigenvalue spectrum by considering $\boldsymbol{Q} + \|\boldsymbol{Q}\|_2\boldsymbol{I}$ does not work. Thus we apply NPM for estimating $\boldsymbol{P}_m$ in our case, which is known to converge as long as the singular value gap of $\boldsymbol{P}_m$ is large enough and norms of the noise matrices at each iterate are properly bounded (Balcan et al., 2016; Hardt and Price, 2014).

Description of the algorithms. The pseudocode of our algorithm is shown in Algorithms 1 and 2. The goal of Algorithm 1 is to estimate $\boldsymbol{N} = [\boldsymbol{P}_m|\widetilde{\boldsymbol{f}}]$ as accurately as possible, as mentioned above. Lines 5–14 do the estimation of $\boldsymbol{P}_m$ and $\boldsymbol{f}$; line 11 is the noisy power method to find $\boldsymbol{P}_m$, lines 12–13 are the estimation of $\widetilde{\boldsymbol{f}}$, and line 14 is the concatenation of the estimates of $\boldsymbol{P}_m$ and $\widetilde{\boldsymbol{f}}$. With the estimated $\boldsymbol{N}$ from Algorithm 1, Algorithm 2 performs the usual noisy power method on $\boldsymbol{\Pi_N \Sigma \Pi_N}$, as in Eqn. (4). We note that the memory complexity of Algorithm 2 is $\mathcal{O}(d \max(m,k))$.

At time step u of Algorithm 1, for each $s \in \{0,1\}$, $b^{(s)}$ is the number of data points $\boldsymbol{x}_i$'s such that $s_i = s$, $\boldsymbol{m}^{(s)}$ is the term used for estimation of the group-wise sample mean of $\boldsymbol{x}_i$'s, and $\boldsymbol{C}^{(s)}$ is for the group-wise (mean-augmented) sample covariance. Their forms are as follows and can be computed incrementally in the streaming setup.

$$b^{(s)} = \sum_{i=1}^{b} \mathbb{1}[s_i = s], \quad \boldsymbol{m}^{(s)} = \frac{1}{b^{(s)}} \sum_{i=1}^{b} \mathbb{1}[s_i = s]\boldsymbol{x}_i, \quad \boldsymbol{C}^{(s)} = \frac{1}{b^{(s)}} \sum_{i=1}^{b} \mathbb{1}[s_i = s]\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \boldsymbol{W}_{u-1}, \tag{5}$$

where we set the last two quantities to **0** when $b^{(s)} = 0$.

Note that as $b^{(s)}$ itself is random, this presents some technical challenges in the proofs of the theoretical guarantees. For instance, the above estimators for the mean and covariance are biased. Still, by properly using peeling argument and matrix concentration inequalities as well as perturbation theories (Golub and Loan, 2013; Tropp, 2015), we could sufficiently bound their errors. Informally speaking, we show that line 11 corresponds to the noisy power method for the matrix $\boldsymbol{Q}$, which incurs the memory complexity $\mathcal{O}(dm)$.

5.2 Previous Approaches are not Suitable for Streaming Setup

All the existing approaches to fair PCA required the full knowledge of $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_0$, or even the full data matrix $\boldsymbol{V}$. Olfat and Aswani (2019) needed $\boldsymbol{f}$ and $\boldsymbol{Q}$ to formulate the convex matrix constraints for their semi-positive definite programming (SDP), which is then solved with commercial SDP solver; Lee et al. (2022) needed $\boldsymbol{V}$ to compute the derivative of their maximum mean discrepancy (MMD) penalty term, which by the way require $\mathcal{O}(d^2)$ to compute kernel Gram matrices. One may hope that the PCA-type approach taken by Kleindessner et al. (2023) may be easily extendable to our streaming setting by just using the standard techniques (e.g., using matrix-vector products) from streaming PCA (Mitliagkas et al., 2013). Indeed they also proposed a similar relaxation of the covariance constraint, leading to a closed-form solution. However, their formulation is not memory-efficient and, more importantly, is not as applicable to our streaming setting as our formulation; see Appendix C for more discussions on this.

6 FNPM is a PAFO-Learnable Algorithm

We now show that our proposed memory-efficient algorithm, FNPM, is actually a PAFO-learning algorithm in that with certain sample complexity, it satisfies the definition of PAFO-learnability. The proofs of all the theoretical results stated here are deferred to Appendix D.

To use proper matrix concentration inequalities for our error term analysis, various streaming PCA literature adapt some assumption on the underlying data distribution, e.g., sub-Gaussianity (Bienstock et al., 2022; Jain et al., 2016; Yang et al., 2018). Here, we adapt the following assumption to our data generation process in terms of the data point conditioned on the sensitive attribute:

Assumption 6.1. *Consider our data generation process $s \sim \mathrm{Bernoulli}(p)$ and $\boldsymbol{x}|s \sim \mathcal{D}_s$. Then, for some scalars $\sigma, V, \mathcal{M}, \mathcal{V} > 0$ and $\sigma_0, \sigma_1 \in (0, \sigma)$, the followings hold: for each $s' \in \{0, 1\}$,*

- $\boldsymbol{x}|s = s' \in \mathrm{nSG}(\sigma_{s'})$, $\mathbb{P}\left[\|\boldsymbol{x}\boldsymbol{x}^\intercal - (\boldsymbol{\Sigma}_{s'} + \boldsymbol{\mu}_{s'}\boldsymbol{\mu}_{s'}^\intercal)\|_2 \leq \mathcal{M} | s = s'\right] = 1$,
- $\|\boldsymbol{\Sigma}_{s'} + \boldsymbol{\mu}_{s'}\boldsymbol{\mu}_{s'}^\intercal\|_2 \leq V$ *and* $\mathrm{Var}\left(\boldsymbol{x}\boldsymbol{x}^\intercal | s = s'\right) \leq \mathcal{V}$,

where nSG *refers to norm-subGaussianity*[3]. *Moreover, there exist $f_{\min} \in (0, 1)$, $f_{\max} \in (f_{\min}, \infty)$, and $p_{\min} \in (0, 0.5]$ such that*

$$\|(\boldsymbol{I} - \boldsymbol{P}_m\boldsymbol{P}_m^\intercal)\boldsymbol{f}\|_2 \in \{0\} \cup [f_{\min}, f_{\max}], \quad \|\boldsymbol{f}\|_2 \in [0, f_{\max}], \quad \textit{and} \quad \{p_0, p_1\} \subset [p_{\min}, 1 - p_{\min}].$$

For the purpose of theoretical discussions, we consider a second assumption on the eigenspectrum of the underlying data distributions:

Assumption 6.2. *Fix $m, k \in \mathbb{N}$. There exist $\Delta_{m,\nu}, \Delta_{k,\kappa}, K_{m,\nu}, K_{k,\kappa} \in (0, \infty)$ such that for $(\mathcal{D}_0, \mathcal{D}_1) \in \mathcal{P}_d^2$, the following hold: $\nu_m - \nu_{m+1} > \Delta_{m,\nu}$, $\kappa_k - \kappa_{k+1} > \Delta_{k,\kappa}$, $\nu_m < K_{m,\nu}$, and $\kappa_k < K_{k,\kappa}$, where $\nu_1 \geq \cdots \geq \nu_d \geq 0$ and $\kappa_1 \geq \cdots \geq \kappa_d \geq 0$ are the singular values of $\boldsymbol{Q}$ and $\Pi_{\boldsymbol{N}}\boldsymbol{\Sigma}\Pi_{\boldsymbol{N}}$, respectively.*

We start by establishing the sample complexity bounds of Algorithms 1 and 2 based on the convergence bound for noisy power method (NPM) by Hardt and Price (2014). Recall that NPM is an algorithm for finding top-r eigenvectors (in magnitude) of a symmetric and *not necessarily psd* matrix $\boldsymbol{A}$ under a random noise $\boldsymbol{Z}$, by update $\boldsymbol{V}_{t+1} \leftarrow \mathtt{QR}(\boldsymbol{A}\boldsymbol{V}_t + \boldsymbol{Z}_t)$, where $\boldsymbol{V}_t \in St(d, r)$ is the iterate. We start by recalling their meta-sample complexity result for NPM, which we have slightly reformulated for our convenience:

Lemma 6.1 (Corollary 1.1 of Hardt and Price (2014))**.** *Let $1 \leq r < d$, $\epsilon \in (0, 1/2)$ and $\delta \in (0, 2e^{-cd})$, where c is an absolute constant*[4]*. Let $\boldsymbol{L}_r$ be the top-r eigenvectors (in magnitude) of the* **symmetric** *(not necessarily PSD) matrix $\boldsymbol{A}$ and denote its* **singular values** *by $\xi_1 \geq \cdots \geq \xi_d \geq 0$. Assume that the noise matrices $\boldsymbol{Z}_t \in \mathbb{R}^{d \times r}$ satisfy*

$$5\left\|\boldsymbol{Z}_t\right\|_2 \leq \epsilon(\xi_r - \xi_{r+1}) \quad \textit{and} \quad 5\left\|\boldsymbol{L}_r^\intercal\boldsymbol{Z}_t\right\|_2 \leq \frac{\delta(\xi_r - \xi_{r+1})}{2\sqrt{dr}}, \quad \forall t \geq 1. \tag{6}$$

Then, after $T = \Theta\left(\frac{\xi_r}{\xi_r - \xi_{r+1}} \log\left(\frac{d}{\epsilon\delta}\right)\right)$ steps of NPM, we have $\|(\boldsymbol{I} - \boldsymbol{V}_T\boldsymbol{V}_T^\intercal)\boldsymbol{L}_r\|_2 \leq \epsilon$ with probability at least $1 - \delta$.

[3] $\boldsymbol{y}$ is $\mathrm{nSG}(\sigma)$ if $\mathbb{P}[\|\boldsymbol{y} - \mathbb{E}\boldsymbol{y}\| \geq t] \leq 2e^{-\frac{t^2}{2\sigma^2}}$.

[4] It depends polynomially only in the sub-Gaussian moment of the data distribution $\mathcal{D}$; see Theorem 1.1 of Rudelson and Vershynin (2009).

First, we prove that the $\boldsymbol{W}_U$ resulting from Algorithm 1 (NPM for the covariance gap) converges to the true value. The noise matrix in this case is $\boldsymbol{Z}_{u,1} := (\boldsymbol{C}^{(1)} - \boldsymbol{C}^{(0)}) - \boldsymbol{Q}\boldsymbol{W}_{u-1}$, where $\boldsymbol{C}^{(s)}$ is as defined in Eqn. (5). The following theorem asserts that with large enough batch size b, the error matrices are sufficiently bounded such that the NPM iterates converge:

Theorem 6.1. *Let $\epsilon, \delta \in (0,1)$. It is sufficient to choose the block size b in Algorithm 1 as*

$$b = \Omega\left(\frac{\mathcal{V}}{\Delta_{m,\nu}^2 p_{\min}}\left(\frac{dm}{\delta^2}\log\frac{m}{\delta} + \frac{1}{\epsilon^2}\log\frac{d}{\delta}\right) + \frac{\mathcal{M}^2}{\mathcal{V}p_{\min}}\log\frac{d}{\delta}\right) \tag{7}$$

to make the following hold with probability at least $1 - \frac{\delta}{8}$:

$$5\left\|\boldsymbol{Z}_{u,1}\right\|_2 \leq \epsilon\Delta_{m,\nu} \quad \textit{and} \quad 5\left\|\boldsymbol{P}_m^\intercal \boldsymbol{Z}_{u,1}\right\|_2 \leq \frac{\delta\Delta_{m,\nu}}{2\sqrt{dm}}, \qquad \forall u \geq 1, \tag{8}$$

where we recall that $\boldsymbol{P}_m$ is the top-m (in magnitude) eigenvectors of $\boldsymbol{Q}$.

Let $\widehat{\boldsymbol{N}} = \boldsymbol{N}_U$ be the final estimate of the true $\boldsymbol{N}$ outputted by Algorithm 1. For Algorithm 2, the noise matrix is $\boldsymbol{Z}_{t,2} := \left(\Pi_{\widehat{\boldsymbol{N}}}\widehat{\boldsymbol{\Sigma}}_t\Pi_{\widehat{\boldsymbol{N}}} - \Pi_{\boldsymbol{N}}\boldsymbol{\Sigma}\Pi_{\boldsymbol{N}}\right)\boldsymbol{V}_t$, where $\widehat{\boldsymbol{\Sigma}}_t := \frac{1}{B}\sum_{j=(t-1)B+1}^{tB}\boldsymbol{x}_j\boldsymbol{x}_j^\intercal$ is the sample covariance at time step t of Algorithm 2. Similarly, with a large enough batch size B, we have the following theorem:

Theorem 6.2. *Let $\epsilon, \delta \in (0,1)$. Suppose that $\left\|(\boldsymbol{I} - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal)\boldsymbol{N}\right\|_2 \leq \frac{\Delta_{k,\kappa}}{20V}\min\left(\epsilon, \frac{\delta}{2\sqrt{dk}}\right)$. Then, it is sufficient to choose the block size B in Algorithm 2 as*

$$B = \Omega\left(\frac{\mathcal{V} + V^2}{\Delta_{k,\kappa}^2}\left(\frac{dk}{\delta^2}\log\frac{k}{\delta} + \frac{1}{\epsilon^2}\log\frac{d}{\delta}\right) + \frac{\mathcal{M}^2}{\mathcal{V} + V^2}\log\frac{d}{\delta}\right), \tag{9}$$

to make the following hold with probability at least $1 - \frac{\delta}{4}$:

$$5\left\|\boldsymbol{Z}_{t,2}\right\|_2 \leq \epsilon\Delta_{k,\kappa} \quad \textit{and} \quad 5\left\|\boldsymbol{U}_k^\intercal \boldsymbol{Z}_{t,2}\right\|_2 \leq \frac{\delta\Delta_{k,\kappa}}{2\sqrt{dk}}, \qquad \forall t \geq 1, \tag{10}$$

where we recall that $\boldsymbol{U}_k$ is the top-k eigenvectors of $\Pi_{\boldsymbol{N}}\boldsymbol{\Sigma}\Pi_{\boldsymbol{N}}$.

Combining the convergence results together, we can prove the following PAFO-learnability guarantee in the memory-limited, streaming setting:

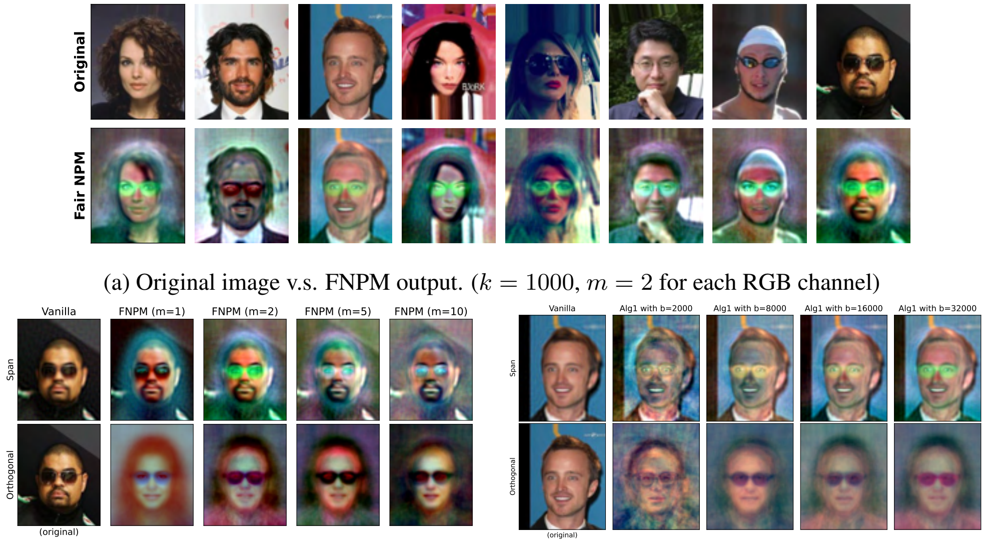
Theorem 6.3. *Let $d, m, k \in \mathbb{N}$ be fixed. Consider a collection $\mathcal{F}_d \subset \mathcal{P}_d \times \mathcal{P}_d \times (0,1)$ satisfying Assumptions 6.1 and 6.2. Then, $\mathcal{F}_d$ is PAFO-learnable for streaming PCA with our FNPM, where the* <u>*sufficient*</u> *number of samples is given as $N_{\mathcal{F}_d}(\varepsilon_1, \varepsilon_2, \delta) = N_1 + N_2$, with*

$$N_1 \gtrsim \frac{1}{p_{\min}}\left\{\frac{K_{m,\nu}\mathcal{V}}{\Delta_{m,\nu}^3}\alpha_f^2\frac{1}{1 + \mathbb{1}_f\frac{f_{\min}^2}{f_{\max}^2}}\frac{1}{\eta_k^2} + \frac{\sigma^2}{f_{\min}^2}\frac{1}{\eta_k^2}\mathbb{1}_f + \frac{K_{m,\nu}\mathcal{M}^2}{\Delta_{m,\nu}}\right\}\left(\log\frac{\alpha_f d}{\eta_k\delta}\right)^2, \tag{Algorithm 1}$$

$$N_2 \gtrsim \left(\frac{K_{k,\kappa}(\mathcal{V} + V^2)}{\Delta_{k,\kappa}^3}\left(\frac{dk}{\delta^2} + \frac{k^2V^2}{\varepsilon_1^2}\right) + \frac{K_{k,\kappa}\mathcal{M}^2}{\Delta_{k,\kappa}(\mathcal{V} + V^2)}\right)\left(\log\frac{dkV}{\varepsilon_1\delta}\right)^2, \tag{Algorithm 2}$$

where $\eta_k := \min\left(\varepsilon_2, \frac{\Delta_{k,\kappa}}{kV^2}\varepsilon_1, \frac{\Delta_{k,\kappa}}{V\sqrt{dk}}\delta\right)$, $\mathbb{1}_f := \mathbb{1}[\boldsymbol{f} \in \mathrm{col}(\boldsymbol{P}_m)]$, and $\alpha_f := \frac{1}{1+\mathbb{1}_f\frac{f_{\min}}{f_{\max}}}$.

Let us take a moment to digest the sample complexity. The second term N_2, which arises from Algorithm 2, is as one would expect from the usual noisy power method (Hardt and Price, 2014); there is no dependency on p, m nor $\boldsymbol{Q}$. The first term N_1, which arises from Algorithm 1, shows the "price" of pursuing fairness. Note that if $p_{\min} = 0$, i.e., if one of the two groups is never sampled, then the sample complexity is infinite, and the learnability does not hold; this aligns with our intuition, as we need a certain number of samples from *both* of the sensitive groups. Its dependency is also quite natural, as the minimum expected number of samples from either group depends linearly on $p_{\min}$. Also, noting the form of η_k, N_1 depends heavily on the hyperparameters and problem-dependent constants of Phase 2 (optimality), most notably, $\Delta_{k,\kappa}$, ε_1. This is because the estimation quality of $\boldsymbol{N}$ from Algorithm 1 directly impacts the optimality quality of Algorithm 2.



(a) Original image v.s. FNPM output. ($k = 1000$, $m = 2$ for each RGB channel)

(b) Ablation of m, intensity of fairness constraint.

(c) Ablation on the estimation of $\boldsymbol{N}$ (Alg. 1).

Figure 1: Experimental results on full-resolution **CelebA** dataset. In (b) and (c), the rows below visualize the orthogonal projection of images to estimated unfair subspace $\mathrm{col}(\boldsymbol{N})$.

Furthermore, observe that N_1 is divided into two cases, depending on whether the true mean difference $\boldsymbol{f} \in \mathrm{col}(\boldsymbol{P}_m)$ or not. When $\boldsymbol{f} \in \mathrm{col}(\boldsymbol{P}_m)$, the first term N_1 is burdened with additional dependencies on $f_{\min}, f_{\max}$, and σ. Precisely speaking, N_1 scales inversely with its norm squared ($f_{\min}^2$) and linearly with the subGaussianity constant (σ^2); this is because the QR-decomposition is done at the last step of Algorithm 1, which "amplifies" the sine error, proportional to $\|\boldsymbol{f}\|$, and larger the deviation, additional samples that one would need for the estimation of $\boldsymbol{f}$. However, regardless of $\boldsymbol{f}$, as long as the singular value gaps are strictly positive, FNPM is a PAFO-learning algorithm with a well-defined sample complexity.

7 Experiments

We next evaluate the efficacy of our proposed FNPM on the CelebA dataset (Liu et al., 2015b). It has been considered for Kleindessner et al. (2023) to show the superior efficiency of their fair PCA algorithm compared to previous approaches (Lee et al., 2022; Olfat and Aswani, 2019; Ravfogel et al., 2022a). However, even in Kleindessner et al. (2023), the images were resized and greyscaled, reducing the dimension from the original 218×178=38,804 to 80×80=6,400. Indeed, it was impossible to load all 162,770 original images in training set to the memory at once, while they require a full dataset to run each step of their algorithm. Thus, we use the *original* resolution, full-color CelebA dataset. We implement our FNPM using Python JAX NumPy Module (Bradbury et al., 2023; Harris et al., 2020) and Pytorch (Paszke et al., 2017). All experiments were performed on Apple 2020 Mac mini M1 with 16GB RAM.

Although CelebA is not streaming in nature, we intend to show that transforming it to one and using our memory-efficient approach allow us to *scale up* fair PCA. We consider the streaming setting of Since there are three channels of color, we run FNPM channel-wise but in *parallel* as usual in vision tasks (Priorov et al., 2013). For each channel of colors, we project the data onto a 1000-dimensional subspace while nullifying $m = 2$ leading eigenvectors of covariance difference.

The resulting images are displayed in Figure 1a. Here, we consider 'Eyeglasses' as a sensitive attribute to divide groups. We adopt the predefined train-validation split and run our algorithm only on the training set for 5 iterations with block sizes of $b = B = 32,000$. Then, using the output $\boldsymbol{V}$ of FNPM, we project images selected from the validation set. We observe that we have images of faces wearing colorful glasses by nullifying some of the leading eigenvectors of covariance difference. Especially for the images with sunglasses originally, their glasses get blurred, and "virtual" eyes are added to them. Not only these, but we also provide more results on the other attributes in Appendix E.

Varying m. Recall from our formulation that m is the hyperparameter that the learner controls on how much covariance fairness to impose; higher m means more fairness, and vice versa. In Figure 1b, we compare the projected images by varying m. The result shows that the more leading eigenvectors of $\boldsymbol{Q}$ get nullified, the more features related to sensitive attributes get erased. The images on the bottom row of Figure 1b are actually the orthogonally projected images onto the column space of estimated $\widehat{\boldsymbol{N}}$; more visual features in original images appear there as m increases.

Effect of estimation error of $\boldsymbol{N}$ on that of $\boldsymbol{V}$. As our statistical guarantee shows, one of the main factors that belong to the estimation error of $\boldsymbol{V}$ is the estimation error of $\boldsymbol{N}$ from Algorithm 1. To inspect this effect practically, we vary the block size $b \in \{2000, 8000, 16000, 32000\}$ used in Algorithm 1 while fixing the number of iterations U as 5; the results are displayed in Figure 1c. The inaccurate estimate of $\boldsymbol{N}$ due to the small block size b results in images of bad quality. However, if the $\boldsymbol{N}$ is estimated well enough, we get a clearer image with precisely shaped colored glasses.

8 Other Related Works

Fairness in ML. Research-wise, there are roughly two directions in algorithmic fairness. One direction is to propose a suitable and meaningful fairness definition (Dwork et al., 2012; Feldman et al., 2015; Hardt et al., 2016). The other direction is to develop *efficient* fair algorithms, although often the fairness constraint forces the algorithm to be much more inefficient than its unfair counterpart, or it calls for a need for a completely new algorithmic approach. There are also different ways of imposing fairness in an ML pipeline, such as learning fair pre-processing (Biswas and Rajan, 2021), fair in-processing (Roh et al., 2021; Wan et al., 2023; Zafar et al., 2019), and more. The reader is encouraged to check Barocas et al. (2019) for a more comprehensive treatment of this subject.

Fair online/streaming Learning. Bechavod et al. (2020); Gillen et al. (2018) have studied individual fairness in online learning in a learning theoretic framework, even when the underlying metric is unavailable. Stemming from the concept of fair clustering as proposed in Chierichetti et al. (2017), Bera et al. (2022); Schmidt et al. (2020) have studied imposing demographic parity on clustering in the streaming setting. Such fairness has been considered in various other streaming problems such as online selection (Correa et al., 2021), streaming submodular optimization (El Halabi et al., 2020), and diversity maximization (Wang et al., 2022). Quite surprisingly, demographic parity (or any other concept of fairness) has never been considered in the setting of streaming PCA.

Streaming PCA. Without the fairness constraint, streaming PCA has been studied much from statistics and the machine learning community. Two prominent algorithms have been studied; the noisy power method (Mitliagkas et al., 2013) and Oja's method (Oja, 1982). Much work has been done in improving the theoretical guarantees of streaming PCA (Balcan et al., 2016; Hardt and Price, 2014; Jain et al., 2016; Liang, 2023), improving the algorithm itself (Xu, 2023; Yun, 2018), or extending the guarantees to various different settings (Balzano et al., 2018; Bienstock et al., 2022; Kumar and Sarkar, 2023). Memory-limited, streaming versions of somewhat related problems, such as community detection (Yun et al., 2014) and low-rank matrix completion (Yun et al., 2015), have been tackled as well using similar spectral techniques as PCA (e.g., power method). However, to the best of our knowledge, fairness (regardless of the definition) has never been considered in this context of streaming PCA, which we tackle in this work and which we believe is of great importance.

9 Conclusion

In this work, we tackled the two outstanding problems of the existing fair PCA literature. From the theoretical side, we illustrated a new formulation of fair PCA based on the "Null It Out" approach and then provided a novel statistical framework called PAFO-learnability of Fair PCA. From the practical side, we addressed the memory-limited circumstances by proposing a new problem setting called fair streaming PCA and a memory-efficient two-stage algorithm called FNPM. Based on these, we established a statistical guarantee that our algorithm achieves the PAFO-learnability for fair streaming PCA. Lastly, we ran experiments on the CelebA dataset to certify the scalability of our method.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their helpful comments and suggestions. We also thank Gwangsu Kim (Jeonbuk National University) for the helpful discussions. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST); No. 2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics).

References

Hervé Abdi and Lynne J. Williams. Principal component analysis. *WIREs Computational Statistics*, 2(4):433–459, 2010.

Nivasini Ananthakrishnan, Shai Ben-David, Tosca Lechner, and Ruth Urner. Identifying regions of trusted predictions. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 2125–2134. PMLR, 27–30 Jul 2021.

Maria-Florina Balcan, Simon Shaolei Du, Yining Wang, and Adams Wei Yu. An Improved Gap-Dependency Analysis of the Noisy Power Method. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 284–309, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

Laura Balzano, Yuejie Chi, and Yue M. Lu. Streaming PCA and Subspace Tracking: The Missing Data Case. *Proceedings of the IEEE*, 106(8):1293–1310, 2018.

Solon Barocas and Andrew D. Selbst. Big Data's Disparate Impact. *104 California Law Review 671*, 2016.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019.

Yahav Bechavod, Christopher Jung, and Steven Z. Wu. Metric-Free Individual Fairness in Online Learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 11214–11225. Curran Associates, Inc., 2020.

Suman K. Bera, Syamantak Das, Sainyam Galhotra, and Sagar Sudhir Kale. Fair K-Center Clustering in MapReduce and Streaming Settings. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1414–1422, New York, NY, USA, 2022. Association for Computing Machinery.

Daniel Bienstock, Minchan Jeong, Apurv Shukla, and Se-Young Yun. Robust Streaming PCA. *Advances in Neural Information Processing Systems*, 35:4231–4243, 2022.

Sumon Biswas and Hridesh Rajan. Fair Preprocessing: Towards Understanding Compositional Fairness of Data Transformers in Machine Learning Pipeline. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2021, page 981–993, New York, NY, USA, 2021. Association for Computing Machinery.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2023. URL http://github.com/google/jax.

Luiz Chamon and Alejandro Ribeiro. Probably Approximately Correct Constrained Learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 16722–16735. Curran Associates, Inc., 2020.

Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair Clustering Through Fairlets. In *Advances in Neural Information Processing Systems*, volume 30, pages 5036–5044. Curran Associates, Inc., 2017.

Jose Correa, Andres Cristi, Paul Duetting, and Ashkan Norouzi-Fard. Fairness and Bias in Online Selection. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2112–2121. PMLR, 18–24 Jul 2021.

Chandler Davis and W. M. Kahan. Some new bounds on perturbation of subspaces. *Bulletin of the American Mathematical Society*, 75(4):863 – 868, 1969.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211—218, 1936.

Marwa El Halabi, Slobodan Mitrović, Ashkan Norouzi-Fard, Jakab Tardos, and Jakub M Tarnawski. Fairness in Streaming Submodular Maximization: Algorithms and Hardness. In *Advances in Neural Information Processing Systems*, volume 33, pages 13609–13622. Curran Associates, Inc., 2020.

Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, New York, NY, USA, 2015. Association for Computing Machinery.

Mina Ghashami, Daniel J. Perry, and Jeff Phillips. Streaming Kernel Principal Component Analysis. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1365–1374, Cadiz, Spain, 09–11 May 2016. PMLR.

Stephen Gillen, Christopher Jung, Michael Kearns, and Aaron Roth. Online Learning with an Unknown Fairness Metric. In *Advances in Neural Information Processing Systems*, volume 31, pages 2605–2614. Curran Associates, Inc., 2018.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, 4 edition, 2013.

Moritz Hardt and Eric Price. The Noisy Power Method: A Meta Algorithm with Applications. In *Advances in Neural Information Processing Systems*, volume 27, pages 2861–2869. Curran Associates, Inc., 2014.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 29, pages 3323–3331. Curran Associates, Inc., 2016.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

Max Hopkins, Daniel M. Kane, Shachar Lovett, and Gaurav Mahajan. Do PAC-Learners Learn the Marginal Distribution? *arXiv preprint arXiv:2302.06285*, 2023.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

De Huang, Jonathan Niles-Weed, and Rachel Ward. Streaming k-PCA: Efficient guarantees for Oja's algorithm, beyond rank-one updates. In *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 2463–2498. PMLR, 15–19 Aug 2021.

Prateek Jain, Chi Jin, Sham M. Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming PCA: Matching Matrix Bernstein and Near-Optimal Finite Sample Guarantees for Oja's Algorithm. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1147–1164, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan. A Short Note on Concentration Inequalities for Random Vectors with SubGaussian Norm. *arXiv preprint arXiv:1902.03736*, 2019.

Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis*. Pearson, 6 edition, 2008.

Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), 2016.

Mohammad Mahdi Kamani, Farzin Haddadpour, Rana Forsati, and Mehrdad Mahdavi. Efficient fair principal component analysis. *Machine Learning*, 111(10):3671–3702, 2022.

L Kirchner, J. Larson, S. Mattu, and J. Angwin. Machine bias. ProPublica, 2016.

René F. Kizilcec and Hansol Lee. Algorithmic Fairness in Education. *arXiv preprint arXiv:2007.05443*, 2021.

Matthäus Kleindessner, Michele Donini, Chris Russell, and Muhammad Bilal Zafar. Efficient fair PCA for fair representation learning. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5250–5270. PMLR, 25–27 Apr 2023.

Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled Cubic Regularization for Non-convex Optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1895–1904. PMLR, 06–11 Aug 2017.

Syamantak Kumar and Purnamrita Sarkar. Streaming PCA for Markovian Data. *arXiv preprint arXiv:2305.02456*, 2023.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. Fairness without Demographics through Adversarially Reweighted Learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 728–740. Curran Associates, Inc., 2020.

Junghyun Lee, Gwangsu Kim, Mahbod Olfat, Mark Hasegawa-Johnson, and Chang D. Yoo. Fast and Efficient MMD-Based Fair PCA via Optimization over Stiefel Manifold. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7363–7371, Jun. 2022.

Xun Li, Dongsheng Chen, Weipan Xu, Haohui Chen, Junjun Li, and Fan Mo. Explainable dimensionality reduction (XDR) to unbox AI 'black box' models: A study of AI perspectives on the ethnic styles of village dwellings. *Humanities and Social Sciences Communications*, 10(1):35, Jan 2023.

Xin Liang. On the optimality of the Oja's algorithm for online PCA. *Statistics and Computing*, 33(3): 62, Mar 2023.

Edo Liberty. Simple and Deterministic Matrix Sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 581–588, New York, NY, USA, 2013. Association for Computing Machinery.

Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. Fast Differentially Private Matrix Factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, page 171–178, New York, NY, USA, 2015a. Association for Computing Machinery.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015b.

Albert W. Marshall, Ingram Olkin, and Barry C. Arnold. *Inequalities: Theory of Majorization and Its Applications*. Springer Series in Statistics. Springer New York, 2 edition, 2011.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, 54(6), jul 2021.

L. Mirsky. A trace inequality of John von Neumann. *Monatshefte für Mathematik*, 79(4):303–306, Dec 1975.

Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory Limited, Streaming PCA. In *Advances in Neural Information Processing Systems*, volume 26, pages 2886–2894. Curran Associates, Inc., 2013.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, Nov 1982.

Erkki Oja and Juha Karhunen. On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix. *Journal of Mathematical Analysis and Applications*, 106: 69–84, 1985.

Matt Olfat and Anil Aswani. Convex Formulations for Fair Principal Component Analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 663–670, Jul. 2019.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS 2017 Workshop Autodiff*, 2017.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 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, October 2014. Association for Computational Linguistics.

Andrey Priorov, Kirill Tumanov, Vladimir Volokhov, Evgeny Sergeev, and Ivan Mochalov. Applications of image filtration based on principal component analysis and nonlocal image processing. *IAENG International Journal of Computer Science*, 40(2):62–80, 2013.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 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, July 2020. Association for Computational Linguistics.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. 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, 17–23 Jul 2022a.

Shauli Ravfogel, Francisco Vargas, Yoav Goldberg, and Ryan Cotterell. Adversarial Concept Erasure in Kernel Space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6034–6055, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics.

Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. FairBatch: Batch Selection for Model Fairness. In *International Conference on Learning Representations*, 2021.

Mark Rudelson and Roman Vershynin. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics*, 62(12):1707–1739, 2009.

Samira Samadi, Uthaipon Tantipongpipat, Jamie H Morgenstern, Mohit Singh, and Santosh Vempala. The Price of Fair PCA: One Extra dimension. In *Advances in Neural Information Processing Systems*, volume 31, pages 10999–11010. Curran Associates, Inc., 2018.

Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Fair coresets and streaming algorithms for fair k-means. In *Approximation and Online Algorithms*, pages 232–251, Cham, 2020. Springer International Publishing.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, 07 1998.

S. Shalev-Schwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Uthaipon Tantipongpipat, Samira Samadi, Mohit Singh, Jamie H Morgenstern, and Santosh Vempala. Multi-Criteria Dimensionality Reduction with Applications to Fairness. In *Advances in Neural Information Processing Systems*, volume 32, pages 15161–15171. Curran Associates, Inc., 2019.

Erico Tjoa and Cuntai Guan. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2021.

Joel A. Tropp. An Introduction to Matrix Concentration Inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Enayat Ullah, Poorya Mianjy, Teodor Vanislavov Marinov, and Raman Arora. Streaming Kernel PCA with $\tilde{O}(\sqrt{n})$ Random Features. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

John von Neumann. Some Matrix-Inequalities and Metrization of Matrix-Space. *Tomsk University Review*, 1:286–300, 1937.

Hieu Vu, Toan Tran, Man-Chung Yue, and Viet Anh Nguyen. Distributionally Robust Fair Principal Components via Geodesic Descents. In *International Conference on Learning Representations*, 2022.

Mingyang Wan, Daochen Zha, Ninghao Liu, and Na Zou. In-Processing Modeling Techniques for Machine Learning Fairness: A Survey. *ACM Transactions on Knowledge Discovery from Data*, 17 (3), mar 2023.

Chuang Wang and Yue M. Lu. Online learning for sparse PCA in high dimensions: Exact dynamics and phase transitions. In *2016 IEEE Information Theory Workshop (ITW)*, pages 186–190, 2016.

Ji Wang, Ding Lu, Ian Davidson, and Zhaojun Bai. Scalable Spectral Clustering with Group Fairness Constraints. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 6613–6629. PMLR, 25–27 Apr 2023.

Yanhao Wang, Francesco Fabbri, and Michael Mathioudakis. Streaming Algorithms for Diversity Maximization with Fairness Constraints. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 41–53, Los Alamitos, CA, USA, 2022. IEEE Computer Society.

Zhiqiang Xu. On the Accelerated Noise-Tolerant Power Method. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 7147–7175. PMLR, 25–27 Apr 2023.

Zhiqiang Xu and Ping Li. Faster Noisy Power Method. In *Proceedings of The 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pages 1138–1164. PMLR, 29 Mar–01 Apr 2022.

Puyudi Yang, Cho-Jui Hsieh, and Jane-Ling Wang. History PCA: A new algorithm for streaming PCA. *arXiv preprint arXiv:1802.05447*, 2018.

Wenzhuo Yang and Huan Xu. Streaming Sparse Principal Component Analysis. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 494–503, Lille, France, 07–09 Jul 2015. PMLR.

Se-Young Yun. Noisy Power Method with Grassmann Average. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 709–712, 2018.

Se-Young Yun, Marc Lelarge, and Alexandre Proutiere. Streaming, Memory Limited Algorithms for Community Detection. In *Advances in Neural Information Processing Systems*, volume 27, pages 3167–3175. Curran Associates, Inc., 2014.

Se-Young Yun, Marc Lelarge, and Alexandre Proutiere. Fast and Memory Optimal Low-Rank Matrix Approximation. In *Advances in Neural Information Processing Systems*, volume 28, pages 3166–3185. Curran Associates, Inc., 2015.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness Constraints: A Flexible Approach for Fair Classification . *Journal of Machine Learning Research*, 20(75):1–42, 2019.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning Fair Representations. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 325–333, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Tianxiang Zhao, Enyan Dai, Kai Shu, and Suhang Wang. Towards Fair Classifiers Without Sensitive Attributes: Exploring Biases in Related Features. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 1433–1442, New York, NY, USA, 2022. Association for Computing Machinery.

Contents

Appendix

A Broader Impacts, Limitations, and Future Directions

A.1 Broader Impacts

This work proposes a dimensionality reduction method that addresses memory efficiency *and* fairness while providing a statistical guarantee. By identifying and nullifying the “unfair direction” inherent in the data distribution, we offer an alternative approach to fair PCA. We anticipate that our approach will motivate researchers to explore other dimensionality reduction techniques (*e.g.,* auto-encoder) with fairness constraints. Significantly, our contribution includes a rigorous theoretical framework, PAFO learnability, which enables sample complexity analysis of fair PCA. We envision the potential for our theory to be further generalized to broader contexts, including alternative definitions of fairness and optimality of different algorithms for fair machine learning.

On the application side, the memory efficiency of our method can facilitate the scalability of fair PCA, making it viable for processing high-dimensional datasets, even in scenarios where data points arrive in a streaming fashion. One possible application of our approach is data pre-/post-processing to alleviate unfairness, such as generating fair word embeddings by eliminating sensitive attribute information through orthogonal projection. For more detailed discussions on potential future directions, please refer to below.

A.2 Limitations and Future Directions

Here, we list some of our work’s limitations and possible extensions/future directions.

Other PCA fairness notions. Our definition of fairness in PCA only covers group fairness with two demographic groups via fair presentation learning, which was also the case for previous works on fair PCA (Kleindessner et al., 2023; Lee et al., 2022; Olfat and Aswani, 2019). As we’ve noted in Section 2, the naïve way of extending this to multiple groups is via the one-vs.-all approach, which may be computationally inefficient. We also mention that none of the works have yet to consider the notion of individual fairness (Dwork et al., 2012) in the context of PCA, for which we do not have a definitive answer.

Knowledge of sensitive attribute. Our framework requires the *full* knowledge of the sensitive attribute s for all data points $\boldsymbol{x}$’s, which was also the case for all the previous works (Kleindessner et al., 2023; Lee et al., 2022; Olfat and Aswani, 2019). But, the assumption of such knowledge may not be feasible in the real world due to privacy or legal reasons (Lahoti et al., 2020; Zhao et al., 2022), or even just due to some extrinsic noises. Considering the case where the dataset may lack sensitive attributes for some or all of the data points is an important future direction.

Making the algorithm anytime. In our formulation of fair PCA, our algorithm is two-phase, with the first phase as a “burn-in” period for estimating the unfair subspace. Thus, it is not an anytime algorithm in that if the algorithm stops whilst in the first phase, then the resulting $\boldsymbol{V}$ is random and completely uninformative. Designing an anytime variant of our algorithm is an important future direction. One possible way to achieve that is to consider other streaming PCA algorithms such as Oja’s method (Oja and Karhunen, 1985) or accelerated NPM (Xu, 2023; Xu and Li, 2022).

Improving gap dependence. Our algorithm’s current sample complexity analysis relies on the analysis of NPM by Hardt and Price (2014), which relies on the immediate singular value gap, $\sigma_k - \sigma_{k+1}$. Balcan et al. (2016) showed that considering greater iteration rank $q \geq k$, *i.e.,* by considering optimization variable of greater size, leads to a better gap dependency: from $\sigma_k - \sigma_{k+1}$ to $\sigma_k - \sigma_{q+1}$. However, this is incompatible with our current definition of PAFO-learnability (Definition 4.2) because with greater iteration rank, the solution $\boldsymbol{V}^\star$ to which the explained variance should be compared against is not clear. The problem is that the sample complexity guarantee of Balcan et al. (2016); Hardt and Price (2014) is derived in terms of the *sine* angle between the top k-eigenspace of the true covariance and the estimated q-dimensional subspace. Clearing this up would allow for a better theoretical guarantee in sample complexity and thus an important future direction.

Kernelizing our framework Kernel PCA (Schölkopf et al., 1998) is a cornerstone in modern machine learning that has lent itself to be an inspiration to many applications, both theory and practice-wise. Unlike previous fair PCA works (Kleindessner et al., 2023; Olfat and Aswani, 2019), in which the authors provided a kernelized version of their fair PCA algorithm, both our statistical framework and memory-efficient algorithm for streaming setting do not readily extend to the kernelized version. Algorithmically, to tackle the streaming setting, one may take inspiration from streaming kernel PCA (Ghashami et al., 2016; Ullah et al., 2018), which was in turn inspired by matrix sketching (Liberty, 2013).

Extending to other settings. In a similar spirit, extending our formulation of fair PCA to other streaming settings such as sparse (Wang and Lu, 2016; Yang and Xu, 2015), nonstationary (Bienstock et al., 2022), or even distributionally robust settings (Vu et al., 2022) would also be interesting.

More experiments Last but not least, we've only performed experiments on synthetic, CelebA, and the UCI datasets. It would be interesting to try our framework (either the statistical formulation, the streaming setting, or both) on datasets from other domains, such as NLP and graphs. GloVe vectors (Pennington et al., 2014) has been used as a benchmark in a similar task called "concept erasure" (Ravfogel et al., 2020, 2022b); group fairness in spectral clustering of graphs has been studied as well (Kleindessner et al., 2023; Wang et al., 2023).

B Full Pseudo-code of Algorithm 1

Algorithm 3: `UnfairSubspace`

```
Input: Block size $b$, Number of iterations $U$;
Output: A matrix $\widehat{\boldsymbol{N}}$ with orthonormal columns;
$\boldsymbol{W}_0 = \mathtt{QR}(\mathcal{N}(0,1)^{d\times m})$;
$(\overline{\boldsymbol{m}}^{(0)}, \overline{\boldsymbol{m}}^{(1)}, B^{(0)}, B^{(1)}) = (\mathbf{0}_d, \mathbf{0}_d, 0, 0)$;
for $u \in [U]$ do
    $(\boldsymbol{m}^{(0)}, \boldsymbol{m}^{(1)}, \boldsymbol{C}^{(0)}, \boldsymbol{C}^{(1)}) = (\mathbf{0}_d, \mathbf{0}_d, \mathbf{0}_{d\times m}, \mathbf{0}_{d\times m})$;
    $(b^{(0)}, b^{(1)}) = (0, 0)$;
    for $j \in [b]$ do
        Receive $(s, \boldsymbol{x})$;    $b^{(s)} \leftarrow b^{(s)} + 1$;
        $\boldsymbol{m}^{(s)} \leftarrow \boldsymbol{m}^{(s)} + \frac{1}{b}\boldsymbol{x}$;
        $\boldsymbol{C}^{(s)} \leftarrow \boldsymbol{C}^{(s)} + \frac{1}{b}\boldsymbol{x}\boldsymbol{x}^\intercal \boldsymbol{W}_{u-1}$;
    $\boldsymbol{W}_u \leftarrow \mathtt{QR}\left(b^{(0)}\boldsymbol{C}^{(1)} - b^{(1)}\boldsymbol{C}^{(0)}\right)$;
    foreach $s \in \{0, 1\}$ do
        $\overline{\boldsymbol{m}}^{(s)} \leftarrow \frac{B^{(s)}\overline{\boldsymbol{m}}^{(s)} + b^{(s)}\boldsymbol{m}^{(s)}}{B^{(s)} + b^{(s)}}$;
        $B^{(s)} \leftarrow B^{(s)} + b^{(s)}$;
$\boldsymbol{g} \leftarrow \frac{1}{Ub}(B^{(0)}\overline{\boldsymbol{m}}^{(1)} - B^{(1)}\overline{\boldsymbol{m}}^{(0)})$;
$\boldsymbol{g} \leftarrow \boldsymbol{g} - \boldsymbol{W}_U \boldsymbol{W}_U^\intercal \boldsymbol{g}$;
if $\|\boldsymbol{g}\|_2$ is too close to 0 then $\widehat{\boldsymbol{N}} = \boldsymbol{W}_U$ else $\widehat{\boldsymbol{N}} = \left[\boldsymbol{W}_U \,\middle|\, \widetilde{\boldsymbol{g}}\right]$, with $\widetilde{\boldsymbol{g}} = \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2}$ ;
return $\widehat{\boldsymbol{N}}$
```

C More Detailed Comparison to Kleindessner et al. (2023)

C.1 Their Approach

Kleindessner et al. (2023) considered the following formulation of fair PCA:

$$\max_{\boldsymbol{V} \in St(d,k)} \operatorname{tr}(\boldsymbol{V}^\intercal \boldsymbol{X}^\intercal \boldsymbol{X} \boldsymbol{V}), \quad \text{subject to } \boldsymbol{f}^\intercal \boldsymbol{V} = \boldsymbol{0} \ \wedge \ \boldsymbol{V}^\intercal (\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_0) \boldsymbol{V} = \boldsymbol{0}, \tag{11}$$

and proposed a reasonable approximation of the covariance constraint, which we briefly describe here. Let us write $\boldsymbol{Q}' = \boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_0$ for brevity. As implicitly assumed in Kleindessner et al. (2023), let us assume that $\boldsymbol{f} \neq \boldsymbol{0}$. The mean constraint is dealt with first. Denoting $\boldsymbol{N_f} \in St(d, d-1)$ to be the matrix whose columns form a basis $(d-1)$-dimensional nullspace of $\boldsymbol{f}$, the mean constraint is satisfied if and only if $\boldsymbol{V}$ is of the form $\boldsymbol{N_f U}$ with $U \in St(d-1, k)$ as the intermediate optimization variable. With this first reparametrization, the optimization now becomes

$$\max_{\boldsymbol{U} \in St(d-1,k)} \operatorname{tr}(\boldsymbol{U}^\intercal \boldsymbol{N_f}^\intercal \boldsymbol{X}^\intercal \boldsymbol{X} \boldsymbol{N_f} \boldsymbol{U}), \quad \text{subject to } \boldsymbol{U}^\intercal \boldsymbol{N_f}^\intercal \boldsymbol{Q}' \boldsymbol{N_f} \boldsymbol{U} = \boldsymbol{0}, \tag{12}$$

which now has only the covariance constraint.

To deal with the possible infeasibility of the covariance constraint, Kleindessner et al. (2023) proposed the following approach: letting $\boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'} \in St(d-1, l)$ be the matrix whose columns are the l smallest eigenvectors of $\boldsymbol{N_f}^\intercal \boldsymbol{Q}' \boldsymbol{N_f}$ (in magnitude), $\boldsymbol{U}$ only needs to nullify the eigenspace spanned by the remaining $d-1-l$ eigenvectors. To see which terms are ignored with the relaxation, let $\sum_{i=1}^d q_i' \boldsymbol{v}' \boldsymbol{v}_i'^\intercal$ be the eigenvalue decomposition of $\boldsymbol{N_f}^\intercal \boldsymbol{Q}' \boldsymbol{N_f}$ with $|q_1'| \geq |q_2'| \geq \cdots |q_d'| \geq 0$. This approach essentially ignores the constraint $\boldsymbol{U}^\intercal \boldsymbol{E}_l \boldsymbol{U} = \boldsymbol{0}$, where

$$\boldsymbol{E}_l = \boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}^\intercal \boldsymbol{N_f}^\intercal \boldsymbol{Q}' \boldsymbol{N_f} \boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'} = \sum_{i=d-l}^{d-1} q_i' \boldsymbol{v}' \boldsymbol{v}_i'^\intercal. \tag{13}$$

The number $l \in \{k, \cdots, d-1\}$ controls how much the group-conditional covariances will be equalized; a smaller l means that the covariance constraint is enforced more stringently and vice versa. Ultimately, the learner can control l as a hyperparameter to create a trade-off between fairness and the explained variance. Moreover, *if* $|q_i'|$'s are negligible for all $i \geq d-1-l$, then $\boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}^\intercal \boldsymbol{N_f}^\intercal \boldsymbol{Q}' \boldsymbol{N_f} \boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'} \approx \boldsymbol{0}$, and the relaxation becomes tighter.

As any $\boldsymbol{U}$ of the form $\boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'} \boldsymbol{\Lambda}$ satisfies the relaxed covariance constraint, a second reparameterization in terms of the new optimization variable $\boldsymbol{\Lambda} \in St(l, k)$ gives us

$$\max_{\boldsymbol{\Lambda} \in St(l,k)} \operatorname{tr}(\boldsymbol{\Lambda}^\intercal \boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}^\intercal \boldsymbol{N_f}^\intercal \boldsymbol{X}^\intercal \boldsymbol{X} \boldsymbol{N_f} \boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'} \boldsymbol{\Lambda}), \tag{14}$$

which can be solved via the standard SVD-based approach for vanilla PCA. Then, the final solution is obtained as $\boldsymbol{V}^* = \boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'} \boldsymbol{N_f} \boldsymbol{\Lambda}^*$.

C.2 Unsuitability for the Streaming Setting

In Section 5.2, we provided a rough overview of why existing approaches to fair PCA (Kleindessner et al., 2023; Lee et al., 2022; Olfat and Aswani, 2019) are not amenable to our streaming setting. Especially as the approach of Kleindessner et al. (2023) (described above) is almost like a PCA, one may wonder if standard techniques used in streaming PCA (Mitliagkas et al., 2013) can be used. Here, we argue in detail why that is *not* the case, which is in sharp contrast to our approach and our reformulation of fair PCA that led to a memory-efficient fair streaming PCA algorithm.

Memory constraint. For streaming PCA without fairness constraints, the main objective could be written as $\operatorname{tr}(\boldsymbol{V}^\intercal \boldsymbol{X}^\intercal \boldsymbol{X} \boldsymbol{V}) = \sum_{i=1}^N \operatorname{tr}\left(\boldsymbol{V}^\intercal \boldsymbol{x}_i \boldsymbol{x}_i^\intercal \boldsymbol{V}\right)$, which is easily amenable to *memory-limited* algorithm such as noisy power method (Mitliagkas et al., 2013) or stochastic optimization (Oja, 1982). Both approaches utilize the fact that instead of storing $d \times d$ matrices, one only needs to store matrix-vector product of size $d \times 1$, e.g., $\boldsymbol{V}^\intercal \boldsymbol{x}_i$. This is not the case for the approach of Kleindessner et al. (2023). To see this, consider Eqn. (12) without the covariance constraint, i.e., fair PCA with only the mean constraint. Even here, although the objective can be written as $\sum_{i=1}^N \operatorname{tr}\left(\boldsymbol{U}^\intercal \boldsymbol{N_f}^\intercal \boldsymbol{x}_i \boldsymbol{x}_i^\intercal \boldsymbol{N_f} \boldsymbol{U}\right)$, we must know $\boldsymbol{N_f}$ in order to proceed further. Moreover, as $\boldsymbol{N_f}$ is of size $\mathcal{O}(d^2)$, it cannot be stored nor estimated explicitly. When the covariance constraint is also taken into account, although the matrix in question $\boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}$ is of dimension $(d-1) \times l$, which is within the memory constraint if $l = \mathcal{O}(1)$, the computation of $\boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}$ *requires* the knowledge of $\boldsymbol{N_f}$. This is because Kleindessner et al. (2023) dealt with the two constraints sequentially (mean first, then covariance), which forced the reparametrization to be done twice and, more importantly, coupling the memory requirement for the computation of $\boldsymbol{N_f}$ and $\boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}$.

Statistical consideration. The statistical guarantee (global convergence, sample complexity) of streaming PCA algorithms is often obtained by appropriately bounding the error term and using proper matrix concentration inequalities (Tropp, 2015). For example, for the sample complexity guarantee of noisy power method (Balcan et al., 2016; Hardt and Price, 2014), as the learner performs power method on the empirical covariance $\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^\intercal$ instead of the true covariance $\boldsymbol{\Sigma}$, the proof proceeds by first showing that the empirical covariance is close enough to the true covariance (i.e., the norm of their error term is sufficiently bounded), which then implies that the variance of the estimation isn't too high. Thus to analyze the error bound of the final iterates $\boldsymbol{V}$ when the approach of Kleindessner et al. (2023) is extended to a streaming setting, one would have to bound the estimation error of $\boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}^\intercal \boldsymbol{N_f}^\intercal \boldsymbol{\Sigma} \boldsymbol{N_f} \boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}$. There are three sources of estimation error: $\boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}, \boldsymbol{N_f}^\intercal, \boldsymbol{\Sigma}$. Recalling that $\boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}$ consists of the *eigenvectors* of $\boldsymbol{N_f}^\intercal \boldsymbol{Q}' \boldsymbol{N_f}$, one can see that the estimation error of $\boldsymbol{M}_{\boldsymbol{f},\boldsymbol{Q}'}$ is actually nontrivially dependent on the estimation quality of *both* $\boldsymbol{N}$ and $\boldsymbol{Q}'$. Here, we say nontrivially because the error isn't simply bounded in a linear sense via the usual triangle inequality; it requires rather intricate techniques involving eigenvector perturbation theory (Davis and Kahan, 1969; Golub and Loan, 2013), which may require additional assumptions on eigenvalue gaps. As one can see later in the proof, our approach considers both constraints simultaneously and thus allows for a quite simple theoretical analysis.

D Proofs of Theorem 6.1, 6.2, and 6.3

D.1 Notations and Assumptions

We recall some notation needed for the proof. The (mean-augmented) covariance difference is $\boldsymbol{Q} = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal | s = 1] - E[\boldsymbol{x}\boldsymbol{x}^\intercal | s = 0] = \boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_0 + \boldsymbol{\mu}_1\boldsymbol{\mu}_1^\intercal - \boldsymbol{\mu}_0\boldsymbol{\mu}_0^\intercal$; let $\boldsymbol{P}_m$ be the matrix whose columns are top m eigenvectors of $\boldsymbol{Q}$ in magnitude of eigenvalues. Every stream data is sampled as $s \sim \mathrm{Bernoulli}(p)$ and $\boldsymbol{x}|s \sim \mathcal{D}_s$, where $\mathcal{D}_s$ has mean $\boldsymbol{\mu}_s$ and covariance $\boldsymbol{\Sigma}_s$, or written more compactly as $\boldsymbol{x} \sim \mathcal{D} := p_0\mathcal{D}_0 + p_1\mathcal{D}_1$. We denote $p_0 := 1 - p$ and $p_1 := p$. Here we list the quantities that are mainly used in our theoretical analysis:

- Counting data points for each sensitive attribute s:
  - For each s, $b_u^{(s)} = \sum_{i=(u-1)b+1}^{ub} \mathbb{1}[s_i = s]$; $\quad B^{(s)} = \sum_{i=1}^{Ub} \mathbb{1}[s_i = s]$
- Estimates of group-conditional means:
  - $\overline{\boldsymbol{m}}_s = \frac{1}{UB^{(s)}} \sum_{i=1}^{Ub} \mathbb{1}[s_i = s]\boldsymbol{x}_i$ if $B^{(s)} > 0$, $\boldsymbol{0}$ otherwise
- Estimate of the group-conditional mean gap:
  - $\boldsymbol{g} = \overline{\boldsymbol{m}}_1 - \overline{\boldsymbol{m}}_0$
- Estimate of the group-conditional (mean-augmented) covariances:
  - $\boldsymbol{A}_u^{(s)} = \frac{1}{b_u^{(s)}} \sum_{i=(u-1)b+1}^{ub} \mathbb{1}[s_i = s]\boldsymbol{x}_i\boldsymbol{x}_i^\intercal$ if $b_u^{(s)} > 0$, $\boldsymbol{0}$ otherwise
- Estimate of the group-conditional covariance gap:
  - $\boldsymbol{G}_u = \boldsymbol{A}_u^{(1)} - \boldsymbol{A}_u^{(0)}$
- $\boldsymbol{W}_u \in St(d, n)$ (optimization variable)
- $\boldsymbol{C}_u^{(s)} = \boldsymbol{A}_u^{(s)}\boldsymbol{W}_{u-1}$
- $\boldsymbol{F}_u = \boldsymbol{C}_u^{(1)} - \boldsymbol{C}_u^{(0)} = \boldsymbol{G}_u\boldsymbol{W}_{u-1}$
- $\boldsymbol{F}_u \overset{\mathtt{QR}}{=} \boldsymbol{W}_u\boldsymbol{R}_u \quad (\boldsymbol{R}_u \in \mathbb{R}^{n\times n})$

If the context is clear, we often omit the time variable "u."

Recall the assumptions on the data distribution, which we will assume throughout the proof:

Assumption 6.1. *Consider our data generation process $s \sim \mathrm{Bernoulli}(p)$ and $\boldsymbol{x}|s \sim \mathcal{D}_s$. Then, for some scalars $\sigma, V, \mathcal{M}, \mathcal{V} > 0$ and $\sigma_0, \sigma_1 \in (0, \sigma)$, the followings hold: for each $s' \in \{0, 1\}$,*

- $\boldsymbol{x}|s = s' \in \mathrm{nSG}(\sigma_{s'})$, $\mathbb{P}\left[\|\boldsymbol{x}\boldsymbol{x}^\intercal - (\boldsymbol{\Sigma}_{s'} + \boldsymbol{\mu}_{s'}\boldsymbol{\mu}_{s'}^\intercal)\|_2 \le \mathcal{M} | s = s'\right] = 1$,
- $\|\boldsymbol{\Sigma}_{s'} + \boldsymbol{\mu}_{s'}\boldsymbol{\mu}_{s'}^\intercal\|_2 \le V$ *and* $\mathrm{Var}\left(\boldsymbol{x}\boldsymbol{x}^\intercal | s = s'\right) \le \mathcal{V}$,

where nSG *refers to norm-subGaussianity*[5]. *Moreover, there exist $f_{\min} \in (0, 1)$, $f_{\max} \in (f_{\min}, \infty)$, and $p_{\min} \in (0, 0.5]$ such that*

$$\|(\boldsymbol{I} - \boldsymbol{P}_m\boldsymbol{P}_m^\intercal)\boldsymbol{f}\|_2 \in \{0\} \cup [f_{\min}, f_{\max}], \;\; \|\boldsymbol{f}\|_2 \in [0, f_{\max}], \;\; \textit{and} \;\; \{p_0, p_1\} \subset [p_{\min}, 1 - p_{\min}].$$

Assumption 6.2. *Fix $m, k \in \mathbb{N}$. There exist $\Delta_{m,\nu}, \Delta_{k,\kappa}, K_{m,\nu}, K_{k,\kappa} \in (0, \infty)$ such that for $(\mathcal{D}_0, \mathcal{D}_1) \in \mathcal{P}_d^2$, the following hold: $\nu_m - \nu_{m+1} > \Delta_{m,\nu}$, $\kappa_k - \kappa_{k+1} > \Delta_{k,\kappa}$, $\nu_m < K_{m,\nu}$, and $\kappa_k < K_{k,\kappa}$, where $\nu_1 \ge \cdots \ge \nu_d \ge 0$ and $\kappa_1 \ge \cdots \ge \kappa_d \ge 0$ are the singular values of $\boldsymbol{Q}$ and $\Pi_{\boldsymbol{N}}\boldsymbol{\Sigma}\Pi_{\boldsymbol{N}}$, respectively.*

We provide some intuitions on the assumptions that we impose here. Assumption 6.1 consists of three parts. The first part, which involves $M, \mathcal{M}, V$ and $\mathcal{V}$, ensures that the maximum deviation in mean and covariance of each $\mathcal{D}_s$ are well bounded; this is critical in allowing for us to use proper matrix concentration inequalities (to be described in the next subsection) and has been used in various streaming PCA literature (Bienstock et al., 2022; Huang et al., 2021; Jain et al., 2016). The second part, which involves f_{min} and f_{max}, imposes a bound on the maximum mean separation,

[5] $\boldsymbol{y}$ is nSG(σ) if $\mathbb{P}[\|\boldsymbol{y} - \mathbb{E}\boldsymbol{y}\| \ge t] \le 2e^{-\frac{t^2}{2\sigma^2}}$.

$\boldsymbol{f} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$, ℓ_2-wise and angle-wise, respectively. If the mean difference can be arbitrarily large, then the ℓ_2-estimation error that one has to achieve becomes arbitrarily small; precisely speaking, $\|\boldsymbol{f}\|_2$ acts as a Lipschitz constant. On the other hand, if the mean difference can be arbitrarily small, then the angle-wise estimation error becomes arbitrarily large. The last part, which involves p_{min}, ensures that both groups are selected with some positive, nonvanishing probability. Assumption 6.2 is standard in streaming PCA literature to ensure convergence; indeed, if the singular value gap is zero, a definitive convergence result can never be obtained, as the ground-truth solution becomes vague.

D.2 Matrix/Vector Concentration Inequalities

Before moving on to our proof, we review some useful concentration inequalities for our theoretical analysis.

Definition D.1 (Variance of random matrix)**.** *For a zero-mean random matrix $\boldsymbol{Z}$, its variance is defined as*

$$\mathrm{Var}(\boldsymbol{Z}) = \max\left(\left\|\mathbb{E}\left[\boldsymbol{Z}\boldsymbol{Z}^\intercal\right]\right\|_2, \left\|\mathbb{E}\left[\boldsymbol{Z}^\intercal\boldsymbol{Z}\right]\right\|_2\right).$$

Proposition D.1 (Matrix Bernstein inequality (Theorem 6.6.1 of Tropp (2015)))**.** *Consider a finite collection $\{\boldsymbol{Y}_j\}_{j=1}^b$ of independent matrices with the same size ($d_1 \times d_2$). Suppose they are zero mean and they have uniformly bounded singular values, i.e.,*

$$\mathbb{E}[\boldsymbol{Y}_j] = \boldsymbol{0} \quad \textit{and} \quad \|\boldsymbol{Y}_j\| \leq \mathcal{M} \quad \textit{for each } j \in [b].$$

Let $\mathcal{V}$ be an upper bound of matrix variance, $\mathcal{V} \geq \mathrm{Var}\left(\boldsymbol{Y}_j\right)$ for all $j \in [b]$. Then, for all $x \geq 0$,

$$\mathbb{P}\left(\left\|\frac{1}{b}\sum_{j=1}^{b}\boldsymbol{Y}_j\right\| \geq x\right) \leq (d_1 + d_2)\exp\left(\frac{-bx^2}{2(\mathcal{V} + \mathcal{M}x/3)}\right).$$

In particular, if $0 \leq x \leq \frac{3\mathcal{V}}{\mathcal{M}}$,

$$\mathbb{P}\left(\left\|\frac{1}{b}\sum_{j=1}^{b}\boldsymbol{Y}_j\right\| \geq x\right) \leq (d_1 + d_2)\exp\left(\frac{-bx^2}{4\mathcal{V}}\right).$$

Proposition D.2 (Vector Hoeffding inequality (Corollary 7 of Jin et al. (2019)))**.** *There exists an absolute constant $\mathfrak{c}$ such that if $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_b$ be independent random vectors with common dimension d, and assume the following:*

$$\mathbb{E}[\boldsymbol{y}_i] = 0, \quad \boldsymbol{y}_i \in \mathrm{nSG}(\sigma).$$

Then, for any $x \geq 0$,

$$\mathbb{P}\left(\left\|\frac{1}{b}\sum_{j=1}^{b}\boldsymbol{y}_j\right\|_2 \geq x\right) \leq 2d\exp\left(-\frac{bx^2}{\mathfrak{c}^2\sigma^2}\right).$$

Remark D.1. *With an additional assumption that the random variables are bounded, we can also consider using vector Bernstein inequality (Lemma 18 of Kohler and Lucchi (2017)), which removes the factor of dimension d from the RHS of the concentration inequality.*

D.3 Proof of Theorem 6.1 - Bounding Error in Covariance Gap

Theorem D.1. *For any $\delta > 0$, choose $b \geq 4\left(\frac{1}{p_1} + \frac{1}{p_0}\right)\max\left(1, \frac{8\mathcal{M}^2}{9\mathcal{V}}\right)\log\frac{16(d+m)}{\delta}$. Then, given $\boldsymbol{W}_{u-1}$, the following holds with probability at least $1 - \frac{\delta}{8}$:*

$$\left\|\boldsymbol{F}_u - \boldsymbol{Q}\boldsymbol{W}_{u-1}\right\|_2 \leq \mathcal{E}_{d+m}^{(\mathrm{Q})},$$
$$\left\|\boldsymbol{P}_m^\intercal\left(\boldsymbol{F}_u - \boldsymbol{Q}\boldsymbol{W}_{u-1}\right)\right\|_2 \leq \mathcal{E}_{2m}^{(\mathrm{Q})},$$

where

$$\mathcal{E}_y^{(\mathrm{Q})} \triangleq \sqrt{\frac{32\mathcal{V}}{b}\log\frac{16y}{\delta}}\left(\frac{1}{\sqrt{p_1}} + \frac{1}{\sqrt{p_0}}\right).$$

Proof. For brevity, fix the time index u and omit it whenever it is clear, and fix a $\boldsymbol{W} = \boldsymbol{W}_{u-1} \in St(d,m)$. Note that $(s_i, \boldsymbol{x}_i)$'s are i.i.d. samples. Consider independent random matrices

$$\boldsymbol{Y}_i^{(s)} = \left(\boldsymbol{x}_i \boldsymbol{x}_i^\intercal - (\boldsymbol{\Sigma}_s + \boldsymbol{\mu}_s \boldsymbol{\mu}_s^\intercal)\right) \boldsymbol{W} \quad (i \in \{u(b-1)+1, \dots, ub\}).$$

For each $S \subseteq [b]$, define an event $E_S := \{s_i = \mathbb{1}[i \in S] \ \forall i \in [b]\}$. Note that $\mathbb{P}[E_S] = p_1^{|S|} p_0^{b-|S|}$. To exploit this, we first apply a peeling argument as follows: for any $c_0, c_1 \in (0,1)$ with $c_0 + c_1 = 1$,

$$\begin{aligned}
&\mathbb{P}\left(\|\boldsymbol{F}_u - \boldsymbol{Q}\boldsymbol{W}_{u-1}\|_2 \geq x\right) \\
&= \sum_{n=0}^{b} \sum_{S \in \binom{[b]}{n}} \mathbb{P}\left(\|\boldsymbol{F}_u - \boldsymbol{Q}\boldsymbol{W}_{u-1}\|_2 \geq x | E_S]\right) \mathbb{P}\left(E_S\right) \\
&= \sum_{n=1}^{b-1} \sum_{S \in \binom{[b]}{n}} \mathbb{P}\left(\left\| \frac{1}{n} \sum_{i \in S} \boldsymbol{Y}_i^{(1)} - \frac{1}{b-n} \sum_{i \notin S} \boldsymbol{Y}_i^{(0)} \right\|_2 \geq x \,\middle|\, E_S \right) \mathbb{P}\left(E_S\right) \\
&\quad + \mathbb{P}\left(\left\| \frac{1}{b} \sum_{i=1}^{b} \boldsymbol{Y}_i^{(0)} \right\|_2 \geq x \,\middle|\, s_1 = \dots = s_b = 0 \right) \mathbb{P}\left(s_1 = \dots = s_b = 0\right) \\
&\quad + \mathbb{P}\left(\left\| \frac{1}{b} \sum_{i=1}^{b} \boldsymbol{Y}_i^{(1)} \right\|_2 \geq x \,\middle|\, s_1 = \dots = s_b = 1 \right) \mathbb{P}\left(s_1 = \dots = s_b = 1\right) \\
&\leq \sum_{n=1}^{b-1} \sum_{S \in \binom{[b]}{n}} \mathbb{P}\left(\left\| \frac{1}{n} \sum_{i \in S} \boldsymbol{Y}_i^{(1)} \right\|_2 \geq c_1 x \textbf{ or } \left\| \frac{1}{b-n} \sum_{i \notin S} \boldsymbol{Y}_i^{(0)} \right\|_2 \geq c_0 x \,\middle|\, E_S \right) \mathbb{P}\left(E_S\right) \\
&\quad + (d+m) p_0^b \exp\left(-\frac{bx^2}{4\mathcal{V}}\right) + (d+m) p_1^b \exp\left(-\frac{bx^2}{4\mathcal{V}}\right) \\
&\leq \sum_{n=1}^{b-1} p_1^n p_0^{b-n} \sum_{S \in \binom{[b]}{n}} \left\{ \mathbb{P}\left(\left\| \frac{1}{n} \sum_{i \in S} \boldsymbol{Y}_i^{(1)} \right\|_2 \geq c_1 x \,\middle|\, E_S \right) + \mathbb{P}\left(\left\| \frac{1}{b-n} \sum_{i \notin S} \boldsymbol{Y}_i^{(0)} \right\|_2 \geq c_0 x \,\middle|\, E_S \right) \right\} \\
&\quad + (d+m) p_0^b \exp\left(-\frac{b c_0^2 x^2}{4\mathcal{V}}\right) + (d+m) p_1^b \exp\left(-\frac{b c_1^2 x^2}{4\mathcal{V}}\right).
\end{aligned}$$

Now, we make the crucial observation that *conditioned* on the event E_S, both $\left\{\boldsymbol{Y}_i^{(1)}\right\}_{i \in S}$ and $\left\{\boldsymbol{Y}_i^{(0)}\right\}_{i \in S^\complement}$ are sets of identically distributed $d \times m$ random matrices that are 1. conditionally independent and 2. conditionally zero-mean. In addition, with Assumption 6.1, we have that for each $i \in S$

$$\mathbb{P}\left[\left\|\boldsymbol{Y}_i^{(1)}\right\|_2 \leq \mathcal{M} | i \in S\right] = 1, \quad \operatorname{Var}\left(\boldsymbol{Y}_i^{(1)} | i \in S\right) \leq \mathcal{V}.$$

(and analogously for each $i \notin S$). Thus, applying the matrix Bernstein inequality[6] (Proposition D.1), we obtain the following: when $0 \leq x \leq \frac{3\mathcal{V}}{\mathcal{M}}$,

$$\begin{aligned}
\mathbb{P}\left(\|\boldsymbol{F}_u - \boldsymbol{Q}\boldsymbol{W}_{u-1}\|_2 \geq x\right) &\leq (d+m) \sum_{n=1}^{b-1} p_1^n p_0^{b-n} \sum_{S \in \binom{[b]}{n}} \left\{ \exp\left(-\frac{n c_1^2 x^2}{4\mathcal{V}}\right) + \exp\left(-\frac{(b-n) c_0^2 x^2}{4\mathcal{V}}\right) \right\} \\
&\quad + (d+m) p_0^b \exp\left(-\frac{b c_0^2 x^2}{4\mathcal{V}}\right) + (d+m) p_1^b \exp\left(-\frac{b c_1^2 x^2}{4\mathcal{V}}\right) \\
&\leq (d+m) \sum_{n=0}^{b} \binom{b}{n} p_1^n p_0^{b-n} \left\{ \exp\left(-\frac{n c_1^2 x^2}{4\mathcal{V}}\right) + \exp\left(-\frac{(b-n) c_0^2 x^2}{4\mathcal{V}}\right) \right\}
\end{aligned}$$

[6] Precisely, we use its conditional version where the means and the variances are replaced with their conditional counterparts.

$$= (d+m)\left\{\left(p_0 + p_1 e^{-\frac{c_1^2 x^2}{4\mathcal{V}}}\right)^b + \left(p_1 + p_0 e^{-\frac{c_0^2 x^2}{4\mathcal{V}}}\right)^b\right\} < \frac{\delta}{8}.$$

For this to occur, it suffices for both terms to be bounded by $\frac{\delta}{16}$. Let us consider only the first term, as the second term follows from symmetry. Taking the log, we have

$$\log\left((1-p_1) + p_1 e^{-\frac{c_1^2 x^2}{4\mathcal{V}}}\right) \leq \frac{1}{b}\log\frac{\delta}{16(d+m)}.$$

Using $\log(1+x) \leq x$, it suffices to have

$$p_1\left(e^{-\frac{c_1^2 x^2}{4\mathcal{V}}} - 1\right) \leq \frac{1}{b}\log\frac{\delta}{16(d+m)}.$$

Using $x \geq 1 - e^{-2x}$ for $x \leq 1/2$, it now suffices to have

$$\frac{8\mathcal{V}}{bp_1c_1^2}\log\frac{16(d+m)}{\delta} \leq x^2 \leq \min\left(\frac{2\mathcal{V}}{c_1^2}, \frac{9\mathcal{V}^2}{\mathcal{M}^2}\right).$$

Combining this with the other term and, for simplicity[7] choosing $c_0 = c_1 = 1/2$, we have our desired statement.

The other inequality follows analogously; the only difference is that the dimension prefactor of Bernstein inequality changes from $d+m$ to $2m$. □

We now finish the proof of Theorem 6.1. Following the notation of Lemma 6.1, $\boldsymbol{Z}_t = \boldsymbol{F}_u - \boldsymbol{Q}\boldsymbol{W}_{u-1}$.

D.4 Proof of Theorem 6.2 - Bounding the Final Error

Recall that the iterates are $\boldsymbol{V}_{t+1} = \mathtt{QR}(\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\widehat{\boldsymbol{\Sigma}}_t\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\boldsymbol{V}_t)$, where $\widehat{\boldsymbol{N}} = \boldsymbol{N}_U$ is the output of Algorithm 1 (or 3), $\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}} := \boldsymbol{I} - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal$, and $\widehat{\boldsymbol{\Sigma}}_t := \frac{1}{B}\sum_{j=(t-1)B+1}^{tB}\boldsymbol{x}_j\boldsymbol{x}_j^\intercal$ is the sample covariance at time step t of Algorithm 2. This can be rewritten as

$$\boldsymbol{V}_{t+1} = \mathtt{QR}(\boldsymbol{\Pi}_{\boldsymbol{N}}\boldsymbol{\Sigma}\boldsymbol{\Pi}_{\boldsymbol{N}}\boldsymbol{V}_t + \boldsymbol{Z}_{t,2}),$$

where $\boldsymbol{Z}_{t,2} = (\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}} - \boldsymbol{\Pi}_{\boldsymbol{N}}\boldsymbol{\Sigma}\boldsymbol{\Pi}_{\boldsymbol{N}})\boldsymbol{V}_t$ is the noise matrix and $\boldsymbol{\Sigma} = \sum_{s\in\{0,1\}} p_s(\boldsymbol{\Sigma}_s + \boldsymbol{\mu}_s\boldsymbol{\mu}_s^\intercal)$. By Assumption 6.1, we have that $\|\boldsymbol{\Sigma}\|_2 \leq V$.

The following lemma states that the sine angle and the corresponding Grassmannian (projection) distance are the same:

Lemma D.1. $\|\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\boldsymbol{N}\|_2 = \|\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}} - \boldsymbol{\Pi}_{\boldsymbol{N}}\|_2$.

Proof. By Theorem 2.5.1 of Golub and Loan (2013), we have that

$$\|\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}} - \boldsymbol{\Pi}_{\boldsymbol{N}}\|_2 = \|(\widehat{\boldsymbol{N}}^\perp)^\intercal\boldsymbol{N}\|_2 \overset{(*)}{=} \left\|\widehat{\boldsymbol{N}}^\perp(\widehat{\boldsymbol{N}}^\perp)^\intercal\boldsymbol{N}\right\|_2 = \|\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\boldsymbol{N}\|_2,$$

where $(*)$ follows from the observation that for any column-orthonormal matrices A and B,

$$\|\boldsymbol{A}\boldsymbol{A}^\intercal\boldsymbol{B}\|_2 \leq \|\boldsymbol{A}^\intercal\boldsymbol{B}\|_2 = \|\boldsymbol{A}^\intercal\boldsymbol{A}\boldsymbol{A}^\intercal\boldsymbol{B}\|_2 \leq \|\boldsymbol{A}\boldsymbol{A}^\intercal\boldsymbol{B}\|_2.$$

□

Fix some $\boldsymbol{V} \in St(d,k)$, and from the design of our algorithm, $\widehat{\boldsymbol{N}}$ is also fixed. By the triangle inequality, our given assumption, above lemma, and the fact that $\|\boldsymbol{V}\|_2 = 1$, we have

$$\begin{aligned}\|\boldsymbol{Z}_{t,2}\|_2 &\leq \left\|\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma})\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\boldsymbol{V}\right\|_2 + 2\|\boldsymbol{\Sigma}\|_2\|\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}} - \boldsymbol{\Pi}_{\boldsymbol{N}}\|_2 \\ &\leq \left\|\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma})\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\boldsymbol{V}\right\|_2 + \frac{\Delta_{k,\kappa}}{10}\min\left(\epsilon, \frac{\delta}{2\sqrt{dk}}\right).\end{aligned}$$

[7] One could try to optimize for c_0, c_1 to obtain an "optimal" sample complexities. However, from some preliminary computations, this seems to be not worth pursuing, and we conjecture that it would yield the same asymptotic dependency as when $c_0 = c_1 = 1/2$.

Theorem D.2. *For any $\delta > 0$, choose $B \geq \frac{4}{9}\frac{\mathcal{M}^2}{\mathcal{V}+V^2}\log\frac{4(d+k)}{\delta}$. Then, given $\boldsymbol{V}$, the following hold with probability at least $1-\delta$:*

$$\left\|\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}(\widehat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma})\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\boldsymbol{V}\right\|_2 \leq \mathcal{E}_{d+k}^{(\Sigma)},$$
$$\left\|\boldsymbol{U}_k^\intercal\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}(\widehat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma})\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\boldsymbol{V}\right\|_2 \leq \mathcal{E}_{2k}^{(\Sigma)},$$

where

$$\mathcal{E}_y^{(\Sigma)} \triangleq \sqrt{\frac{4(\mathcal{V}+V^2)}{B}\log\frac{4y}{\delta}}.$$

Proof. Note that $\boldsymbol{x}_i$'s are i.i.d. samples from $\mathcal{D}$. Let

$$\boldsymbol{Y}_i = \boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\left(\boldsymbol{x}_i\boldsymbol{x}_i^\intercal - \boldsymbol{\Sigma}\right)\boldsymbol{\Pi}_{\widehat{\boldsymbol{N}}}\boldsymbol{V} \quad (i \in \{(t-1)B+1,\ldots,tB\}).$$

Then, $\boldsymbol{Y}_i$ are i.i.d. zero-mean $d \times k$ random matrices across i. In addition, with Assumption 6.1,

$$\mathbb{P}\left[\|\boldsymbol{x}\boldsymbol{x}^\intercal - \boldsymbol{\Sigma}\|_2 \leq \mathcal{M}\right] = \sum_{s' \in \{0,1\}} \mathbb{P}\left[\|\boldsymbol{x}\boldsymbol{x}^\intercal - \boldsymbol{\Sigma}\|_2 \leq \mathcal{M} \,|\, s = s'\right]\mathbb{P}[s=s'] = 1,$$

and

$$\begin{aligned}\mathrm{Var}(\boldsymbol{x}\boldsymbol{x}^\intercal) &= \sum_{s' \in \{0,1\}} \left\{\mathrm{Var}(\boldsymbol{x}\boldsymbol{x}^\intercal|s=s') + \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal|s=s']^2\right\}\mathbb{P}[s=s'] - \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal]^2 \\ &\preceq (\mathcal{V}+V^2)\boldsymbol{I},\end{aligned}$$

where $\preceq$ is the Loewner order and with a slight abuse of notation, we denote $\mathrm{Var}(\boldsymbol{X}) = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\intercal]$. Combined, we have that $\|\boldsymbol{Y}_i\|_2 \leq \mathcal{M}$ a.s. and $\mathrm{Var}(\boldsymbol{Y}_i) \leq \mathcal{V}+V^2$. Applying the matrix Bernstein inequality (Proposition D.1), we obtain the following: when $0 \leq x \leq \frac{3(\mathcal{V}+V^2)}{\mathcal{M}}$,

$$\mathbb{P}\left(\left\|\frac{1}{B}\sum_{i=1}^{B}\boldsymbol{Y}_i\right\|_2 \geq x\right) \leq (d+k)\exp\left(-\frac{Bx^2}{4(\mathcal{V}+V^2)}\right) \leq \frac{\delta}{4}.$$

Solving for x, we have

$$\frac{4(\mathcal{V}+V^2)}{B}\log\frac{4(d+k)}{\delta} \leq x^2 \leq 9\left(\frac{(\mathcal{V}+V^2)}{\mathcal{M}}\right)^2,$$

from which the statement naturally follows.

This follows analogously for $\boldsymbol{U}_k^\intercal\boldsymbol{Z}_{t,2}$; the only difference is that the dimension prefactor of Bernstein inequality changes from $d+k$ to $2k$. □

D.5 Bounding the Estimation Error of Projected Mean Difference

Recall that

Theorem D.3. *For any $\delta > 0$, choose $Ub \geq 2\left(\frac{1}{p_0}+\frac{1}{p_1}\right)\log\frac{4d}{\delta}$. Then, the following holds with probability at least $1-\delta$:*

$$\|\boldsymbol{g}-\boldsymbol{f}\|_2 \leq \mathcal{E}^{(\mathrm{f})} \triangleq 2\mathfrak{c}\left(\frac{\sigma_0}{\sqrt{p_0}}+\frac{\sigma_1}{\sqrt{p_1}}\right)\sqrt{\frac{2}{Ub}\log\frac{4d}{\delta}}.$$

Proof. Again, note that $(s_i, \boldsymbol{x}_i)$'s are i.i.d. samples. Consider independent random vectors

$$\boldsymbol{y}_i^{(s)} = \boldsymbol{x}_i - \boldsymbol{\mu}_s \quad (i \in \{1,\ldots,Ub\}).$$

Recall that for each $S \subseteq [b]$, $E_S = \{s_i = \mathbb{1}[i \in S]\ \forall i \in [b]\}$ is an event satisfying $\mathbb{P}[E_S] = p_1^{|S|}p_0^{b-|S|}$. To exploit this, we now apply the peeling argument as follows: for any $c_0, c_1 \in (0,1)$ with $c_0 + c_1 = 1$,

$\mathbb{P}\left(\|\boldsymbol{g}-\boldsymbol{f}\|_2 \geq x\right)$

$$
\begin{aligned}
&= \sum_{n=0}^{Ub} \sum_{S \in \binom{[Ub]}{n}} \mathbb{P}\left(\|\boldsymbol{g} - \boldsymbol{f}\|_2 \geq x | E_S]\right) \mathbb{P}\left(E_S\right) \\
&= \sum_{n=1}^{b-1} \sum_{S \in \binom{[Ub]}{n}} \mathbb{P}\left(\left\| \frac{1}{n} \sum_{i \in S} \boldsymbol{y}_i^{(1)} - \frac{1}{Ub-n} \sum_{i \notin S} \boldsymbol{y}_i^{(0)} \right\|_2 \geq x \,\middle|\, E_S \right) \mathbb{P}\left(E_S\right) \\
&\quad + \mathbb{P}\left(\left\| \frac{1}{Ub} \sum_{i=1}^{Ub} \boldsymbol{y}_i^{(0)} \right\|_2 \geq x \,\middle|\, s_1 = \cdots = s_{Ub} = 0 \right) \mathbb{P}\left(s_1 = \cdots = s_{Ub} = 0\right) \\
&\quad + \mathbb{P}\left(\left\| \frac{1}{Ub} \sum_{i=1}^{Ub} \boldsymbol{y}_i^{(1)} \right\|_2 \geq x \,\middle|\, s_1 = \cdots = s_{Ub} = 1 \right) \mathbb{P}\left(s_1 = \cdots = s_{Ub} = 1\right) \\
&\leq \sum_{n=1}^{Ub-1} \sum_{S \in \binom{[Ub]}{n}} \mathbb{P}\left(\left\| \frac{1}{n} \sum_{i \in S} \boldsymbol{y}_i^{(1)} \right\|_2 \geq c_1 x \textbf{ or } \left\| \frac{1}{Ub-n} \sum_{i \notin S} \boldsymbol{y}_i^{(0)} \right\|_2 \geq c_0 x \,\middle|\, E_S \right) \mathbb{P}\left(E_S\right) \\
&\quad + 2dp_0^{Ub} \exp\left(-\frac{Ubx^2}{\mathfrak{c}^2\sigma_0^2}\right) + 2dp_1^{Ub} \exp\left(-\frac{Ubx^2}{\mathfrak{c}^2\sigma_1^2}\right) \\
&\leq \sum_{n=1}^{Ub-1} p_1^n p_0^{Ub-n} \sum_{S \in \binom{[Ub]}{n}} \left\{ \mathbb{P}\left(\left\| \frac{1}{n} \sum_{i \in S} \boldsymbol{y}_i^{(1)} \right\|_2 \geq c_1 x \,\middle|\, E_S \right) + \mathbb{P}\left(\left\| \frac{1}{Ub-n} \sum_{i \notin S} \boldsymbol{y}_i^{(0)} \right\|_2 \geq c_0 x \,\middle|\, E_S \right) \right\} \\
&\quad + 2dp_0^{Ub} \exp\left(-\frac{Ubx^2}{\mathfrak{c}^2\sigma_0^2}\right) + 2dp_1^{Ub} \exp\left(-\frac{Ubx^2}{\mathfrak{c}^2\sigma_1^2}\right).
\end{aligned}
$$

Now, we make the crucial observation that *conditioned* on the event E_S, both $\left\{\boldsymbol{y}_i^{(1)}\right\}_{i \in S}$ and $\left\{\boldsymbol{y}_i^{(0)}\right\}_{i \in S^\complement}$ are sets of identically distributed d-dimensional random vectors that are 1. conditionally independent and 2. conditionally zero-mean. In addition, with Assumption 6.1, we have that for each $i \in S$, $\boldsymbol{y}_i^{(1)} \in \mathrm{nSG}(\sigma_1)$ and analogously for each $i \notin S$.

Thus, applying the vector Bernstein inequality (Proposition D.2), we obtain the following: for $x > 0$,

$$
\begin{aligned}
\mathbb{P}\left(\|\boldsymbol{g} - \boldsymbol{f}\|_2 \geq x\right) &\leq 2d \sum_{n=1}^{Ub-1} p_1^n p_0^{Ub-n} \sum_{S \in \binom{[Ub]}{n}} \left\{ \exp\left(-\frac{nc_1^2x^2}{\mathfrak{c}^2\sigma_1^2}\right) + \exp\left(-\frac{(Ub-n)c_0^2x^2}{\mathfrak{c}^2\sigma_0^2}\right) \right\} \\
&\quad + 2dp_0^{Ub} \exp\left(-\frac{Ubx^2}{\mathfrak{c}^2\sigma_0^2}\right) + 2dp_1^{Ub} \exp\left(-\frac{Ubx^2}{\mathfrak{c}^2\sigma_1^2}\right) \\
&\leq 2d \sum_{n=0}^{Ub} \binom{Ub}{n} p_1^n p_0^{Ub-n} \left\{ \exp\left(-\frac{nc_1^2x^2}{\mathfrak{c}^2\sigma_1^2}\right) + \exp\left(-\frac{(Ub-n)c_0^2x^2}{\mathfrak{c}^2\sigma_0^2}\right) \right\} \\
&= 2d \left\{ \left(p_0 + p_1 e^{-\frac{c_1^2x^2}{\mathfrak{c}^2\sigma_1^2}}\right)^{Ub} + \left(p_1 + p_0 e^{-\frac{c_0^2x^2}{\mathfrak{c}^2\sigma_1^2}}\right)^{Ub} \right\} < \delta.
\end{aligned}
$$

It suffices for both terms in the parenthesis to be bounded by $\frac{\delta}{2}$. Let us consider only the first term, as the second term follows from symmetry. Taking the log, we have

$$
\log\left((1-p_1) + p_1 e^{-\frac{c_1^2x^2}{\mathfrak{c}^2\sigma_1^2}}\right) \leq \frac{1}{Ub} \log \frac{\delta}{4d}.
$$

Using $\log(1+x) \leq x$, it suffices to have

$$
p_1 \left(e^{-\frac{c_1^2x^2}{\mathfrak{c}^2\sigma_1^2}} - 1\right) \leq \frac{1}{Ub} \log \frac{\delta}{4d}.
$$

Using $x \geq 1 - e^{-2x}$ for $x \leq 1/2$, it now suffices to have

$$\frac{2\mathfrak{c}^2\sigma_1^2}{Ubp_1c_1^2}\log\frac{4d}{\delta} \leq x^2 \leq \frac{\mathfrak{c}^2\sigma_1^2}{c_1^2}.$$

Combining this with the other term and, for simplicity (see the footnote on pg. 25), choosing $c_0 = c_1 = 1/2$, we have our desired statement. □

D.6 Proof of Theorem 6.3 - Sample Complexity for PAFO-learnability

Let us fix some $\varepsilon_1, \varepsilon_2, \delta \in (0,1)$. From the definition of PAFO-learnability (Definition 4.2), with a large enough sample size, we must guarantee the following with probability at least $1-\delta$:

$$\operatorname{tr}\left(\boldsymbol{V}^\intercal\boldsymbol{\Sigma}\boldsymbol{V}\right) \geq \operatorname{tr}\left(\boldsymbol{V}^{\star\intercal}\boldsymbol{\Sigma}\boldsymbol{V}^\star\right) - \varepsilon_1, \quad \|\boldsymbol{N}\boldsymbol{N}^\intercal\boldsymbol{V}\|_2 \leq \varepsilon_2.$$

Let $\widehat{\boldsymbol{N}}$ be the final estimate of the unfair subspace $\boldsymbol{N}$ from Algorithm 1, and let $\boldsymbol{V}$ be the final estimate of $\boldsymbol{U}_k = \boldsymbol{V}^\star$, whose columns are top k eigenvectors of $\boldsymbol{\Pi_N\Sigma\Pi_N}$.

We first recall the von Neumann trace inequality (Mirsky, 1975; von Neumann, 1937):

Lemma D.2 (Theorem H.1.g of Marshall et al. (2011))**.** *If $\boldsymbol{A}, \boldsymbol{B}$ are two $n \times n$ Hermitian matrices, then*

$$\operatorname{tr}(\boldsymbol{AB}) \leq \sum_{i=1}^{n} \lambda_i(\boldsymbol{A})\lambda_i(\boldsymbol{B}),$$

where $\lambda_i(\cdot)$ is the i-th smallest eigenvalue.

Using this, we can write the optimality (first inequality) as follows:

$$\begin{aligned}
\operatorname{tr}\left(\boldsymbol{U}_k^\intercal\boldsymbol{\Sigma}\boldsymbol{U}_k\right) - \operatorname{tr}\left(\boldsymbol{V}^\intercal\boldsymbol{\Sigma}\boldsymbol{V}\right) &= \operatorname{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{U}_k\boldsymbol{U}_k^\intercal - \boldsymbol{V}\boldsymbol{V}^\intercal)\right) \\
&\overset{(a)}{\leq} \sum_{i=1}^{d} \lambda_i(\boldsymbol{\Sigma})\lambda_i(\boldsymbol{U}_k\boldsymbol{U}_k^\intercal - \boldsymbol{V}\boldsymbol{V}^\intercal) \\
&\overset{(b)}{=} \sum_{i=1}^{2k} \lambda_i(\boldsymbol{\Sigma})\lambda_i(\boldsymbol{U}_k\boldsymbol{U}_k^\intercal - \boldsymbol{V}\boldsymbol{V}^\intercal) \\
&\leq 2k\|\boldsymbol{\Sigma}\|_2\|\boldsymbol{U}_k\boldsymbol{U}_k^\intercal - \boldsymbol{V}\boldsymbol{V}^\intercal\|_2 \\
&\leq 2kV\|(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^\intercal)\boldsymbol{U}_k\|_2 \leq \varepsilon_1.
\end{aligned}$$

Here, (a) follows from the von Neumann trace inequality, and (b) follows from the fact that $\boldsymbol{U}_k\boldsymbol{U}_k^\intercal - \boldsymbol{V}\boldsymbol{V}^\intercal$ is a symmetric matrix of rank at most $2k$. Thus, for optimality, it suffices to ensure that $\|(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^\intercal)\boldsymbol{U}_k\|_2 \leq \frac{\varepsilon_1}{2kV}$.

Similarly, for fairness (second inequality), it suffices to ensure that $\|(\boldsymbol{I} - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal)\boldsymbol{N}\|_2 \leq \varepsilon_2$, as

$$\|\boldsymbol{N}\boldsymbol{N}^\intercal\boldsymbol{V}\|_2 \overset{(*)}{=} \|\boldsymbol{N}^\intercal\Pi_{\widehat{\boldsymbol{N}}}\boldsymbol{V}\|_2 \leq \left\|\boldsymbol{N}^\intercal(\boldsymbol{I} - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal)\right\|_2 = \left\|(\boldsymbol{I} - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal)\boldsymbol{N}\right\|_2 \leq \varepsilon_2$$

where $(*)$ follows from $\boldsymbol{V} = \Pi_{\widehat{\boldsymbol{N}}}\boldsymbol{V}$ (due to the design of our algorithm) and $\boldsymbol{N} \in St(d,m)$.

From Lemma 6.1 and Theorem 6.2, *given* that

$$\|(\boldsymbol{I} - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal)\boldsymbol{N}\|_2 \lesssim \eta_k = \eta_k(\varepsilon_1, \varepsilon_2, \delta) \triangleq \min\left(\varepsilon_2, \frac{\Delta_{k,\kappa}}{kV^2}\varepsilon_1, \frac{\Delta_{k,\kappa}}{V\sqrt{dk}}\delta\right), \tag{15}$$

a total of N_2 samples are sufficient in Algorithm 2 for ensuring ε_1-optimality with probability at least $1 - \frac{\delta}{4}$, where

$$N_2 \gtrsim \left(\frac{K_{k,\kappa}(\mathcal{V} + V^2)}{\Delta_{k,\kappa}^3}\left(\frac{dk}{\delta^2} + \frac{k^2V^2}{\varepsilon_1^2}\right) + \frac{K_{k,\kappa}\mathcal{M}^2}{\Delta_{k,\kappa}(\mathcal{V} + V^2)}\right)\left(\log\frac{dkV}{\varepsilon_1\delta}\right)^2. \tag{16}$$

We now focus on obtaining the sample complexity for Algorithm 1 to satisfy Eqn. (15) with probability at least $1 - \frac{\delta}{4}$. Then combining those and simplifying gives us the desired statement.

From hereon and forth, let us write $\boldsymbol{W} = \boldsymbol{W}_U$ and $\boldsymbol{P} = \boldsymbol{P}_m$ for notational simplicity. Also, denote $\boldsymbol{f}' = (\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\intercal)\boldsymbol{f}$, $\widetilde{\boldsymbol{f}} = \frac{\boldsymbol{f}'}{\|\boldsymbol{f}'\|_2}$, $\boldsymbol{g}' = (\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\intercal)\boldsymbol{g}$, and $\widetilde{\boldsymbol{g}} = \frac{\boldsymbol{g}'}{\|\boldsymbol{g}'\|_2}$.

First assume that $\boldsymbol{f}' \neq \boldsymbol{0}$. Then with probability at least $1 - \frac{\delta}{8}$,

$$\begin{aligned}\|(\boldsymbol{I} - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal)\boldsymbol{N}\|_2 = \|\boldsymbol{N}\boldsymbol{N}^\intercal - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal\|_2 &\le \|\boldsymbol{P}\boldsymbol{P}^\intercal - \boldsymbol{W}\boldsymbol{W}^\intercal\|_2 + \|\widetilde{\boldsymbol{f}}\widetilde{\boldsymbol{f}}^\intercal - \widetilde{\boldsymbol{g}}\widetilde{\boldsymbol{g}}^\intercal\|_2 \\ &= \|(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\intercal)\boldsymbol{P}\|_2 + \|\widetilde{\boldsymbol{f}}\widetilde{\boldsymbol{f}}^\intercal - \widetilde{\boldsymbol{g}}\widetilde{\boldsymbol{g}}^\intercal\|_2.\end{aligned}$$

To bound the second term, which is basically $\sin\theta(\boldsymbol{f}', \boldsymbol{g}')$, we introduce the following lemma that provides a general bound of sine angle between a pair of vectors with a small ℓ_2-distance.

Lemma D.3. *Consider two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$ and $\epsilon \in \left(0, \frac{\|\boldsymbol{a}\|_2}{2}\right)$. Suppose $\|\boldsymbol{a} - \boldsymbol{b}\|_2 \le \epsilon$. Then, if we denote the (acute) angle between $\boldsymbol{a}$ and $\boldsymbol{b}$ as $\theta(\boldsymbol{a}, \boldsymbol{b})$, the following holds:*

$$\sin\theta(\boldsymbol{a}, \boldsymbol{b}) \le \frac{\sqrt{2}\epsilon}{\|\boldsymbol{a}\|_2}.$$

The ℓ_2-distance between $\boldsymbol{f}'$ and $\boldsymbol{g}'$ is then bounded as follows:

$$\begin{aligned}\|\boldsymbol{f}' - \boldsymbol{g}'\|_2 &= \|(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\intercal)(\boldsymbol{f} - \boldsymbol{g}) + (\boldsymbol{W}\boldsymbol{W}^\intercal - \boldsymbol{P}\boldsymbol{P}^\intercal)\boldsymbol{f}\|_2 \\ &\le 2\|\boldsymbol{f} - \boldsymbol{g}\|_2 + \|\boldsymbol{W}\boldsymbol{W}^\intercal - \boldsymbol{P}\boldsymbol{P}^\intercal\|_2 \|\boldsymbol{f}\|_2.\end{aligned}$$

Combining them in the case of $\boldsymbol{f}' \neq \boldsymbol{0}$, it is sufficient to ensure that

$$\|(\boldsymbol{I} - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal)\boldsymbol{N}\|_2 \le \frac{2f_{\max}}{f_{\min}}\|(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\intercal)\boldsymbol{P}\|_2 + \frac{2\|\boldsymbol{f} - \boldsymbol{g}\|_2}{f_{\min}} \lesssim \eta_k,$$

where we recall from Assumption 6.1 that $\|\boldsymbol{f}'\|_2 \in \{0\} \cup [f_{\min}, f_{\max}]$ and $\|\boldsymbol{f}\|_2 \in [0, f_{\max}]$.

From Lemma 6.1, Theorem 6.1, and Theorem D.3, a total of M_1 samples are sufficient to ensure the above with probability at least $1 - \frac{\delta}{4}$, where

$$M_1 \gtrsim \frac{1}{p_{\min}}\left\{\frac{K_{m,\nu}\mathcal{V}}{\Delta_{m,\nu}^3}\frac{f_{\max}^2}{f_{\min}^2}\frac{1}{\eta_k^2} + \frac{\sigma^2}{f_{\min}^2}\frac{1}{\eta_k^2} + \frac{K_{m,\nu}\mathcal{M}^2}{\Delta_{m,\nu}}\right\}\left(\log\left(\frac{d}{\eta_k\delta}\frac{f_{\max}}{f_{\min}}\right)\right)^2. \tag{17}$$

Here, we recall that $\eta_k = \min\left(\varepsilon_2, \frac{\Delta_{k,\kappa}}{kV^2}\varepsilon_1, \frac{\Delta_{k,\kappa}}{V\sqrt{dk}}\delta\right)$.

Now assume that $\boldsymbol{f}' = \boldsymbol{0}$. Then with probability at least $1 - \frac{\delta}{8}$,

$$\|(\boldsymbol{I} - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal)\boldsymbol{N}\|_2 = \|\boldsymbol{N}\boldsymbol{N}^\intercal - \widehat{\boldsymbol{N}}\widehat{\boldsymbol{N}}^\intercal\|_2 = \|\boldsymbol{P}\boldsymbol{P}^\intercal - \boldsymbol{W}\boldsymbol{W}^\intercal\|_2.$$

From Lemma 6.1 and Theorem 6.1, a total of M_2 samples are sufficient in Algorithm 1 to ensure $\|\boldsymbol{P}\boldsymbol{P}^\intercal - \boldsymbol{W}\boldsymbol{W}^\intercal\|_2 \lesssim \eta_k$ with probability at least $1 - \frac{\delta}{8}$, where

$$M_2 \gtrsim \frac{1}{p_{\min}}\left\{\frac{K_{m,\nu}\mathcal{V}}{\Delta_{m,\nu}^3}\frac{1}{\eta_k^2} + \frac{K_{m,\nu}\mathcal{M}^2}{\Delta_{m,\nu}}\right\}\left(\log\frac{d}{\eta_k\delta}\right)^2. \tag{18}$$

Combining Eqn. (17) and (18), we have our desired statement.

Lastly, we provide the missing proof for our lemma:

Proof of Lemma D.3. Observe that $\|\boldsymbol{b}\|_2 \ge \|\boldsymbol{a}\|_2 - \epsilon \ge \|\boldsymbol{a}\|_2/2$ and

$$2(\|\boldsymbol{a}\|_2\|\boldsymbol{b}\|_2 - \langle\boldsymbol{a}, \boldsymbol{b}\rangle) \le \|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}\|_2^2 - 2\langle\boldsymbol{a}, \boldsymbol{b}\rangle = \|\boldsymbol{a} - \boldsymbol{b}\|_2^2 \le \epsilon^2.$$

Thus,

$$\begin{aligned}\sin^2\theta(\boldsymbol{a}, \boldsymbol{b}) &= (1 + \cos\theta(\boldsymbol{a}, \boldsymbol{b}))(1 - \cos\theta(\boldsymbol{a}, \boldsymbol{b})) \le 2(1 - \cos\theta(\boldsymbol{a}, \boldsymbol{b})) \\ &= 2\frac{\|\boldsymbol{a}\|_2\|\boldsymbol{b}\|_2 - \langle\boldsymbol{a}, \boldsymbol{b}\rangle}{\|\boldsymbol{a}\|_2\|\boldsymbol{b}\|_2} \le \frac{2\epsilon^2}{\|\boldsymbol{a}\|_2^2}.\end{aligned}$$

□

E More Experiments

E.1 Synthetic Example

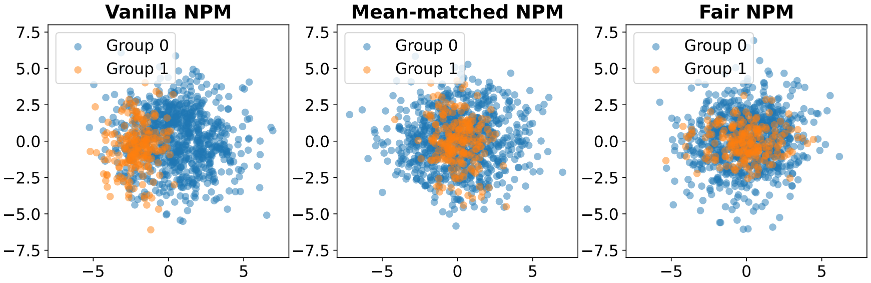


Figure 2: **Synthetic Example**: Vanilla NPM v.s. Mean-matched NPM v.s. FNPM (ours).

We randomly generated two different group-conditional distributions as 10-dimensional multivariate Gaussians with different mean vectors $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$, satisfying $\boldsymbol{\mu} = (1-p)\boldsymbol{\mu}_0 + p\boldsymbol{\mu}_1 = \mathbf{0}$, and different covariance matrices $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$. We choose the sampling probability parameter p as 0.2, which induces asymmetric sampling between two sensitive attributes. The covariance matrices are designed so that both of their eigenvalue spectra, $\{\sigma_1, \dots, \sigma_d\}$, have *power-law* decay as many practical datasets do (Liu et al., 2015a), *i.e.,* $\sigma_j = \Theta(j^{-\alpha})$ for some decay parameter $\alpha \geq 1$.

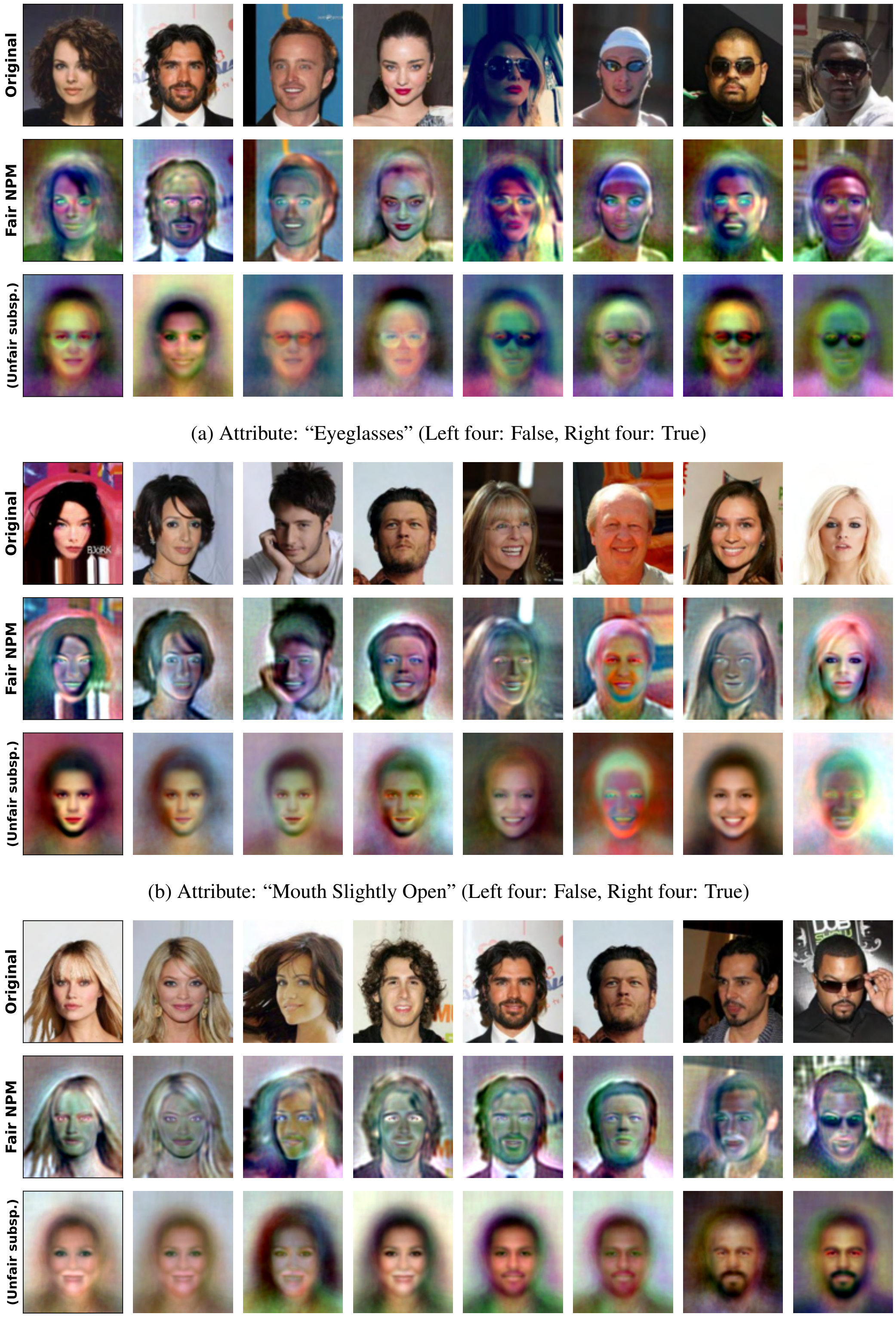
We first run and compare three different algorithms: vanilla NPM (without any fairness constraint), mean-matched NPM (with only constraint $\boldsymbol{V}^\intercal \boldsymbol{f} = \mathbf{0}$), and FNPM with $m = 3$. To ease the visualization, we project the sampled distributions onto a 2-dimensional subspace (*i.e.,* running 2-PCA). After running three algorithms for ten iterations and with a block size of $b = B = 1000$, we randomly sample 1000 data points and visualize the results of projecting the data points in Figure 2. In particular, for FNPM, we run 50 iterations for unfair subspace estimation (Algorithm 1) and run the other 50 iterations for PCA (Algorithm 2). We observe that FNPM does indeed enforce both mean-matching and (partial) covariance-matching, despite the setting being streaming and having asymmetric sampling probability.

E.2 Additional Results on the CelebA Dataset

We now provide additional experimental results on the CelebA dataset (Liu et al., 2015b). We consider three choices of sensitive attributes in the CelebA dataset: “Eyeglasses”, “Mouth Slightly Open”, and “Goattee”. In all the figures, False is when the sensitive attribute is absent; True is otherwise.

Figure 3 presents the main results for the new attributes. The first row shows the original images, the second row shows the images after projecting them to $\mathrm{col}(\boldsymbol{V}^\star)$, and the third row shows the images after projecting them to the estimated unfair subspace, $\mathrm{col}(\widehat{\boldsymbol{N}})$. Here, both $\widehat{\boldsymbol{N}}$ and $\boldsymbol{V}^\star$ are obtained from our FNPM; specifically speaking, they are obtained from Algorithm 1 and Algorithm 2, respectively. We remark that while we’ve used $m = 2$ for Figure 1a in the main text, here we use $m = 5$.

We then perform the two ablation studies as described in the main text, one on the dimension m of unfair subspace and another on the block size b of Algorithm 1 (or 3), for the additional sensitive attributes considered. In Figure 4, we vary $m \in \{1, 2, 5, 10\}$: as m increases, more features of images are “erased”, making the images from the two sensitive groups less distinguishable. At the same time, more semantically meaningful features are erased as well, resulting in rather “alien-like” images. In Figure 5, we vary $b \in \{32000, 8000, 3200, 1600\}$: as b increases, we have a more accurate estimation of unfair subspace $\mathrm{col}(\boldsymbol{N})$, resulting in more indistinguishable images (in sensitive attributes). As soon as the batch size b exceeds a certain threshold (e.g., 8000 for “Goattee”), $\mathrm{col}(\boldsymbol{N})$ is estimated very well, and the unfair features are cleanly recovered, as it can be seen in the bottom rows.

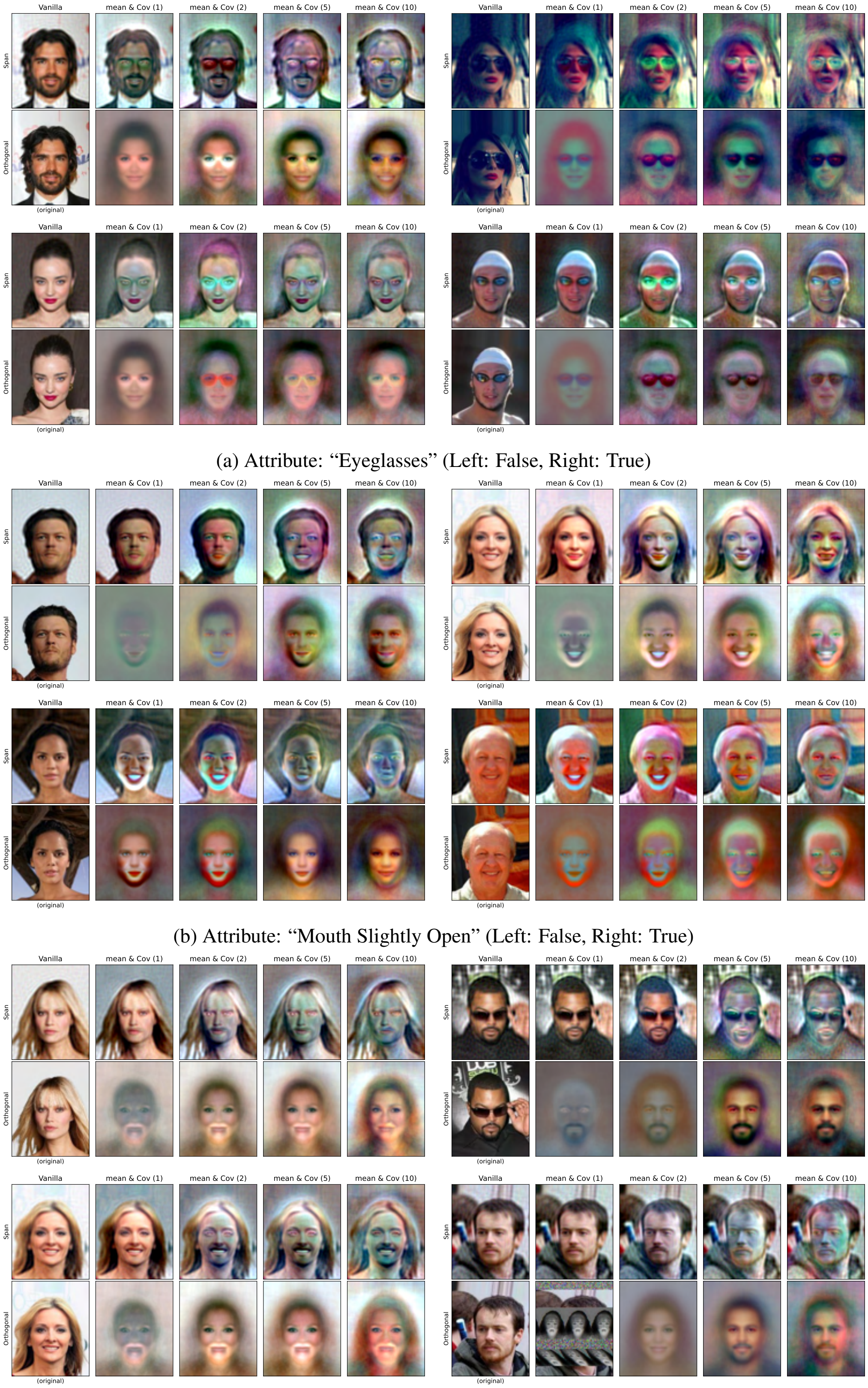


(a) Attribute: “Eyeglasses” (Left four: False, Right four: True)

(b) Attribute: “Mouth Slightly Open” (Left four: False, Right four: True)

(c) Attribute: “Goatee” (Left four: False, Right four: True)

Figure 3: CelebA, Additional results ($m = 5$).

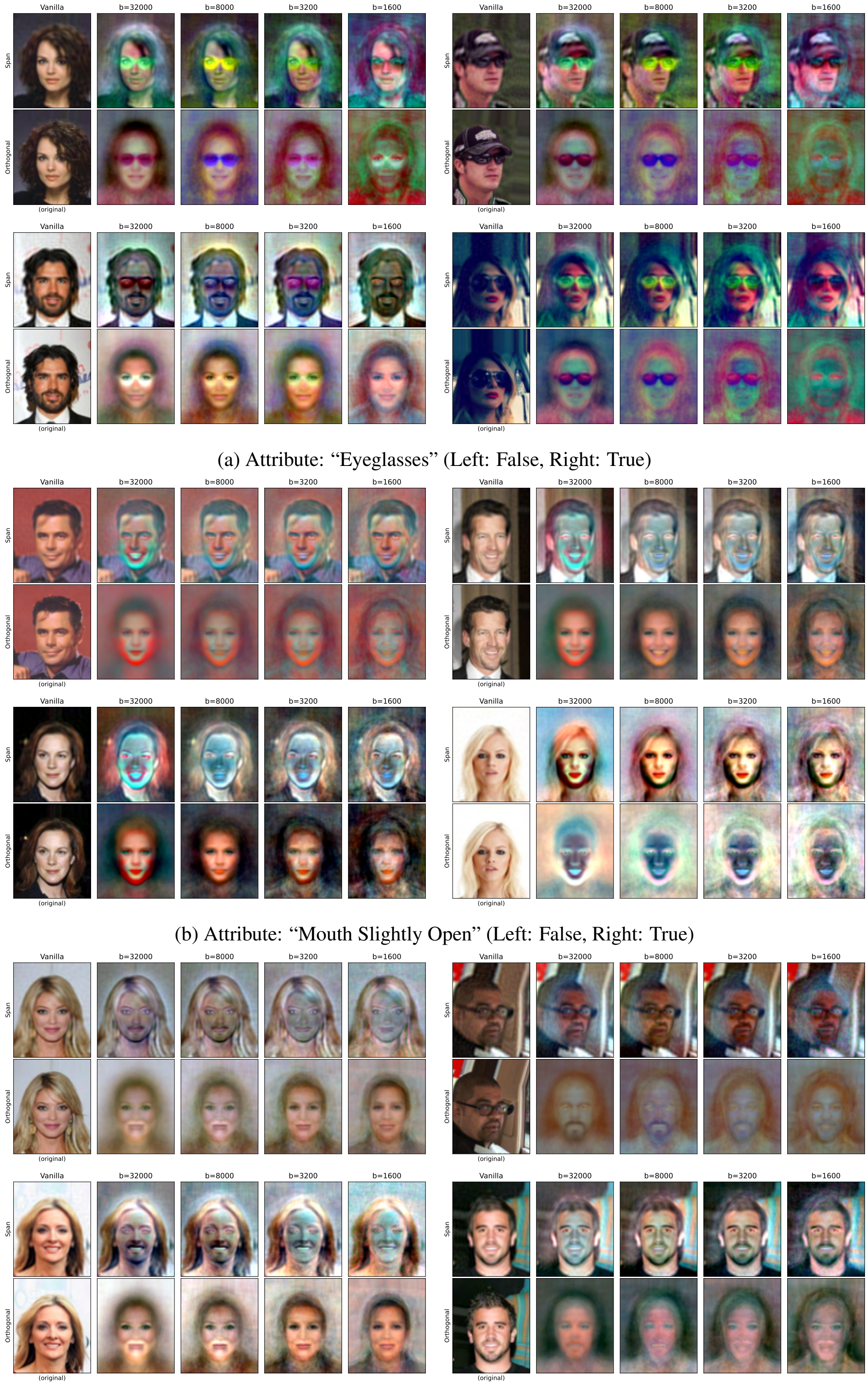


(a) Attribute: “Eyeglasses” (Left: False, Right: True)

(b) Attribute: “Mouth Slightly Open” (Left: False, Right: True)

(c) Attribute: “Goatee” (Left: False, Right: True)

Figure 4: CelebA, Additional ablation on $m \in \{1, 2, 5, 10\}$



(a) Attribute: “Eyeglasses” (Left: False, Right: True)

(b) Attribute: “Mouth Slightly Open” (Left: False, Right: True)

(c) Attribute: “Goatee” (Left: False, Right: True)

Figure 5: CelebA, Additional ablation on $b \in \{32000, 8000, 3200, 1600\}$