# Efficient Stochastic Algorithms for Canonical Correlation Analysis with Variance Reduction and Preconditioning

Chris Junchi Li[◇]

Department of Electrical Engineering and Computer Sciences[◇]
University of California, Berkeley

October 8, 2024

**Abstract**

Canonical Correlation Analysis (CCA) is a widely-used statistical method to extract shared structures from paired datasets. However, traditional methods for solving CCA become computationally expensive as dataset size and dimensionality increase. In this paper, we propose two novel stochastic optimization algorithms that guarantee global convergence: Alternating Least Squares (ALS) with variance reduction and a Shift-and-Invert preconditioning method. These algorithms are designed to efficiently handle high-dimensional, large-scale data, leveraging stochastic gradient-based updates to reduce computational overhead. Our theoretical analysis demonstrates their favorable time complexities, achieving faster convergence compared to previous stochastic approaches. Numerical experiments on real-world datasets show the superiority of our methods in terms of both accuracy and speed.

**Keywords:** Canonical Correlation Analysis, Stochastic Optimization, Variance Reduction, Global Convergence, Shift-and-Invert Preconditioning

## 1 Introduction

Canonical Correlation Analysis (CCA) is a fundamental technique in multivariate statistics, widely used to uncover linear relationships between two sets of random variables. Its applications span diverse fields such as neuroscience, machine learning, and genomics, where identifying shared patterns or latent structures between datasets is crucial. Traditional CCA approaches often rely on solving eigenvalue problems, which involve costly matrix decompositions. This becomes computationally prohibitive when dealing with large-scale or high-dimensional data, where forming and decomposing large covariance matrices is infeasible.

To address the scalability issues posed by such datasets, stochastic optimization methods have gained attention as promising alternatives. Techniques such as Stochastic Gradient Descent (SGD) and its variants can process small batches of data at each iteration, allowing efficient learning in large-scale settings. However, many existing stochastic methods for CCA either lack rigorous convergence guarantees or suffer from slow convergence rates, limiting their practical utility.

This paper presents two novel, globally convergent stochastic optimization algorithms for solving CCA: (1) an Alternating Least Squares (ALS) method enhanced with variance reduction and (2) a Shift-and-Invert preconditioning method that accelerates convergence by transforming the original problem into simpler subproblems. These algorithms are designed to efficiently handle large datasets and high-dimensional scenarios, where computational efficiency and global convergence are critical. Our contributions include both theoretical analysis and empirical validation on real-world datasets, demonstrating the efficiency and scalability of the proposed methods.

**Backgrounds** Canonical correlation analysis (CCA, [Hotelling, 1936]) and its extensions are ubiquitous techniques in scientific research areas for revealing the common sources of variability in multiple views of the same phenomenon. In CCA, the training set consists of paired observations from two views, denoted $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$, where $N$ is the training set size, $\mathbf{x}_i \in \mathbb{R}^{d_x}$ and $\mathbf{y}_i \in \mathbb{R}^{d_y}$ for $i = 1, \ldots, N$. We also denote the data matrices for each view[1] by $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d_x \times N}$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \mathbb{R}^{d_y \times N}$, and $d := d_x + d_y$. The objective of CCA is to find linear projections of each view such that the correlation between the projections is maximized:

$$\max_{\mathbf{u},\mathbf{v}} \quad \mathbf{u}^\top \mathbf{\Sigma}_{xy} \mathbf{v} \qquad \text{s.t.} \quad \mathbf{u}^\top \mathbf{\Sigma}_{xx} \mathbf{u} = \mathbf{v}^\top \mathbf{\Sigma}_{yy} \mathbf{v} = 1 \tag{1}$$

where $\mathbf{\Sigma}_{xy} = \frac{1}{N}\mathbf{X}\mathbf{Y}^\top$ is the cross-covariance matrix, $\mathbf{\Sigma}_{xx} = \frac{1}{N}\mathbf{X}\mathbf{X}^\top + \gamma_x \mathbf{I}$ and $\mathbf{\Sigma}_{yy} = \frac{1}{N}\mathbf{Y}\mathbf{Y}^\top + \gamma_y \mathbf{I}$ are the auto-covariance matrices, and $(\gamma_x, \gamma_y) \geq 0$ are regularization parameters [Vinod, 1976].

We denote by $(\mathbf{u}^*, \mathbf{v}^*)$ the global optimum of (1), which can be computed in closed-form. Define

$$\mathbf{T} := \mathbf{\Sigma}_{xx}^{-\frac{1}{2}} \mathbf{\Sigma}_{xy} \mathbf{\Sigma}_{yy}^{-\frac{1}{2}} \in \mathbb{R}^{d_x \times d_y} \tag{2}$$

and let $(\boldsymbol{\phi}, \boldsymbol{\psi})$ be the (unit-length) left and right singular vector pair associated with $\mathbf{T}$'s largest singular value $\rho_1$. Then the optimal objective value, i.e., the canonical correlation between the views, is $\rho_1$, achieved by $(\mathbf{u}^*, \mathbf{v}^*) = (\mathbf{\Sigma}_{xx}^{-\frac{1}{2}} \boldsymbol{\phi}, \mathbf{\Sigma}_{yy}^{-\frac{1}{2}} \boldsymbol{\psi})$. Note that

$$\rho_1 = \|\mathbf{T}\| \leq \left\|\mathbf{\Sigma}_{xx}^{-\frac{1}{2}} \mathbf{X}\right\| \left\|\mathbf{\Sigma}_{yy}^{-\frac{1}{2}} \mathbf{Y}\right\| \leq 1$$

Furthermore, we are guaranteed to have $\rho_1 < 1$ if $(\gamma_x, \gamma_y) > 0$.

For large and high dimensional datasets, it is time and memory consuming to first explicitly form the matrix $\mathbf{T}$ (which requires eigen-decomposition of the covariance matrices) and then compute its singular value decomposition (SVD). For such datasets, it is desirable to develop stochastic algorithms that have efficient updates, converges fast, and takes advantage of the input sparsity. There have been recent attempts to solve (1) based on stochastic gradient descent (SGD) methods [Ma et al., 2015, Wang et al., 2015, Xie et al., 2015], but none of these work provides rigorous convergence analysis for their stochastic CCA algorithms.

The main result of this paper is the proposal of two globally convergent meta-algorithms for solving (1), namely, alternating least squares (ALS, Algorithm 2) and shift-and-invert preconditioning (SI, Algorithm 4), both of which transform the original problem (1) into sequences of least squares problems that need only be solved approximately. We instantiate the meta algorithms with state-of-the-art SGD methods and obtain efficient stochastic optimization algorithms for CCA.

In order to measure the alignments between an approximate solution $(\mathbf{u}, \mathbf{v})$ and the optimum $(\mathbf{u}^*, \mathbf{v}^*)$, we assume that $\mathbf{T}$ has a positive singular value gap $\Delta := \rho_1 - \rho_2 \in (0, 1]$ so its top left and right singular vector pair is unique (up to a change of sign).

Table 1 summarizes the time complexities of several algorithms for achieving $\eta$-suboptimal alignments, where $\widetilde{\kappa} = \frac{\max_i \max\left(\|\mathbf{x}_i\|^2, \|\mathbf{y}_i\|^2\right)}{\min(\sigma_{\min}(\mathbf{\Sigma}_{xx}), \sigma_{\min}(\mathbf{\Sigma}_{yy}))}$ is the upper bound of condition numbers of least squares problems solved in all cases.[2] We use the notation $\widetilde{\mathcal{O}}(\cdot)$ to hide poly-logarithmic dependencies (see Sec. 3.1.1 and Sec. 3.2.3 for the hidden factors). Each time complexity may be preferrable in certain regime depending on the parameters of the problem.

[1] We assume that $\mathbf{X}$ and $\mathbf{Y}$ are centered at the origin for notational simplicity; if they are not, we can center them as a pre-processing operation.

[2] For the ALS meta-algorithm, its enough to consider a per-view conditioning. And when using AGD as the least squares solver, the time complexities dependends on $\sigma_{\max}(\mathbf{\Sigma}_{xx})$ instead, which is less than $\max_i \|x_i\|^2$.

**Table 1.** Time complexities of different algorithms for achieving $\eta$-suboptimal solution $(\mathbf{u}, \mathbf{v})$ to CCA, i.e., $\min\left((\mathbf{u}^\top \mathbf{\Sigma}_{xx}\mathbf{u}^*)^2,\ (\mathbf{v}^\top \mathbf{\Sigma}_{yy}\mathbf{v}^*)^2\right) \geq 1-\eta$. GD=gradient descent, AGD=accelerated GD, SVRG=stochastic variance reduced gradient, ASVRG=accelerated SVRG. Note ASVRG provides speedup over SVRG only when $\widetilde{\kappa} > N$, and we show the dominant term in its complexity.

| Algorithm | Least squares solver | Time complexity |
|---|---|---|
| `AppGrad` [Ma et al., 2015] | GD | $\widetilde{\mathcal{O}}\left(dN\widetilde{\kappa}\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\cdot\log\left(\frac{1}{\eta}\right)\right)$ (local) |
| `CCALin` [Ge et al., 2016] | AGD | $\widetilde{\mathcal{O}}\left(dN\sqrt{\widetilde{\kappa}}\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\cdot\log\left(\frac{1}{\eta}\right)\right)$ |
| This work: Alternating least squares (ALS) | AGD | $\widetilde{\mathcal{O}}\left(dN\sqrt{\widetilde{\kappa}}\left(\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\right)^2\cdot\log^2\left(\frac{1}{\eta}\right)\right)$ |
| | SVRG | $\widetilde{\mathcal{O}}\left(d(N+\widetilde{\kappa})\left(\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\right)^2\cdot\log^2\left(\frac{1}{\eta}\right)\right)$ |
| | ASVRG | $\widetilde{\mathcal{O}}\left(d\sqrt{N\widetilde{\kappa}}\left(\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\right)^2\cdot\log^2\left(\frac{1}{\eta}\right)\right)$ |
| This work: Shift-and-invert preconditioning (SI) | AGD | $\widetilde{\mathcal{O}}\left(dN\sqrt{\widetilde{\kappa}}\sqrt{\frac{1}{\rho_1-\rho_2}}\cdot\log^2\left(\frac{1}{\eta}\right)\right)$ |
| | SVRG | $\widetilde{\mathcal{O}}\left(d\left(N+(\widetilde{\kappa}\frac{1}{\rho_1-\rho_2})^2\right)\cdot\log^2\left(\frac{1}{\eta}\right)\right)$ |
| | ASVRG | $\widetilde{\mathcal{O}}\left(dN^{\frac{3}{4}}\sqrt{\widetilde{\kappa}}\sqrt{\frac{1}{\rho_1-\rho_2}}\cdot\log^2\left(\frac{1}{\eta}\right)\right)$ |

**Contributions** The key contributions of this paper are as follows:

- We develop two novel stochastic optimization algorithms for CCA that ensure global convergence.
- We provide a rigorous theoretical analysis of the time complexities and convergence rates of both algorithms.
- We present empirical evidence demonstrating the algorithms' superior performance on real-world datasets.

**Organization** The remainder of this paper is structured as follows. In Section 2, we provide a review of related work on CCA and stochastic optimization methods. In Section 3, we introduce the proposed ALS and SI algorithms, providing theoretical analysis and proofs of global convergence. Section 4 presents experimental results on various datasets. Finally, Section 5 concludes the paper and outlines future research directions.

**Notations** We use $\sigma_i(\mathbf{A})$ to denote the $i$-th largest singular value of a matrix $\mathbf{A}$, and use $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$ to denote the largest and smallest singular values of $\mathbf{A}$ respectively.

## 2 Motivation: Alternating least squares

Our solution to (1) is inspired by the alternating least squares (ALS) formulation of CCA [Golub & Zha, 1995, Algorithm 5.2], as shown in Algorithm 1. Let the nonzero singular values of $\mathbf{T}$ be $1 \geq \rho_1 \geq \rho_2 \geq \cdots \geq \rho_r > 0$, where $r = \mathrm{rank}(\mathbf{T}) \leq \min(d_x, d_y)$, and the corresponding (unit-length) left and right singular vector pairs be $(\mathbf{a}_1, \mathbf{b}_1), \ldots, (\mathbf{a}_r, \mathbf{b}_r)$, with $\mathbf{a}_1 = \boldsymbol{\phi}$ and $\mathbf{b}_1 = \boldsymbol{\psi}$. Define

$$\mathbf{C} = \begin{bmatrix} \mathbf{0} & \mathbf{T} \\ \mathbf{T}^\top & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{d\times d} \tag{3}$$

**Algorithm 1** Alternating least squares for CCA.

**Input:** Data matrices $\mathbf{X} \in \mathbb{R}^{d_x \times N}$, $\mathbf{Y} \in \mathbb{R}^{d_y \times N}$, regularization parameters $(\gamma_x, \gamma_y)$.

Initialize $\widetilde{\mathbf{u}}_0 \in \mathbb{R}^{d_x}$, $\widetilde{\mathbf{v}}_0 \in \mathbb{R}^{d_y}$. $\quad \{\widetilde{\boldsymbol{\phi}}_0,\ \widetilde{\boldsymbol{\psi}}_0\}$

$\mathbf{u}_0 \leftarrow \widetilde{\mathbf{u}}_0 / \sqrt{\widetilde{\mathbf{u}}_0^\top \boldsymbol{\Sigma}_{xx} \widetilde{\mathbf{u}}_0}, \quad \mathbf{v}_0 \leftarrow \widetilde{\mathbf{v}}_0 / \sqrt{\widetilde{\mathbf{v}}_0^\top \boldsymbol{\Sigma}_{yy} \widetilde{\mathbf{v}}_0}$ $\quad \{\boldsymbol{\phi}_0 \leftarrow \widetilde{\boldsymbol{\phi}}_0 / \left\|\widetilde{\boldsymbol{\phi}}_0\right\|,\ \boldsymbol{\psi}_0 \leftarrow \widetilde{\boldsymbol{\psi}}_0 / \left\|\widetilde{\boldsymbol{\psi}}_0\right\|\}$

**for** $t = 1, 2, \ldots, T$ **do**

$\widetilde{\mathbf{u}}_t \leftarrow \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{xy} \mathbf{v}_{t-1}$ $\quad \{\widetilde{\boldsymbol{\phi}}_t \leftarrow \boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}} \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-\frac{1}{2}} \boldsymbol{\psi}_{t-1}\}$

$\widetilde{\mathbf{v}}_t \leftarrow \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Sigma}_{xy}^\top \mathbf{u}_{t-1}$ $\quad \{\widetilde{\boldsymbol{\psi}}_t \leftarrow \boldsymbol{\Sigma}_{yy}^{-\frac{1}{2}} \boldsymbol{\Sigma}_{xy}^\top \boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}} \boldsymbol{\phi}_{t-1}\}$

$\mathbf{u}_t \leftarrow \widetilde{\mathbf{u}}_t / \sqrt{\widetilde{\mathbf{u}}_t^\top \boldsymbol{\Sigma}_{xx} \widetilde{\mathbf{u}}_t}, \quad \mathbf{v}_t \leftarrow \widetilde{\mathbf{v}}_t / \sqrt{\widetilde{\mathbf{v}}_t^\top \boldsymbol{\Sigma}_{yy} \widetilde{\mathbf{v}}_t}$ $\quad \{\boldsymbol{\phi}_t \leftarrow \widetilde{\boldsymbol{\phi}}_t / \left\|\widetilde{\boldsymbol{\phi}}_t\right\|,\ \boldsymbol{\psi}_t \leftarrow \widetilde{\boldsymbol{\psi}}_t / \left\|\widetilde{\boldsymbol{\psi}}_t\right\|\}$

**end for**

**Output:** $(\mathbf{u}_T, \mathbf{v}_T) \to (\mathbf{u}^*, \mathbf{v}^*)$ as $T \to \infty$. $\quad \{(\boldsymbol{\phi}_T, \boldsymbol{\psi}_T) \to (\boldsymbol{\phi}, \boldsymbol{\psi})\}$

It is straightforward to check that the nonzero eigenvalues of $\mathbf{C}$ are:

$$\rho_1 \geq \cdots \geq \rho_r \geq -\rho_r \geq \cdots \geq -\rho_1$$

with corresponding eigenvectors $\frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{b}_1 \end{bmatrix}, \ \ldots, \ \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{a}_r \\ \mathbf{b}_r \end{bmatrix}, \ \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{a}_r \\ -\mathbf{b}_r \end{bmatrix}, \ \ldots, \ \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{a}_1 \\ -\mathbf{b}_1 \end{bmatrix}$.

The key observation is that Algorithm 1 effectively runs a variant of power iterations on $\mathbf{C}$ to extract its top eigenvector. To see this, make the following change of variables

$$\boldsymbol{\phi}_t = \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \mathbf{u}_t, \qquad \boldsymbol{\psi}_t = \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}} \mathbf{v}_t, \qquad \widetilde{\boldsymbol{\phi}}_t = \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \widetilde{\mathbf{u}}_t, \qquad \widetilde{\boldsymbol{\psi}}_t = \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}} \widetilde{\mathbf{v}}_t \tag{4}$$

Then we can equivalently rewrite the steps of Algorithm 1 in the new variables as in $\{\}$ of each line.

Observe that the iterates are updated as follows from step $t-1$ to step $t$:

$$\begin{bmatrix} \widetilde{\boldsymbol{\phi}}_t \\ \widetilde{\boldsymbol{\psi}}_t \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{0} & \mathbf{T} \\ \mathbf{T}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_{t-1} \\ \boldsymbol{\psi}_{t-1} \end{bmatrix}, \qquad \begin{bmatrix} \boldsymbol{\phi}_t \\ \boldsymbol{\psi}_t \end{bmatrix} \leftarrow \begin{bmatrix} \widetilde{\boldsymbol{\phi}}_t / ||\widetilde{\boldsymbol{\phi}}_t|| \\ \widetilde{\boldsymbol{\psi}}_t / ||\widetilde{\boldsymbol{\psi}}_t|| \end{bmatrix} \tag{5}$$

Except for the special normalization steps which rescale the two sets of variables separately, Algorithm 1 is very similar to the power iterations [Golub & van Loan, 1996].

We show the convergence rate of ALS below (see its proof in Appendix A). The first measure of progress is the alignment of $\boldsymbol{\phi}_t$ to $\boldsymbol{\phi}$ and the alignment of $\boldsymbol{\psi}_t$ to $\boldsymbol{\psi}$, i.e., $(\boldsymbol{\phi}_t^\top \boldsymbol{\phi})^2 = (\mathbf{u}_t^\top \boldsymbol{\Sigma}_{xx} \mathbf{u}^*)^2$ and $(\boldsymbol{\psi}_t^\top \boldsymbol{\psi})^2 = (\mathbf{v}_t^\top \boldsymbol{\Sigma}_{yy} \mathbf{v}^*)^2$. The maximum value for such alignments is 1, achieved when the iterates completely align with the optimal solution. The second natural measure of progress is the objective of (1), i.e., $\mathbf{u}_t^\top \boldsymbol{\Sigma}_{xy} \mathbf{v}_t$, with the maximum value being $\rho_1$.

**Theorem 1** (Convergence of Algorithm 1)**.** *Let* $\mu := \min\left((\mathbf{u}_0^\top \boldsymbol{\Sigma}_{xx} \mathbf{u}^*)^2,\ (\mathbf{v}_0^\top \boldsymbol{\Sigma}_{yy} \mathbf{v}^*)^2\right) > 0$.[3] *Then for* $t \geq \lceil \frac{\rho_1^2}{\rho_1^2 - \rho_2^2} \log\left(\frac{1}{\mu\eta}\right) \rceil$, *we have in Algorithm 1 that* $\min\left((\mathbf{u}_t^\top \boldsymbol{\Sigma}_{xx} \mathbf{u}^*)^2,\ (\mathbf{v}_t^\top \boldsymbol{\Sigma}_{yy} \mathbf{v}^*)^2\right) \geq 1 - \eta$, *and* $\mathbf{u}_t^\top \boldsymbol{\Sigma}_{xy} \mathbf{v}_t \geq \rho_1(1 - 2\eta)$.

[3] One can show that $\mu$ is bounded away from 0 with high probability using random initialization $(\mathbf{u}_0, \mathbf{v}_0)$.

**Algorithm 2** The alternating least squares (ALS) meta-algorithm for CCA.

**Input:** Data matrices $\mathbf{X} \in \mathbb{R}^{d_x \times N}$, $\mathbf{Y} \in \mathbb{R}^{d_y \times N}$, regularization parameters $(\gamma_x, \gamma_y)$.

Initialize $\widetilde{\mathbf{u}}_0 \in \mathbb{R}^{d_x}$, $\widetilde{\mathbf{v}}_0 \in \mathbb{R}^{d_y}$.

$\widetilde{\mathbf{u}}_0 \leftarrow \widetilde{\mathbf{u}}_0 / \sqrt{\widetilde{\mathbf{u}}_0^\top \mathbf{\Sigma}_{xx} \widetilde{\mathbf{u}}_0}$, $\widetilde{\mathbf{v}}_0 \leftarrow \widetilde{\mathbf{v}}_0 / \sqrt{\widetilde{\mathbf{v}}_0^\top \mathbf{\Sigma}_{yy} \widetilde{\mathbf{v}}_0}$, $\mathbf{u}_0 \leftarrow \widetilde{\mathbf{u}}_0$, $\mathbf{v}_0 \leftarrow \widetilde{\mathbf{v}}_0$

**for** $t = 1, 2, \ldots, T$ **do**

Solve $\min_{\mathbf{u}} \; f_t(\mathbf{u}) := \frac{1}{2N} \left\| \mathbf{u}^\top \mathbf{X} - \mathbf{v}_{t-1}^\top \mathbf{Y} \right\|^2 + \frac{\gamma_x}{2} \|\mathbf{u}\|^2$ with initialization $\widetilde{\mathbf{u}}_{t-1}$, and output approximate solution $\widetilde{\mathbf{u}}_t$ satisfying $f_t(\widetilde{\mathbf{u}}_t) \leq \min_{\mathbf{u}} \; f_t(\mathbf{u}) + \epsilon$.

Solve $\min_{\mathbf{v}} \; g_t(\mathbf{v}) := \frac{1}{2N} \left\| \mathbf{v}^\top \mathbf{Y} - \mathbf{u}_{t-1}^\top \mathbf{X} \right\|^2 + \frac{\gamma_y}{2} \|\mathbf{v}\|^2$ with initialization $\widetilde{\mathbf{v}}_{t-1}$, and output approximate solution $\widetilde{\mathbf{v}}_t$ satisfying $g_t(\widetilde{\mathbf{v}}_t) \leq \min_{\mathbf{v}} \; g_t(\mathbf{v}) + \epsilon$.

$\mathbf{u}_t \leftarrow \widetilde{\mathbf{u}}_t / \sqrt{\widetilde{\mathbf{u}}_t^\top \mathbf{\Sigma}_{xx} \widetilde{\mathbf{u}}_t}$, $\mathbf{v}_t \leftarrow \widetilde{\mathbf{v}}_t / \sqrt{\widetilde{\mathbf{v}}_t^\top \mathbf{\Sigma}_{yy} \widetilde{\mathbf{v}}_t}$

**end for**

**Output:** $(\mathbf{u}_T, \mathbf{v}_T)$ is the approximate solution to CCA.

**Remarks** We have assumed a nonzero singular value gap in Theorem 1 to obtain linear convergence in both the alignments and the objective. When there exists no singular value gap, the top singular vector pair is not unique and it is no longer meaningful to measure the alignments. Nonetheless, it is possible to extend our proof to obtain sublinear convergence for the objective in this case.

Observe that, besides the steps of normalization to unit length, the basic operation in each iteration of Algorithm 1 is of the form $\widetilde{\mathbf{u}}_t \leftarrow \mathbf{\Sigma}_{xx}^{-1} \mathbf{\Sigma}_{xy} \mathbf{v}_{t-1} = (\frac{1}{N} \mathbf{X}\mathbf{X}^\top + \gamma_x \mathbf{I})^{-1} \frac{1}{N} \mathbf{X}\mathbf{Y}^\top \mathbf{v}_{t-1}$, which is equivalent to solving the following regularized least squares (ridge regression) problem

$$\min_{\mathbf{u}} \; \frac{1}{2N} \left\| \mathbf{u}^\top \mathbf{X} - \mathbf{v}_{t-1}^\top \mathbf{Y} \right\|^2 + \frac{\gamma_x}{2} \|\mathbf{u}\|^2 \equiv \min_{\mathbf{u}} \; \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \left| \mathbf{u}^\top \mathbf{x}_i - \mathbf{v}_{t-1}^\top \mathbf{y}_i \right|^2 + \frac{\gamma_x}{2} \|\mathbf{u}\|^2 \tag{6}$$

In the next section, we show that, to maintain the convergence of ALS, it is unnecessary to solve the least squares problems exactly. This enables us to use state-of-the-art SGD methods for solving (6) to sufficient accuracy, and to obtain a globally convergent stochastic algorithm for CCA.

# 3 Our algorithms

## 3.1 Algorithm I: Alternating least squares (ALS) with variance reduction

Our first algorithm consists of two nested loops. The outer loop runs inexact power iterations while the inner loop uses advanced stochastic optimization methods, e.g., stochastic variance reduced gradient (SVRG, [Johnson & Zhang, 2013]) to obtain approximate matrix-vector multiplications. A sketch of our algorithm is provided in Algorithm 2. We make the following observations from this algorithm.

$$\widetilde{\mathbf{u}}_t \leftarrow \widetilde{\mathbf{u}}_{t-1} - 2\xi \, \mathbf{X}(\mathbf{X}^\top \widetilde{\mathbf{u}}_{t-1} - \mathbf{Y}^\top \mathbf{v}_{t-1})/N, \qquad \mathbf{u}_t \leftarrow \widetilde{\mathbf{u}}_t / \sqrt{\widetilde{\mathbf{u}}_t^\top \mathbf{\Sigma}_{xx} \widetilde{\mathbf{u}}_t}$$

$$\widetilde{\mathbf{v}}_t \leftarrow \widetilde{\mathbf{v}}_{t-1} - 2\xi \, \mathbf{Y}(\mathbf{Y}^\top \widetilde{\mathbf{v}}_{t-1} - \mathbf{X}^\top \mathbf{u}_{t-1})/N, \qquad \mathbf{v}_t \leftarrow \widetilde{\mathbf{v}}_t / \sqrt{\widetilde{\mathbf{v}}_t^\top \mathbf{\Sigma}_{yy} \widetilde{\mathbf{v}}_t}$$

where $\xi > 0$ is the stepsize (assuming $\gamma_x = \gamma_y = 0$). This coincides with the `AppGrad` algorithm of [Ma et al., 2015, Algorithm 3], for which only local convergence is shown. Since the objectives $f_t(\mathbf{u})$ and $g_t(\mathbf{v})$ decouple over training samples, it is convenient to apply SGD methods to them. This observation motivated the stochastic CCA algorithms of [Ma et al., 2015, Wang et al., 2015]. We note however, no global convergence guarantee was shown for these stochastic CCA algorithms, and the key to our convergent algorithm is to solve the least squares problems to *sufficient* accuracy.

**Warm-start** Observe that for different $t$, the least squares problems $f_t(\mathbf{u})$ only differ in their targets as $\mathbf{v}_t$ changes over time. Since $\mathbf{v}_{t-1}$ is close to $\mathbf{v}_t$ (especially when near convergence), we may use $\widetilde{\mathbf{u}}_t$ as initialization for minimizing $f_{t+1}(\mathbf{u})$ with an iterative algorithm.

**Normalization** At the end of each outer loop, Algorithm 2 implements exact normalization of the form $\mathbf{u}_t \leftarrow \widetilde{\mathbf{u}}_t / \sqrt{\widetilde{\mathbf{u}}_t^\top \mathbf{\Sigma}_{xx} \widetilde{\mathbf{u}}_t}$ to ensure the constraints, where $\widetilde{\mathbf{u}}_t^\top \mathbf{\Sigma}_{xx} \widetilde{\mathbf{u}}_t = \frac{1}{N}(\widetilde{\mathbf{u}}_t^\top \mathbf{X})(\widetilde{\mathbf{u}}_t^\top \mathbf{X})^\top + \gamma_x \left\|\widetilde{\mathbf{u}}_t\right\|^2$ requires computing the projection of the training set $\widetilde{\mathbf{u}}_t^\top \mathbf{X}$. However, this does not introduce extra computation because we also compute this projection for the batch gradient used by SVRG (at the beginning of time step $t+1$). In contrast, the stochastic algorithms of [Ma et al., 2015, Wang et al., 2015] (possibly adaptively) estimate the covariance matrix from a minibatch of training samples and use the estimated covariance for normalization. This is because their algorithms perform normalizations after each update and thus need to avoid computing the projection of the entire training set frequently. But as a result, their inexact normalization steps introduce noise to the algorithms.

**Input sparsity** For high dimensional sparse data (such as those used in natural language processing [Lu & Foster, 2014]), an advantage of gradient based methods over the closed-form solution is that the former takes into account the input sparsity. For sparse inputs, the time complexity of our algorithm depends on $nnz(\mathbf{X}, \mathbf{Y})$, i.e., the total number of nonzeros in the inputs instead of $dN$.

**Canonical ridge** When $(\gamma_x, \gamma_y) > 0$, $f_t(\mathbf{u})$ and $g_t(\mathbf{v})$ are guaranteed to be strongly convex due to the $\ell_2$ regularizations, in which case SVRG converges linearly. It is therefore beneficial to use small nonzero regularization for improved computational efficiency, especially for high dimensional datasets where inputs $\mathbf{X}$ and $\mathbf{Y}$ are approximately low-rank.

**Theorem 2** (Convergence of Algorithm 2)**.** *Fix* $T \geq \lceil \frac{\rho_1^2}{\rho_1^2 - \rho_2^2} \log\left(\frac{2}{\mu\eta}\right) \rceil$*, and set* $\epsilon(T) \leq \frac{\eta^2 \rho_r^2}{128} \cdot \left(\frac{(2\rho_1/\rho_r)-1}{(2\rho_1/\rho_r)^T - 1}\right)^2$ *in Algorithm 2. Then we have* $\mathbf{u}_T^\top \mathbf{\Sigma}_{xx} \mathbf{u}_T = \mathbf{v}_T^\top \mathbf{\Sigma}_{yy} \mathbf{v}_T = 1$, $\min\left((\mathbf{u}_T^\top \mathbf{\Sigma}_{xx} \mathbf{u}^*)^2,\ (\mathbf{v}_T^\top \mathbf{\Sigma}_{yy} \mathbf{v}^*)^2\right) \geq 1 - \eta$, *and* $\mathbf{u}_T^\top \mathbf{\Sigma}_{xy} \mathbf{v}_T \geq \rho_1(1 - 2\eta)$.

### 3.1.1 Stochastic optimization of regularized least squares

We now discuss the inner loop of Algorithm 2, which approximately solves problems of the form (6). Owing to the finite-sum structure of (6), several stochastic optimization methods such as SAG [Schmidt et al., 2013] SDCA [Shalev-Shwartz & Zhang, 2013] and SVRG [Johnson & Zhang, 2013], provide linear convergence rates. All these algorithms can be readily applied to (6); we choose SVRG since it is memory efficient and easy to implement. We also apply the recently developed accelerations techniques for first order optimization methods [Frostig et al., 2015, Lin et al., 2015] to obtain an accelerated SVRG (ASVRG) algorithm. We give the sketch of SVRG for (6) in Appendix C.

Note that $f(\mathbf{u}) = \frac{1}{N}\sum_{i=1}^{N} f^i(\mathbf{u})$ where each component $f^i(\mathbf{u}) = \frac{1}{2}\left|\mathbf{u}^\top \mathbf{x}_i - \mathbf{v}^\top \mathbf{y}_i\right|^2 + \frac{\gamma_x}{2}\|\mathbf{u}\|^2$ is $\|\mathbf{x}_i\|^2$-smooth, and $f(\mathbf{u})$ is $\sigma_{\min}(\mathbf{\Sigma}_{xx})$-strongly convex[4] with $\sigma_{\min}(\mathbf{\Sigma}_{xx}) \geq \gamma_x$. We show in Appendix D that the initial suboptimality for minimizing $f_t(\mathbf{u})$ is upper-bounded by constant when using the warm-starts. We quote the convergence rates of SVRG [Johnson & Zhang, 2013] and ASVRG [Lin et al., 2015] below.

**Lemma 1.** *The SVRG algorithm [Johnson & Zhang, 2013] finds a vector $\widetilde{\mathbf{u}}$ satisfying[5] $\mathbb{E}[f(\widetilde{\mathbf{u}})] - \min_{\mathbf{u}} f(\mathbf{u}) \leq \epsilon$ in time $\mathcal{O}\left(d_x\left(N+\kappa_x\right)\log\left(\frac{1}{\epsilon}\right)\right)$ where $\kappa_x = \frac{\max_i\|\mathbf{x}_i\|^2}{\sigma_{\min}(\mathbf{\Sigma}_{xx})}$. The ASVRG algorithm [Lin et al., 2015] finds a such solution in time $\mathcal{O}\left(d_x\sqrt{N\kappa_x}\log\left(\frac{1}{\epsilon}\right)\right)$.*

**Remarks** As mentioned in [Lin et al., 2015], the acceleration version provides speedup over normal SVRG only when $\kappa_x > N$ and we only show the dominant term in the above complexity.

By combining the iteration complexity of the outer loop (Theorem 2) and the time complexity of the inner loop (Lemma 1), we obtain the total time complexity of $\widetilde{\mathcal{O}}\left(d\left(N+\kappa\right)\left(\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\right)^2\cdot\log^2\left(\frac{1}{\eta}\right)\right)$ for ALS+SVRG and $\widetilde{\mathcal{O}}\left(d\sqrt{N\kappa}\left(\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\right)^2\cdot\log^2\left(\frac{1}{\eta}\right)\right)$ for ALS+ASVRG, where $\kappa := \max\left(\frac{\max_i\|\mathbf{x}_i\|^2}{\sigma_{\min}(\mathbf{\Sigma}_{xx})}, \frac{\max_i\|\mathbf{y}_i\|^2}{\sigma_{\min}(\mathbf{\Sigma}_{yy})}\right)$ and $\widetilde{\mathcal{O}}(\cdot)$ hides poly-logarithmic dependences on $\frac{1}{\mu}$ and $\frac{1}{\rho_r}$. Our algorithm does not require the initialization to be close to the optimum and converges globally. For comparison, the locally convergent `AppGrad` has a time complexity [Ma et al., 2015, Theorem 2.1] of $\widetilde{\mathcal{O}}\left(dN\kappa'\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\cdot\log\left(\frac{1}{\eta}\right)\right)$, where $\kappa' := \max\left(\frac{\sigma_{\max}(\mathbf{\Sigma}_{xx})}{\sigma_{\min}(\mathbf{\Sigma}_{xx})}, \frac{\sigma_{\max}(\mathbf{\Sigma}_{yy})}{\sigma_{\min}(\mathbf{\Sigma}_{yy})}\right)$. Note, in this complexity, the dataset size $N$ and the least squares condition number $\kappa'$ are multiplied together because `AppGrad` essentially uses batch gradient descent as the least squares solver. Within our framework, we can use accelerated gradient descent (AGD, [Nesterov, 2004]) instead and obtain a globally convergent algorithm with a total time complexity of $\widetilde{\mathcal{O}}\left(dN\sqrt{\kappa'}\left(\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\right)^2\cdot\log^2\left(\frac{1}{\eta}\right)\right)$.

### 3.2 Algorithm II: Shift-and-invert preconditioning (SI) with variance reduction

The second algorithm is inspired by the shift-and-invert preconditioning method for PCA [Garber & Hazan, 2015, Jin et al., 2015]. Instead of running power iterations on $\mathbf{C}$ as defined in (3), we will be running power iterations on

$$\mathbf{M}_\lambda = (\lambda\mathbf{I}-\mathbf{C})^{-1} = \begin{bmatrix} \lambda\mathbf{I} & -\mathbf{T} \\ -\mathbf{T}^\top & \lambda\mathbf{I} \end{bmatrix}^{-1} \in \mathbb{R}^{d\times d} \tag{7}$$

where $\lambda > \rho_1$. It is straightforward to check that $\mathbf{M}_\lambda$ is positive definite and its eigenvalues are:

$$\frac{1}{\lambda-\rho_1} \geq \cdots \geq \frac{1}{\lambda-\rho_r} \geq \cdots \geq \frac{1}{\lambda+\rho_r} \geq \cdots \geq \frac{1}{\lambda+\rho_1}$$

with eigenvectors $\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{a}_1\\ \mathbf{b}_1\end{bmatrix}, \ldots, \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{a}_r\\ \mathbf{b}_r\end{bmatrix}, \ldots, \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{a}_r\\ -\mathbf{b}_r\end{bmatrix}, \ldots, \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{a}_1\\ -\mathbf{b}_1\end{bmatrix}$.

The main idea behind shift-and-invert power iterations is that when $\lambda-\rho_1 = c(\rho_1-\rho_2)$ with $c \sim \mathcal{O}(1)$, the relative eigenvalue gap of $\mathbf{M}_\lambda$ is large and so power iterations on $\mathbf{M}_\lambda$ converges quickly. Our shift-and-invert preconditioning (SI) meta-algorithm for CCA is sketched in Algorithm 4 (in Appendix D due to space limit) and it proceeds in two phases.

[4] We omit the regularization in these constants, which are typically very small, to have concise expressions.

[5] The expectation is taken over random sampling of component functions. High probability error bounds can be obtained using the Markov's inequality.

### 3.2.1 Phase I: shift-and-invert preconditioning for eigenvectors of $\mathbf{M}_\lambda$

Using an estimate of the singular value gap $\widetilde{\Delta}$ and starting from an over-estimate of $\rho_1$ ($1+\widetilde{\Delta}$ suffices), the algorithm gradually shrinks $\lambda_{(s)}$ towards $\rho_1$ by crudely estimating the leading eigenvector/eigenvalues of each $\mathbf{M}_{\lambda_{(s)}}$ along the way and shrinking the gap $\lambda_{(s)}-\rho_1$, until we reach a $\lambda_{(f)} \in (\rho_1, \rho_1 + c(\rho_1-\rho_2))$ where $c \sim \mathcal{O}(1)$. Afterwards, the algorithm fixes $\lambda_{(f)}$ and runs inexact power iterations on $\mathbf{M}_{\lambda_{(f)}}$ to obtain an accurate estimate of its leading eigenvector. Note in this phase, power iterations implicitly operate on the concatenated variables $\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_t \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_t \end{bmatrix}$ and $\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_t \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_t \end{bmatrix}$ in $\mathbb{R}^d$ (but without ever computing $\mathbf{\Sigma}_{xx}^{\frac{1}{2}}$ and $\mathbf{\Sigma}_{yy}^{\frac{1}{2}}$).

**Matrix-vector multiplication** The matrix-vector multiplications in Phase I have the form

$$\begin{bmatrix} \widetilde{\mathbf{u}}_t \\ \widetilde{\mathbf{v}}_t \end{bmatrix} \leftarrow \begin{bmatrix} \lambda\mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda\mathbf{\Sigma}_{yy} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Sigma}_{xx} & \\ & \mathbf{\Sigma}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{t-1} \\ \mathbf{v}_{t-1} \end{bmatrix} \tag{8}$$

where $\lambda$ varies over time in order to locate $\lambda_{(f)}$. This is equivalent to solving

$$\begin{bmatrix} \widetilde{\mathbf{u}}_t \\ \widetilde{\mathbf{v}}_t \end{bmatrix} \leftarrow \min_{\mathbf{u},\mathbf{v}} \frac{1}{2}\begin{bmatrix}\mathbf{u}^\top \mathbf{v}^\top\end{bmatrix} \begin{bmatrix} \lambda\mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda\mathbf{\Sigma}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} - \mathbf{u}^\top\mathbf{\Sigma}_{xx}\mathbf{u}_{t-1} - \mathbf{v}^\top\mathbf{\Sigma}_{yy}\mathbf{v}_{t-1}$$

And as in ALS, this least squares problem can be further written as finite-sum:

$$\min_{\mathbf{u},\mathbf{v}} \quad h_t(\mathbf{u},\mathbf{v}) = \frac{1}{N}\sum_{i=1}^N h_t^i(\mathbf{u},\mathbf{v}) \qquad \text{where} \tag{9}$$

$$h_t^i(\mathbf{u},\mathbf{v}) = \frac{1}{2}\begin{bmatrix}\mathbf{u}^\top \mathbf{v}^\top\end{bmatrix} \begin{bmatrix} \lambda\left(\mathbf{x}_i\mathbf{x}_i^\top + \gamma_x\mathbf{I}\right) & -\mathbf{x}_i\mathbf{y}_i^\top \\ -\mathbf{y}_i\mathbf{x}_i^\top & \lambda\left(\mathbf{y}_i\mathbf{y}_i^\top + \gamma_y\mathbf{I}\right) \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} - \mathbf{u}^\top\mathbf{\Sigma}_{xx}\mathbf{u}_{t-1} - \mathbf{v}^\top\mathbf{\Sigma}_{yy}\mathbf{v}_{t-1}.$$

We could directly apply SGD methods to this problem as before.

**Normalization** The normalization steps in Phase I have the form

$$\begin{bmatrix} \mathbf{u}_t \\ \mathbf{v}_t \end{bmatrix} \leftarrow \sqrt{2}\begin{bmatrix} \widetilde{\mathbf{u}}_t \\ \widetilde{\mathbf{v}}_t \end{bmatrix} \Big/ \sqrt{\widetilde{\mathbf{u}}_t^\top\mathbf{\Sigma}_{xx}\widetilde{\mathbf{u}}_t + \widetilde{\mathbf{v}}_t^\top\mathbf{\Sigma}_{yy}\widetilde{\mathbf{v}}_t}$$

and so the following remains true for the normalized iterates in Phase I:

$$\mathbf{u}_t^\top\mathbf{\Sigma}_{xx}\mathbf{u}_t + \mathbf{v}_t^\top\mathbf{\Sigma}_{yy}\mathbf{v}_t = 2, \qquad \text{for} \quad t = 1,\ldots,T \tag{10}$$

Unlike the normalizations in ALS, the iterates $\mathbf{u}_t$ and $\mathbf{v}_t$ in Phase I do *not* satisfy the original CCA constraints, and this is taken care of in Phase II.

We have the following convergence guarantee for Phase I (see its proof in Appendix E).

**Theorem 3** (Convergence of Algorithm 4, Phase I)**.** *Let* $\Delta = \rho_1 - \rho_2 \in (0,1]$, *and* $\widetilde{\mu} := \frac{1}{4}\left(\mathbf{u}_0^\top\mathbf{\Sigma}_{xx}\mathbf{u}^* + \mathbf{v}_0^\top\mathbf{\Sigma}_{yy}\mathbf{v}^*\right)^2 > 0$, *and* $\widetilde{\Delta} \in [c_1\Delta, c_2\Delta]$ *where* $0 < c_1 \le c_2 \le 1$. *Set* $m_1 = \lceil 8\log\left(\frac{16}{\widetilde{\mu}}\right)\rceil$, $m_2 = \lceil \frac{5}{4}\log\left(\frac{128}{\widetilde{\mu}\eta^2}\right)\rceil$, *and* $\widetilde{\epsilon} \le \min\left(\frac{1}{3084}\left(\frac{\widetilde{\Delta}}{18}\right)^{m_1-1}, \frac{\eta^4}{4^{10}}\left(\frac{\widetilde{\Delta}}{18}\right)^{m_2-1}\right)$ *in Algorithm 4. Then the* $(\mathbf{u}_T, \mathbf{v}_T)$ *output by Phase I of Algorithm 4 satisfies* (10) *and*

$$\frac{1}{4}(\mathbf{u}_T^\top\mathbf{\Sigma}_{xx}\mathbf{u}^* + \mathbf{v}_T^\top\mathbf{\Sigma}_{yy}\mathbf{v}^*)^2 \ge 1 - \frac{\eta^2}{64} \tag{11}$$

*and the number of calls to the least squares solver of* $h_t(\mathbf{u},\mathbf{v})$ *is* $\mathcal{O}\left(\log\left(\frac{1}{\widetilde{\mu}}\right)\log\left(\frac{1}{\Delta}\right) + \log\left(\frac{1}{\widetilde{\mu}\eta^2}\right)\right)$.

### 3.2.2 Phase II: final normalization

In order to satisfy the CCA constraints, we perform a last normalization

$$\widehat{\mathbf{u}} \leftarrow \mathbf{u}_T / \sqrt{\mathbf{u}_T^\top \mathbf{\Sigma}_{xx} \mathbf{u}_T}, \qquad \widehat{\mathbf{v}} \leftarrow \mathbf{v}_T / \sqrt{\mathbf{v}_T^\top \mathbf{\Sigma}_{yy} \mathbf{v}_T} \tag{12}$$

And we output $(\widehat{\mathbf{u}}, \widehat{\mathbf{v}})$ as our final approximate solution to (1). We show that this step does not cause much loss in the alignments, as stated below (see it proof in Appendix F).

**Theorem 4** (Convergence of Algorithm 4, Phase II)**.** *Let Phase I of Algorithm 4 outputs* $(\mathbf{u}_T, \mathbf{v}_T)$ *that satisfy* (11)*. Then after* (12)*, we obtain an approximate solution* $(\widehat{\mathbf{u}}, \widehat{\mathbf{v}})$ *to* (1) *such that* $\widehat{\mathbf{u}}^\top \mathbf{\Sigma}_{xx} \widehat{\mathbf{u}} = \widehat{\mathbf{v}}^\top \mathbf{\Sigma}_{yy} \widehat{\mathbf{v}} = 1$*,* $\min\left((\widehat{\mathbf{u}}^\top \mathbf{\Sigma}_{xx} \mathbf{u}^*)^2, (\widehat{\mathbf{v}}^\top \mathbf{\Sigma}_{yy} \mathbf{v}^*)^2\right) \geq 1 - \eta$*, and* $\widehat{\mathbf{u}}^\top \mathbf{\Sigma}_{xy} \widehat{\mathbf{v}} \geq \rho_1 (1 - 2\eta)$*.*

### 3.2.3 Time complexity

We have shown in Theorem 3 that Phase I only approximately solves a small number of instances of (9). The normalization steps (10) require computing the projections of the traning set which are reused for computing batch gradients of (9). The final normalization (12) is done only once and costs $\mathcal{O}(dN)$. Therefore, the time complexity of our algorithm mainly comes from solving the least squares problems (9) using SGD methods in a blackbox fashion. And the time complexity for SGD methods depends on the condition number of (9). Denote

$$\mathbf{Q}_\lambda = \begin{bmatrix} \lambda \mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda \mathbf{\Sigma}_{yy} \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \lambda \mathbf{I} & -\mathbf{T} \\ -\mathbf{T}^\top & \lambda \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \end{bmatrix} \tag{13}$$

It is clear that

$$\sigma_{\max}(\mathbf{Q}_\lambda) \leq (\lambda + \rho_1) \cdot \max\left(\sigma_{\max}(\mathbf{\Sigma}_{xx}), \sigma_{\max}(\mathbf{\Sigma}_{yy})\right),$$
$$\sigma_{\min}(\mathbf{Q}_\lambda) \geq (\lambda - \rho_1) \cdot \min\left(\sigma_{\min}(\mathbf{\Sigma}_{xx}), \sigma_{\min}(\mathbf{\Sigma}_{yy})\right).$$

We have shown in the proof of Theorem 3 that $\frac{\lambda+\rho_1}{\lambda-\rho_1} \leq \frac{9}{\widetilde{\Delta}} \leq \frac{9}{c_1 \Delta}$ throughout Algorithm 4 (cf. Lemma 6, Appendix E.2), and thus the condtion number for AGD is $\frac{\sigma_{\max}(\mathbf{Q}_\lambda)}{\sigma_{\min}(\mathbf{Q}_\lambda)} \leq \frac{9/c_1}{\rho_1 - \rho_2} \widetilde{\kappa}'$, where $\widetilde{\kappa}' := \frac{\max(\sigma_{\max}(\mathbf{\Sigma}_{xx}), \sigma_{\max}(\mathbf{\Sigma}_{yy}))}{\min(\sigma_{\min}(\mathbf{\Sigma}_{xx}), \sigma_{\min}(\mathbf{\Sigma}_{yy}))}$. For SVRG/ASVRG, the relevant condition number depends on the gradient Lipschitz constant of individual components. We show in Appendix G (Lemma 8) that the relevant condition number is at most $\frac{9/c_1}{\rho_1 - \rho_2} \widetilde{\kappa}$, where $\widetilde{\kappa} := \frac{\max_i \max\left(\|\mathbf{x}_i\|^2, \|\mathbf{y}_i\|^2\right)}{\min(\sigma_{\min}(\mathbf{\Sigma}_{xx}), \sigma_{\min}(\mathbf{\Sigma}_{yy}))}$. An interesting issue for SVRG/ASVRG is that, depending on the value of $\lambda$, the independent components $h_t^i(\mathbf{u}, \mathbf{v})$ may be nonconvex. If $\lambda \geq 1$, each component is still guaranteed to by convex; otherwise, some components might be non-convex, with the overall average $\frac{1}{N} \sum_{i=1}^N \mathbf{h}_t^i$ being convex. In the later case, we use the modified analysis of SVRG [Garber & Hazan, 2015, Appendix B] for its time complexity. We use warm-start in SI as in ALS, and the initial suboptimality for each subproblem can be bounded similarly.

The total time complexities of our SI meta-algorithm are given in Table 1. Note that $\widetilde{\kappa}$ (or $\widetilde{\kappa}'$) and $\frac{1}{\rho_1 - \rho_2}$ are multiplied together, giving the effective condition number. When using SVRG as the least squares solver, we obtain the total time complexity of $\widetilde{\mathcal{O}}\left(d(N + \widetilde{\kappa} \frac{1}{\rho_1 - \rho_2}) \cdot \log^2\left(\frac{1}{\eta}\right)\right)$ if all components are convex, and $\widetilde{\mathcal{O}}\left(d(N + (\widetilde{\kappa} \frac{1}{\rho_1 - \rho_2})^2) \cdot \log^2\left(\frac{1}{\eta}\right)\right)$ otherwise. When using ASVRG, we have $\widetilde{\mathcal{O}}\left(d\sqrt{N}\sqrt{\widetilde{\kappa}}\sqrt{\frac{1}{\rho_1 - \rho_2}} \cdot \log^2\left(\frac{1}{\eta}\right)\right)$ if all components are convex, and $\widetilde{\mathcal{O}}\left(dN^{\frac{3}{4}}\sqrt{\widetilde{\kappa}}\sqrt{\frac{1}{\rho_1 - \rho_2}} \cdot \log^2\left(\frac{1}{\eta}\right)\right)$ otherwise. Here $\widetilde{\mathcal{O}}(\cdot)$ hides poly-logarithmic dependences on $\frac{1}{\widetilde{\mu}}$ and $\frac{1}{\widetilde{\Delta}}$. It is remarkable that the SI meta-algorithm is able to separate the dependence of dataset size $N$ from other parameters in the time complexities.

**Parallel work** In a parallel work [Ge et al., 2016], the authors independently proposed a similar ALS algorithm[6], and they solve the least squares problems using AGD. The time complexity of their algorithm for extracting the first canonical correlation is $\widetilde{\mathcal{O}}\left(dN\sqrt{\kappa'}\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\cdot\log\left(\frac{1}{\eta}\right)\right)$, which has linear dependence on $\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\log\left(\frac{1}{\eta}\right)$ (so their algorithm is linearly convergent, but our complexity for ALS+AGD has quadratic dependence on this factor), but typically worse dependence on $N$ and $\kappa'$ (see remarks in Section 3.1.1). Moreover, our SI algorithm tends to significantly outperform ALS theoretically and empirically. It is future work to remove extra $\log\left(\frac{1}{\eta}\right)$ dependence in our analysis.

**Extension to multi-dimensional projections** To extend our algorithms to $L$-dimensional projections, we can extract the dimensions sequentially and remove the explained correlation from $\mathbf{\Sigma}_{xy}$ each time we extract a new dimension [Witten et al., 2009]. For the ALS meta-algorithm, a cleaner approach is to extract the $L$ dimensions simultaneously using (inexact) orthogonal iterations [Golub & van Loan, 1996], in which case the subproblems become multi-dimensional regressions and our normalization steps are of the form $\mathbf{U}_t \leftarrow \widetilde{\mathbf{U}}_t(\widetilde{\mathbf{U}}_t^\top\mathbf{\Sigma}_{xx}\widetilde{\mathbf{U}}_t)^{-\frac{1}{2}}$ (the same normalization is used by [Ma et al., 2015, Wang et al., 2015]). Such normalization involves the eigenvalue decomposition of a $L\times L$ matrix and can be solved exactly as we typically look for low dimensional projections. Our analysis for $L=1$ can be extended to this scenario and the convergence rate of ALS will depend on the gap between $\rho_L$ and $\rho_{L+1}$.

We demonstrate the proposed algorithms, namely `ALS-VR`, `ALS-AVR`, `SI-VR`, and `SI-AVR`, abbreviated as "meta-algorithm – least squares solver" (VR for SVRG, and AVR for ASVRG) on three real-world datasets: Mediamill [Snoek et al., 2006] ($N=3\times10^4$), JW11 [Westbury, 1994] ($N=3\times10^4$), and MNIST [LeCun et al., 1998] ($N=6\times10^4$). We compare our algorithms with batch `AppGrad` and its stochastic version `s-AppGrad` [Ma et al., 2015], as well as the `CCALin` algorithm in parallel work [Ge et al., 2016]. For each algorithm, we compare the canonical correlation estimated by the iterates at different number of passes over the data with that of the exact solution by SVD. For each dataset, we vary the regularization parameters $\gamma_x=\gamma_y$ over $\{10^{-5},10^{-4},10^{-3},10^{-2}\}$ to vary the least squares condition numbers, and larger regularization leads to better conditioning. We plot the suboptimality in objective vs. # passes for each algorithm in Figure 1. Experimental details (e.g. SVRG parameters) are given in Appendix H.

We make the following observations from the results. First, the proposed stochastic algorithms significantly outperform batch gradient based methods `AppGrad`/`CCALin`. This is because the least squares condition numbers for these datasets are large, and SVRG enable us to decouple dependences on the dataset size $N$ and the condition number $\kappa$ in the time complexity. Second, `SI-VR` converges faster than `ALS-VR` as it further decouples the dependence on $N$ and the singular value gap of $\mathbf{T}$. Third, inexact normalizations keep the `s-AppGrad` algorithm from converging to an accurate solution. Finally, ASVRG improves over SVRG when the the condition number is large.

# 4 Conclusion

We study the stochastic optimization of Canonical Correlation Analysis (CCA), which presents challenges due to its nonconvexity and coupling across training samples. Despite recent efforts to address this with stochastic gradient-based algorithms, prior work lacked global convergence guarantees.

[6]Our arxiv preprint for the ALS meta-algorithm was posted before their paper got accepted by ICML 2016.

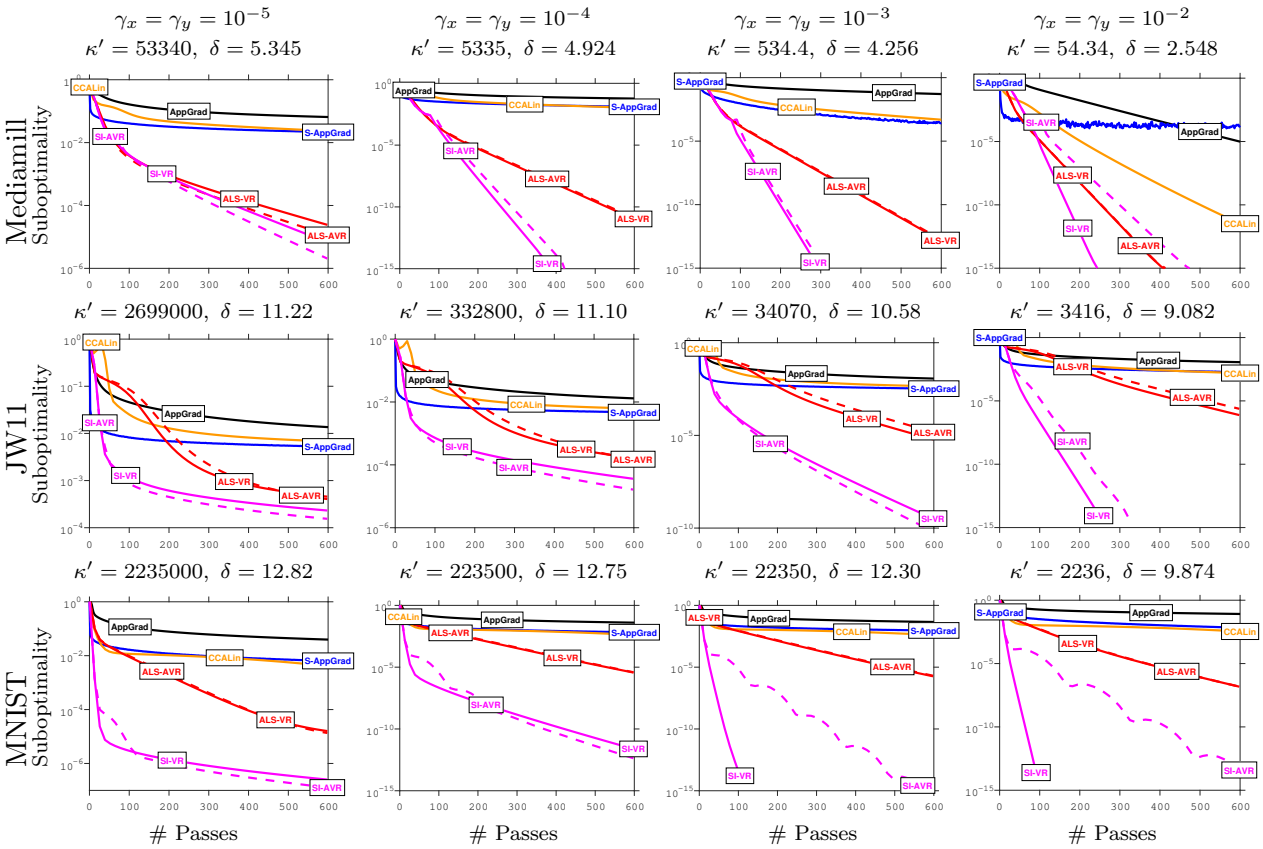


**Figure 1.** Comparison of suboptimality vs. # passes for different algorithms. For each dataset and regularization parameters $(\gamma_x, \gamma_y)$, we give $\kappa' = \max\left(\frac{\sigma_{\max}(\mathbf{\Sigma}_{xx})}{\sigma_{\min}(\mathbf{\Sigma}_{xx})}, \frac{\sigma_{\max}(\mathbf{\Sigma}_{yy})}{\sigma_{\min}(\mathbf{\Sigma}_{yy})}\right)$ and $\delta = \frac{\rho_1^2}{\rho_1^2 - \rho_2^2}$.

Inspired by the alternating least squares and power iteration methods for CCA, as well as shift-and-invert techniques for Principal Component Analysis (PCA), we proposed two globally convergent meta-algorithms: Alternating Least Squares (ALS) with variance reduction and Shift-and-Invert preconditioning. These transform the original problem into sequences of least squares problems that are solved approximately, allowing significant improvements in time complexity over previous methods. Our experimental results further validated the superior performance of these methods on large, real-world datasets.

# References


[Arora et al., 2012] Arora, R., Cotter, A., Livescu, K., & Srebro, N. (2012). Stochastic optimization for pca and pls. In *ALLERTON*.

[Balsubramani et al., 2013] Balsubramani, A., Dasgupta, S., & Freund, Y. (2013). The fast convergence of incremental pca. In *NIPS*.

[Frostig et al., 2015] Frostig, R., Ge, R., Kakade, S., & Sidford, A. (2015). Un-regularizing: Approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *ICML*.

[Garber & Hazan, 2015] Garber, D. & Hazan, E. (2015). Fast and simple pca via convex optimization. *arXiv*.

[Ge et al., 2016] Ge, R., Jin, C., Kakade, S., Netrapalli, P., & Sidford, A. (2016). Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis. *arXiv*.

[Golub & van Loan, 1996] Golub, G. & van Loan, C. (1996). *Matrix Computations*. third edition.

[Golub & Zha, 1995] Golub, G. & Zha, H. (1995). The canonical correlations of matrix pairs and their numerical computation. In *Linear Algebra for Signal Processing* (pp. 27–49).

[Hotelling, 1936] Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3/4), 321–377.

[Jin et al., 2015] Jin, C., Kakade, S., Musco, C., Netrapalli, P., & Sidford, A. (2015). Robust shift-and-invert preconditioning: Faster and more sample efficient algorithms for eigenvector computation.

[Johnson & Zhang, 2013] Johnson, R. & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11), 2278–2324.

[Lin et al., 2015] Lin, H., Mairal, J., & Harchaoui, Z. (2015). A universal catalyst for first-order optimization. In *NIPS*.

[Lu & Foster, 2014] Lu, Y. & Foster, D. (2014). Large scale canonical correlation analysis with iterative least squares. In *NIPS*.

[Ma et al., 2015] Ma, Z., Lu, Y., & Foster, D. (2015). Finding linear structure in large datasets with scalable canonical correlation analysis. In *ICML*.

[Nesterov, 2004] Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization. A Basic Course*. Springer.

[Schmidt et al., 2013] Schmidt, M., Roux, N. L., & Bach, F. (2013). *Minimizing finite sums with the stochastic average gradient*. Technical Report HAL 00860051, École Normale Supérieure.

[Shalev-Shwartz & Zhang, 2013] Shalev-Shwartz, S. & Zhang, T. (2013). Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*.

[Shamir, 2015] Shamir, O. (2015). A stochastic pca and svd algorithm with an exponential convergence rate. In *ICML*.

[Snoek et al., 2006] Snoek, C., Worring, M., van Gemert, J., Geusebroek, J., & Smeulders, A. (2006). The challenge problem for automated detection of 101 semantic concepts in multimedia. In *MULTIMEDIA*.

[Vinod, 1976] Vinod, H. D. (1976). Canonical ridge and econometrics of joint production. *J. Econometrics*.

[Wang et al., 2015] Wang, W., Arora, R., Srebro, N., & Livescu, K. (2015). Stochastic optimization for deep cca via nonlinear orthogonal iterations. In *ALLERTON*.

[Warmuth & Kuzmin, 2008] Warmuth, M. & Kuzmin, D. (2008). Randomized online pca algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*.

[Westbury, 1994] Westbury, J. (1994). *X-Ray Microbeam Speech Production Database User's Handbook*.

[Witten et al., 2009] Witten, D., Tibshirani, R., & Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*.

[Xie et al., 2015] Xie, B., Liang, Y., & Song, L. (2015). Scale up nonlinear component analysis with doubly stochastic gradients. In *NIPS*.

[Yger et al., 2012] Yger, F., Berar, M., Gasso, G., & Rakotomamonjy, A. (2012). Adaptive canonical correlation analysis based on matrix manifolds. In *ICML*.

# A Proof of Theorem 1

*Proof.* It is easy to see that by the end of the first iteration of Algorithm 1, $\widetilde{\boldsymbol{\psi}}_1$ and $\boldsymbol{\psi}_1$ lie in the span of $\{\mathbf{b}_i\}_{i=1}^r$, while $\widetilde{\boldsymbol{\phi}}_1$ and $\boldsymbol{\phi}_1$ lie in the span of $\{\mathbf{a}_i\}_{i=1}^r$. And therefore they remain in these spaces for all $t \geq 1$.

Let us first focus on $\boldsymbol{\phi}_t$. For $t \geq 2$, we observe that

$$\boldsymbol{\phi}_t = \mathbf{T}\boldsymbol{\psi}_{t-1} / \left\|\widetilde{\boldsymbol{\phi}}_t\right\| = \mathbf{T}\mathbf{T}^\top \boldsymbol{\phi}_{t-2} / \left(\left\|\widetilde{\boldsymbol{\phi}}_t\right\| \cdot \left\|\widetilde{\boldsymbol{\psi}}_{t-1}\right\|\right)$$

Since $\left\|\boldsymbol{\phi}_{t-2}\right\| = \|\boldsymbol{\phi}_t\| = 1$, it is equivalent to using the following updates:

$$\boldsymbol{\phi}_t \leftarrow \mathbf{T}\mathbf{T}^\top \boldsymbol{\phi}_{t-2}, \qquad \boldsymbol{\phi}_t \leftarrow \boldsymbol{\phi}_t / \|\boldsymbol{\phi}_t\|$$

This indicates that, Algorithm 1 runs the standard power iterations on $\mathbf{T}\mathbf{T}^\top$ to generate the $\{\boldsymbol{\phi}_t\}_{t\geq 1}$ sequence for every two steps.

**(i)** For $t = 2, 4, \ldots$, we have $\boldsymbol{\phi}_t = \frac{(\mathbf{T}\mathbf{T}^\top)^{\frac{t}{2}}\boldsymbol{\phi}_0}{\left\|(\mathbf{T}\mathbf{T}^\top)^{\frac{t}{2}}\boldsymbol{\phi}_0\right\|}$. Let $\mathbf{M} = \mathbf{T}\mathbf{T}^\top$, whose nonzero eigenvalues are $\rho_1^2 \geq \rho_2^2 \geq \cdots \geq \rho_r^2 > 0$, with corresponding eigenvectors $\mathbf{a}_1, \ldots, \mathbf{a}_r$. Then, for $i = 1, \ldots, r$,

$$\begin{aligned}
(\mathbf{a}_i^\top \boldsymbol{\phi}_t)^2 &= \frac{\left(\mathbf{a}_i^\top \mathbf{M}^{\frac{t}{2}}\boldsymbol{\phi}_0\right)^2}{\left\|\mathbf{M}^{\frac{t}{2}}\boldsymbol{\phi}_0\right\|^2} = \frac{\left(\mathbf{a}_i^\top \mathbf{M}^{\frac{t}{2}}\boldsymbol{\phi}_0\right)^2}{\boldsymbol{\phi}_0^\top \mathbf{M}^t \boldsymbol{\phi}_0} = \frac{\left(\rho_i^t \mathbf{a}_i^\top \boldsymbol{\phi}_0\right)^2}{\sum_{j=1}^r \rho_j^{2t}(\mathbf{a}_j^\top \boldsymbol{\phi}_0)^2} = \frac{\left(\mathbf{a}_i^\top \boldsymbol{\phi}_0\right)^2}{\sum_{j=1}^r \left(\frac{\rho_j^2}{\rho_i^2}\right)^t (\mathbf{a}_j^\top \boldsymbol{\phi}_0)^2} \\
&\leq \frac{\left(\mathbf{a}_i^\top \boldsymbol{\phi}_0\right)^2}{\left(\frac{\rho_1^2}{\rho_i^2}\right)^t (\mathbf{a}_1^\top \boldsymbol{\phi}_0)^2} = \frac{\left(\mathbf{a}_i^\top \boldsymbol{\phi}_0\right)^2}{(\mathbf{a}_1^\top \boldsymbol{\phi}_0)^2}\left(\frac{\rho_i^2}{\rho_1^2}\right)^t = \frac{\left(\mathbf{a}_i^\top \boldsymbol{\phi}_0\right)^2}{(\mathbf{a}_1^\top \boldsymbol{\phi}_0)^2}\left(1 - \frac{\rho_1^2 - \rho_i^2}{\rho_1^2}\right)^t \\
&\leq \frac{\left(\mathbf{a}_i^\top \boldsymbol{\phi}_0\right)^2}{(\mathbf{a}_1^\top \boldsymbol{\phi}_0)^2} \exp\left(-\frac{\rho_1^2 - \rho_i^2}{\rho_1^2} t\right)
\end{aligned}$$

**(ii)** For $t = 1, 3, \ldots$, we have $\boldsymbol{\phi}_t = \frac{(\mathbf{T}\mathbf{T}^\top)^{\frac{t-1}{2}}\mathbf{T}\boldsymbol{\psi}_0}{\left\|(\mathbf{T}\mathbf{T}^\top)^{\frac{t-1}{2}}\mathbf{T}\boldsymbol{\psi}_0\right\|}$. Let $\mathbf{N} = \mathbf{T}^\top\mathbf{T}$, whose nonzero eigenvalues are $\rho_1^2 \geq \rho_2^2 \geq \cdots \geq \rho_r^2 > 0$, with corresponding eigenvectors $\mathbf{b}_1, \ldots, \mathbf{b}_r$. Then, for $i = 1, \ldots, r$,

$$\begin{aligned}
(\mathbf{a}_i^\top \boldsymbol{\phi}_t)^2 &= \frac{\left(\mathbf{a}_i^\top (\mathbf{T}\mathbf{T}^\top)^{\frac{t-1}{2}}\mathbf{T}\boldsymbol{\psi}_0\right)^2}{\left\|(\mathbf{T}\mathbf{T}^\top)^{\frac{t-1}{2}}\mathbf{T}\boldsymbol{\psi}_0\right\|^2} = \frac{\left((\mathbf{T}^\top \mathbf{a}_i)^\top \mathbf{N}^{\frac{t-1}{2}}\boldsymbol{\psi}_0\right)^2}{\boldsymbol{\psi}_0^\top \mathbf{N}^t \boldsymbol{\psi}_0} = \frac{\left(\rho_i^t \mathbf{b}_i^\top \boldsymbol{\psi}_0\right)^2}{\sum_{j=1}^r \rho_j^{2t}(\mathbf{b}_j^\top \boldsymbol{\psi}_0)^2} \\
&\leq \frac{\left(\mathbf{b}_i^\top \boldsymbol{\psi}_0\right)^2}{(\mathbf{b}_1^\top \boldsymbol{\psi}_0)^2} \exp\left(-\frac{\rho_1^2 - \rho_i^2}{\rho_1^2} t\right).
\end{aligned}$$

Given $\delta \in (0, 1)$, define $S(\delta) = \{i : \rho_i^2 > (1-\delta)\rho_1^2\}$. For $\delta_1, \delta_2 \in (0, 1)$, define

$$T(\delta_1, \delta_2) := \lceil \frac{1}{\delta_1} \log\left(\frac{1}{\mu\delta_2}\right) \rceil$$

For all $i \notin S(\delta_1)$, when $t > T(\delta_1, \delta_2)$, it holds that $(\mathbf{a}_i^\top \boldsymbol{\phi}_t)^2 \leq \delta_2 (\mathbf{a}_i^\top \boldsymbol{\phi}_0)^2$ if $t$ is even, and $(\mathbf{a}_i^\top \boldsymbol{\phi}_t)^2 \leq \delta_2 (\mathbf{b}_i^\top \boldsymbol{\psi}_0)^2$ if $t$ is odd. In both cases, we have $\sum_{i\in S(\delta_1)} (\mathbf{a}_i^\top \boldsymbol{\phi}_t)^2 \geq 1 - \delta_2$.

When there exists a postive singular value gap, i.e., $\rho_1 - \rho_2 > 0$, set $\delta_1 = (\rho_1^2 - \rho_2^2)/\rho_1^2$ and thus $S(\delta_1) = 1$. Futhermore, set $\delta_2 = \eta$ and we obtain $(\mathbf{a}_1^\top \boldsymbol{\phi}_t)^2 \geq 1 - \eta$.

The proof for $\boldsymbol{\psi}_t$ is completely analogous. To obtain the bound on the objective, we have

$$\begin{aligned}
\mathbf{u}_t^\top \boldsymbol{\Sigma}_{xy} \mathbf{v}_t = \boldsymbol{\phi}_t^\top \mathbf{T} \boldsymbol{\psi}_t &= \rho_1 (\boldsymbol{\phi}_t^\top \mathbf{a}_1)(\boldsymbol{\psi}_t^\top \mathbf{b}_1) + \sum_{i=2}^{r} \rho_i (\boldsymbol{\phi}_t^\top \mathbf{a}_i)(\boldsymbol{\psi}_t^\top \mathbf{b}_i) \\
&\geq \rho_1 (\boldsymbol{\phi}_t^\top \mathbf{a}_1)(\boldsymbol{\psi}_t^\top \mathbf{b}_1) - \rho_1 \sum_{i=2}^{r} \left| \boldsymbol{\phi}_t^\top \mathbf{a}_i \right| \left| \boldsymbol{\psi}_t^\top \mathbf{b}_i \right| \\
&\geq \rho_1 (1 - \eta) - \rho_1 \sqrt{\sum_{i=2}^{r} \left( \boldsymbol{\phi}_t^\top \mathbf{a}_i \right)^2} \sqrt{\sum_{i=2}^{r} \left( \boldsymbol{\psi}_t^\top \mathbf{b}_i \right)^2} \\
&\geq \rho_1 (1 - \eta) - \rho_1 \eta = \rho_1 (1 - 2\eta)
\end{aligned}$$

where we have used the Cauchy-Schwarz inequality in the second inequality. □

# B Proof of Theorem 2

From now on, we distinguish the iterates of our stochastic algorithm (Algorithm 2) from the iterates of the exact power iterations (Algorithm 1) and denote the latter with asterisks, i.e., $\widetilde{\mathbf{u}}_t^*$ and $\widetilde{\mathbf{v}}_t^*$ for the unnormalized iterates and $\mathbf{u}_t^*$ and $\mathbf{v}_t^*$ for the normalized iterates. We denote the exact optimum of $f_t(\mathbf{u})$ and $g_t(\mathbf{v})$ by $\overline{\mathbf{u}}_t$ and $\overline{\mathbf{v}}_t$ respectively.

The following lemma bounds the distance between the iterates of inexact and exact power iterations.

**Lemma 2.** *Assume that Algorithm 1 and Algorithm 2 start with the same initialization, i.e., $\widetilde{\mathbf{u}}_0 = \widetilde{\mathbf{u}}_0^*$ and $\widetilde{\mathbf{v}}_0 = \widetilde{\mathbf{v}}_0^*$. Then, for $t \geq 1$, the unnormalized iterates of Algorithm 2 satisfy*

$$\max \left( \left\| \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \widetilde{\mathbf{u}}_t - \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \widetilde{\mathbf{u}}_t^* \right\|, \ \left\| \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}} \widetilde{\mathbf{v}}_t - \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}} \widetilde{\mathbf{v}}_t^* \right\| \right) \leq \widetilde{S}_t$$

*where*

$$\widetilde{S}_t := \sqrt{2\epsilon} \, \frac{(2\rho_1/\rho_r)^t - 1}{(2\rho_1/\rho_r) - 1}$$

*Furthermore, for $t \geq 1$, the normalized iterates of Algorithm 2 satisfy*

$$\max \left( \left\| \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \mathbf{u}_t - \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \mathbf{u}_t^* \right\|, \ \left\| \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}} \mathbf{v}_t - \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}} \mathbf{v}_t^* \right\| \right) \leq S_t := \frac{2\widetilde{S}_t}{\rho_r}$$

*Proof.* We focus on the $\{\widetilde{\mathbf{u}}_t\}_{t \geq 0}$ and $\{\mathbf{u}_t\}_{t \geq 0}$ sequences below; the proof for $\{\widetilde{\mathbf{v}}_t\}_{t \geq 0}$ and $\{\mathbf{v}_t\}_{t \geq 0}$ is completely analogous.

We prove the bound for unnormalized iterates by induction. First, the case for $t = 1$ holds trivially. For $t \geq 2$, we can bound the error of the unnormalized iterates using the exact solution to $f_t(\mathbf{u})$:

$$\left\| \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \widetilde{\mathbf{u}}_t - \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \widetilde{\mathbf{u}}_t^* \right\| \leq \left\| \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \widetilde{\mathbf{u}}_t - \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \overline{\mathbf{u}}_t \right\| + \left\| \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \overline{\mathbf{u}}_t - \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} \widetilde{\mathbf{u}}_t^* \right\| \tag{14}$$

For the first term of (14), notice $f_t(\mathbf{u})$ is a quadratic function with minimum achieved at $\overline{\mathbf{u}}_t = \mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xy}\mathbf{v}_{t-1}$. For the approximate solution $\widetilde{\mathbf{u}}_t$, we have

$$f_t(\widetilde{\mathbf{u}}_t) - f_t(\overline{\mathbf{u}}_t) = \frac{1}{2}(\widetilde{\mathbf{u}}_t - \overline{\mathbf{u}}_t)^\top \mathbf{\Sigma}_{xx}(\widetilde{\mathbf{u}}_t - \overline{\mathbf{u}}_t) = \frac{1}{2}\left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_t - \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}_t\right\|^2 \leq \epsilon$$

It then follows that $\left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_t - \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}_t\right\| \leq \sqrt{2\epsilon}$.

The second term of (14) is concerned with the error due to inexact target in the least squares problem $f_t(\mathbf{u})$ as $\mathbf{v}_{t-1}$ is different from $\mathbf{v}_{t-1}^*$. We can bound it as

$$\begin{aligned}
\left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}_t - \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_t^*\right\| &= \left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xy}\mathbf{v}_{t-1} - \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xy}\mathbf{v}_{t-1}^*\right\| \\
&= \left\|\left(\mathbf{\Sigma}_{xx}^{-\frac{1}{2}}\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-\frac{1}{2}}\right)\left(\mathbf{\Sigma}_{yy}^{\frac{1}{2}}(\mathbf{v}_{t-1} - \mathbf{v}_{t-1}^*)\right)\right\| \\
&\leq \|\mathbf{T}\|\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-1} - \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-1}^*\right\| = \rho_1\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-1} - \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-1}^*\right\| \qquad (15)
\end{aligned}$$

In view of the update rule of our algorithm and the triangle inequality, we have

$$\begin{aligned}
&\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-1} - \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-1}^*\right\| \\
\leq &\left\|\frac{\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}\right\|} - \frac{\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\|}\right\| + \left\|\frac{\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\|} - \frac{\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\|}\right\| \\
= &\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}\right\|\left|\frac{1}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}\right\|} - \frac{1}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\|}\right| + \frac{1}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\|}\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1} - \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\| \\
= &\frac{1}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\|}\left|\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\| - \left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}\right\|\right| + \frac{1}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\|}\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1} - \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\| \\
\leq &\frac{2}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\|}\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1} - \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\| \leq \frac{2\widetilde{S}_{t-1}}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\|} \qquad (16)
\end{aligned}$$

We now bound $\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\|$ from below. Since $t \geq 2$, we have

$$\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^* = \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{\Sigma}_{yy}^{-1}\mathbf{\Sigma}_{xy}^\top\mathbf{u}_{t-2}^* = \left(\mathbf{\Sigma}_{yy}^{-\frac{1}{2}}\mathbf{\Sigma}_{xy}^\top\mathbf{\Sigma}_{xx}^{-\frac{1}{2}}\right)\left(\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_{t-2}^*\right) = \mathbf{T}^\top\left(\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_{t-2}^*\right)$$

Now, $\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_{t-2}^*$ corresponds to $\boldsymbol{\phi}_{t-2}$ in Algorithm 1, which has unit length and lies in the span of $\{\mathbf{a}_1, \ldots, \mathbf{a}_r\}$, so we have

$$\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_{t-1}^*\right\| = \left\|\mathbf{T}^\top\boldsymbol{\phi}_{t-2}\right\| \geq \rho_r$$

Combining (14), (15) and (16) gives

$$\begin{aligned}\left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_t - \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_t^*\right\| &\leq \sqrt{2\epsilon} + \frac{2\rho_1}{\rho_r}\cdot\widetilde{S}_{t-1} = \sqrt{2\epsilon} + \frac{2\rho_1}{\rho_r}\cdot\sqrt{2\epsilon}\frac{(2\rho_1/\rho_r)^{t-1}-1}{(2\rho_1/\rho_r)-1} \\ &= \sqrt{2\epsilon}\,\frac{(2\rho_1/\rho_r)^{t}-1}{(2\rho_1/\rho_r)-1} = \widetilde{S}_t\end{aligned}$$

The bound for normalized iterates follows from (16). □

***Proof of Theorem 2***. We prove the theorem by relating the iterates of inexact power iterations to those of exact power iterations.

Assume the same initialization as in Lemma 2. First observe that

$$\begin{aligned}(\mathbf{u}_t^\top\boldsymbol{\Sigma}_{xx}\mathbf{u}^*)^2 &= \left((\mathbf{u}_t^*)^\top\boldsymbol{\Sigma}_{xx}\mathbf{u}^* + (\mathbf{u}_t-\mathbf{u}_t^*)^\top\boldsymbol{\Sigma}_{xx}\mathbf{u}^*\right)^2 \\ &\geq \left((\mathbf{u}_t^*)^\top\boldsymbol{\Sigma}_{xx}\mathbf{u}^*\right)^2 + 2\left((\mathbf{u}_t^*)^\top\boldsymbol{\Sigma}_{xx}\mathbf{u}^*\right)\left((\mathbf{u}_t-\mathbf{u}_t^*)^\top\boldsymbol{\Sigma}_{xx}\mathbf{u}^*\right) \\ &\geq \left((\mathbf{u}_t^*)^\top\boldsymbol{\Sigma}_{xx}\mathbf{u}^*\right)^2 - 2\left|\left(\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}(\mathbf{u}_t-\mathbf{u}_t^*)\right)^\top\left(\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}^*\right)\right| \\ &\geq \left((\mathbf{u}_t^*)^\top\boldsymbol{\Sigma}_{xx}\mathbf{u}^*\right)^2 - 2\left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_t - \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_t^*\right\| \qquad (17)\end{aligned}$$

where we have used the fact that $\left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_t\right\| = \left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_t^*\right\| = \left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}^*\right\| = 1$ and the Cauchy-Schwarz inequality in the last two steps.

Applying Theorem 1 with $T \geq \lceil\frac{\rho_1^2}{\rho_1^2-\rho_2^2}\log\left(\frac{2}{\mu\eta}\right)\rceil$, we have that $\left((\mathbf{u}_T^*)^\top\boldsymbol{\Sigma}_{xx}\mathbf{u}^*\right)^2 \geq 1-\eta/2$. On the other hand, in view of Lemma 2, we have for the specified $\epsilon$ value in Algorithm 2 that $\left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T - \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T^*\right\| \leq S_T = \eta/4$. Plugging these two bounds into (17) gives the desired result.

The proof for $\mathbf{v}_T$ is completely analogous. □

# C SVRG for minimizing $f(\mathbf{u})$

We provide the pseudo-code of SVRG for solving the least squares problem (6) below.

# D Initial suboptimality of warm-starts in Algorithm 2

At time step $t$, we initialize the least squares problem $f_t(\mathbf{u})$ with the unnormalized iterate $\widetilde{\mathbf{u}}_{t-1}$ from the previous time step. We now bound the suboptimality of this initialization. Observe that the minimum of $f_t(\mathbf{u})$ is achieved by $\overline{\mathbf{u}}_t = \boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\mathbf{v}_{t-1}$, and that

$$f_t(\widetilde{\mathbf{u}}_{t-1}) - f_t(\overline{\mathbf{u}}_t) = \frac{1}{2}(\widetilde{\mathbf{u}}_{t-1}-\overline{\mathbf{u}}_t)^\top\boldsymbol{\Sigma}_{xx}(\widetilde{\mathbf{u}}_{t-1}-\overline{\mathbf{u}}_t) = \frac{1}{2}\left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_{t-1} - \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}_t\right\|^2$$

Applying the triangle inequality, we have for $t=1$ that

$$\left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_0 - \boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}_1\right\| \leq \left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_0\right\| + \left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}_1\right\| \leq 1 + \left\|\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\mathbf{v}_0\right\|$$

**Algorithm 3** SVRG for $\min_{\mathbf{u}} \ f(\mathbf{u}) := \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{2}\left|\mathbf{u}^\top\mathbf{x}_i - \mathbf{v}^\top\mathbf{y}_i\right|^2 + \frac{\gamma_x}{2}\|\mathbf{u}\|^2\right)$.

**Input:** Stepsize $\xi$.
  Initialize $\mathbf{u}_{(0)} \in \mathbb{R}^{d_x}$.
  **for** $j = 1, 2, \ldots, M$ **do**
    $\mathbf{w}_0 \leftarrow \mathbf{u}_{(j-1)}$
    Evaluate the batch gradient $\nabla f(\mathbf{w}_0) = \mathbf{X}(\mathbf{X}^\top\mathbf{w}_0 - \mathbf{Y}^\top\mathbf{v})/N + \gamma_x\mathbf{w}_0$
    **for** $t = 1, 2, \ldots, m$ **do**
      Randomly pick $i_t$ from $\{1, \ldots, N\}$
      $\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \xi\left((\mathbf{x}_{i_t}\mathbf{x}_{i_t}^\top + \gamma_x\mathbf{I})(\mathbf{w}_{t-1} - \mathbf{w}_0) + \nabla f(\mathbf{w}_0)\right)$
    **end for**
    $\mathbf{u}_{(j)} \leftarrow \mathbf{w}_t$ for randomly chosen $t \in \{1, \ldots, m\}$.
  **end for**
**Output:** $\mathbf{u}_{(M)}$ is the approximate solution.

$$= 1 + \left\|\mathbf{T}\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_0\right\| \leq 1 + \|\mathbf{T}\|\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_0\right\| = 1 + \rho_1 \leq 2$$

where we have used facts that $\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{u}}_0\right\| = \left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_0\right\| = 1$ due to the initial normalizations.

And we have for $t \geq 2$ that

$$\begin{aligned}
\left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_{t-1} - \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}_t\right\| &\leq \left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_{t-1} - \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}_{t-1}\right\| + \left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}_{t-1} - \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}_t\right\| \\
&\leq \sqrt{2\epsilon} + \left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xy}\mathbf{v}_{t-2} - \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xy}\mathbf{v}_{t-1}\right\| \\
&= \sqrt{2\epsilon} + \left\|\mathbf{T}\left(\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-2} - \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-1}\right)\right\| \\
&\leq \sqrt{2\epsilon} + \|\mathbf{T}\|\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-2} - \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-1}\right\| \\
&\leq \sqrt{2\epsilon} + 2\rho_1 \leq \sqrt{2\epsilon} + 2
\end{aligned}$$

where we have used the fact that $\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-2}\right\| = \left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_{t-1}\right\| = 1$ in the last inequality.

Therefore, for all $t \geq 1$, the ration between initial suboptimality and required accuracy is

$$\frac{f_t(\widetilde{\mathbf{u}}_{t-1}) - f_t(\overline{\mathbf{u}}_t)}{\epsilon} \sim \frac{2}{\epsilon}$$

Our shift-and-invert preconditioning (SI) meta-algorithm is detailed in Algorithm 4.

# E Proof of Theorem 3

The proof of Theorem 3 closely follows that of [Garber & Hazan, 2015, Theorem 4.2]. And we will need a few lemmas on the convergence of inexact power iterations.

**Algorithm 4** The shift-and-invert preconditioning meta-algorithm for CCA.

**Input:** Data matrices $\mathbf{X}$, $\mathbf{Y}$, regularization parameters $(\gamma_x, \gamma_y)$, an estimate $\widetilde{\Delta}$ for $\Delta = \rho_1 - \rho_2$.

Initialize $\widetilde{\mathbf{u}}_0 \in \mathbb{R}^{d_x},\ \widetilde{\mathbf{v}}_0 \in \mathbb{R}^{d_y}$

$\mathbf{u}_0 \leftarrow \widetilde{\mathbf{u}}_0 / \sqrt{\widetilde{\mathbf{u}}_0^\top \mathbf{\Sigma}_{xx} \widetilde{\mathbf{u}}_0}, \qquad \mathbf{v}_0 \leftarrow \widetilde{\mathbf{v}}_0 / \sqrt{\widetilde{\mathbf{v}}_0^\top \mathbf{\Sigma}_{yy} \widetilde{\mathbf{v}}_0}$

**// Phase I: shift-and-invert preconditioning for eigenvectors of $\mathbf{M}_\lambda$**

$s \leftarrow 0, \qquad \lambda_{(0)} \leftarrow 1 + \widetilde{\Delta}$

**repeat**

$s \leftarrow s + 1$

**for** $t = (s-1)m_1 + 1, \ldots, sm_1$ **do**

Optimize the least squares problem

$$\min_{\mathbf{u},\mathbf{v}} \ h_t(\mathbf{u}, \mathbf{v}) := \frac{1}{2} \begin{bmatrix} \mathbf{u}^\top \mathbf{v}^\top \end{bmatrix} \begin{bmatrix} \lambda_{(s-1)} \mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda_{(s-1)} \mathbf{\Sigma}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} - \mathbf{u}^\top \mathbf{\Sigma}_{xx} \mathbf{u}_{t-1} - \mathbf{v}^\top \mathbf{\Sigma}_{yy} \mathbf{v}_{t-1}$$

and output an approximate solution $(\widetilde{\mathbf{u}}_t, \widetilde{\mathbf{v}}_t)$ satisfying $h_t(\widetilde{\mathbf{u}}_t, \widetilde{\mathbf{v}}_t) \le \min_{\mathbf{u},\mathbf{v}} h_t(\mathbf{u}, \mathbf{v}) + \widetilde{\epsilon}$.

Normalization: $\begin{bmatrix} \mathbf{u}_t \\ \mathbf{v}_t \end{bmatrix} \leftarrow \sqrt{2} \begin{bmatrix} \widetilde{\mathbf{u}}_t \\ \widetilde{\mathbf{v}}_t \end{bmatrix} \Big/ \sqrt{\widetilde{\mathbf{u}}_t^\top \mathbf{\Sigma}_{xx} \widetilde{\mathbf{u}}_t + \widetilde{\mathbf{v}}_t^\top \mathbf{\Sigma}_{yy} \widetilde{\mathbf{v}}_t}$

**end for**

Optimize the least squares problem

$$\min_{\mathbf{w}} \ l_s(\mathbf{w}) := \frac{1}{2} \mathbf{w}^\top \begin{bmatrix} \lambda_{(s-1)} \mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda_{(s-1)} \mathbf{\Sigma}_{yy} \end{bmatrix} \mathbf{w} - \mathbf{w}^\top \begin{bmatrix} \mathbf{\Sigma}_{xx} \mathbf{u}_{sm_1} \\ \mathbf{\Sigma}_{yy} \mathbf{v}_{sm_1} \end{bmatrix}$$

and output an approximate solution $\mathbf{w}_s$ satisfying $l_s(\mathbf{w}_s) \le \min_{\mathbf{w}} l_s(\mathbf{w}) + \widetilde{\epsilon}$.

$\Delta_s \leftarrow \frac{1}{2} \cdot \dfrac{1}{\frac{1}{2}\begin{bmatrix} \mathbf{u}_{sm_1}^\top \mathbf{v}_{sm_1}^\top \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{xx} & \\ & \mathbf{\Sigma}_{yy} \end{bmatrix} \mathbf{w}_s - 2\sqrt{\widetilde{\epsilon}/\widetilde{\Delta}}}, \qquad \lambda_{(s)} \leftarrow \lambda_{(s-1)} - \frac{\Delta_s}{2}$

**until** $\Delta_{(s)} \le \widetilde{\Delta}$

$\lambda_{(f)} \leftarrow \lambda_{(s)}$

**for** $t = sm_1 + 1, sm_1 + 2, \ldots, sm_1 + m_2$ **do**

Optimize the least squares problem

$$\min_{\mathbf{u},\mathbf{v}} \ h_t(\mathbf{u}, \mathbf{v}) := \frac{1}{2} \begin{bmatrix} \mathbf{u}^\top \mathbf{v}^\top \end{bmatrix} \begin{bmatrix} \lambda_{(f)} \mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda_{(f)} \mathbf{\Sigma}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} - \mathbf{u}^\top \mathbf{\Sigma}_{xx} \mathbf{u}_{t-1} - \mathbf{v}^\top \mathbf{\Sigma}_{yy} \mathbf{v}_{t-1}$$

and output an approximate solution $(\widetilde{\mathbf{u}}_t, \widetilde{\mathbf{v}}_t)$ satisfying $h_t(\widetilde{\mathbf{u}}_t, \widetilde{\mathbf{v}}_t) \le \min_{\mathbf{u},\mathbf{v}} h_t(\mathbf{u}, \mathbf{v}) + \widetilde{\epsilon}$.

Normalization: $\begin{bmatrix} \mathbf{u}_t \\ \mathbf{v}_t \end{bmatrix} \leftarrow \sqrt{2} \begin{bmatrix} \widetilde{\mathbf{u}}_t \\ \widetilde{\mathbf{v}}_t \end{bmatrix} \Big/ \sqrt{\widetilde{\mathbf{u}}_t^\top \mathbf{\Sigma}_{xx} \widetilde{\mathbf{u}}_t + \widetilde{\mathbf{v}}_t^\top \mathbf{\Sigma}_{yy} \widetilde{\mathbf{v}}_t}$

**end for**

**// Phase II: Final normalization**

$T \leftarrow sm_1 + m_2, \qquad \widehat{\mathbf{u}} \leftarrow \mathbf{u}_T / \sqrt{\mathbf{u}_T^\top \mathbf{\Sigma}_{xx} \mathbf{u}_T}, \qquad \widehat{\mathbf{v}} \leftarrow \mathbf{v}_T / \sqrt{\mathbf{v}_T^\top \mathbf{\Sigma}_{yy} \mathbf{v}_T}$

**Output:** $(\widehat{\mathbf{u}}, \widehat{\mathbf{v}})$ is the approximate solution to CCA.

## E.1 Auxiliary lemmas

Define the condition number of $\mathbf{M}_\lambda$ as

$$\kappa_\lambda := \frac{\sigma_1(\mathbf{M}_\lambda)}{\sigma_d(\mathbf{M}_\lambda)} = \frac{\frac{1}{\lambda-\rho_1}}{\frac{1}{\lambda+\rho_1}} = \frac{\lambda+\rho_1}{\lambda-\rho_1}$$

and the inverse relative spectral gap of $\mathbf{M}_\lambda$ as

$$\delta_\lambda := \frac{\sigma_1(\mathbf{M}_\lambda)}{\sigma_1(\mathbf{M}_\lambda) - \sigma_2(\mathbf{M}_\lambda)} = \frac{\frac{1}{\lambda-\rho_1}}{\frac{1}{\lambda-\rho_1} - \frac{1}{\lambda-\rho_2}} = \frac{\lambda-\rho_2}{\rho_1-\rho_2}$$

The first lemma states the convergence of exact power iterations, paralleling [Garber & Hazan, 2015, Theorem A.1].

**Lemma 3** (Convergence of exact power iterations)**.** *Fix $\alpha > 0$. For the exact power iterations on $\mathbf{M}_\lambda$ where*

$$\begin{aligned}
\begin{bmatrix} \widetilde{\mathbf{u}}_t^* \\ \widetilde{\mathbf{v}}_t^* \end{bmatrix} &\leftarrow \begin{bmatrix} \lambda\mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda\mathbf{\Sigma}_{yy} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Sigma}_{xx} & \\ & \mathbf{\Sigma}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{t-1}^* \\ \mathbf{v}_{t-1}^* \end{bmatrix}, \\
\begin{bmatrix} \mathbf{u}_t^* \\ \mathbf{v}_t^* \end{bmatrix} &\leftarrow \sqrt{2} \begin{bmatrix} \widetilde{\mathbf{u}}_t^* \\ \widetilde{\mathbf{v}}_t^* \end{bmatrix} \Big/ \sqrt{(\widetilde{\mathbf{u}}_t^*)^\top \mathbf{\Sigma}_{xx} \widetilde{\mathbf{u}}_t^* + (\widetilde{\mathbf{v}}_t^*)^\top \mathbf{\Sigma}_{yy} \widetilde{\mathbf{v}}_t^*}, \qquad \textit{for} \quad t = 1, \ldots, m
\end{aligned}$$

*and $\mu' := \frac{1}{4}\left((\mathbf{u}_0^*)^\top \mathbf{\Sigma}_{xx} \mathbf{u}^* + (\mathbf{v}_0^*)^\top \mathbf{\Sigma}_{yy} \mathbf{v}^*\right)^2 > 0$, we have*

- *(crude regime)*

$$\frac{1}{2}\left[(\mathbf{u}_t^*)^\top \mathbf{\Sigma}_{xx}^{\frac{1}{2}},\ (\mathbf{v}_t^*)^\top \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\right] \mathbf{M}_\lambda \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} \mathbf{u}_t^* \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \mathbf{v}_t^* \end{bmatrix} \geq (1-\alpha) \cdot \sigma_1(\mathbf{M}_\lambda)$$

  *for $t \geq \lceil \frac{1}{\alpha} \log\left(\frac{2}{\mu'\alpha}\right) \rceil$,*

- *(accurate regime)*

$$\frac{1}{4}\left((\mathbf{u}_t^*)^\top \mathbf{\Sigma}_{xx} \mathbf{u}^* + (\mathbf{v}_t^*)^\top \mathbf{\Sigma}_{yy} \mathbf{v}^*\right)^2 \geq 1 - \alpha$$

  *for $t \geq \lceil \frac{\delta_\lambda}{2} \log\left(\frac{1}{\mu'\alpha}\right) \rceil$.*

The second lemma bounds the distances between the iterates of inexact and exact power iterations, paralleling [Garber & Hazan, 2015, Lemma 4.1]. Recall that the $(\widetilde{\mathbf{u}}_t, \widetilde{\mathbf{v}}_t)$ in Algorithm 4 satisfies $h_t(\widetilde{\mathbf{u}}_t, \widetilde{\mathbf{v}}_t) \leq \min_{\mathbf{u},\mathbf{v}} h_t(\mathbf{u}, \mathbf{v}) + \widetilde{\epsilon}$. Let $(\overline{\mathbf{u}}_t, \overline{\mathbf{v}}_t)$ be the exact minimum of $h_t$. Then we have

$$\begin{aligned}
& h_t(\widetilde{\mathbf{u}}_t, \widetilde{\mathbf{v}}_t) - h_t(\overline{\mathbf{u}}_t, \overline{\mathbf{v}}_t) \\
=& \frac{1}{2}\begin{bmatrix} (\widetilde{\mathbf{u}}_t - \overline{\mathbf{u}}_t)^\top & (\widetilde{\mathbf{v}}_t - \overline{\mathbf{v}}_t)^\top \end{bmatrix} \begin{bmatrix} \lambda\mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda\mathbf{\Sigma}_{yy} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{u}}_t - \overline{\mathbf{u}}_t \\ \widetilde{\mathbf{v}}_t - \overline{\mathbf{v}}_t \end{bmatrix} \\
=& \frac{1}{2}\begin{bmatrix} (\widetilde{\mathbf{u}}_t - \overline{\mathbf{u}}_t)^\top & (\widetilde{\mathbf{v}}_t - \overline{\mathbf{v}}_t)^\top \end{bmatrix} \begin{bmatrix} \lambda\mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda\mathbf{\Sigma}_{yy} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{u}}_t - \overline{\mathbf{u}}_t \\ \widetilde{\mathbf{v}}_t - \overline{\mathbf{v}}_t \end{bmatrix}
\end{aligned}$$

$$
\begin{aligned}
=&\frac{1}{2}\begin{bmatrix}(\widetilde{\mathbf{u}}_t-\overline{\mathbf{u}}_t)^\top\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} & (\widetilde{\mathbf{v}}_t-\overline{\mathbf{v}}_t)^\top\boldsymbol{\Sigma}_{yy}^{\frac{1}{2}}\end{bmatrix}\begin{bmatrix}\lambda\mathbf{I} & -\mathbf{T}\\ -\mathbf{T}^\top & \lambda\mathbf{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}(\widetilde{\mathbf{u}}_t-\overline{\mathbf{u}}_t)\\ \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}}(\widetilde{\mathbf{v}}_t-\overline{\mathbf{v}}_t)\end{bmatrix}\\
=&\frac{1}{2}\begin{bmatrix}(\widetilde{\mathbf{u}}_t-\overline{\mathbf{u}}_t)^\top\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}} & (\widetilde{\mathbf{v}}_t-\overline{\mathbf{v}}_t)^\top\boldsymbol{\Sigma}_{yy}^{\frac{1}{2}}\end{bmatrix}\mathbf{M}_\lambda^{-1}\begin{bmatrix}\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}(\widetilde{\mathbf{u}}_t-\overline{\mathbf{u}}_t)\\ \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}}(\widetilde{\mathbf{v}}_t-\overline{\mathbf{v}}_t)\end{bmatrix}\le\widetilde{\epsilon} \qquad (18)
\end{aligned}
$$

**Lemma 4** (Power iterations with inexact matrix-vector multiplications)**.** *Consider the inexact power iterations on* $\mathbf{M}_\lambda$ *where*

$$
\begin{aligned}
&(\widetilde{\mathbf{u}}_t,\widetilde{\mathbf{v}}_t) \qquad \textit{satisfies} \qquad (18),\\
&\begin{bmatrix}\mathbf{u}_t\\ \mathbf{v}_t\end{bmatrix}\leftarrow\sqrt{2}\begin{bmatrix}\widetilde{\mathbf{u}}_t\\ \widetilde{\mathbf{v}}_t\end{bmatrix}\Big/\sqrt{\widetilde{\mathbf{u}}_t^\top\boldsymbol{\Sigma}_{xx}\widetilde{\mathbf{u}}_t+\widetilde{\mathbf{v}}_t^\top\boldsymbol{\Sigma}_{yy}\widetilde{\mathbf{v}}_t}, \qquad \textit{for} \quad t=1,\dots,m
\end{aligned}
$$

*Compare these iterates with those of the exact power iterations described in Lemma 3 using the same initialization* $\widetilde{\mathbf{u}}_0=\widetilde{\mathbf{u}}_0^*$, $\widetilde{\mathbf{v}}_0=\widetilde{\mathbf{v}}_0^*$. *Then, for* $t\ge 0$, *the unnormalized iterates satisfy*

$$
\left\|\frac{1}{\sqrt{2}}\begin{bmatrix}\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_t\\ \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_t\end{bmatrix}-\frac{1}{\sqrt{2}}\begin{bmatrix}\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_t^*\\ \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_t^*\end{bmatrix}\right\|\le\widetilde{R}_t
$$

*where*

$$
\widetilde{R}_t:=\sqrt{\sigma_1(\mathbf{M}_\lambda)\cdot\widetilde{\epsilon}}\cdot\frac{(2\kappa_\lambda)^t-1}{2\kappa_\lambda-1}
$$

*while the normalized iterates satisfy*

$$
\left\|\frac{1}{\sqrt{2}}\begin{bmatrix}\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_t\\ \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_t\end{bmatrix}-\frac{1}{\sqrt{2}}\begin{bmatrix}\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_t^*\\ \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_t^*\end{bmatrix}\right\|\le R_t:=\frac{2\widetilde{R}_t}{\sigma_d(\mathbf{M}_\lambda)}
$$

The third lemma states the convergence of inexact power iterations, paralleling [Garber & Hazan, 2015, Theorem 4.1].

**Lemma 5** (Convergence of inexact power iterations)**.** *Fix* $\alpha>0$. *Consider the inexact power iterations described in Lemma 4.*

- *(crude regime) Let* $t_1=\lceil\frac{2}{\alpha}\log\left(\frac{4}{\mu'\alpha}\right)\rceil$. *Fix* $T\ge t_1$, *and set* $\widetilde{\epsilon}(T)=\frac{\alpha^2\cdot\sigma_d(\mathbf{M}_\lambda)}{64\kappa_\lambda}\left(\frac{2\kappa_\lambda-1}{(2\kappa_\lambda)^T-1}\right)^2$. *Then we have*

$$
\frac{1}{2}\begin{bmatrix}\mathbf{u}_T^\top\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}, & \mathbf{v}_T^\top\boldsymbol{\Sigma}_{yy}^{\frac{1}{2}}\end{bmatrix}\mathbf{M}_\lambda\begin{bmatrix}\boldsymbol{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T\\ \boldsymbol{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_T\end{bmatrix}\ge(1-\alpha)\cdot\sigma_1(\mathbf{M}_\lambda)
$$

- *(accurate regime) Let* $t_2=\lceil\frac{\delta(\mathbf{M}_\lambda)}{2}\log\left(\frac{2}{\mu'\alpha}\right)\rceil$. *Fix* $T\ge t_2$, *and set* $\widetilde{\epsilon}(T)=\frac{\alpha^2\cdot\sigma_d(\mathbf{M}_\lambda)}{64\kappa_\lambda}\left(\frac{2\kappa_\lambda-1}{(2\kappa_\lambda)^T-1}\right)^2$. *Then we have*

$$
\frac{1}{4}\left(\mathbf{u}_T^\top\boldsymbol{\Sigma}_{xx}\mathbf{u}^*+\mathbf{v}_T^\top\boldsymbol{\Sigma}_{yy}\mathbf{v}^*\right)^2\ge 1-\alpha
$$

For brevity, let us define the following short-hands:

$$\widetilde{\mathbf{r}}_t = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} \widetilde{\mathbf{u}}_t \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \widetilde{\mathbf{v}}_t \end{bmatrix}, \qquad \mathbf{r}_t = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} \mathbf{u}_t \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \mathbf{v}_t \end{bmatrix}, \qquad \overline{\mathbf{r}}_t = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} \overline{\mathbf{u}}_t \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \overline{\mathbf{v}}_t \end{bmatrix},$$

$$\widetilde{\mathbf{r}}_t^* = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} \widetilde{\mathbf{u}}_t^* \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \widetilde{\mathbf{v}}_t^* \end{bmatrix}, \qquad \mathbf{r}_t^* = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} \mathbf{u}_t^* \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \mathbf{v}_t^* \end{bmatrix}, \qquad \mathbf{r}^* = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} \mathbf{u}^* \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \mathbf{v}^* \end{bmatrix}$$

All these vectors are in $\mathbb{R}^d$ and have length 1.

Observe that the matrix-vector multiplication (8) is equivalent to

$$\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} \widetilde{\mathbf{u}}_t \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \widetilde{\mathbf{v}}_t \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \lambda \mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda \mathbf{\Sigma}_{yy} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} \mathbf{u}_{t-1} \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \mathbf{v}_{t-1} \end{bmatrix}$$

and

$$\begin{aligned}
&\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \lambda \mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda \mathbf{\Sigma}_{yy} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \end{bmatrix} \\
=&\begin{bmatrix} \mathbf{\Sigma}_{xx}^{-\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{-\frac{1}{2}} \end{bmatrix}^{-1} \begin{bmatrix} \lambda \mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda \mathbf{\Sigma}_{yy} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{-\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{-\frac{1}{2}} \end{bmatrix}^{-1} \\
=&\left( \begin{bmatrix} \mathbf{\Sigma}_{xx}^{-\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \lambda \mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda \mathbf{\Sigma}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{xx}^{-\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{-\frac{1}{2}} \end{bmatrix} \right)^{-1} \\
=&\begin{bmatrix} \lambda \mathbf{I} & -\mathbf{\Sigma}_{xx}^{-\frac{1}{2}} \mathbf{\Sigma}_{xy} \mathbf{\Sigma}_{yy}^{-\frac{1}{2}} \\ -\mathbf{\Sigma}_{yy}^{-\frac{1}{2}} \mathbf{\Sigma}_{xy}^\top \mathbf{\Sigma}_{xx}^{-\frac{1}{2}} & \lambda \mathbf{I} \end{bmatrix}^{-1} \\
=&\mathbf{M}_\lambda
\end{aligned}$$

Then the updates for exact power iterations can be written as

$$\widetilde{\mathbf{r}}_t^* \leftarrow \mathbf{M}_\lambda \mathbf{r}_{t-1}^*, \qquad \mathbf{r}_t^* \leftarrow \widetilde{\mathbf{r}}_t^* / \left\| \widetilde{\mathbf{r}}_t^* \right\|, \qquad t = 1, \ldots$$

and the updates for inexact power iterations can be written as

$$\widetilde{\mathbf{r}}_t \approx \mathbf{M}_\lambda \mathbf{r}_{t-1}, \qquad \mathbf{r}_t \leftarrow \widetilde{\mathbf{r}}_t / \left\| \widetilde{\mathbf{r}}_t \right\|, \qquad t = 1, \ldots$$

Note we have according to (18) that

$$\widetilde{\epsilon} \geq (\widetilde{\mathbf{r}}_t - \overline{\mathbf{r}}_t)^\top \mathbf{M}_\lambda^{-1} (\widetilde{\mathbf{r}}_t - \overline{\mathbf{r}}_t) \geq \sigma_d(\mathbf{M}_\lambda^{-1}) \cdot \left\| \widetilde{\mathbf{r}}_t - \overline{\mathbf{r}}_t \right\|^2 = \frac{1}{\sigma_1(\mathbf{M}_\lambda)} \cdot \left\| \widetilde{\mathbf{r}}_t - \overline{\mathbf{r}}_t \right\|^2$$

or equivalently

$$\left\| \widetilde{\mathbf{r}}_t - \overline{\mathbf{r}}_t \right\| \leq \sqrt{\sigma_1(\mathbf{M}_\lambda) \cdot \epsilon} \tag{19}$$

***Proof of Lemma 3.*** Recall that the eigenvectors of $\mathbf{M}_\lambda$ are:

$$\lambda_1 := \frac{1}{\lambda - \rho_1} > \lambda_2 := \frac{1}{\lambda - \rho_2} \geq \cdots \geq \lambda_{d-1} := \frac{1}{\lambda + \rho_2} \geq \lambda_d := \frac{1}{\lambda + \rho_1}$$

with corresponding eigenvectors

$$\mathbf{e}_1 = \mathbf{r}^* = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{b}_1 \end{bmatrix},\ \mathbf{e}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{a}_2 \\ \mathbf{b}_2 \end{bmatrix},\ \ldots,\ \mathbf{e}_{d-1} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{a}_2 \\ -\mathbf{b}_2 \end{bmatrix},\ \mathbf{e}_d = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{a}_1 \\ -\mathbf{b}_1 \end{bmatrix}$$

By the update rule of exact power iterations, it holds that for $i = 1, \ldots, d$ that

$$\begin{aligned}
(\mathbf{e}_i^\top \mathbf{r}_t^*)^2 &= \frac{\left(\mathbf{e}_i^\top \mathbf{M}_\lambda^t \mathbf{r}_0^*\right)^2}{\left\|\mathbf{M}_\lambda^t \mathbf{r}_0^*\right\|^2} = \frac{\left(\mathbf{e}_i^\top \mathbf{M}_\lambda^t \mathbf{r}_0\right)^2}{(\mathbf{r}_0^*)^\top \mathbf{M}_\lambda^{2t} \mathbf{r}_0^*} = \frac{\left(\lambda_i^t \mathbf{e}_i^\top \mathbf{r}_0^*\right)^2}{\sum_{j=1}^d \lambda_j^{2t} \left(\mathbf{e}_j^\top \mathbf{r}_0^*\right)^2} = \frac{\left(\mathbf{e}_i^\top \mathbf{r}_0^*\right)^2}{\sum_{j=1}^d \left(\frac{\lambda_j}{\lambda_i}\right)^{2t} \left(\mathbf{e}_j^\top \mathbf{r}_0^*\right)^2} \\
&\le \frac{\left(\mathbf{e}_i^\top \mathbf{r}_0^*\right)^2}{\left(\frac{\lambda_1}{\lambda_i}\right)^{2t} \left(\mathbf{e}_1^\top \mathbf{r}_0^*\right)^2} = \frac{\left(\mathbf{e}_i^\top \mathbf{r}_0^*\right)^2}{\left(\mathbf{e}_1^\top \mathbf{r}_0^*\right)^2} \left(\frac{\lambda_i}{\lambda_1}\right)^{2t} = \frac{\left(\mathbf{e}_i^\top \mathbf{r}_0^*\right)^2}{\widetilde{\mu}} \left(1 - \frac{\lambda_1 - \lambda_i}{\lambda_1}\right)^{2t} \\
&\le \frac{\left(\mathbf{e}_i^\top \mathbf{r}_0^*\right)^2}{\widetilde{\mu}} \cdot \exp\left(-2\frac{\lambda_1 - \lambda_i}{\lambda_1} t\right)
\end{aligned}$$

Given $\delta \in (0,1)$, define $S(\delta) = \{i : \lambda_i > (1-\delta)\lambda_1\}$. For $\delta_1, \delta_2 \in (0,1)$, define

$$T(\delta_1, \delta_2) := \lceil \frac{1}{2\delta_1} \log\left(\frac{1}{\widetilde{\mu}\delta_2}\right) \rceil$$

For all $i \notin S(\delta_1)$, when $t > T(\delta_1, \delta_2)$, it holds that $(\mathbf{e}_i^\top \mathbf{r}_t^*)^2 \le \delta_2 (\mathbf{e}_i^\top \mathbf{r}_0^*)^2$, and thus in particular $\sum_{i \in S(\alpha/2)} \left(\mathbf{e}_i^\top \mathbf{r}_t^*\right)^2 \ge 1 - \delta_2$.

Part one (crude regime) of the lemma now follows by noticing that, by setting $\delta_1 = \delta_2 = \frac{\alpha}{2}$ we have that for $t \ge T\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$, it holds that

$$(\mathbf{r}_t^*)^\top \mathbf{M}_\lambda \mathbf{r}_t^* = \sum_{i=1}^d \lambda_i \left(\mathbf{e}_i^\top \mathbf{r}_t^*\right)^2 \ge \sum_{i \in S(\alpha/2)} \left(1 - \frac{\alpha}{2}\right) \lambda_1 \left(\mathbf{e}_i^\top \mathbf{r}_t^*\right)^2 \ge \left(1 - \frac{\alpha}{2}\right)^2 \lambda_1 \ge (1-\alpha)\,\lambda_1$$

For the second part (accurate regime) of the lemma, note that $S\left(\frac{\lambda_1 - \lambda_2}{\lambda_1}\right) = \{1\}$. Thus for all $t \ge T\left(\frac{\lambda_1 - \lambda_2}{\lambda_1}, \alpha\right)$, it holds that $(\mathbf{e}_1^\top \mathbf{r}_t^*)^2 \ge 1 - \alpha$.

□

***Proof of Lemma 4.*** We prove the bound for unnormalized iterates by induction. The case for $t = 1$ holds trivially. For $t \ge 2$, we can bound the error of the unnormalized iterates using the exact solution to $\widetilde{h}_t$:

$$\|\widetilde{\mathbf{r}}_t - \widetilde{\mathbf{r}}_t^*\| \le \|\widetilde{\mathbf{r}}_t - \overline{\mathbf{r}}_t\| + \|\overline{\mathbf{r}}_t - \widetilde{\mathbf{r}}_t^*\| \tag{20}$$

The second term of (20) is concerned with the error due to inexact target in the least squares problem $h_t(\mathbf{u}, \mathbf{v})$ as $\begin{bmatrix} \mathbf{u}_{t-1} \\ \mathbf{v}_{t-1} \end{bmatrix}$ is different from $\begin{bmatrix} \mathbf{u}_{t-1}^* \\ \mathbf{v}_{t-1}^* \end{bmatrix}$. We can bound this term as

$$\begin{aligned}
\|\overline{\mathbf{r}}_t - \widetilde{\mathbf{r}}_t^*\| &= \left\|\mathbf{M}_\lambda \mathbf{r}_{t-1} - \mathbf{M}_\lambda \mathbf{r}_{t-1}^*\right\| \le \|\mathbf{M}_\lambda\| \cdot \left\|\mathbf{r}_{t-1} - \mathbf{r}_{t-1}^*\right\| \\
&= \sigma_1(\mathbf{M}_\lambda) \cdot \left\|\mathbf{r}_{t-1} - \mathbf{r}_{t-1}^*\right\|
\end{aligned} \tag{21}$$

In view of the update rule of our algorithm and the triangle inequality, we have

$$\left\|\mathbf{r}_{t-1} - \mathbf{r}_{t-1}^*\right\|$$

$$
\begin{aligned}
&\leq \left\| \frac{\widetilde{\mathbf{r}}_{t-1}}{\|\widetilde{\mathbf{r}}_{t-1}\|} - \frac{\widetilde{\mathbf{r}}_{t-1}}{\|\widetilde{\mathbf{r}}^*_{t-1}\|} \right\| + \left\| \frac{\widetilde{\mathbf{r}}_{t-1}}{\|\widetilde{\mathbf{r}}^*_{t-1}\|} - \frac{\widetilde{\mathbf{r}}^*_{t-1}}{\|\widetilde{\mathbf{r}}^*_{t-1}\|} \right\| \\
&= \|\widetilde{\mathbf{r}}_{t-1}\| \left| \frac{1}{\|\widetilde{\mathbf{r}}_{t-1}\|} - \frac{1}{\|\widetilde{\mathbf{r}}^*_{t-1}\|} \right| + \frac{1}{\|\widetilde{\mathbf{r}}^*_{t-1}\|} \left\| \widetilde{\mathbf{r}}_{t-1} - \widetilde{\mathbf{r}}^*_{t-1} \right\| \\
&= \frac{1}{\|\widetilde{\mathbf{r}}^*_{t-1}\|} \left| \|\widetilde{\mathbf{r}}^*_{t-1}\| - \|\widetilde{\mathbf{r}}_{t-1}\| \right| + \frac{1}{\|\widetilde{\mathbf{r}}^*_{t-1}\|} \left\| \widetilde{\mathbf{r}}_{t-1} - \widetilde{\mathbf{r}}^*_{t-1} \right\| \\
&\leq \frac{2}{\|\widetilde{\mathbf{r}}^*_{t-1}\|} \left\| \widetilde{\mathbf{r}}_{t-1} - \widetilde{\mathbf{r}}^*_{t-1} \right\| \leq \frac{2\widetilde{R}_{t-1}}{\|\widetilde{\mathbf{r}}^*_{t-1}\|}
\end{aligned} \tag{22}
$$

For $t \geq 2$, we have $\widetilde{\mathbf{r}}^*_{t-1} = \mathbf{M}_\lambda \mathbf{r}^*_{t-2}$ and $\left\|\mathbf{r}^*_{t-2}\right\| = 1$, and thus

$$
\left\|\widetilde{\mathbf{r}}^*_{t-1}\right\| \geq \sigma_d(\mathbf{M}_\lambda).
$$

Combining (20), (21) and (22) gives

$$
\|\widetilde{\mathbf{r}}_t - \widetilde{\mathbf{r}}^*_t\| \leq \sqrt{\sigma_1(\mathbf{M}_\lambda) \cdot \epsilon} + 2\kappa_\lambda \widetilde{R}_{t-1} = \widetilde{R}_t.
$$

The bound for normalized iterates follows from (22). □

***Proof of Lemma 5.*** For the first item (crude regime), observe that

$$
\mathbf{r}_t^\top \mathbf{M}_\lambda \mathbf{r}_t = (\mathbf{r}^*_t)^\top \mathbf{M}_\lambda \mathbf{r}^*_t + \left( (\mathbf{r}^*_t)^\top \mathbf{M}_\lambda \mathbf{r}^*_t - \mathbf{r}_t^\top \mathbf{M}_\lambda \mathbf{r}_t \right) \tag{23}
$$

and that

$$
\begin{aligned}
\left| (\mathbf{r}^*_t)^\top \mathbf{M}_\lambda (\mathbf{r}^*_t) - \mathbf{r}_t^\top \mathbf{M}_\lambda \mathbf{r}_t \right| &= \left| \left( \mathbf{M}_\lambda^{\frac{1}{2}} \mathbf{r}^*_t + \mathbf{M}_\lambda^{\frac{1}{2}} \mathbf{r}_t \right)^\top \left( \mathbf{M}_\lambda^{\frac{1}{2}} \mathbf{r}^*_t - \mathbf{M}_\lambda^{\frac{1}{2}} \mathbf{r}_t \right) \right| \\
&\leq \left\| \mathbf{M}_\lambda^{\frac{1}{2}} \mathbf{r}^*_t + \mathbf{M}_\lambda^{\frac{1}{2}} \mathbf{r}_t \right\| \left\| \mathbf{M}_\lambda^{\frac{1}{2}} \mathbf{r}^*_t - \mathbf{M}_\lambda^{\frac{1}{2}} \mathbf{r}_t \right\| \\
&\leq \left\| \mathbf{M}_\lambda^{\frac{1}{2}} \right\| \|\mathbf{r}^*_t + \mathbf{r}_t\| \left\| \mathbf{M}_\lambda^{\frac{1}{2}} \right\| \|\mathbf{r}^*_t - \mathbf{r}_t\| \\
&\leq \|\mathbf{M}_\lambda\| \left( \|\mathbf{r}^*_t\| + \|\mathbf{r}_t\| \right) \|\mathbf{r}^*_t - \mathbf{r}_t\| \\
&= 2\sigma_1(\mathbf{M}_\lambda) \cdot \|\mathbf{r}^*_t - \mathbf{r}_t\|
\end{aligned}
$$

Our choices of $T$ and $\widetilde{\epsilon}$ make sure that $(\mathbf{r}^*_T)^\top \mathbf{M}_\lambda \mathbf{r}^*_T \geq (1 - \frac{\alpha}{2}) \cdot \sigma_1(\mathbf{M}_\lambda)$ by Lemma 3 and that $\|\mathbf{r}^*_T - \mathbf{r}_T\| \leq R_T = \alpha/4$ by Lemma 4. Continuing from (23), we have

$$
\mathbf{r}_T^\top \mathbf{M}_\lambda \mathbf{r}_T \geq \left(1 - \frac{\alpha}{2}\right) \cdot \sigma_1(\mathbf{M}_\lambda) - \frac{\alpha}{2} \cdot \sigma_1(\mathbf{M}_\lambda) = (1 - \alpha) \cdot \sigma_1(\mathbf{M}_\lambda)
$$

For the second item (accurate regime), observe that

$$
(\mathbf{r}_t^\top \mathbf{r}^*)^2 = \left( (\mathbf{r}^*_t)^\top \mathbf{r}^* + (\mathbf{r}_t - \mathbf{r}^*_t)^\top \mathbf{r}^* \right)^2 \geq \left( (\mathbf{r}^*_t)^\top \mathbf{r}^* \right)^2 - 2 \|\mathbf{r}_t - \mathbf{r}^*_t\| \tag{24}
$$

Our choices of $T$ and $\widetilde{\epsilon}$ make sure that $\left( (\mathbf{r}^*_T)^\top \mathbf{r}^* \right)^2 \geq 1 - \frac{\alpha}{2}$ by Lemma 3 and that $\|\mathbf{r}^*_T - \mathbf{r}_T\| \leq R_T = \alpha/4$ by Lemma 4. Continuing from (24), we have

$$
(\mathbf{r}_T^\top \mathbf{r}^*)^2 \geq 1 - \frac{\alpha}{2} - \frac{\alpha}{2} = 1 - \alpha
$$

□

## E.2 Iteration complexity of Algorithm 4

Observe that, the **for** loops within the **repeat-until** loop, as well as the final **for** loop in Algorithm 4 are running inexact power iterations on $\mathbf{M}_{\lambda_{(s)}}$ and $\mathbf{M}_{\lambda_{(f)}}$ for $m_1$ and $m_2$ inexact matrix-vector multiplication respectively. And the convergence of inexact power iterations is provided by Lemma 4.

For each iteration of the **repeat-until** loop, we work in the crude regime and only require $\mathbf{r}_{sm_1}$ to give a constant multiple estimate of $\mathbf{M}_{\lambda_{(s)}}$. The lemma below shows an important property of $\Delta_s$ which is used to locate $\lambda_{(f)}$, and the number of iterations needed to reach $\lambda_{(f)}$.

**Lemma 6** (Iteration complexity of the **repeat-until** loop in Algorithm 4)**.** *Suppose that* $\widetilde{\Delta} \in [c_1\Delta,\ c_2\Delta]$ *where* $c_2 \leq 1$. *Set* $m_1 = \lceil 8\log\left(\frac{16}{\mu'}\right)\rceil$ *and* $\widetilde{\epsilon} \leq \frac{1}{3084}\left(\frac{\widetilde{\Delta}}{18}\right)^{m_1-1}$ *in Algorithm 4. Then for all* $s \geq 1$ *it holds that*

$$\frac{1}{2}(\lambda_{(s-1)} - \rho_1) \leq \Delta_s \leq \lambda_{(s-1)} - \rho_1$$

*upon exiting this loop, the* $\lambda_{(f)}$ *satisfies*

$$\rho_1 + \frac{\widetilde{\Delta}}{4} \leq \lambda_{(f)} \leq \rho_1 + \frac{3\widetilde{\Delta}}{2} \tag{25}$$

*and the number of iterations run by the* ***repeat-until*** *loop is* $\log\left(\frac{1}{\widetilde{\Delta}}\right)$.

*Proof.* Let $\overline{\sigma}$ be an upper bound of all $\sigma_1(\mathbf{M}_{\lambda_{(s)}})$ used in the **repeat-until** loop, i.e.,

$$\overline{\sigma} \geq \sigma_1(\mathbf{M}_{\lambda_{(s)}}), \qquad s = 1, 2, \ldots$$

And suppose for now that throughout the loop, $\widetilde{\epsilon}$ satisfies

$$\sqrt{\overline{\sigma}\widetilde{\epsilon}} \leq \frac{\sigma_1\left(\mathbf{M}_{\lambda_{(s-1)}}\right)}{8} \tag{26}$$

Set $\alpha = \frac{1}{4}$ in Lemma 4 (crude regime), and with our choice of $m_1$ and

$$\widetilde{\epsilon} \leq \frac{\sigma_d(\mathbf{M}_{\lambda_{(s)}})}{1024\kappa_{\lambda_{(s)}}}\left(\frac{2\kappa_{\lambda_{(s)}} - 1}{\left(2\kappa_{\lambda_{(s)}}\right)^{m_1} - 1}\right)^2 \tag{27}$$

we have

$$\mathbf{r}_{sm_1}^\top \mathbf{M}_{\lambda_{(s-1)}} \mathbf{r}_{sm_1} \geq \frac{3}{4}\sigma_1(\mathbf{M}_{\lambda_{(s-1)}}) \tag{28}$$

In view of the definition of the vector $\mathbf{w}_s$ in Algorithm 4, and following the same argument in (18), we have

$$\left\| \frac{\mathbf{z}_s}{\sqrt{2}} - \mathbf{M}_{\lambda_{(s-1)}} \mathbf{r}_{sm_1} \right\| \leq \sqrt{\sigma_1(\mathbf{M}_{\lambda_{(s-1)}}) \cdot \widetilde{\epsilon}}$$

where $\mathbf{z}_s = \begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} & \\ & \mathbf{\Sigma}_{yy}^{\frac{1}{2}} \end{bmatrix} \mathbf{w}_s$.

Then for every iteration of the **repeat-until** loop, it holds that

$$\begin{aligned}
&\frac{1}{2}\begin{bmatrix}\mathbf{u}_{sm_1}^\top & \mathbf{v}_{sm_1}^\top\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma}_{xx} & \\ & \mathbf{\Sigma}_{yy}\end{bmatrix}\mathbf{w}_s \\
=&\mathbf{r}_{sm_1}^\top\left(\frac{\mathbf{z}_s}{\sqrt{2}}\right) = \mathbf{r}_{sm_1}^\top \mathbf{M}_{\lambda_{(s-1)}}\mathbf{r}_{sm_1} + \mathbf{r}_{sm_1}^\top\left(\frac{\mathbf{z}_s}{\sqrt{2}} - \mathbf{M}_{\lambda_{(s-1)}}\mathbf{r}_{sm_1}\right) \\
\in&\left[\mathbf{r}_{sm_1}^\top \mathbf{M}_{\lambda_{(s-1)}}\mathbf{r}_{sm_1} - \sqrt{\sigma_1(\mathbf{M}_{\lambda_{(s-1)}})\cdot\widetilde{\epsilon}},\ \mathbf{r}_{sm_1}^\top \mathbf{M}_{\lambda_{(s-1)}}\mathbf{r}_{sm_1} + \sqrt{\sigma_1(\mathbf{M}_{\lambda_{(s-1)}})\cdot\widetilde{\epsilon}}\right] \\
\in&\left[\mathbf{r}_{sm_1}^\top \mathbf{M}_{\lambda_{(s-1)}}\mathbf{r}_{sm_1} - \sqrt{\overline{\sigma}\widetilde{\epsilon}},\ \mathbf{r}_{sm_1}^\top \mathbf{M}_{\lambda_{(s-1)}}\mathbf{r}_{sm_1} + \sqrt{\overline{\sigma}\widetilde{\epsilon}}\right]
\end{aligned}$$

where we have used the Cauchy-Schwarz inequality in the second step.

In view of (26) and (28), it follows that

$$\begin{aligned}
&\frac{1}{2}\begin{bmatrix}\mathbf{u}_{sm_1}^\top & \mathbf{v}_{sm_1}^\top\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma}_{xx} & \\ & \mathbf{\Sigma}_{yy}\end{bmatrix}\mathbf{w}_s - \sqrt{\overline{\sigma}\widetilde{\epsilon}} \\
\in&\left[\mathbf{r}_{sm_1}^\top \mathbf{M}_{\lambda_{(s-1)}}\mathbf{r}_{sm_1} - 2\sqrt{\overline{\sigma}\widetilde{\epsilon}},\ \mathbf{r}_{sm_1}^\top \mathbf{M}_{\lambda_{(s-1)}}\mathbf{r}_{sm_1}\right] \\
\in&\left[\frac{1}{2}\sigma_1(\mathbf{M}_{\lambda_{(s-1)}}),\ \sigma_1(\mathbf{M}_{\lambda_{(s-1)}})\right]
\end{aligned}$$

By the definition of $\Delta_s$ in Algorithm 4 and the fact that $\sigma_1(\mathbf{M}_{\lambda_{(s-1)}}) = \frac{1}{\lambda_{(s-1)}-\rho_1}$, we have

$$\Delta_s = \frac{1}{2}\cdot\frac{1}{\frac{1}{2}\begin{bmatrix}\mathbf{u}_{sm_1}^\top & \mathbf{v}_{sm_1}^\top\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma}_{xx} & \\ & \mathbf{\Sigma}_{yy}\end{bmatrix}\mathbf{w}_s - \sqrt{\overline{\sigma}\widetilde{\epsilon}}} \in \left[\frac{1}{2}\left(\lambda_{(s-1)}-\rho_1\right),\ \lambda_{(s-1)}-\rho_1\right] \tag{29}$$

And as a result,

$$\lambda_{(s)} = \lambda_{(s-1)} - \frac{\Delta_s}{2} \geq \lambda_{(s-1)} - \frac{1}{2}\left(\lambda_{(s-1)}-\rho_1\right) = \frac{\lambda_{(s-1)}+\rho_1}{2}$$

and thus by induction (note $\lambda_{(0)} \geq \rho_1$) we have $\lambda_{(s)} \geq \rho_1$ throughout the **repeat-until** loop.

From (29) we also obtain

$$\lambda_{(s)} - \rho_1 = \lambda_{(s-1)} - \rho_1 - \frac{\Delta_s}{2} \leq \lambda_{(s-1)} - \rho_1 - \frac{1}{4}\left(\lambda_{(s-1)}-\rho_1\right) = \frac{3}{4}\left(\lambda_{(s-1)}-\rho_1\right)$$

To sum up, $\lambda_{(s)}$ approaches $\rho_1$ from above and the gap between $\lambda_{(s)}$ and $\rho_1$ reduces at the geometric rate of $\frac{3}{4}$. Thus after at most $t_3 = \lceil \log_{3/4}\left(\frac{\widetilde{\Delta}}{\lambda_{(0)}-\rho_1}\right)\rceil \sim \mathcal{O}\left(\log\left(\frac{1}{\widetilde{\Delta}}\right)\right)$ iterations, we reach a $\lambda_{(t_3)}$ such that $\lambda_{(t_3)} - \rho_1 \leq \widetilde{\delta}$. And in view of (29), the **repeat-until** loop exits in the next iteration. Hence, the overall number of iterations is at most $t_3 + 1 = \mathcal{O}\left(\frac{1}{\widetilde{\Delta}}\right)$.

We now analyze $\lambda_{(f)}$ and derive the interval it lies in. Note that $\Delta_f \leq \widetilde{\Delta}$ and $\Delta_{f-1} > \widetilde{\Delta}$ by the exiting condition. In view of (29), we have

$$\lambda_{(f)} - \rho_1 = \lambda_{(f-1)} - \rho_1 - \frac{\Delta_f}{2} \leq 2\Delta_f - \frac{\Delta_f}{2} = \frac{3\Delta_f}{2} \leq \frac{3\widetilde{\Delta}}{2}$$

On the other hand,

$$\lambda_{(f)} - \rho_1 = \lambda_{(f-1)} - \rho_1 - \frac{\Delta_f}{2} \geq \lambda_{(f-1)} - \rho_1 - \frac{1}{2}\left(\lambda_{(f-1)} - \rho_1\right) = \frac{1}{2}\left(\lambda_{(f-1)} - \rho_1\right) \tag{30}$$

If $f = 1$, then by our choice of $\lambda_{(0)}$ we have that $\lambda_{(f)} - \rho_1 \geq \widetilde{\Delta}$. Otherwise, by unfolding (30) one more time, we have that

$$\lambda_{(f)} - \rho_1 \geq \frac{1}{4}\left(\lambda_{(f-2)} - \rho_1\right) \geq \frac{\Delta_{f-1}}{4} \geq \frac{\widetilde{\Delta}}{4}$$

Thus in both case, we have that $\lambda_{(f)} - \rho_1 \geq \frac{\widetilde{\Delta}}{4}$ holds.

It remains to give an explicit bound on $\widetilde{\epsilon}$ based on the two requirements (26) and (27). Since the $\lambda_{(s)}$ values are monotonically non-increasing and lower-bounded by $\rho_1 + \frac{\widetilde{\Delta}}{4}$, we have

$$\max_s \ \sigma_1(\mathbf{M}_{\lambda_{(s)}}) = \sigma_1(\mathbf{M}_{\lambda_{(f)}}) = \frac{1}{\lambda_{(f)} - \rho_1} \leq \frac{4}{\widetilde{\Delta}} =: \overline{\sigma}$$

and

$$\begin{aligned}\min_s \ \sigma_1(\mathbf{M}_{\lambda_{(s)}}) = \sigma_1(\mathbf{M}_{\lambda_{(0)}}) &= \frac{1}{\lambda_{(0)} - \rho_1} = \frac{1}{1 + \widetilde{\Delta} - \rho_1} \\ &\geq \frac{1}{1 + c_2\Delta - \Delta} \geq 1 + (1 - c_2)\Delta \geq 1 + \frac{1 - c_2}{c_2}\widetilde{\Delta} := \underline{\sigma}\end{aligned}$$

where the first inequality holds since by definition of $\Delta$ it follows that $\rho_1 = \rho_2 + \Delta \geq \Delta$.

Therefore, for the assumption (26) to hold, we just need

$$\left(\frac{\underline{\sigma}}{8\sqrt{\overline{\sigma}}}\right)^2 = \frac{\left(1 + \frac{1-c_2}{c_2}\widetilde{\Delta}\right)^2}{64 \cdot \frac{4}{\widetilde{\Delta}}} \geq \frac{1}{64 \cdot \frac{4}{\widetilde{\Delta}}} = \frac{\widetilde{\Delta}}{256} \geq \widetilde{\epsilon} \tag{31}$$

We now derive a lower bound of the right hand side of (27). Notice

$$\kappa_{\lambda_{(s)}} = \frac{\lambda_{(s)} + \rho_1}{\lambda_{(s)} - \rho_1} = 1 + \frac{2\rho_1}{\lambda_{(s)} - \rho_1} \leq 1 + 2\rho_1\overline{\sigma} \leq 1 + 2\overline{\sigma} \leq \frac{9}{\widetilde{\Delta}} \tag{32}$$

On the other hand,

$$\sigma_d(\mathbf{M}_{\lambda_{(s)}}) \geq \sigma_d(\mathbf{M}_{\lambda_{(0)}}) = \frac{1}{\lambda_{(0)} + \rho_1} = \frac{1}{1 + \widetilde{\Delta} + \rho_1} \geq \frac{1}{3}$$

As a result, we have

$$\begin{aligned}\frac{\sigma_d(\mathbf{M}_{\lambda_{(s)}})}{1024\kappa_{\lambda_{(s)}}}\left(\frac{2\kappa_{\lambda_{(s)}} - 1}{\left(2\kappa_{\lambda_{(s)}}\right)^{m_1} - 1}\right)^2 &\geq \frac{1}{3084 \cdot \frac{9}{\widetilde{\Delta}}}\left(\frac{2\frac{9}{\widetilde{\Delta}} - 1}{\left(2\frac{9}{\widetilde{\Delta}}\right)^{m_1} - 1}\right)^2 \geq \frac{\left(\frac{17}{\widetilde{\Delta}}\right)^2}{3084 \cdot \frac{9}{\widetilde{\Delta}} \cdot \left(\frac{18}{\widetilde{\Delta}}\right)^{m_1}} \\ &\geq \frac{1}{3084}\left(\frac{\widetilde{\Delta}}{18}\right)^{m_1 - 1}\end{aligned} \tag{33}$$

Our final bound on $\widetilde{\epsilon}$ chooses the smaller of (31) and (33). □

For the final **for** loop of Algorithm 4, we work in the accurate regime of power iterations.

**Lemma 7** (Iteration complexity of the final **for** loop in Algorithm 4)**.** *Suppose that* $\widetilde{\Delta} \in [c_1\Delta, c_2\Delta]$ *where* $c_2 \leq 1$*. Set* $m_2 = \lceil \frac{5}{4} \log\left(\frac{128}{\widetilde{\mu}\eta^2}\right) \rceil$ *and* $\widetilde{\epsilon} \leq \frac{\eta^4}{4^{10}} \left(\frac{\widetilde{\Delta}}{18}\right)^{m_2-1}$ *in Algorithm 4. Then the* $(\mathbf{u}_T, \mathbf{v}_T)$ *output by Phase I satisfies*

$$\frac{1}{4}(\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}^* + \mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}^*)^2 \geq 1 - \frac{\eta^2}{64} \tag{34}$$

*Proof.* Notice when $\lambda = \rho_1 + c(\rho_1 - \rho_2)$, we have

$$\delta(\mathbf{M}_\lambda) = \frac{\sigma_1(\mathbf{M}_\lambda)}{\sigma_1(\mathbf{M}_\lambda) - \sigma_2(\mathbf{M}_\lambda)} = \frac{\frac{1}{\lambda - \rho_1}}{\frac{1}{\lambda-\rho_1} - \frac{1}{\lambda-\rho_2}} = \frac{\lambda - \rho_2}{\rho_1 - \rho_2} = \frac{\rho_1 + c(\rho_1 - \rho_2) - \rho_2}{\rho_1 - \rho_2} = c + 1$$

In view of (25), $\lambda_{(f)} - \rho_1 \leq \frac{3}{2}\widetilde{\Delta} \leq \frac{3c_2}{2}\Delta \leq \frac{3}{2}\Delta$, and thus $\delta(\mathbf{M}_{\lambda_{(f)}}) \leq \frac{5}{2}$.

Set $\alpha = \frac{\eta^2}{64}$ in Lemma 4 (accurate regime), and with our choice of $m_2$ and

$$\widetilde{\epsilon} \leq \frac{\eta^4 \cdot \sigma_d(\mathbf{M}_{\lambda_{(f)}})}{64^3 \cdot \kappa_{\lambda_{(f)}}} \left(\frac{2\kappa_{\lambda_{(f)}} - 1}{\left(2\kappa_{\lambda_{(f)}}\right)^{m_2} - 1}\right)^2 \tag{35}$$

we are guaranteed to obtained the desired alignment.

We now give a lower bound of the right hand side of (35). First,

$$\sigma_d(\mathbf{M}_{\lambda_{(f)}}) = \frac{1}{\lambda_{(f)} + \rho_1} \geq \frac{1}{\rho_1 + \frac{3}{2}\Delta + \rho_1} \geq \frac{1}{4}$$

Recall that we have proved in (32) that $\kappa_{\lambda_{(f)}} \leq \frac{9}{\widetilde{\Delta}}$. Following a derivation similar to that of (33), we have

$$\frac{\eta^4 \cdot \sigma_d(\mathbf{M}_{\lambda_{(f)}})}{64^3 \cdot \kappa_{\lambda_{(f)}}} \left(\frac{2\kappa_{\lambda_{(f)}} - 1}{\left(2\kappa_{\lambda_{(f)}}\right)^{m_2} - 1}\right)^2 \geq \frac{\eta^4}{4^{10}} \left(\frac{\widetilde{\Delta}}{18}\right)^{m_2-1} \tag{36}$$

and this explains the $\epsilon$ we set in the lemma. □

***Proof of Theorem 3.*** As shown in Lemma 7, the **repeat-until** loop runs $\mathcal{O}\left(\log\left(\frac{1}{\widetilde{\Delta}}\right)\right) \sim \mathcal{O}\left(\log\left(\frac{1}{\Delta}\right)\right)$ iterations, and inside each iteration, we run $m_1$ approximate matrix-vector multiplications. On the other hand, the final **for** loop runs $m_2$ approximate matrix-vector multiplications. By the definitions of $m_1$ and $m_2$, the total number of invocations of approximate matrix-vector multiplications/least squares problems is

$$m_1 \cdot \log\left(\frac{1}{\Delta}\right) + m_2 \sim \mathcal{O}\left(\log\left(\frac{1}{\widetilde{\mu}}\right)\log\left(\frac{1}{\Delta}\right) + \log\left(\frac{1}{\widetilde{\mu}\eta^2}\right)\right) \sim \widetilde{\mathcal{O}}(1)$$

□

## F Proof of Theorem 4

*Proof.* Notice that the eigenvectors of $\mathbf{M}_\lambda$ form an orthonormal bases of $\mathbb{R}^{d_x+d_y}$. Thus when (34) holds, i.e., the alignment between $\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_T \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_T \end{bmatrix}$ and $\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}^* \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}^* \end{bmatrix}$ is large, the alignments between $\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_T \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_T \end{bmatrix}$ and other eigenvectors have to be small. In particular, the alignment bewteen $\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widetilde{\mathbf{u}}_T \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widetilde{\mathbf{v}}_T \end{bmatrix}$ and the tailing eigenvector $\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}^* \\ -\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}^* \end{bmatrix}$ has to be small:

$$(\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}^* - \mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}^*)^2 \le \frac{\eta^2}{16} \tag{37}$$

From (37) and (34), we have respectively

$$-\frac{\eta}{4} \le \left|\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}^*\right| - \left|\mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}^*\right| \le \frac{\eta}{4},$$
$$\left|\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}^*\right| + \left|\mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}^*\right| \ge 2\sqrt{1-\frac{\eta^2}{64}} \ge 2\left(1-\frac{\eta}{8}\right)$$

where we have used the fact that $\sqrt{1-x} \ge 1-\sqrt{x}$ for $x \in [0,1]$ in the second inequality.

Averaging the above two inequalities gives

$$\left|\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}^*\right| \ge 1-\frac{\eta}{4}, \qquad \left|\mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}^*\right| \ge 1-\frac{\eta}{4}$$

Finally,

$$\begin{aligned}
(\widehat{\mathbf{u}}^\top \mathbf{\Sigma}_{xx}\mathbf{u}^*)^2 + (\widehat{\mathbf{v}}^\top \mathbf{\Sigma}_{yy}\mathbf{v}^*)^2 &= \frac{(\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}^*)^2}{\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}_T} + \frac{(\mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}^*)^2}{\mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}_T} \\
&\ge (1-\frac{\eta}{4})^2\left(\frac{1}{\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}_T} + \frac{1}{\mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}_T}\right) \\
&\ge \left(1-\frac{\eta}{4}\right)^2 \frac{4}{\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}_T + \mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}_T} \\
&\ge 2\left(1-\frac{\eta}{2}\right) = 2-\eta
\end{aligned}$$

where we have used the fact that $\frac{1}{x}+\frac{1}{y} \ge \frac{4}{x+y}$ in the first inequality, and (10) in the second inequality. Then the theorem follows from the fact that $(\widehat{\mathbf{u}}^\top \mathbf{\Sigma}_{xx}\mathbf{u}^*)^2$ and $(\widehat{\mathbf{v}}^\top \mathbf{\Sigma}_{yy}\mathbf{v}^*)^2$ can be at most 1. □

## G Condition number of $h_t$ for SVRG

**Lemma 8.** *Throughout Algorithm 4, the condition number of $h_t$ for SVRG is at most $\frac{9/c}{\Delta}\widetilde{\kappa}$, where*

$$\widetilde{\kappa} := \frac{\max_i \max\left(\|\mathbf{x}_i\|^2, \|\mathbf{y}_i\|^2\right)}{\min\left(\sigma_{\min}(\mathbf{\Sigma}_{xx}), \sigma_{\min}(\mathbf{\Sigma}_{yy})\right)}$$

**Table 2:** Brief summary of datasets.

| Datasets | Description | $d_x$ | $d_y$ | $N$ |
|---|---|---|---|---|
| Mediamill | Image and its labels | 100 | 120 | 30,000 |
| JW11 | Acoustic and articulation measurements | 273 | 112 | 30,000 |
| MNIST | Left and right halves of images | 392 | 392 | 60,000 |

*Proof.* The gradient Lipschitz constant of $h_t^i(\mathbf{u}, \mathbf{v})$ is bounded by the largest eigenvalue (in absolute value) of its Hessian[7]

$$\mathbf{Q}_\lambda^i = \begin{bmatrix} \lambda \mathbf{x}_i \mathbf{x}_i^\top & -\mathbf{x}_i \mathbf{y}_i^\top \\ -\mathbf{y}_i \mathbf{x}_i^\top & \lambda \mathbf{y}_i \mathbf{y}_i^\top \end{bmatrix}$$

and the largest eigenvalue is defined as

$$\max_{\mathbf{g}_x \in \mathbb{R}^{d_x}, \mathbf{g}_y \mathbb{R}^{d_y}} \beta := \left| [\mathbf{g}_x^\top, \mathbf{g}_y^\top] \mathbf{Q}_\lambda^i \begin{bmatrix} \mathbf{g}_x \\ \mathbf{g}_y \end{bmatrix} \right| \qquad \text{s.t.} \quad \|\mathbf{g}_x\|^2 + \|\mathbf{g}_y\|^2 = 1$$

We have

$$\begin{aligned}
\beta &= \left| \lambda (\mathbf{g}_x^\top \mathbf{x}_i)^2 + \lambda (\mathbf{g}_y^\top \mathbf{y}_i)^2 - 2 (\mathbf{g}_x^\top \mathbf{x}_i)(\mathbf{g}_y^\top \mathbf{y}_i) \right| \\
&\leq \lambda (\mathbf{g}_x^\top \mathbf{x}_i)^2 + \lambda (\mathbf{g}_y^\top \mathbf{y}_i)^2 + 2 \left| \mathbf{g}_x^\top \mathbf{x}_i \right| \left| \mathbf{g}_y^\top \mathbf{y}_i \right| \\
&\leq \lambda (\mathbf{g}_x^\top \mathbf{x}_i)^2 + \lambda (\mathbf{g}_y^\top \mathbf{y}_i)^2 + (\mathbf{g}_x^\top \mathbf{x}_i)^2 + (\mathbf{g}_y^\top \mathbf{y}_i)^2 \\
&= (\lambda + 1) \left( (\mathbf{g}_x^\top \mathbf{x}_i)^2 + (\mathbf{g}_y^\top \mathbf{y}_i) \right) \\
&\leq (\lambda + 1) \left( \|\mathbf{g}_x\|^2 \|\mathbf{x}_i\|^2 + \|\mathbf{g}_y\|^2 \|\mathbf{y}_i\|^2 \right) \\
&\leq (\lambda + 1) \max \left( \|\mathbf{x}_i\|^2, \|\mathbf{y}_i\|^2 \right)
\end{aligned}$$

where we have used the Cauchy-Schwarz inequality and the constraint in the third and the last inequality respectively.

It only remains to bound $\frac{\lambda+1}{\lambda-\rho}$. Note that we have shown in Lemma 6 that $\lambda \geq \rho_1 + \frac{\widetilde{\Delta}}{4}$ throughout Algorithm 4, and thus

$$\frac{\lambda + 1}{\lambda - \rho} = 1 + \frac{1 + \rho}{\lambda - \rho} \leq 1 + \frac{2}{\lambda - \rho} \leq 1 + 2\frac{4}{\widetilde{\Delta}} \leq \frac{9}{\widetilde{\Delta}} \leq \frac{9/c_1}{\Delta}$$

□

# H More details of the experiments

The statistics of these datasets are summaized in Table 2. These datasets have also been used by [Ma et al., 2015, Wang et al., 2015] for demonstrating their stochastic CCA algorithms.

We now provide additional details for the experiments. For `s-AppGrad`, both gradient and normalization steps are estimated with mini-batchs of 100 samples (the authors of [Ma et al., 2015] suggest that the mini-batch size shall be at least the same magnitude as the dimensionality of the

[7] We omit the regularization terms, which are typically very small, to have concise expressions.

CCA projection). For `SI-VR` and `SI-AVR`, within the **repeat-until** loop, we apply SVRG with $M = 2$ epochs to approximately find the top eigenvector $\mathbf{w}_s$, and SVRG with $M = 2$ epochs to approximately calculate its top eigenvalue of $\mathbf{M}_{\lambda_{(s)}}$ as $\mathbf{w}_s^T \mathbf{M}_{\lambda_{(s)}} \mathbf{w}_s$. We exit the **repeat-until** loop when $\Delta_s \leq 0.06$. Afterwards, for the fixed $\lambda_{(f)}$, we apply SVRG to solve every least squares problems with $M = 4$ epochs. Each epoch of SVRG includes a batch gradient evaluation and $m = N$ stochastic gradient steps. We set the step size according to the smoothness for each least squares solver, i.e., $\frac{1}{\sigma_{\max}(\mathbf{\Sigma}_{xx})}$ for GD/AGD in `AppGrad`/`s-AppGrad`/`CCALin`, and $\frac{1}{\max_i \|\mathbf{x}_i\|^2}$ for SVRG/ASVRG in our algorithms.

# I Other related work

Recent years have witnessed continuous efforts to scale up fundamental methods such as principal component analysis (PCA) and partial least squares with stochastic/online updates [Warmuth & Kuzmin, 2008, Arora et al., 2012, Balsubramani et al., 2013, Shamir, 2015, Xie et al., 2015, Garber & Hazan, 2015, Jin et al., 2015]. But as pointed out by [Arora et al., 2012], the CCA objective is more challenging due to the constraints.

[Yger et al., 2012] proposed an adaptive CCA algorithm with efficient online updates based on matrix manifolds defined by the constraints. However, the goal of their algorithm is anomaly detection for streaming data with a varying distribution, rather than to optimize the CCA objective on a given dataset. Similar to our algorithms, the stochastic CCA algorithms of [Ma et al., 2015, Wang et al., 2015] are motivated by the ALS formulation. [Xie et al., 2015] proposed a stochastic algorithm based on the Lagrangian formulation of the objective (1). None of these online/stochastic algorithms have rigorous global convergence guarantee.