Journal of Machine Learning Research 22 (2021) 1-71 Submitted 1/20; Revised 4/21; Published 8/21

Integrated Principal Components Analysis

Tiffany M. Tang TIFFANY.TANG@BERKELEY.EDU
Department of Statistics
University of California, Berkeley
Berkeley, CA 94720, USA

Genevera I. Allen GALLEN@RICE.EDU
Department of Electrical and Computer Engineering
Rice University
Houston, TX 77005, USA

Editor: Garvesh Raskutti

Abstract

Data integration, or the strategic analysis of multiple sources of data simultaneously, can often lead to discoveries that may be hidden in individualistic analyses of a single data source. We develop a new unsupervised data integration method named Integrated Principal Components Analysis (iPCA), which is a model-based generalization of PCA and serves as a practical tool to find and visualize common patterns that occur in multiple data sets. The key idea driving iPCA is the matrix-variate normal model, whose Kronecker product covariance structure captures both individual patterns within each data set and joint patterns shared by multiple data sets. Building upon this model, we develop several penalized (sparse and non-sparse) covariance estimators for iPCA, and using geodesic convexity, we prove that our non-sparse iPCA estimator converges to the global solution of a non-convex problem. We also demonstrate the practical advantages of iPCA through extensive simulations and a case study application to integrative genomics for Alzheimer's disease. In particular, we show that the joint patterns extracted via iPCA are highly predictive of a patient's cognition and Alzheimer's diagnosis.

Keywords: data integration, multi-view data, matrix-variate normal, dimension reduction, integrative genomics

1. Introduction

The recent growth in both data volume and variety drives the need for principled data integration methods that can analyze multiple sources of data simultaneously. For instance, meteorologists must integrate data from satellites, ground-based sensors, and numerical models for forecasting (Ghil and Malanotte-Rizzoli, 1991). Audio and video are often combined for surveillance as well as speech recognition (Shivappa et al., 2010); and as new high-throughput technologies arise in biology, scientists are leveraging information from multiple genomic sources to better understand complex biological processes (Huang et al., 2017). By exploiting the commonalities and diversity of information from different data sets, data integration methods have the potential to provide a holistic and perhaps more realistic model of the phenomena at hand.

In this work, we focus on facilitating unsupervised learning tasks such as pattern recognition, dimension reduction, and visualization for integrated data in various applications. More specifically, we consider the multi-view data setting, where we observe multiple data sources (or matrices) with features of different types that are measured on the same set of samples. Our goal in this setting is to develop a practical statistical data integration method that 1) leverages multiple data sources to discover and visualize dominant joint patterns among the samples that are common across all data sets; 2) generalizes the classical principal components analysis (PCA), thereby inheriting its nice properties including easily-interpretable visualizations, a unique solution, and nested, orthogonal components that can be quickly obtained all at once; and 3) has provable statistical and optimization guarantees to begin bridging the gap between the theory and practice of data integration.

To this end, we propose Integrated Principal Components Analysis (iPCA), which extends a model-based framework of the classical Principal Components Analysis (PCA) to integrated data. As such, iPCA inherits the many advantages and nice familiar interpretations of PCA. The key idea here behind iPCA is to leverage a new but natural connection between data integration and the matrix-variate normal model. This enables us to flexibly model a rich set of dependencies among features and samples simultaneously, as is often crucial in integrated data. Building upon this model, we develop novel penalized covariance estimators for iPCA including a new geodesically convex penalty with both theoretical and practical advantages. While the main contributions of this work are methodological and applied in nature, we also begin to study the theoretical properties of our approach. In particular, we show that our non-sparse iPCA estimator converges to the global solution of a non-convex problem using geodesic convexity, and in the Appendix, we show that our sparse iPCA estimator consistently estimates the underlying joint subspace. Finally, through simulations and a careful case study application to integrative genomics for Alzheimer's disease, we demonstrate the superior empirical performance of iPCA for integrative exploratory data analysis, joint pattern recognition, and visualization. Specifically, in our case study, iPCA is able to identify joint patterns that are biologically-meaningful and can separate patients by their cognitive capabilities and Alzheimer's diagnosis. This interesting finding and application to Alzheimer's disease not only demonstrates the practical utility of iPCA but also points to promising hypotheses for follow-up experiments in Alzheimer's research.

1.1 Related Work

Currently, existing data integration methods for unsupervised learning tend to fall into one of two categories—the matrix factorizations or the multiblock PCA family—but each have major practical limitations that we address with iPCA.

First, matrix factorizations, including coupled matrix and tensor (CMTF) factorizations (Singh and Gordon, 2008; Acar et al., 2014) and Joint and Individual Variation Explained (JIVE) (Lock et al., 2013), each solve an optimization problem to factorize the integrated data sets into a low-rank joint variation matrix, encoding the shared patterns, and a low-rank individual variation matrix, encoding the patterns specific each data set. Matrix factorizations are well-liked due to their simplicity and computational feasibility, but a practical challenge with these methods is that they do not have a unique solution and heavily depend on the ranks of the factorized matrices. That is, the top factors from CMTF and JIVE are non-nested and can change drastically depending on the user-specified rank. Since the "optimal" rank is almost always unknown and must be specified a priori, this poses significant interpretability challenges as the practitioner could end up with many different, but equally valid, solutions corresponding to different choices of ranks and random seeds. On a related front, the generalized SVD (GSVD) (Alter et al., 2003; Ponnapalli et al., 2011) provides an exact matrix decomposition for integrated data that does not depend on the matrix rank. Nevertheless, it is limited in scope since the GSVD assumes each matrix has full row rank, excluding problems with both high- and low-dimensional data sets.

On the other hand, the multiblock PCA family, which includes methods such as Multiple Factor Analysis (MFA) (Escofier and Pages, 1994; Abdi et al., 2013) and Consensus PCA (Westerhuis et al., 1998), works to generalize PCA to the integrated data regime and does not suffer from the rank-dependence or interpretability issues as in matrix factorizations. In these multiblock PCA methods, each data matrix is normalized according to a specific procedure, and then PCA is performed on the normalized concatenated data. However, these normalization schemes are often ad-hoc, and it is unclear which scheme works best and for which situation. Closely related to this is Distributed PCA (Fan et al., 2017), which integrates data that are stored across multiple servers and implicitly assumes that the data are *i.i.d.* across the different servers. This assumption differs from our target setting, where we allow for heterogeneity among the different sources (or servers in the distributed context).

To date, an unsupervised data integration method, which both generalizes PCA and automatically determines the best way to normalize for the different scales and signal strengths between sources, does not exist. Motivated by these limitations, we develop iPCA as a proper generalization of PCA, thus inheriting its many properties and advantages (e.g., a unique solution, easily-interpretable visualizations, and nested, orthogonal principal components) unlike the matrix factorizations; and unlike the multiblock PCA methods, iPCA automatically adjusts for the different scales and signals between data sources without the need for a specified normalization scheme. The two main building blocks of iPCA are PCA and the matrix-variate normal distribution, which we review next.

1.2 Principal Components Analysis

Given a column-centered data set $\mathbf{X} \in \mathbb{R}^{n \times p}$ with n samples and p features, recall that PCA finds orthogonal directions $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^p$, which maximize the covariance $\mathbf{\Delta} \in \mathcal{S}^p_{++}$, where $\mathcal{S}^p_{++}$ is the set of $p \times p$ positive definite matrices. That is, for each $j = 1, \ldots, m$,

$$\mathbf{v}_j = \underset{\mathbf{v} \in \mathbb{R}^p}{\operatorname{argmax}} \; \mathbf{v}^T \mathbf{\Delta} \, \mathbf{v} \quad \text{subject to} \;\; \mathbf{v}^T \mathbf{v} = 1, \;\; \mathbf{v}^T \mathbf{v}_i = 0 \; \forall \, i < j. \tag{1}$$

It is well-known that the PC loading $\mathbf{v}_j$ is the eigenvector of $\mathbf{\Delta}$ with the j^{th} largest eigenvalue, and its corresponding PC score is $\mathbf{u}_j := \mathbf{X} \, \mathbf{v}_j$. In practice, since the population covariance $\mathbf{\Delta}$ is typically unknown, the sample version of PCA plugs in an estimate $\hat{\mathbf{\Delta}} := \frac{1}{n} \mathbf{X}^T \mathbf{X}$ for $\mathbf{\Delta}$ in (1). It follows that the PC loadings are the eigenvectors of $\hat{\mathbf{\Delta}}$, and the PC scores are the scaled eigenvectors of $\hat{\mathbf{\Sigma}} := \frac{1}{p} \mathbf{X} \, \mathbf{X}^T$. To later establish the link between iPCA and PCA, we point out that $\hat{\mathbf{\Delta}}$ is the maximum likelihood estimator (MLE) of $\mathbf{\Delta}$ under the multivariate normal model $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{iid}{\sim} N(\mathbf{0}, \mathbf{\Delta})$ and $\hat{\mathbf{\Sigma}}$ is the MLE of $\mathbf{\Sigma}$ under $\mathbf{x}'_1, \ldots, \mathbf{x}'_p \overset{iid}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{x}_i$ is the i^{th} row of $\mathbf{X}$, and $\mathbf{x}'_j$ is the j^{th} column of $\mathbf{X}$. Thus, there is a dual row/column interpretation of the PCA model. While this is not the only way of viewing PCA, this formulation illustrates two points which we explore further in iPCA: (1) PCA finds linear projections of the data that maximize the variance under a multivariate normal model; (2) eigenvectors correspond to the dominant (or variance-maximizing) patterns in the data.

1.3 Matrix-variate Normal Model

Laid out in Gupta and Nagar (1999) and Dawid (1981), the matrix-variate normal distribution is an extension of the multivariate normal distribution such that the matrix is the unit of study. Formally, we say $\mathbf{X} \in \mathbb{R}^{n \times p}$ follows a matrix-variate normal distribution and write $\mathbf{X} \sim N_{n,p}(\mathbf{M}, \mathbf{\Sigma} \otimes \mathbf{\Delta})$ if $\text{vec}(\mathbf{X}^T)$ follows a multivariate normal distribution with a Kronecker product covariance structure, $\text{vec}(\mathbf{X}^T) \sim N(\text{vec}(\mathbf{M}^T), \mathbf{\Sigma} \otimes \mathbf{\Delta})$. Here, $\text{vec}(\mathbf{X}) \in \mathbb{R}^{np}$ is the column vector formed by stacking the columns of $\mathbf{X}$ below one another. We call $\mathbf{M} \in \mathbb{R}^{n \times p}$ the mean matrix, $\mathbf{\Sigma} \in \mathcal{S}^n_{++}$ the row covariance matrix, and $\mathbf{\Delta} \in \mathcal{S}^p_{++}$ the column covariance matrix.

Put differently, the row covariance matrix $\mathbf{\Sigma}$ encodes the dependencies between rows of $\mathbf{X}$ while the column covariance matrix $\mathbf{\Delta}$ encodes the dependencies among columns, i.e., $\mathbf{X}_{i,\cdot} \sim N(\mathbf{M}_{i,\cdot}, \mathbf{\Sigma}_{ii}\,\mathbf{\Delta})$ and $\mathbf{X}_{\cdot,j} \sim N(\mathbf{M}_{\cdot,j}, \mathbf{\Delta}_{jj}\,\mathbf{\Sigma})$. It can also be shown that if $\mathbf{\Sigma} = \mathbf{I}$ and $\mathbf{M} = \mathbf{0}$, we are in the familiar multivariate normal setting, $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{iid}{\sim} N(\mathbf{0}, \mathbf{\Delta})$, and if $\mathbf{\Delta} = \mathbf{I}$ and $\mathbf{M} = \mathbf{0}$, then $\mathbf{x}'_1, \ldots, \mathbf{x}'_p \overset{iid}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$. The matrix-variate normal model, however, is far more general than the multivariate normal. While the multivariate normal can only model relationships between elements of a single row or a single column in $\mathbf{X}$, the matrix-variate normal can model relationships between elements from different rows and columns. With this level of flexibility, the matrix-variate normal has proven to be a versatile tool in various contexts such as graphical models (Yin and Li, 2012; Tsiligkaridis et al., 2013; Zhou, 2014a), spatio-temporal models (Greenewald and Hero, 2015), and transposable models (Allen and Tibshirani, 2010). Our work on iPCA is the first to consider the matrix-variate normal model in light of data integration.

1.4 Outline

Building upon the matrix-variate normal model, we introduce iPCA as a proper generalization of PCA to the integrated data regime in Section 2. In Section 3, we discuss covariance estimation methods for iPCA and begin to study some of their theoretical properties. We then demonstrate the strong empirical performance of iPCA in Section 4 through simulations and a real data application to integrative genomics for Alzheimer's disease. Finally, we conclude with a discussion of iPCA in Section 5.

2. Integrated PCA

At its core, iPCA, like PCA, is an unsupervised tool for exploratory data analysis, pattern recognition, and visualization. Unlike PCA however, iPCA aims to extract dominant joint patterns which are *common* to multiple data sets, not necessarily the variance-maximizing patterns since they might be specific to one data set. These joint patterns are typically of considerable interest to practitioners as its common occurrence in multiple data sets

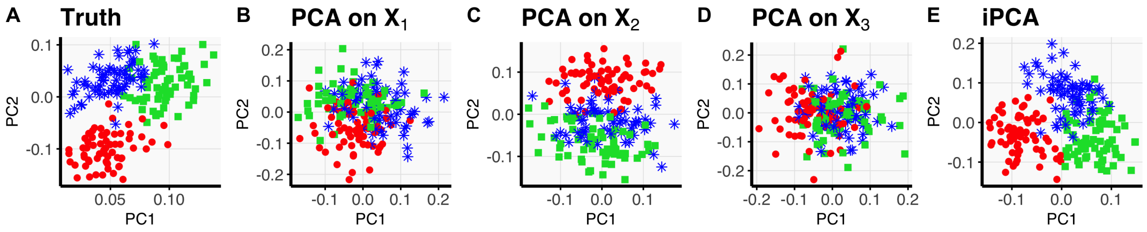

Figure 1: Coupled matrices $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ with $n = 200$, $p_1 = 300$, $p_2 = 500$, $p_3 = 400$ were simulated from the iPCA model (2). Here, $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \mathbf{\Delta}_2, \mathbf{\Delta}_3$ were taken to be as in the base simulation described in Section 4.1. (A) plots the top two eigenvectors of $\mathbf{\Sigma}$. In separate PCA analyses (B-D), the individual signal in each data set masks the true joint signal, but (E) iPCA (using the multiplicative Frobenius estimator) exploits the integrated data structure and recovers the true joint signal.

may point to some foundational mechanism or structure. For instance, scientists may be more interested in uncovering the patterns (or clusters) of patients who have similar gene expression levels and miRNA expression levels than those patients with similar gene expression levels alone.

Figure 1 illustrates a motivating example for when iPCA is advantageous. In the example, strong dependencies among features obscure the true joint patterns among the samples so that the true joint signal is not the variance-maximizing direction. As a result, applying PCA separately to each of the data sets (panels B-D) fails to reveal the joint signal. iPCA can better recover the joint signal because it exploits the known integrated data structure and extracts the shared information among all three data sets simultaneously.

Generally speaking, iPCA finds these joint patterns by modeling the dependencies between and within data sets via the matrix-variate normal model. The inherent Kronecker product covariance structure enables us to decompose the total covariance of each data matrix into two components—an individual column covariance structure which is unique to each data set and a joint row covariance structure which is shared among all data sets. The joint row covariance structure is our primary interest, and maximizing this joint variation will yield the dominant patterns which are common across all data sets. In the following sections, we will introduce iPCA and provide interpretations and intuition into the model.

2.1 Population Model of iPCA

Suppose we observe K coupled data matrices, $\mathbf{X}_1, \ldots, \mathbf{X}_K$, of dimensions $n \times p_1, \ldots, n \times p_K$, where n is the number of samples and p_k is the number of features in $\mathbf{X}_k$. Throughout this paper, we let $p := \sum_{k=1}^{K} p_k$ and $\tilde{\mathbf{X}} := [\mathbf{X}_1, \ldots, \mathbf{X}_K]$. Since iPCA is primarily interested in the data variation, let us assume that each data matrix $\mathbf{X}_k$ has been mean centered so that each column of $\mathbf{X}_k$ has a mean of 0. Suppose also that each of the data matrices are measured on the same n samples and that all rows of $\mathbf{X}_k$ are perfectly aligned (see

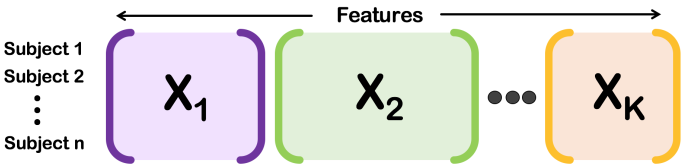

Figure 2: Integrated Data Setting for iPCA: We observe K different but coupled data matrices, each with a distinct set of features that are measured on the same set of n samples. Assume that the rows align.

Figure 2). Under the iPCA model, we assume that each data set $\mathbf{X}_k$ arises from a matrix-variate normal distribution,

$$\mathbf{X}_k \overset{ind.}{\sim} N_{n,p_k}(\mathbf{0},\ \mathbf{\Sigma} \otimes \mathbf{\Delta}_k), \qquad (k = 1, \dots, K) \tag{2}$$

where $\mathbf{\Sigma}$ is an $n \times n$ row covariance matrix that is jointly shared by all data matrices, and $\mathbf{\Delta}_k$ is a $p_k \times p_k$ column covariance matrix that is specific to $\mathbf{X}_k$. We next provide additional intuition into the model parameters $\mathbf{\Sigma}, \mathbf{\Delta}_1, \dots, \mathbf{\Delta}_K$ and unpack the iPCA modeling assumptions.

Feature Dependencies $\mathbf{\Delta}_k$: By properties of the matrix-variate normal, we can interpret $\mathbf{\Delta}_k$ as describing the dependence structure among features in $\mathbf{X}_k$, giving rise to feature patterns unique to $\mathbf{X}_k$. As a consequence of the iPCA model in (2), this feature (or column) dependence is revealed as

$$[\mathbf{X}_k]_{i,\cdot} \sim N(\mathbf{0},\ \mathbf{\Sigma}_{ii}\, \mathbf{\Delta}_k), \qquad (k = 1, \dots, K,\ i = 1, \dots, n). \tag{3}$$

In other words, each sample or row in $\mathbf{X}_k$ has its own variance scaling factor given by $\mathbf{\Sigma}_{ii}$ while all dependencies among the features are captured by $\mathbf{\Delta}_k$.

Sample Dependencies $\mathbf{\Sigma}$: Analogously, we can interpret $\mathbf{\Sigma}$ as describing the common row dependence structure, corresponding to patterns among the samples that are shared by all K data sets. According to the iPCA model in (2), the sample (or row) dependence manifests itself as

$$[\mathbf{X}_k]_{\cdot,j} \sim N(\mathbf{0},\ [\mathbf{\Delta}_k]_{jj}\, \mathbf{\Sigma}), \qquad (k = 1, \dots, K,\ j = 1, \dots, p_k). \tag{4}$$

Here, each feature or column in $\mathbf{X}_k$ gets its own variance scaling factor $[\mathbf{\Delta}_k]_{jj}$ while all dependencies among the samples are given by $\mathbf{\Sigma}$.

iPCA Through a Whitening Lens: In addition to these marginal interpretations of $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$, it is also important to gain intuition into how $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$ interact together within the iPCA model. The simplest way to do so is through a whitening perspective, where we note that the iPCA model in (2) can be equivalently rewritten as follows: for each $k = 1, \dots, K$,

$$\mathbf{\Sigma}^{-1/2}\, \mathbf{X}_k\, \mathbf{\Delta}_k^{-1/2} = \mathbf{Z}_k, \qquad \text{where } [\mathbf{Z}_k]_{ij} \overset{iid}{\sim} N(0, 1). \tag{5}$$

This is to say that after whitening each data matrix $\mathbf{X}_k$ of both its feature dependencies in $\mathbf{\Delta}_k$ and the shared row dependencies in $\mathbf{\Sigma}$, the remainder is distributed *i.i.d.* from a standard normal distribution. Thus, all dependencies in $\mathbf{X}_k$ must be captured by $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$. This assumption may seem strong at first glance, but it is a simple generalization of the usual assumptions in PCA.

To see this, notice another equivalence relation to the iPCA model (2) based on whitening: for each $k = 1, \ldots, K$,

$$\mathbf{Y}_k := \mathbf{X}_k\, \mathbf{\Delta}_k^{-1/2} \overset{ind.}{\sim} N_{n,p_k}(\mathbf{0},\, \mathbf{\Sigma} \otimes \mathbf{I}_{p_k}), \tag{6}$$

or equivalently, in vector-form,

$$[\mathbf{X}_k\, \mathbf{\Delta}_k^{-1/2}]_{\cdot,j} \overset{iid}{\sim} N(\mathbf{0},\, \mathbf{\Sigma}) \quad \text{for each } j = 1, \ldots, p_k. \tag{7}$$

That is, in the idealistic scenario when population covariances are known, the iPCA model simply states that each data set $\mathbf{X}_k$, after being whitened of its feature dependencies $\mathbf{\Delta}_k$, follows a normal distribution with the common row covariance $\mathbf{\Sigma}$. This is not a new assumption. In fact, it is the primary modeling assumption if we were to apply classical PCA to the concatenated centered and whitened data $[\mathbf{Y}_1, \ldots, \mathbf{Y}_K]$. Further, like PCA, while iPCA assumes and works best with normally-distributed data, it is still practically effective in the non-normal regime. We provide empirical evidence of this robustness in Section 4.

It is important to note, however, that when $\mathbf{\Delta}_k \neq \mathbf{I}$ such as in most cases with real data, the underlying iPCA model greatly differs from that of applying PCA to the concatenated un-whitened data (henceforth referred to as concatenated PCA). While the normal assumption in iPCA is often insignificant in practice, it is crucial to account for $\mathbf{\Delta}_k$ in the integrated data model. Intuitively, the feature dependencies $\mathbf{\Delta}_k$ in iPCA can be viewed as nuisance parameters that obscure the sample dependencies $\mathbf{\Sigma}$ and vice versa, so ignoring the feature dependencies completely as in concatenated PCA can be extremely problematic. The challenge here though is that $\mathbf{\Delta}_k$ and $\mathbf{\Sigma}$ are both unknown in practice and must be estimated. In the setting where we only observe one data set $\mathbf{X}_1$, it is not possible to distinguish or estimate both $\mathbf{\Delta}_1$ and $\mathbf{\Sigma}$, but with multiple observed data sets $\mathbf{X}_1, \ldots, \mathbf{X}_K$, we can and should leverage the additional data to distinguish between the feature and sample dependencies. As we will see later, this is an upshot of iPCA—namely, that iPCA models and estimates both $\mathbf{\Delta}_k$ and $\mathbf{\Sigma}$ concurrently, exploiting the integrated structure while also accounting for the opposition's nuisance effects.

Now under these modeling assumptions from the iPCA model in (2), which we have seen to be generalizations of the classical PCA assumptions, iPCA extends the variance-maximizing ideas of PCA and achieves its objective of finding the dominant joint and individual patterns in the data by maximizing the joint row covariance $\mathbf{\Sigma}$ and individual column covariances $\mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_K$ simultaneously. Namely, for $k = 1, \ldots, K$, iPCA solves

$$\mathbf{u}_i = \underset{\mathbf{u} \in \mathbb{R}^n}{\text{argmax}}\ \mathbf{u}^T \mathbf{\Sigma}\, \mathbf{u} \qquad \text{subject to } \mathbf{u}^T \mathbf{u} = 1,\ \mathbf{u}^T \mathbf{u}_l = 0\ \forall\, l < i, \quad (i = 1, \ldots, n) \tag{8}$$

$$\mathbf{v}_j^k = \underset{\mathbf{v} \in \mathbb{R}^{p_k}}{\text{argmax}}\ \mathbf{v}^T \mathbf{\Delta}_k\, \mathbf{v} \qquad \text{subject to } \mathbf{v}^T \mathbf{v} = 1,\ \mathbf{v}^T \mathbf{v}_l^k = 0\ \forall\, l < j, \quad (j = 1, \ldots, p_k) \tag{9}$$

for which we know the solution to be given by the eigendecompositions of $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$, respectively. That is, $\mathbf{u}_i$ is the eigenvector of $\mathbf{\Sigma}$ with the i^{th} largest eigenvalue, and $\mathbf{v}_j^k$

is the eigenvector of $\mathbf{\Delta}_k$ with the j^{th} largest eigenvalue. Most notably, $\mathbf{u}_1$ maximizes the joint variation and is interpreted as the most dominant pattern among the samples, which occurs in all K data sets. We call the columns of $\mathbf{U} := [\mathbf{u}_1, \dots, \mathbf{u}_n]$ the integrated principal component (iPC) scores and the columns of $\mathbf{V}_k := [\mathbf{v}_1^k, \dots, \mathbf{v}_{p_k}^k]$ the iPC loadings for the k^{th} data set. Note that though the population covariances in (2) are not identifiable (e.g., $\mathbf{\Sigma} \otimes \mathbf{\Delta}_k = c\,\mathbf{\Sigma} \otimes \frac{1}{c}\,\mathbf{\Delta}_k$ for $c \in \mathbb{R}$), the iPC scores and loadings are identifiable since eigenvectors are scale-invariant.

Since we are often most interested in the joint patterns, we primarily plot the iPC scores $\mathbf{U}$ to visualize the joint patterns in sample space. To visualize the individual feature patterns from the k^{th} data set, we can also plot the iPC loadings $\mathbf{V}_k$.

2.2 Sample Version of iPCA

To perform iPCA in practice, we must typically plug in estimators $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}_k$ for $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$ since the population covariances in (2) are almost always unknown. We summarize the sample version of iPCA as follows:

1. **Model** each (column-centered) data set via a matrix-variate normal model: $\mathbf{X}_k \sim N_{n,p_k}(\mathbf{0},\ \mathbf{\Sigma} \otimes \mathbf{\Delta}_k),\ k = 1, \dots, K$.

2. **Estimate** the covariance matrices $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \dots, \mathbf{\Delta}_K$ simultaneously to obtain $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}_1, \dots, \hat{\mathbf{\Delta}}_K$. Methods for covariance estimation will be discussed in Section 3.

3. Compute the **eigenvectors**, say $\hat{\mathbf{U}}$ = eigenvectors of $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{V}}_k$ = eigenvectors of $\hat{\mathbf{\Delta}}_k$. We interpret $\hat{\mathbf{U}}$ as the dominant joint patterns in sample space and $\hat{\mathbf{V}}_k$ as the dominant patterns in feature space which are specific to $\mathbf{X}_k$.

4. **Visualize** and **explore** the dominant joint patterns by plotting the iPC scores $\hat{\mathbf{U}}$ and the dominant individual patterns by plotting the iPC loadings $\hat{\mathbf{V}}_k$.

2.2.1 Variance Explained by iPCA

After performing iPCA, we can also interpret the signal in the obtained iPCs through a notion of variance explained, analogous to that in PCA.

Definition 1 *Assume that $\mathbf{X}_k$ has been column centered. We define the cumulative proportion of variance explained in data set $\mathbf{X}_k$ by the top m iPCs to be*

$$PVE_{k,m} := \frac{\| (\mathbf{U}^{(m)})^T \mathbf{X}_k \mathbf{V}_k^{(m)} \|_F^2}{\| \mathbf{X}_k \|_F^2}, \tag{10}$$

where $\mathbf{U}^{(m)} = [\mathbf{u}_1, \dots, \mathbf{u}_m]$ are the top m iPC scores, and $\mathbf{V}_k^{(m)} = [\mathbf{v}_1^k, \dots, \mathbf{v}_m^k]$ are the top m iPC loadings associated with $\mathbf{X}_k$.

Definition 2 *The marginal proportion of variance explained in data set $\mathbf{X}_k$ by the m^{th} iPC is defined as $MPVE_{k,m} := PVE_{k,m} - PVE_{k,m-1}$.*

We verify in Appendix A that $\text{PVE}_{k,m}$ is a proportion and monotonically increasing as m increases. Aside from being well-defined, we also show in Appendix A that (10) generalizes the cumulative proportion of variance explained in PCA and hence, is a natural definition. Note however that unlike PCA, it may be that $\text{MPVE}_{k,m+1} > \text{MPVE}_{k,m}$ in iPCA. This is because iPCA does not maximize the *total* variance. For example, if $\text{MPVE}_{1,2} > \text{MPVE}_{1,1}$, this simply means that data set $\mathbf{X}_1$ contributed more variation to the joint pattern in iPC2 than in iPC1.

2.3 Connections to Existing Methods

Throughout our development of iPCA, we have established several connections between iPCA and PCA, which demonstrate that iPCA is indeed a natural extension of PCA. We also find it instructive to draw on connections between iPCA and other existing data integration methods to develop an even deeper understanding of iPCA.

2.3.1 Relationship to Multiblock PCA Family

As discussed in Abdi et al. (2013), multiblock PCA methods reduce to performing PCA on the normalized concatenated data $\tilde{\mathbf{X}}' = [\mathbf{X}'_1, \ldots, \mathbf{X}'_K]$, where each $\mathbf{X}_k$ has been normalized to $\mathbf{X}'_k$ according to some procedure. We will show later in Proposition 6 that performing PCA on the unnormalized concatenated data (i.e., concatenated PCA) is a special case of iPCA, where we assume $\boldsymbol{\Delta}_k = \mathbf{I}$ for each k. Proposition 6 can also be easily extended to show that multiblock PCA methods are a special case of iPCA for some fixed $\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_K$, and the exact form of $\boldsymbol{\Delta}_k$ depends on the normalization procedure. For example, since MFA normalizes $\mathbf{X}_k$ by dividing all of its entries by its largest singular value $\sigma_{\max,k}$, MFA is a special case of iPCA, where each $\boldsymbol{\Delta}_k = \sigma_{\max,k}\,\mathbf{I}$. Put differently, MFA assumes $[\mathbf{X}_k\,\sigma_{\max,k}^{-1/2}]_{\cdot,j} \overset{iid}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma})$ whereas iPCA assumes $[\mathbf{X}_k\,\boldsymbol{\Delta}_k^{-1/2}]_{\cdot,j} \overset{iid}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma})$.

This gives rise to another interpretation of iPCA: iPCA is a generalization and unifying framework for the entire multiblock PCA family. However, while the multiblock PCA methods assume that $\boldsymbol{\Delta}_k$ takes a specific form, iPCA does not impose any restrictions on the form of $\boldsymbol{\Delta}_k$ and instead freely estimates the full $\boldsymbol{\Delta}_k$ matrix simultaneously with $\boldsymbol{\Sigma}$. In doing so, iPCA acts as an automatic way of normalizing for the different scales and signals between data sources. Without appropriate normalization, the estimated principal components from multi-block PCA methods such as concatenated PCA will be biased as quantified by the following proposition.

Proposition 3 *Suppose that* $\mathbf{X}_k \sim N_{n,p_k}(\mathbf{0}, \boldsymbol{\Sigma} \otimes \boldsymbol{\Delta}_k)$ *for each* $k = 1, \ldots, K$, *and define* $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_K]$, $\boldsymbol{\Delta} = diag(\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_K)$, *and* $\mathbf{Z} = [\mathbf{Z}_1, \ldots, \mathbf{Z}_K]$, *where* $\mathbf{Z}_k$ *is as defined in* (5) *previously. If* $\mathbf{X}\,\boldsymbol{\Delta}^{-1/2} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ *and* $\mathbf{X} = \tilde{\mathbf{U}}\tilde{\mathbf{D}}\tilde{\mathbf{V}}^T$ *are their respective compact SVDs, then* $\mathbf{U}$ *and* $\tilde{\mathbf{U}}$ *denote the iPCA and concatenated PCA scores, respectively, and*

$$\mathbf{U} - \tilde{\mathbf{U}} = \boldsymbol{\Sigma}^{1/2}\,\mathbf{Z}(\mathbf{V}\mathbf{D}^{-1} - \boldsymbol{\Delta}^{1/2}\,\tilde{\mathbf{V}}\tilde{\mathbf{D}}^{-1}). \tag{11}$$

Moreover, the iPCA and concatenated PCA scores are equal if and only if $\mathbf{V}\mathbf{D}^{-1} - \boldsymbol{\Delta}^{1/2}\,\tilde{\mathbf{V}}\tilde{\mathbf{D}}^{-1}$ *is in the nullspace of* $\mathbf{Z}$ *or* $\boldsymbol{\Delta}^{1/2}\,\tilde{\mathbf{V}} = \mathbf{V}\mathbf{D}^{-1}\tilde{\mathbf{D}}$.

Proposition 3 implies that if the feature dependencies, captured by $\boldsymbol{\Delta}_k$, rotate the data $\mathbf{X}_k$ in a way such that it does not obscure the joint row patterns, then concatenated PCA will work adequately. However, for most situations and examples of $\mathbf{\Delta}_k$ that we expect to find in real data, this is unlikely to occur, and the resulting eigenvectors of concatenated PCA will be severely biased. To avoid this bias, both the sample and feature dependencies must be accounted for in the estimation procedure as done in iPCA.

2.3.2 Relationship to Matrix Factorizations

Coupled matrix factorizations (CMF) decompose each data set $\mathbf{X}_k \in \mathbb{R}^{n \times p_k}$ into the product of low-rank joint factor $\mathbf{U} \in \mathbb{R}^{n \times m}$ and a low-rank individual factor $\mathbf{V}_k \in \mathbb{R}^{m \times p_k}$ so that $\mathbf{X}_k \approx \mathbf{U}\,\mathbf{V}_k$ (Singh and Gordon, 2008; Acar et al., 2014). This approximate factorization is related to iPCA in that our matrix-variate normal model assumes a similar multiplicative structure. Specifically, from (5), the iPCA model can be viewed as $\mathbf{X}_k = \mathbf{\Sigma}^{1/2}\,\mathbf{Z}_k\,\mathbf{\Delta}_k^{1/2}$, where $\mathbf{Z}_k$ is standard normal random matrix. Moreover, an argument similar to Theorem 2 from Hastie et al. (2015) shows that one solution of the CMF optimization problem (with ℓ_2 penalties) is the solution of concatenated PCA and thereby a special case of iPCA.

Despite this connection however, there is a fundamental difference between CMF and iPCA. On the one hand, CMF assumes $\mathbf{X}_k$ can be approximated by a low-rank matrix, and the estimation of the CMF factors actively depends on the pre-specified rank m. On the other hand, the rank of $\mathbf{X}_k$ plays absolutely no role in the iPCA assumptions or the estimation step of iPCA. Consequently, the joint and individual CMF factors can change drastically depending on the pre-specified rank unlike in iPCA. CMF also does not have a unique solution nor enforces orthogonal components whereas iPCA gives unique, nested, orthogonal components that can be interpreted in the same way as in PCA.

In contrast to the multiplicative models of CMF and iPCA, JIVE, which has been commonly used in integrative genomics, assumes an additive model and decomposes coupled data into the sum of a low-rank joint variation matrix, a low-rank individual variation matrix, and an error matrix (Lock et al., 2013). Additive and multiplicative models, being quite different models, are each advantageous in different situations, but as with CMF, the estimation of JIVE depends on the pre-specified ranks of its factors and results in non-nested, rank-dependent joint and individual components.

3. Covariance Estimators for iPCA

We next return to address the covariance estimation step when fitting the iPCA model to data. In Section 3.1, we consider the traditional maximum likelihood approach but find that it suffers from substantial limitations in the integrated data setting. These limitations ultimately drive the need for new estimators, which we develop in Section 3.2.

3.1 Unpenalized Maximum Likelihood Estimators

Guided by the formulation of PCA in Section 1.2, we instinctively try to estimate the iPCA population covariances $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_K$ via maximum likelihood estimation. Under the iPCA population model (2), the log-likelihood function reduces to

$$\ell(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}) \propto p \log |\,\mathbf{\Sigma}^{-1}\,| + n \sum_{k=1}^{K} \log |\,\mathbf{\Delta}_k^{-1}\,| - \sum_{k=1}^{K} \mathrm{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T\right), \quad (12)$$

so by taking partial derivatives of (12) with respect to each parameter, we obtain

Lemma 4 *The unpenalized MLEs of $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \dots, \mathbf{\Delta}_K$ satisfy*

$$\hat{\mathbf{\Sigma}} = \frac{1}{p} \sum_{k=1}^{K} \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T \tag{13}$$

$$\hat{\mathbf{\Delta}}_k = \frac{1}{n} \mathbf{X}_k^T \, \hat{\mathbf{\Sigma}}^{-1} \, \mathbf{X}_k \quad \forall\, k = 1, \dots, K. \tag{14}$$

However, with only one matrix observation per matrix-variate normal model in the iPCA context, existence of the MLE is not guaranteed. In fact, the following theorem essentially implies that the MLE does not exist for all practical purposes.

Theorem 5 *(i) Suppose that $\mathbf{X}_k$ has been column-centered so that each column has a mean of 0 for each $k = 1, \dots, K$. Then the unpenalized MLEs for $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \dots, \mathbf{\Delta}_K$ are not positive definite and hence do not exist.*

(ii) Suppose that $\mathbf{X}_k$ has not been column-centered but that $\operatorname{rank}(\tilde{\mathbf{X}}) = n$ *and* $\operatorname{rank}(\mathbf{X}_k) = p_k$ *for $k = 1, \dots, K$.*

- *(a) If $n \neq p_k$ for some $k = 1, \dots, K$, then the unpenalized log-likelihood function $\ell(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1})$ is unbounded.*
- *(b) If the unpenalized log-likelihood function $\ell(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1})$ is bounded, then the unpenalized MLE for $\mathbf{\Sigma}$, $\mathbf{\Delta}_1, \dots, \mathbf{\Delta}_K$ exist.*

The proof of Theorem 5 also shows that if the unpenalized MLEs for $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \dots, \mathbf{\Delta}_K$ exist, then $p_k = n \leq p$ for each $k = 1, \dots, K$. Thus in summary, if $p_k \neq n$ for some k, $n > p$, or the data matrices have been column-centered to have mean 0, then the unpenalized MLEs do not exist. These severe restrictions motivate new covariance estimators.

For example, one alternative but naive approach is to estimate $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$ by setting their counterparts to $\mathbf{I}$.

Proposition 6 *(i) The MLE for $\mathbf{\Delta}_k$, assuming that $\mathbf{\Sigma} = \mathbf{I}$, is*

$$\hat{\mathbf{\Delta}}_k = \frac{1}{n} \mathbf{X}_k^T \, \mathbf{X}_k \, . \tag{15}$$

(ii) The MLE for $\mathbf{\Sigma}$, assuming $\mathbf{\Delta}_k = \mathbf{I}$ for all $k = 1, \dots, K$, is

$$\hat{\mathbf{\Sigma}} = \frac{1}{p} \sum_{k=1}^{K} \mathbf{X}_k \, \mathbf{X}_k^T = \frac{1}{p} \tilde{\mathbf{X}} \tilde{\mathbf{X}}^T. \tag{16}$$

This approach for estimating $\mathbf{\Sigma}$ is the familiar MLE for the concatenated data $\tilde{\mathbf{X}}$, and hence, performing concatenated PCA is equivalent to a special case of iPCA, where we set $\mathbf{\Delta}_k = \mathbf{I}$ for each k. While this formalizes the connection between PCA and iPCA, we will see in Section 4 that concatenated PCA performs poorly when the data sets are of different scales or when the feature dependencies are stronger than the sample dependencies. In the next section, we discuss more effective methods for estimating the iPCA covariances.

3.2 Penalized Maximum Likelihood Estimators

Given that the unpenalized MLEs do not exist for a large class of problems, one possible solution is to develop penalized MLEs, which solve

$$\begin{aligned}\hat{\mathbf{\Sigma}}^{-1}, \hat{\mathbf{\Delta}}_1^{-1}, \dots, \hat{\mathbf{\Delta}}_K^{-1} = \underset{\substack{\mathbf{\Sigma}^{-1} \succ 0 \\ \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1} \succ 0}}{\text{argmax}} \quad & \Big\{ p \log |\, \mathbf{\Sigma}^{-1} \,| + n \sum_{k=1}^{K} \log |\, \mathbf{\Delta}_k^{-1} \,| \\ & - \sum_{k=1}^{K} \text{tr}\left(\mathbf{\Sigma}^{-1}\, \mathbf{X}_k\, \mathbf{\Delta}_k^{-1}\, \mathbf{X}_k^T \right) \\ & - P(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1}) \Big\}. \qquad (17)\end{aligned}$$

Similar to previous work on the penalized matrix-variate normal log-likelihood (Yin and Li, 2012; Allen and Tibshirani, 2010), we can apply an additive-type penalty and define the additive L_q iPCA penalty to be

$$P_q(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1}) = \lambda_{\mathbf{\Sigma}} \|\, \mathbf{\Sigma}^{-1} \,\|_q + \sum_{k=1}^{K} \lambda_k \|\, \mathbf{\Delta}_k^{-1} \,\|_q,$$

where $\lambda_{\mathbf{\Sigma}}, \lambda_1, \dots, \lambda_K > 0$ are tuning parameters. Though there are many potential choices of norm-penalties here, one natural choice is the additive Frobenius penalty, $\| \cdot \|_q = \| \cdot \|_F^2$, as it is a proper generalization of PCA. That is, performing iPCA with the additive Frobenius penalty is equivalent to PCA in the $K = 1$ case (see Theorem 1 in Allen and Tibshirani 2010). When $K \geq 1$, the additive Frobenius penalty induces a smoothness over the eigenvalues of the covariance matrices and returns a dense full-rank covariance estimator. In the sparse covariance setting, we can instead induce sparsity through the additive L_1 penalty $\| \cdot \|_q = \| \cdot \|_{1,\text{off}}$, where $\|\, \mathbf{A} \,\|_{1,\text{off}} = \sum_{i \neq j} A_{ij}$. Applying the additive L_1 penalty to the inverse covariance matrix is common practice, but one can alternatively apply the additive L_1 penalty to the inverse correlation matrix. This latter approach was adopted in Zhou (2014a) and Rothman et al. (2008). Given the plethora and well-studied nature of sparse graphical model estimation, we leverage existing ideas and tools to show that this approach, namely, the additive L_1 penalty applied to the inverse correlation matrix, consistently estimates the true underlying joint subspace under certain conditions at a rate of $O\left(\sum_{k=1}^{K} \frac{p_k}{p} \sqrt{\frac{\max\{s_{\mathbf{\Sigma}}, 1\} \log(\max\{n, p_k\})}{p_k}} \right)$, where $s_{\mathbf{\Sigma}}$ is the number of non-zero off-diagonal entries in $\mathbf{\Sigma}^{-1}$. However, due to the superior empirical performance of the iPCA Frobenius estimators over the sparse iPCA estimators (see Section 4), we leave the precise theorem statement and discussion of the L_1 subspace consistency result to Appendix F.

Despite the popularity of additive-type penalties in the literature, an overarching downside with these existing penalties in the integrated data regime is that solving (17) with additive penalties is a non-convex problem, for which we can only guarantee convergence to a local solution. Nonetheless, though (17) is non-convex in Euclidean space, Wiesel (2012) showed that the matrix-variate normal log-likelihood is geodesically convex (g-convex) with respect to the manifold of positive definite matrices. G-convexity is a generalized notion of convexity on a Riemannian manifold, and like convexity, all local minima of g-convex functions are globally optimal. Exploiting this idea of g-convexity, we propose a novel type

Algorithm 1 Flip-Flop Algorithm for Multiplicative Frobenius iPCA Estimators

```
Center the columns of $\mathbf{X}_1, \ldots, \mathbf{X}_K$, and initialize $\hat{\boldsymbol{\Sigma}}$, $\hat{\boldsymbol{\Delta}}_1, \ldots, \hat{\boldsymbol{\Delta}}_K$ to be positive definite.
while not converged do
    Take eigendecomposition: $\sum_{k=1}^{K} \mathbf{X}_k \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \mathbf{X}_k^T = \mathbf{U} \, \boldsymbol{\Gamma} \, \mathbf{U}^T$
    Regularize eigenvalues: $\Phi_{ii} = \frac{1}{2p} \left( \Gamma_{ii} + \sqrt{\Gamma_{ii}^2 + 8p \sum_{k=1}^{K} \lambda_k \| \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \|_F^2} \right)$        } Update $\boldsymbol{\Sigma}$
    Update $\hat{\boldsymbol{\Sigma}}^{-1} = \mathbf{U} \, \boldsymbol{\Phi}^{-1} \, \mathbf{U}^T$
    for $k = 1, \ldots, K$ do
        Take eigendecomposition: $\mathbf{X}_k^T \, \hat{\boldsymbol{\Sigma}}^{-1} \, \mathbf{X}_k = \mathbf{V} \, \boldsymbol{\Phi} \, \mathbf{V}^T$
        Regularize eigenvalues: $\Gamma_{ii} = \frac{1}{2n} \left( \Phi_{ii} + \sqrt{\Phi_{ii}^2 + 8n\lambda_k \| \, \hat{\boldsymbol{\Sigma}}^{-1} \, \|_F^2} \right).$        } Update $\boldsymbol{\Delta}_k$
        Update $\hat{\boldsymbol{\Delta}}_k^{-1} = \mathbf{V} \, \boldsymbol{\Gamma}^{-1} \, \mathbf{V}^T$
```

of penalty, named the multiplicative Frobenius iPCA penalty

$$P^*(\boldsymbol{\Sigma}^{-1}, \boldsymbol{\Delta}^{-1}) = \sum_{k=1}^{K} \lambda_k \| \, \boldsymbol{\Sigma}^{-1} \otimes \boldsymbol{\Delta}_k^{-1} \, \|_F^2,$$

which we will show to be g-convex in Theorem 8. Note that since $\| \, \mathbf{A} \otimes \mathbf{B} \, \|_F^2 = \| \, \mathbf{A} \, \|_F^2 \| \, \mathbf{B} \, \|_F^2$, the multiplicative penalty can be rewritten as a product $\| \, \boldsymbol{\Sigma}^{-1} \, \|_F^2 \sum_{k=1}^{K} \lambda_k \| \, \boldsymbol{\Delta}_k^{-1} \, \|_F^2$, giving rise to its name.

Like the additive Frobenius iPCA estimator, the multiplicative Frobenius iPCA estimator is a shrinkage technique that returns a dense covariance estimate with smoothed eigenvalues when $K \geq 1$, and when $K = 1$, it is equivalent to PCA (see Appendix E).

Having introduced several different types of penalized iPCA covariance estimators, namely, the additive Frobenius estimator, multiplicative Frobenius estimator, additive L_1 covariance estimator, and additive L_1 correlation estimator, the question for practitioners becomes how to compute these estimators, how to select the penalty parameters, and which estimator to use in which situation. We discuss each in turn next.

3.2.1 Flip-flop Algorithms for iPCA Estimators

For each of the aforementioned penalties, we can compute the corresponding penalized MLEs via Flip-Flop algorithms (also known as block coordinate descent algorithms), which iteratively optimize over each of the parameters, one at a time, while keeping all other parameters fixed. These algorithms are derived fully in Appendix C.2, but in general, for the Frobenius penalties, each Flip-Flop update has a closed form solution determined by a full eigendecomposition. For the L_1 penalties (also known as the Kronecker Graphical Lasso, Tsiligkaridis et al., 2013), each update can be solved by the graphical lasso (Hsieh et al., 2011). We provide the multiplicative Frobenius Flip-Flop algorithm here in Algorithm 1, and as the other algorithms take similar forms, we leave them to Appendix C.2. The following theorem guarantees numerical convergence of the Flip-Flop algorithms to a local solution for the multiplicative Frobenius, additive Frobenius, and additive L_1 penalties.

Theorem 7 *Suppose that the objective function in* (17) *is bounded below. Suppose also that either (i)* $P(\boldsymbol{\Sigma}^{-1}, \boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_K^{-1})$ *is a differentiable convex function with respect to each coordinate or (ii)* $P(\boldsymbol{\Sigma}^{-1}, \boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_K^{-1}) = P_0(\boldsymbol{\Sigma}^{-1}) + \sum_{k=1}^{K} P_k(\boldsymbol{\Delta}_k^{-1})$*, where* P_i *is* *a (non-differentiable) convex function for each* $k = 1, \ldots, K$*. If either (i) or (ii) holds, then the Flip-Flop algorithm corresponding to* (17) *converges to a stationary point of the objective.*

However, building upon Wiesel (2012) and the notion of g-convexity, we can prove a far stronger result for the multiplicative Frobenius iPCA estimator.

Theorem 8 *The multiplicative Frobenius iPCA estimator is jointly geodesically convex in* $\mathbf{\Sigma}^{-1}$ *and* $\mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}$*. Because of this, the Flip-Flop algorithm for the multiplicative Frobenius iPCA estimator given in Algorithm 1 converges to the global solution.*

There are currently only a handful of non-convex problems where there exists an achievable global solution, so this guarantee that the multiplicative Frobenius iPCA estimator always reaches a global solution is both extremely rare and highly desirable. In Section 4, we will also see that the multiplicative Frobenius iPCA estimator undoubtedly gives the best empirical performance, indicating that in addition to its optimization-theoretic advantages from global convergence, there are significant practical advantages associated with the g-convex penalty. A self-contained review of g-convexity and the proof of Theorem 8 are given in Appendix D.

3.2.2 Tuning Penalty Parameters

To select penalty parameters for (17) in a data-driven manner, we propose a cross-validation-like procedure. The following procedure can also be used to perform iPCA in missing data scenarios.

Let Λ denote the space of penalty parameters, and let $\lambda := (\lambda_{\mathbf{\Sigma}}, \lambda_1, \ldots, \lambda_K)$ be a specific choice of penalty parameters in Λ. The idea is to first randomly leave out scattered elements from each $\mathbf{X}_k$. Then, for each $\lambda \in \Lambda$, impute the missing elements via an Expectation-Maximization-like algorithm, similar to Allen and Tibshirani (2010). Finally, select the λ which minimizes the error between the imputed values and the observed values.

Searching over all combinations of penalty parameters in Λ however can be computationally intractable if K or the data sets themselves are large. In these cases, we can select the penalty parameters in a greedy manner: first fix $\lambda_1, \ldots, \lambda_K$ and optimize over $\lambda_{\mathbf{\Sigma}}$, then fix $\lambda_{\mathbf{\Sigma}}, \lambda_2, \ldots, \lambda_K$ and optimize λ_1, and so forth, and we stop after optimizing λ_K.

Note also that it can be substantially easier and faster to tune the multiplicative Frobenius penalty, which has K penalty parameters, compared to the additive iPCA penalties with $K + 1$ parameters. Because the choice of penalty parameter can significantly impact the empirical performance of iPCA, having one less parameter to tune is an extremely important practical advantage, and we attribute part of the strong empirical performance of the multiplicative Frobenius iPCA estimator, displayed in Section 4, to this advantage.

Details, technical derivations, and numerical results regarding our imputation method and penalty parameter selection are provided in Appendix G.

3.3 Choosing the Type of Penalized iPCA Estimator

As a result of its global convergence guarantee and the reduced complexity in tuning fewer penalty parameters, we strongly recommend using the multiplicative Frobenius estimator in practice. Though we have yet to prove statistical guarantees for the multiplicative Frobenius estimator, the strong empirical performance of the multiplicative Frobenius estimator, seen next in Section 4, firmly supports this recommendation. Even in the sparse setting (see Figure 13), the multiplicative Frobenius estimator performs only slightly worse than the additive L_1 iPCA estimators, for which we have proved subspace consistency guarantees (see Appendix F). This empirically demonstrates the robustness and applicability of the multiplicative Frobenius estimator to a diverse array of problems.

4. Empirical Results

In the following simulations and case study, we evaluate iPCA against individual PCAs on each of the data sets $\mathbf{X}_k$, concatenated PCA, distributed PCA, JIVE, and MFA. Note that many data integration methods from the multiblock PCA family are known to perform similarly to MFA (Abdi et al., 2013), so we only include MFA to minimize redundancy. We also omit CMF as it performs similarly to concatenated PCA and the GSVD since it is not applicable for integrated data with both low-dimensional and high-dimensional data sets.

Our focus here will be on the non-sparse setting while we leave the sparse simulations to Appendix H. Within the class of iPCA estimators, we thus concentrate our attention on the additive and multiplicative Frobenius iPCA estimators in these dense simulations, but to also represent the sparse estimators, we include the most commonly used sparse estimator, the additive L_1 penalty ($\| \cdot \|_{1,\text{off}}$) applied to the inverse covariance matrices. The stopping rule in the Flip-Flop algorithms for the additive and multiplicative Frobenius iPCA estimators is given by $\bar{\boldsymbol{\lambda}}^{1/2} || \hat{\boldsymbol{\Sigma}}_t^{-1} - \hat{\boldsymbol{\Sigma}}_{t-1}^{-1} ||_F / || \hat{\boldsymbol{\Sigma}}_{t-1}^{-1} ||_F < 10^{-6}$, where $\bar{\boldsymbol{\lambda}}$ denotes the mean of the penalty parameters and $\hat{\boldsymbol{\Sigma}}_t$ denotes the estimate of $\boldsymbol{\Sigma}$ in the t^{th} iteration. Due to computational constraints, we stop the L_1 Flip-Flop algorithm after one iteration, and we select the iPCA penalty parameters in a greedy manner, as discussed in Section 3.2.2.

4.1 Simulations

The base simulation is set up as follows: Three coupled data matrices, $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$, with $n = 150$, $p_1 = 300$, $p_2 = 500, p_3 = 400$, were simulated according to the iPCA Kronecker covariance model (2). Here, $\boldsymbol{\Sigma}$ is taken to be a full-rank spiked covariance matrix, where the top two eigenvalues are simulated to be much larger than the rest. These top two factors in $\boldsymbol{\Sigma}$ form the three clusters as shown in Figure 1A. $\boldsymbol{\Delta}_1$ is an autoregressive Toeplitz matrix with entry (i, j) given by $.9^{|i-j|}$; $\boldsymbol{\Delta}_2$ is the observed covariance matrix of miRNA data from TCGA ovarian cancer (The Cancer Genome Atlas Research Network, 2011); and $\boldsymbol{\Delta}_3$ is a block-diagonal matrix with five equally-sized blocks. We also ensured that the largest eigenvalue of each $\boldsymbol{\Delta}_k$ was larger than that of $\boldsymbol{\Sigma}$ so that the joint patterns are intentionally obscured by individualistic patterns. From this base simulation, we systematically varied the parameters—number of samples, number of features, and strength of the joint signal in $\boldsymbol{\Sigma}$ (i.e. $\| \boldsymbol{\Sigma} \|_2$)—one at a time while keeping everything else constant.

We evaluate the performance of various methods using the subspace recovery error: If the true underlying subspace of $\boldsymbol{\Sigma}$ was simulated to be of dimension d with the orthogonal eigenbasis $\mathbf{u}_1, \ldots, \mathbf{u}_d$ and the top d eigenvectors of the estimate $\hat{\boldsymbol{\Sigma}}$ are given by $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_d$, then the subspace recovery error is defined to be $\frac{1}{d} \| \hat{\mathbf{U}} \hat{\mathbf{U}}^T - \mathbf{U} \mathbf{U}^T \|_F^2$, where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_d]$

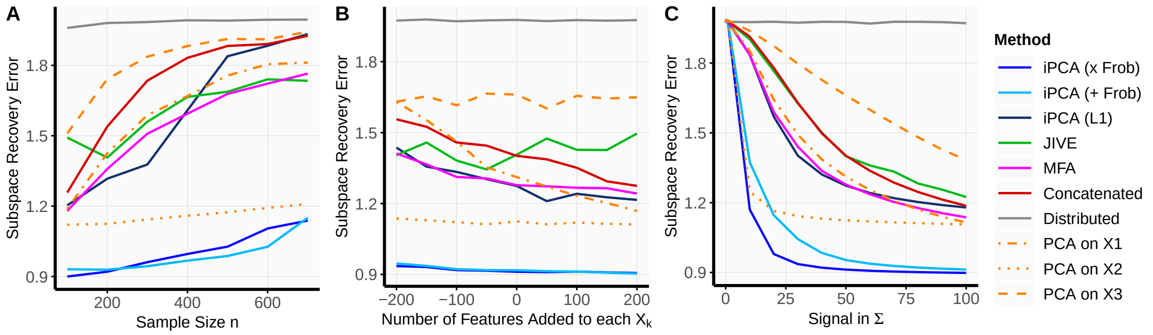

Figure 3: Subspace recovery as simulation pararmeters vary from the base simulation: (A) As the number of samples increases, it becomes more difficult to estimate the joint row subspace; (B) As the number of features increases, it becomes slightly easier to estimate the joint row subspace; (C) Performance drastically improves as the strength of the joint signal in $\mathbf{\Sigma}$ (i.e. the top singular value of $\mathbf{\Sigma}$) increases. Moreover, in almost every scenario, the multiplicative and additive Frobenius iPCA estimators outperform their competitors.

and $\hat{\mathbf{U}} = [\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_d]$. This metric simply quantifies the distance between the true subspace of $\mathbf{\Sigma}$ and the estimated subspace from $\hat{\mathbf{\Sigma}}$. We note that a lower subspace recovery error implies higher estimation accuracy, and in the base simulation, the true subspace of $\mathbf{\Sigma}$ is given by the number of spikes so that $d = 2$. Although there are other metrics like canonical angles, which also quantify the distance between subspaces, these metrics behave similarly to the subspace recovery error and are omitted for brevity.

The average subspace recovery error, measured over 50 trials, from various simulations are shown in Figure 3. We clearly see that the additive and multiplicative Frobenius iPCA estimators consistently outperformed all other methods. Since $\mathbf{\Sigma}$ was not simulated to be sparse, it is no surprise that the Frobenius iPCA estimators outperformed the L_1 iPCA estimator. It is also expected that distributed PCA performs poorly since the $\mathbf{\Delta}_k$'s are not all identical, violating its basic assumption. What may be surprising is that doing PCA on $\mathbf{X}_2$ performed better than its competitors, excluding the Frobenius iPCA estimators. We speculate that this is because the observed covariance $\mathbf{\Delta}_2$ happened to be a very low-rank matrix, and because $\mathbf{\Delta}_2$ was low-rank, the signal from $\mathbf{\Sigma}$ most likely dominated much of the variation in the second PC. Looking ahead at Figure 4A, as Laplacian error was added to the simulated data, PCA on $\mathbf{X}_2$ failed to recover the true signal since the Laplacian error increasingly contributed to the variation in the data. We also point out that MFA always yielded a lower error than concatenated PCA in Figure 3, indicating that there is value in normalizing data sets to be on comparable scales. On the other hand, we must be weary of this normalization process. In the case of these simulations, PCA on $\mathbf{X}_2$ outperformed MFA, illustrating that normalization can sometimes remove important information.

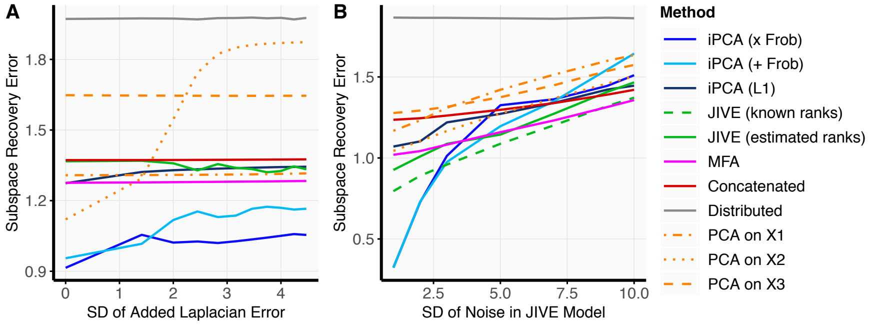

Figure 4: Robustness Simulations: (A) As Laplacian error is increasingly added to the simulated data sets, the Frobenius iPCA estimators appear to be robust to the departures from Gaussianity; (B) As the amount of noise in the JIVE model increases, iPCA seems to be comparable to existing methods, illustrating its relative robustness to departures from the Kronecker product model.

To verify that these simulation results are not heavily dependent by the base simulation setup of $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$, we also ran simulations, varying the dimension d of the true joint subspace $\mathbf{U}$ and the number of data sets K. We provide these results in Appendix H.

Beyond simulating from the iPCA model (2), we check for robustness from the two main iPCA assumptions—normality and separability (i.e., the Kronecker covariance structure). To deviate from normality, we add Laplacian noise to the base simulation setup, and to depart from the Kronecker covariance structure, we simulate data from the JIVE model. The results are summarized in Figure 4, and we leave the simulation details as well as other simulations that demonstrate robustness to Appendix H.

As seen in Figure 4A, the Frobenius iPCA estimators appear to be relatively robust to non-Gaussian noise and outperformed their competitors even as the standard deviation of added Laplacian errors increased. From the simulations under the JIVE model in Figure 4B, we see that as the amount of noise increases, JIVE given the known ranks yields the lowest error, as expected. But similar to how JIVE was comparable to competing methods under the iPCA model (Figure 3), the iPCA estimators are comparable to competing methods for high noise levels under the JIVE model. At low noise levels however, the Frobenius iPCA estimators are surprisingly able to recover the true joint subspace better than JIVE with the known ranks. Further investigation into this peculiar behavior reveals that the Frobenius iPCA estimators give lower subspace recovery errors but much larger approximation errors $\| \mathbf{\Sigma} - \hat{\mathbf{\Sigma}} \|_F^2$, compared to JIVE with the known ranks. This brings up a subtle, but important distinction—iPCA revolves around estimating the underlying subspace, determined by eigenvectors, while JIVE focuses on minimizing the matrix approximation error. These are inherently different objectives, and it is common for iPCA to estimate the eigenvectors well at the cost of a poor matrix approximation due to the regularized eigenvalues.

4.2 Case Study: Integrative Genomics of Alzheimer's Disease

A key motivating example for our research is in integrative genomics, where the goal is to combine multiple genomic sources to gain insights into the genetic basis of diseases. In particular, apart from the APOE gene, little is known about the genomic basis of Alzheimer's disease (AD) and the genes which contribute to dominant expression patterns in AD. In this case study, we delve into the integrative genomics of AD and jointly analyze miRNA expression, gene expression via RNASeq, and DNA methylation data obtained from the Religious Orders Study Memory and Aging Project (ROSMAP) Study (Mostafavi et al., 2018). The ROSMAP study is a longitudinal clinical-pathological cohort study of aging and AD, consisting of 507 subjects, 309 miRNAs, 900 genes, and 1250 CpG (methylation) sites after preprocessing (which we detail in Appendix I). This data is uniquely positioned for the study of AD since its genomics data is collected from post-mortem brain tissue from the dorsolateral prefrontal cortex, an area known to play a critical role in cognitive functions.

For our analysis, we consider two clinical outcomes: clinician's diagnosis and global cognition score. The clinician's diagnosis is the last clinical evaluation prior to the patient's death and is a categorical variable with three levels—Alzheimer's disease (AD), mild cognitive impairment (MCI), and no cognitive impairment (NCI). Global cognition score, a continuous variable, is the average of 19 cognitive tests and is the last cognitive testing score prior to death. While the clinician's diagnosis is sometimes subjective, global cognition score is viewed as a more objective measure of cognition. Our goal is to find common patterns among patients, which occur in all three data sets, and to understand whether these joint patterns are predictive of AD, as measured by the clinician's diagnosis and global cognition score.

To this end, we run iPCA and other existing methods to extract dominant patterns from the ROSMAP data. Figure 5 shows the PC plots obtained from the various methods—each point represents a subject and is colored by either clinician's diagnosis or cognition score.

Since visuals are a subjective measure of performance, we quantify it by taking the top PCs and using them as predictors in a random forest to predict the outcome of interest. The random forest test errors, averaged over 100 random training/test splits, are shown in Figure 6. Here, we see that the joint patterns extracted from iPCA using the multiplicative Frobenius penalty were the most predictive of the clinician's diagnosis of AD and the patient's global cognition score. Moreover, most of the predictive power can be attributed to the first three iPCs, which we visualize in Figure 7A-B. We also note that the top iPCs from iPCA with the multiplicative Frobenius penalties were more predictive than combining the PCs from the three individual PCAs performed on each data set. This showcases empirically that a joint analysis of the integrated data sets can be advantageous over three disparate analyses.

Beyond the high predictive power of iPCA with the multiplicative Frobenius penalty, it is perhaps more important for scientists to be able to interpret the iPCA results. One way is through the proportion of variance explained by the joint iPCs, as defined in Section 2.2.1. Figure 7C shows the marginal proportions for the top 5 iPCs. It reveals that the RNASeq data set contributed the most variation in the joint patterns found by iPC1 and iPC2, and the miRNA data set contributed the most variation in iPC3. More interestingly, even though iPC2 and iPC3 have relatively small variances, iPCA is able to pick out these weak joint signals, which we found to be predictive of AD. This reiterates that the most variable patterns in the data are not necessarily the most relevant patterns for the question at hand. In this case, our goal was to find joint patterns which occur in all three data sets, and since the joint signal is not the most dominant source of variation in each data set, no individualistic PCA analysis would have identified the joint signal found by iPCA.

We conclude our ROSMAP analysis by extracting the top genetic features which are associated to the joint patterns shown in Figure 7. Since iPCA provides an estimate of both $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$, we can select the top features by applying sparse PCA to each $\hat{\mathbf{\Delta}}_k$ obtained from iPCA. Table 1 lists the top miRNAs, genes, and genes affiliated with the selected CpG sites obtained from sparse PCA. Here, we used the `sparseEigen` R package (Benidis et al., 2016) and chose the tuning parameter such that there were only 12 non-zero features.

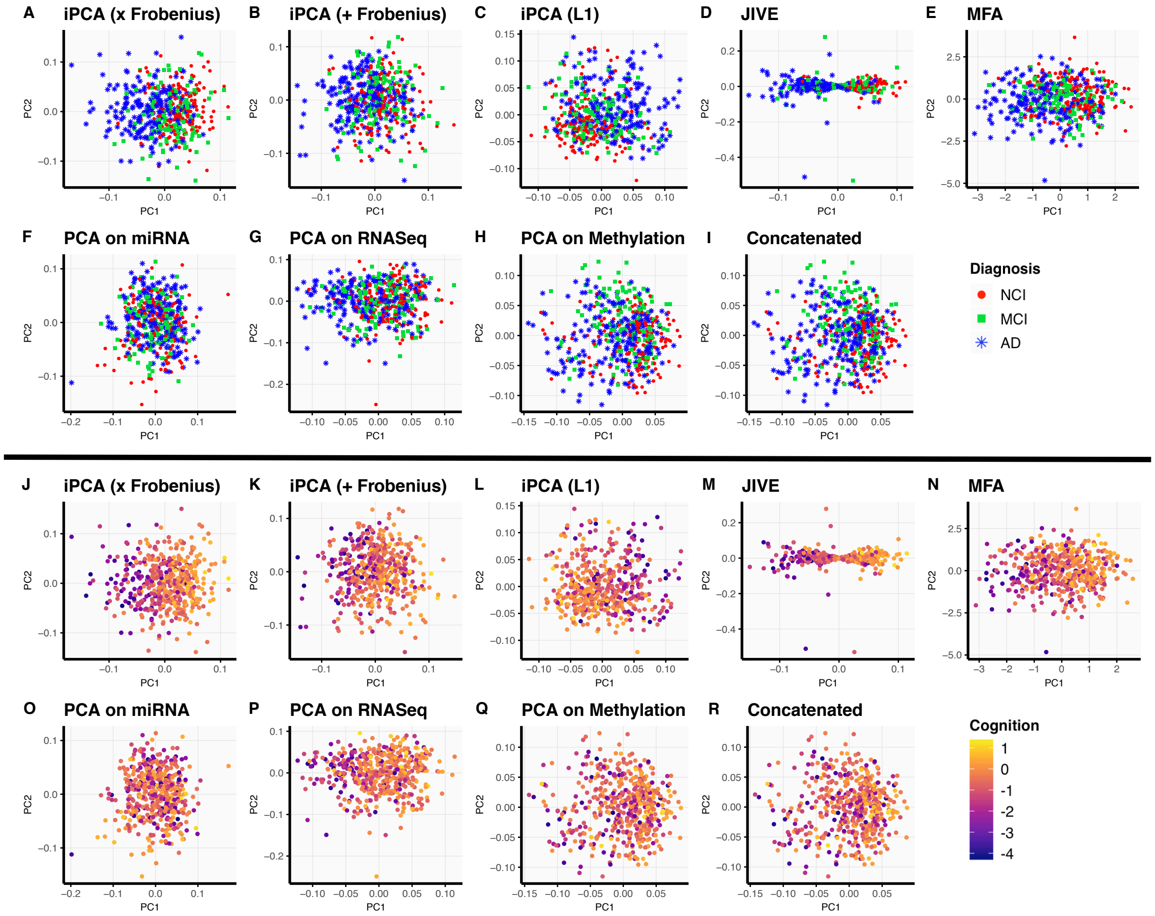

Figure 5: We plot the first two (integrated) principal components from various methods applied to the ROSMAP data. Each point represents a subject, colored by the clinician's diagnosis in panels A-I and by global cognition score in panels J-R.

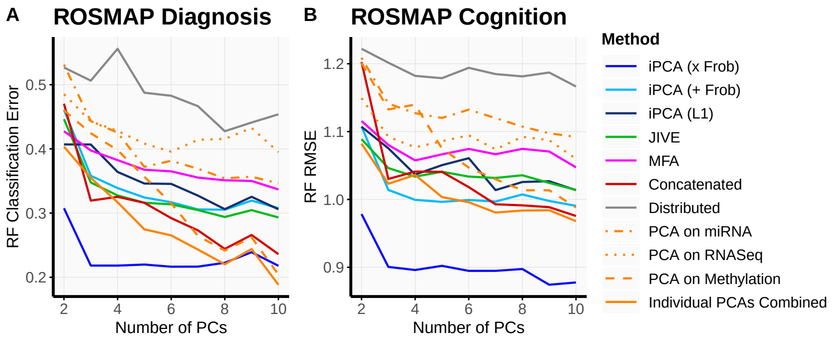

Figure 6: We took the top PCs and used them as predictor variables in a random forest to predict (A) the clinician's diagnosis and (B) the global cognition score. For the random forest, we split the ROSMAP data into a training ($n = 375$) and test set ($n = 132$) and used the default random forest settings in R. The average test error from the random forests over 100 random splits are shown as the number of PCs used in the random forests increases.

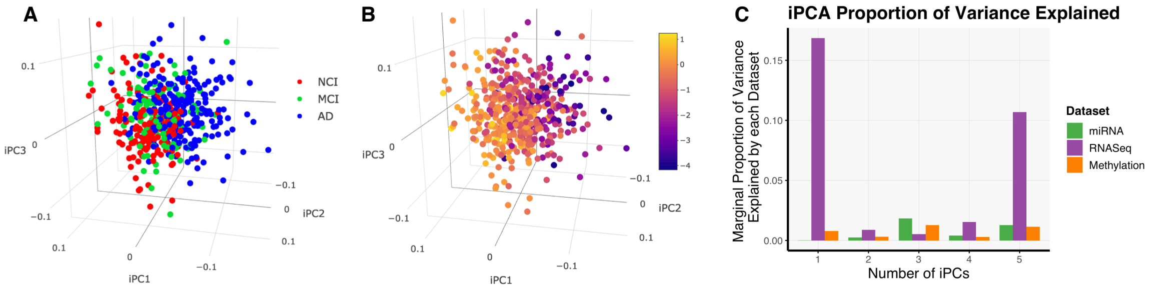

Figure 7: (A)-(B) We show the top 3 iPCs obtained from iPCA with the multiplicative Frobenius estimator; the points are colored by clinician's diagnosis and global cognition score in (A) and (B), respectively. (C) We plot the marginal proportion of variance explained by the top iPCs in each data set in the ROSMAP analysis (using the multiplicative Frobenius iPCA estimator).

Because the RNASeq data contributed most of the variation in iPC1, we did a literature search on the top five genes extracted by sparse PCA on $\hat{\mathbf{\Delta}}_2$. Out of the top five genes, we found evidence in the biological literature, which links four of the five genes (the exception being SVOP) to AD (Carrette et al., 2003; Li et al., 2017; Han et al., 2014; Espuny-Camacho et al., 2017). While this is only a preliminary investigation into the importance of the genetic features obtained from iPCA, it is encouraging evidence and may potentially hint at candidate genes for future research.

	miRNA	**RNASeq**	**Methylation**
1	miR 216a	VGF	TMCO6
2	miR 127 3p	SVOP	PHF3
3	miR 124	PCDHGC5	BRUNOL4
4	miR 30c	ADCYAP1	OSCP1
5	miR 143	LINC01007	GRIN2B
6	miR 27a	FRMPD2L1	CASP9
7	miR 603	SLC30A3	ZFP91; LPXN; ZFP91-CNTF
8	miR 423 3p	NCALD	CNP
9	miR 204	S100A4	YWHAE
10	miR 128	AZGP1	C11orf73
11	miR 193a 3p	PAK1	TMED10
12	ebv miRBART14	MAL2	RELL1

Table 1: Top genetic features obtained by applying Sparse PCA to each $\hat{\mathbf{\Delta}}_k$ in ROSMAP analysis (using the multiplicative Frobenius iPCA estimator)

5. Discussion

As showcased in the simulations and the Alzheimer's disease case study, iPCA is not simply a theoretical construct that generalizes PCA to the integrated data setting. iPCA is also a useful and effective tool in practice to discover interesting joint patterns that are shared across multiple data sets. We believe that iPCA's strong empirical performance is due in part to its flexibility to handle a rich set of dependencies among samples and features concurrently, which is particularly appropriate and necessary for integrated data problems. This flexibility is inherently driven by the underlying matrix-variate normal model, and from a whitening perspective, we can view the iPCA model as a natural generalization of the classical PCA model and assumptions. More specifically, in relation to PCA, iPCA can be viewed as performing PCA on the concatenated feature-whitened data, having estimated the individual feature covariances $\mathbf{\Delta}_k$ and the joint sample covariance $\mathbf{\Sigma}$ simultaneously.

While we discuss many potential penalized iPCA estimators for $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$ in this work, we recommend that practitioners strongly consider using our new multiplicative Frobenius iPCA estimator. The simulations show that the Frobenius penalties are relatively robust to departures from model assumptions, and furthermore, the multiplicative Frobenius penalized estimator requires one less penalty parameter to tune and always converges to the global solution. Similar in spirit to other shrinkage penalties (Ledoit and Wolf, 2004), the multiplicative Frobenius iPCA estimator is a shrinkage technique that induces a smoothness over the eigenvalues of the covariance matrices. However, its multiplicative form is especially unique and well-suited for integrated data problems as it performs automatic reweighting of the integrated data sets and accounts for the concurrent estimation of $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$ in each penalty term. Further investigation into the multiplicative Frobenius penalty and its statistical properties is left for future research.

Moreover, we believe that this work opens the door for research into the theoretical underpinnings of dimension reduction and data integration in ways that other non-model-based methods cannot. Building upon the model-based construction of iPCA, we show in Appendix F that the additive L_1 correlation estimator satisfies one of the first statistical guarantees in the data integration context. However, this is only the beginning and certainly not meant to be the final investigation into theoretical properties of the iPCA estimators and data integration. Minimax results and consistency guarantees for the multiplicative Frobenius iPCA estimator in particular are challenging and will require careful follow up in future work.

Still, there are many other open avenues for future exploration. Analogous to PCA, one might imagine similar fruitful extensions of iPCA to higher-order data, functional data, and other structured applications. One could continue exploring g-convex penalties in different contexts and problems. Another interesting area for future research would be to develop a general framework to prove the consistency of g-convex estimators using the intrinsic manifold space, rather than the Euclidean space. We believe this intersection of g-convexity and statistical theory is a particularly ripe area of future research, but overall, in this work, we developed a theoretically sound and practical tool for performing dimension reduction in the integrated data setting, thus facilitating holistic analyses at a large scale.

Acknowledgments

The authors acknowledge support from NSF DMS-1264058, NSF DMS-1554821 and NSF NeuroNex-1707400. T.T. also acknowledges support from the NSF Graduate Research Fellowship Program DGE-1752814. The authors thank Dr. Joshua Shulman and Dr. David Bennett for help acquiring the ROS/MAP data and acknowledge support from NIH P30AG10161, RF1AG15819, R01AG17917, and R01AG36042 for this data. The authors thank Dr. Zhandong Liu and Dr. Ying-Wooi Wan for help with processing and interpreting the ROS/MAP data. The authors also thank Cole Franks for help with improving and strengthening Theorem 5.

Appendix A. Variance Explained by iPCA

To ensure that the cumulative proportion of variance explained from Definition 1 is a well-defined concept, we check that $\text{PVE}_{k,m}$ is a proportion and is an increasing function as m increases. This implies that the marginal proportion of variance explained given in Definition 2 is also a proportion.

Proposition 9 *The cumulative proportion of variance explained in $\mathbf{X}_k$ by the top m iPCs, as defined in* (10)*, satisfies the following properties: for each $k = 1, \dots, K$ and $m = 1, \dots, \min\{n, p_k\}$,*

(i) $0 \leq PVE_{k,m} \leq 1$*;*

(ii) $PVE_{k,m-1} \leq PVE_{k,m}$*.*

Proof (i) Since the Frobenius norm is always non-negative, it is clear that $\text{PVE}_{k,m} \geq 0$. So it suffices to show that $\text{PVE}_{k,m} \leq 1$, or equivalently, $\| (\mathbf{U}^{(m)})^T \mathbf{X}_k \mathbf{V}_k^{(m)} \|_F^2 \leq \| \mathbf{X}_k \|_F^2$.

By definition of the Frobenius norm, we have that

$$\begin{aligned}
\| (\mathbf{U}^{(m)})^T \mathbf{X}_k \mathbf{V}_k^{(m)} \|_F^2 &= \sum_{i=1}^{m} \sum_{j=1}^{m} \left((\mathbf{U}^{(m)})^T \mathbf{X}_k \mathbf{V}_k^{(m)} \right)_{ij}^2 \\
&= \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\sum_{q=1}^{n} \sum_{r=1}^{p_k} (\mathbf{U}^{(m)})_{iq}^T \mathbf{X}_{k,qr} \mathbf{V}_{k,rj}^{(m)} \right)^2 \\
&\overset{[1]}{\leq} \sum_{i=1}^{n} \sum_{j=1}^{p_k} \left(\sum_{q=1}^{n} \sum_{r=1}^{p_k} \mathbf{U}_{iq}^T \mathbf{X}_{k,qr} \mathbf{V}_{k,rj} \right)^2 \\
&= \| \mathbf{U}^T \mathbf{X}_k \mathbf{V}_k \|_F^2 \\
&\overset{[2]}{=} \| \mathbf{X}_k \|_F^2 .
\end{aligned}$$

Here, [2] holds by the orthogonality of $\mathbf{U}$ and $\mathbf{V}_k$, and [1] follows from the facts that $m \leq \min\{n, p_k\}$, $\mathbf{U}^{(m)}$ and $\mathbf{V}_k^{(m)}$ are precisely the first m columns of $\mathbf{U}$ and $\mathbf{V}_k$ respectively, and the summand is non-negative. This concludes the proof of part (i).

(ii) We follow a similar argument as part (i) to see that

$$\begin{aligned}
\| (\mathbf{U}^{(m-1)})^T \mathbf{X}_k \mathbf{V}_k^{(m-1)} \|_F^2 &= \sum_{i=1}^{m-1} \sum_{j=1}^{m-1} \left(\sum_{q=1}^{n} \sum_{r=1}^{p_k} (\mathbf{U}^{(m-1)})_{iq}^T \mathbf{X}_{k,qr} \mathbf{V}_{k,rj}^{(m-1)} \right)^2 \\
&\leq \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\sum_{q=1}^{n} \sum_{r=1}^{p_k} (\mathbf{U}^{(m)})_{iq}^T \mathbf{X}_{k,qr} \mathbf{V}_{k,rj}^{(m)} \right)^2 \\
&= \| (\mathbf{U}^{(m)})^T \mathbf{X}_k \mathbf{V}_k^{(m)} \|_F^2 .
\end{aligned}$$

This implies that $\text{PVE}_{k,m-1} \leq \text{PVE}_{k,m}$. ■

Next, we claim that Definition 1 is a generalization of the cumulative proportion of variance explained in PCA. Recall that in PCA, if $\mathbf{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$ is the SVD of $\mathbf{X}$, then the cumulative proportion of variance explained by the top m PCs is $(\sum_{i=1}^m d_i^2)/(\sum_{i=1}^p d_i^2)$, where $\mathbf{D} = \text{diag}(d_1, \ldots, d_p)$. We can rewrite this as

$$\frac{\sum_{i=1}^m d_i^2}{\sum_{i=1}^p d_i^2} = \frac{\| \mathbf{D}^{(m)} \|_F^2}{\| \mathbf{X} \|_F^2} = \frac{\| (\mathbf{U}^{(m)})^T \mathbf{X} \mathbf{V}^{(m)} \|_F^2}{\| \mathbf{X} \|_F^2} \tag{18}$$

using properties of the SVD. Since $\mathbf{U}$ are the PC scores, and $\mathbf{V}$ are the PC loadings from PCA, then Definition 1 is indeed a natural definition in the sense that it generalizes the PCA cumulative proportion of variance explained.

Appendix B. Difference Between iPCA and Concatenated PCA

To formalize the difference between iPCA and concatenated PCA, we examine the bias of the eigenvectors from concatenated PCA when sample and feature dependencies simultaneously obscure each other under the matrix-variate normal model.

Proposition 10 *Suppose that* $\mathbf{X}_k \sim N_{n,p_k}(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{\Delta}_k)$ *for each* $k = 1, \dots, K$*, and define* $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_K]$*,* $\mathbf{\Delta} = diag(\mathbf{\Delta}_1, \dots, \mathbf{\Delta}_K)$*, and* $\mathbf{Z} = [\mathbf{Z}_1, \dots, \mathbf{Z}_K]$*, where* $\mathbf{Z}_k$ *is as defined in* (5) *previously. If* $\mathbf{X}\,\mathbf{\Delta}^{-1/2} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ *and* $\mathbf{X} = \tilde{\mathbf{U}}\tilde{\mathbf{D}}\tilde{\mathbf{V}}^T$ *are their respective compact SVDs, then* $\mathbf{U}$ *and* $\tilde{\mathbf{U}}$ *denote the iPCA and concatenated PCA scores, respectively, and*

$$\mathbf{U} - \tilde{\mathbf{U}} = \mathbf{\Sigma}^{1/2}\,\mathbf{Z}(\mathbf{V}\mathbf{D}^{-1} - \mathbf{\Delta}^{1/2}\,\tilde{\mathbf{V}}\tilde{\mathbf{D}}^{-1}). \tag{11}$$

Moreover, the iPCA and concatenated PCA scores are equal if and only if $\mathbf{V}\mathbf{D}^{-1} - \mathbf{\Delta}^{1/2}\,\tilde{\mathbf{V}}\tilde{\mathbf{D}}^{-1}$ *is in the nullspace of* $\mathbf{Z}$ *or* $\mathbf{\Delta}^{1/2}\,\tilde{\mathbf{V}} = \mathbf{V}\mathbf{D}^{-1}\tilde{\mathbf{D}}$*.*

Proof By construction of their methodology, we know that the iPCA scores $\mathbf{U}$ and the concatenated PCA scores $\tilde{\mathbf{U}}$ satisfy the following compact SVDs:

$$\mathbf{X}\,\mathbf{\Delta}^{-1/2} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T,$$
$$\mathbf{X} = \tilde{\mathbf{U}}\tilde{\mathbf{D}}\tilde{\mathbf{V}}^T.$$

We also know from (5) that

$$\mathbf{X} = \mathbf{\Sigma}^{1/2}\,\mathbf{Z}\,\mathbf{\Delta}^{1/2}. \tag{19}$$

Using the orthogonality of the singular vectors and plugging in (19) for $\mathbf{X}$ in the compact SVD formulas then gives

$$\mathbf{U} = \mathbf{\Sigma}^{1/2}\,\mathbf{Z}\,\mathbf{V}\,\mathbf{D}^{-1},$$
$$\tilde{\mathbf{U}} = \mathbf{\Sigma}^{1/2}\,\mathbf{Z}\,\mathbf{\Delta}^{1/2}\,\tilde{\mathbf{V}}\tilde{\mathbf{D}}^{-1}.$$

Therefore,

$$\mathbf{U} - \tilde{\mathbf{U}} = \mathbf{\Sigma}^{1/2}\,\mathbf{Z}\left(\mathbf{V}\,\mathbf{D}^{-1} - \mathbf{\Delta}^{1/2}\,\tilde{\mathbf{V}}\tilde{\mathbf{D}}^{-1}\right).$$

■

Appendix C. Covariance Estimation for iPCA

In this section, we provide the proofs and derivations related to the unpenalized and penalized maximum likelihood estimators under the iPCA model.

C.1 Unpenalized Maximum Likelihood Estimators

We begin this section by deriving the log-likelihood equation associated with the iPCA population model (2).

Recall that the probability density function for each matrix-variate normal model ($k = 1, \dots, K$) is given by

$$f\left(\mathbf{X}_k \,|\, \mathbf{\Sigma}, \mathbf{\Delta}_k\right) = (2\pi)^{-\frac{np_k}{2}}\,|\,\mathbf{\Sigma}\,|^{-\frac{p_k}{2}}\,|\,\mathbf{\Delta}_k\,|^{-\frac{n}{2}}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T\right)\right).$$

Hence, the log-likelihood function is

$$\begin{aligned}\ell(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}) = \sum_{k=1}^{K} \Big[&- \frac{np_k}{2}\log(2\pi) + \frac{p_k}{2}\log|\,\mathbf{\Sigma}^{-1}\,| + \frac{n}{2}\log|\,\mathbf{\Delta}_k^{-1}\,| \\ &- \frac{1}{2}\mathrm{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T\right)\Big] \\ \propto p\log|\,\mathbf{\Sigma}^{-1}\,| &+ n\sum_{k=1}^{K}\log|\,\mathbf{\Delta}_k^{-1}\,| - \sum_{k=1}^{K}\mathrm{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T\right).\end{aligned}$$

The unpenalized MLEs are designed to solve the optimization problem

$$\underset{\mathbf{\Sigma}^{-1},\mathbf{\Delta}_1^{-1},\ldots,\mathbf{\Delta}_K^{-1}}{\mathrm{argmin}} \quad -\ell(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}) \tag{20}$$

Taking partial derivatives with respect to each covariance parameter thus gives the following:

Lemma 11 *The unpenalized MLEs of $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_K$ satisfy*

$$\hat{\mathbf{\Sigma}} = \frac{1}{p}\sum_{k=1}^{K}\mathbf{X}_k\,\hat{\mathbf{\Delta}}_k^{-1}\,\mathbf{X}_k^T \tag{13}$$

$$\hat{\mathbf{\Delta}}_k = \frac{1}{n}\,\mathbf{X}_k^T\,\hat{\mathbf{\Sigma}}^{-1}\,\mathbf{X}_k \quad \forall\, k = 1, \ldots, K. \tag{14}$$

Proof Taking the partial derivatives of the log-likelihood equation with respect to $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}_k^{-1}$ respectively yields the gradient equations:

$$\frac{\partial \ell}{\partial\,\mathbf{\Sigma}^{-1}} = p\,\mathbf{\Sigma} - \sum_{k=1}^{K}\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T$$

$$\frac{\partial \ell}{\partial\,\mathbf{\Delta}_k^{-1}} = n\,\mathbf{\Delta}_k - \mathbf{X}_k^T\,\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,.$$

Setting the gradient equations equal to **0** gives the desired result. ■

Assuming that the unpenalized MLEs exist, we can compute the unpenalized MLEs of $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_K$ via Algorithm 2, which is analogous to the Flip-Flop algorithm provided in Dutilleul (1999).

The next theorem provides very restrictive conditions for which the unpenalized MLEs exist, but in almost all practical cases, the unpenalized maximum likelihood problem is ill-posed for iPCA.

Theorem 5 *(i) Suppose that $\mathbf{X}_k$ has been column-centered so that each column has a mean of 0 for each $k = 1, \ldots, K$. Then the unpenalized MLEs for $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_K$ are not positive definite and hence do not exist.*

(ii) Suppose that $\mathbf{X}_k$ has not been column-centered but that $\mathrm{rank}(\tilde{\mathbf{X}}) = n$ and $\mathrm{rank}(\mathbf{X}_k) = p_k$ for $k = 1, \ldots, K$.

Algorithm 2 Flip-Flop Algorithm for iPCA Unpenalized MLEs

```
Assume that $\mathbf{X}_k$ has been centered appropriately and that the unpenalized MLEs exist.
Initialize $\hat{\mathbf{\Sigma}}$, $\hat{\mathbf{\Delta}}_1, \dots, \hat{\mathbf{\Delta}}_K$ to be symmetric positive definite.
while not converged do
    Update $\hat{\mathbf{\Sigma}} = \frac{1}{p} \sum_{k=1}^{K} \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T$
    for $k = 1, \dots, K$ do
        Update $\hat{\mathbf{\Delta}}_k = \frac{1}{n} \, \mathbf{X}_k^T \, \hat{\mathbf{\Sigma}}^{-1} \, \mathbf{X}_k$
```

(a) If $n \neq p_k$ for some $k = 1, \dots, K$, then the unpenalized log-likelihood function $\ell(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1})$ is unbounded.

(b) If the unpenalized log-likelihood function $\ell(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1})$ is bounded, then the unpenalized MLE for $\mathbf{\Sigma}$, $\mathbf{\Delta}_1, \dots, \mathbf{\Delta}_K$ exist.

Proof (ii) Suppose that $\mathbf{X}_k$ has not been column-centered but that $\text{rank}(\tilde{\mathbf{X}}) = n$ and $\text{rank}(\mathbf{X}_k) = p_k$ for $k = 1, \dots, K$. We will first prove part (b), so assume also that the unpenalized log-likelihood is bounded. From the unpenalized MLEs in Lemma 4, it is easy to see that $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}_1, \dots, \hat{\mathbf{\Delta}}_K$ are symmetric positive semidefinite. We next claim that $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}_1, \dots, \hat{\mathbf{\Delta}}_K$ are full rank and hence positive definite. To prove this claim, we proceed by induction on the unpenalized Flip-Flop iteration counter m. Let $\hat{\mathbf{\Sigma}}^m$ and $\hat{\mathbf{\Delta}}_k^m$ denote the m^{th} Flip-Flop update of $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}_k$, respectively.

Clearly, the base case holds since $\mathbf{\Sigma}^0$ and $\mathbf{\Delta}_1^0, \dots, \mathbf{\Delta}_K^0$ are initialized to be symmetric positive definite. So suppose $\mathbf{\Sigma}^m$ and $\mathbf{\Delta}_1^m \dots, \mathbf{\Delta}_K^m$ are full rank. Then the unpenalized Flip-Flop iterates at the $(m+1)^{th}$-update step are

$$\begin{aligned} \hat{\mathbf{\Sigma}}^{m+1} &= \frac{1}{p} \sum_{k=1}^{K} \mathbf{X}_k (\hat{\mathbf{\Delta}}_k^m)^{-1} \, \mathbf{X}_k^T = \frac{1}{p} \tilde{\mathbf{X}} (\tilde{\mathbf{\Delta}}^m)^{-1} \tilde{\mathbf{X}}^T, \\ \hat{\mathbf{\Delta}}_k^{m+1} &= \frac{1}{n} \, \mathbf{X}_k^T (\hat{\mathbf{\Sigma}}^{m+1})^{-1} \, \mathbf{X}_k, \end{aligned}$$

where $\tilde{\mathbf{X}} = [\mathbf{X}_1, \dots, \mathbf{X}_K]$ and $\tilde{\mathbf{\Delta}}^m = \text{diag}(\hat{\mathbf{\Delta}}_1^m, \dots, \hat{\mathbf{\Delta}}_K^m)$.

Therefore, we have that

$$\text{rank}\left(\hat{\mathbf{\Sigma}}^{m+1}\right) = \text{rank}\left(\tilde{\mathbf{X}} (\tilde{\mathbf{\Delta}}^m)^{-1} \tilde{\mathbf{X}}^T\right) = \text{rank}\left(\tilde{\mathbf{X}} (\tilde{\mathbf{\Delta}}^m)^{-\frac{1}{2}}\right) = \text{rank}(\tilde{\mathbf{X}}) = n.$$

Here, the second equality holds because $\text{rank}(\mathbf{A}^T \, \mathbf{A}) = \text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^T)$. The third equality holds because $(\tilde{\mathbf{\Delta}}^m)^{-\frac{1}{2}}$ is full rank (i.e. rank $= p$) by the inductive hypothesis, and the last equality holds by hypothesis. Thus, $\hat{\mathbf{\Sigma}}^{m+1}$ is positive definite.

Similarly, for each $k = 1, \dots, K$,

$$\text{rank}\left(\hat{\mathbf{\Delta}}_k^{m+1}\right) = \text{rank}\left(\mathbf{X}_k^T (\hat{\mathbf{\Sigma}}^{m+1})^{-1} \, \mathbf{X}_k\right) = \text{rank}(\mathbf{X}_k) = p_k.$$

So for each $k = 1, \dots, K$, $\hat{\mathbf{\Delta}}_k^{m+1}$ is positive definite.

By induction, $\hat{\mathbf{\Sigma}}^m, \hat{\mathbf{\Delta}}_1^m, \dots, \hat{\mathbf{\Delta}}_K^m \succ 0$ for each iterate of Algorithm 2, and by Corollary 21, Algorithm 2 converges to the global solution of (20). Thus, the unpenalized MLEs for $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1}$ exist under the assumptions given in part $(ii)(b)$.

To prove part (a) of (ii), let $\mathbf{X}_1, \ldots, \mathbf{X}_K$ be given, and assume that $n \neq p_k$ for some $k = 1, \ldots, K$. By the rank assumptions, it must be that $p_k \leq n \leq p$ for each $k = 1, \ldots, K$. Define $\mathcal{K}^0 = \{k \in \{1, \ldots, K\} \mid p_k < n\}$ and $\mathcal{K}^1 = \{k \in \{1, \ldots, K\} \mid p_k = n\}$. Note that by assumption, $\mathcal{K}^0$ is not empty, and for each $k \in \mathcal{K}^0$, $\mathbf{X}_k^T$ has a kernel. We can then choose bases for $\mathbb{R}^n$ and $\mathbb{R}^{p_k}$ $(k = 1, \ldots, K)$ such that the first row of $\mathbf{X}_k$ is zero for each $k \in \mathcal{K}^0$ and the first row of $\mathbf{X}_k$ is the first standard basis vector, denoted $\mathbf{f}_{k,1}$, of $\mathbb{R}^{p_k}$ for each $k \in \mathcal{K}^1$. Let $\mathbf{e}_1$ denote the first standard basis vector in $\mathbb{R}^n$.

We will next construct a family of diagonal matrices for $\mathbf{\Sigma}, \mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_K$ that sends the unpenalized log-likelihood to infinity. Specifically, let $\mathbf{\Sigma} = e^{t\,\mathbf{e}_1\,\mathbf{e}_1^T}$, $\mathbf{\Delta}_k = \mathbf{I}_{p_k}$ for $k \in \mathcal{K}^0$, and $\mathbf{\Delta}_k = e^{-t\,\mathbf{f}_{k,1}\,\mathbf{f}_{k,1}^T}$ for $k \in \mathcal{K}^1$, where e denotes the matrix exponential. Then for $k \in \mathcal{K}^0$,

$$\operatorname{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T\right) = \operatorname{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,\mathbf{X}_k^T\right) = \sum_{i=2}^{n} \mathbf{x}_{k,i}^T\,\mathbf{x}_{k,i},$$

and for $k \in \mathcal{K}^1$,

$$\operatorname{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T\right) = 1 + \sum_{i=2}^{n}\left((e^t - 1)(\mathbf{x}_{k,i}^T\,\mathbf{f}_{k,1})^2 + \mathbf{x}_{k,i}^T\,\mathbf{x}_{k,i}\right),$$

where $\mathbf{x}_{k,i}$ denotes the i^{th} row of $\mathbf{X}_k$.

Therefore, for every $t \in \mathbb{R}$, we have that

$$\begin{aligned}
\ell(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}) &= -p \log|\,\mathbf{\Sigma}\,| - n \sum_{k\in\mathcal{K}^0} \log|\,\mathbf{\Delta}_k\,| - n \sum_{k\in\mathcal{K}^1} \log|\,\mathbf{\Delta}_k\,| \\
&\quad - \sum_{k\in\mathcal{K}^0} \operatorname{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T\right) - \sum_{k\in\mathcal{K}^1} \operatorname{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T\right) \\
&= -p\log(e^t) - n\sum_{k\in\mathcal{K}^0} 0 - n \sum_{k\in\mathcal{K}^1} \log(e^{-t}) - \sum_{k\in\mathcal{K}^0}\sum_{i=2}^{n} \mathbf{x}_{k,i}^T\,\mathbf{x}_{k,i} \\
&\quad - \sum_{k\in\mathcal{K}^1}\left(1 + \sum_{i=2}^{n}\left((e^t-1)(\mathbf{x}_{k,i}^T\,\mathbf{f}_{k,1})^2 + \mathbf{x}_{k,i}^T\,\mathbf{x}_{k,i}\right)\right) \\
&= t(n|\mathcal{K}^1| - p) - \sum_{k\in\mathcal{K}^0}\sum_{i=2}^{n} \mathbf{x}_{k,i}^T\,\mathbf{x}_{k,i} \\
&\quad - \sum_{k\in\mathcal{K}^1}\left(1 + \sum_{i=2}^{n}\left((e^t-1)(\mathbf{x}_{k,i}^T\,\mathbf{f}_{k,1})^2 + \mathbf{x}_{k,i}^T\,\mathbf{x}_{k,i}\right)\right).
\end{aligned}$$

As $t \to -\infty$, the first term approaches ∞ since $n|\mathcal{K}^1| < p$ while the second and third terms approach a constant. Thus, $\ell(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1})$ is unbounded, as desired.

(i) Suppose that we have centered each data matrix $\mathbf{X}_k$ to have column means $\mathbf{0}$. Recall that the unpenalized MLEs for $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_K$ are obtained by

$$\begin{aligned}
\hat{\mathbf{\Sigma}} &= \frac{1}{p}\sum_{k=1}^{K} \mathbf{X}_k\,\hat{\mathbf{\Delta}}_k^{-1}\,\mathbf{X}_k^T = \frac{1}{p}\tilde{\mathbf{X}}\tilde{\mathbf{\Delta}}^{-1}\tilde{\mathbf{X}}^T, \\
\hat{\mathbf{\Delta}}_k &= \frac{1}{n}\,\mathbf{X}_k^T\,\hat{\mathbf{\Sigma}}^{-1}\,\mathbf{X}_k,
\end{aligned}$$

where $\tilde{\mathbf{\Delta}} := \operatorname{diag}(\hat{\mathbf{\Delta}}_1, \ldots, \hat{\mathbf{\Delta}}_K)$.

For the sake of contradiction, suppose that there exists $n, p_1, \ldots, p_k$ such that $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}$ are symmetric positive definite. By the same argument as in part $(ii)(b)$, $\operatorname{rank}(\hat{\mathbf{\Sigma}}) = \operatorname{rank}(\tilde{\mathbf{X}})$, but since each $\mathbf{X}_k$ has been centered to have column means $\mathbf{0}$, then the n rows of $\tilde{\mathbf{X}}$ are linearly dependent. Hence, $\operatorname{rank}(\hat{\mathbf{\Sigma}}) = \operatorname{rank}(\tilde{\mathbf{X}}) < n$, which implies that $\hat{\mathbf{\Sigma}}$ can never be positive definite, a contradiction. Therefore, the unpenalized MLEs never exist if each $\mathbf{X}_k$ has be column-centered to have mean $\mathbf{0}$. ■

Remark 12 *Notice that if the unpenalized MLEs for $\mathbf{\Sigma}, \mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_K$ exist, then*

$$\begin{aligned} n &\overset{[1]}{=} \operatorname{rank}(\hat{\mathbf{\Sigma}}) \overset{[2]}{=} \operatorname{rank}(\tilde{\mathbf{X}}) \overset{[3]}{\leq} \min\{n, p\} \\ p_k &\overset{[1]}{=} \operatorname{rank}(\hat{\mathbf{\Delta}}_k) \overset{[2]}{=} \operatorname{rank}(\mathbf{X}_k) \overset{[3]}{\leq} \min\{n, p_k\} \qquad \forall\, k = 1, \ldots, K \end{aligned}$$

where [1] follows from positive definiteness of $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}_k$, [2] follows the same argument as in the proof of Theorem 5, and [3] holds by properties of rank and the dimensions of $\tilde{\mathbf{X}}$ and $\mathbf{X}_k$. By combining these rank constraints with the result of Theorem 5, we see that if the unpenalized MLEs for $\mathbf{\Sigma}, \mathbf{\Delta}_1, \ldots, \mathbf{\Delta}_K$ exist, it must be that $p_k = n \leq p$ for each $k = 1, \ldots, K$.

Proposition 6 can be proved in the same way as Lemma 4, so we omit the proof.

C.2 Penalized Maximum Likelihood Estimators

In this section, we develop Flip-Flop algorithms and analyze the convergence results for both Frobenius and L_1 penalties. For the sake of notation, let

$$\begin{aligned} -\ell_P(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}) = -p \log |\, \mathbf{\Sigma}^{-1} \,| - n \sum_{k=1}^{K} \log |\, \mathbf{\Delta}_k^{-1} \,| + \sum_{k=1}^{K} \operatorname{tr}\left(\mathbf{\Sigma}^{-1}\, \mathbf{X}_k\, \mathbf{\Delta}_k^{-1}\, \mathbf{X}_k^T\right) \\ + P(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}) \end{aligned}$$

We give the overarching framework of the Flip-Flop algorithms in Algorithm 3, and we show in Theorem 7 that Algorithm 3 can be used to find a local solution of (17) for a certain class of penalties, which includes the additive Frobenius, multiplicative Frobenius, and additive L_1 penalties. The main idea behind the proof is to use convexity and view Algorithm 3 as a block coordinate descent algorithm so that each update of the Flip-Flop algorithm is a descent direction.

Theorem 7 *Suppose that the objective function in* (17) *is bounded below. Suppose also that either (i) $P(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1})$ is a differentiable convex function with respect to each coordinate or (ii) $P(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}) = P_0(\mathbf{\Sigma}^{-1}) + \sum_{k=1}^{K} P_k(\mathbf{\Delta}_k^{-1})$, where P_i is a (non-differentiable) convex function for each $k = 1, \ldots, K$. If either (i) or (ii) holds, then the Flip-Flop algorithm corresponding to* (17) *converges to a stationary point of the objective.*

Algorithm 3 Outline of Flip-Flop Algorithm for Penalized iPCA Covariance Estimators

```
Center the columns of $\mathbf{X}_1, \dots, \mathbf{X}_K$, and initialize $\hat{\mathbf{\Sigma}}$, $\hat{\mathbf{\Delta}}_1, \dots, \hat{\mathbf{\Delta}}_K$ to be positive definite.
while not converged do
    Update $\mathbf{\Sigma}$ while fixing all other variables:
        $\hat{\mathbf{\Sigma}}^{-1} = \underset{\mathbf{\Sigma}^{-1} \succ 0}{\text{argmin}} \quad - \ell_P(\mathbf{\Sigma}^{-1}, \hat{\mathbf{\Delta}}_1^{-1}, \dots, \hat{\mathbf{\Delta}}_K^{-1})$
    for $k = 1, \dots, K$ do
        Update $\mathbf{\Delta}_k$ while fixing all other variables:
            $\hat{\mathbf{\Delta}}_k^{-1} = \underset{\mathbf{\Delta}_k^{-1} \succ 0}{\text{argmin}} \quad - \ell_P(\hat{\mathbf{\Sigma}}^{-1}, \hat{\mathbf{\Delta}}_1^{-1}, \dots, \hat{\mathbf{\Delta}}_{k-1}^{-1}, \mathbf{\Delta}_k^{-1}, \hat{\mathbf{\Delta}}_{k+1}^{-1}, \dots, \hat{\mathbf{\Delta}}_K^{-1})$
```

Proof Suppose either $P(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1})$ is a differentiable convex function with respect to each coordinate or $P(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1}) = P_0(\mathbf{\Sigma}^{-1}) + \sum_{k=1}^K P_k(\mathbf{\Delta}_k^{-1})$ where P_i is a (non-differentiable) convex function for each $i = 1, \dots, K$.

Let $\ell(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1}) = p \log |\, \mathbf{\Sigma}^{-1} \,| + n \sum_{k=1}^K \log |\, \mathbf{\Delta}_k^{-1} \,| - \sum_{k=1}^K \text{tr}\left(\mathbf{\Sigma}^{-1} \, \mathbf{X}_k \, \mathbf{\Delta}_k^{-1} \, \mathbf{X}_k^T\right)$. Since the domain of $-\ell$ is open and $-\ell$ is Gateaux-differentiable on its domain, then $-\ell_P$ is regular in the domain of $-\ell_P$ by Lemma 3.1 in Tseng (2001).

Note also that since the log-determinant is a strictly concave function on the set of symmetric positive definite matrices, the trace function is linear, and the penalty term is convex with respect to each coordinate by hypothesis, then

- $-\ell_P$ is strictly convex in $\mathbf{\Sigma}^{-1}$ with $\mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1}$ fixed, and
- for each $k = 1, \dots, K$, $-\ell_P$ is strictly convex in in $\mathbf{\Delta}_k^{-1}$ with $\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_j^{-1}$, $j \neq k$ fixed.

Because $-\ell_P$ is regular and strictly convex with respect to each coordinate, it follows that the Flip-Flop algorithm corresponding to (17) converges to a stationary point of the objective function by Theorem 4.1(c) in Tseng (2001). ■

In the following sections, we will derive the specific form of the Flip-Flop updates for each of the penalized iPCA estimators.

C.2.1 Additive Frobenius Penalized Flip-Flop Estimator

To compute the additive Frobenius penalized estimator, we solve

$$\begin{aligned} \hat{\mathbf{\Sigma}}^{-1}, \hat{\mathbf{\Delta}}_1^{-1}, \dots, \hat{\mathbf{\Delta}}_K^{-1} = \underset{\substack{\mathbf{\Sigma}^{-1} \succ 0 \\ \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1} \succ 0}}{\text{argmax}} \quad & \Big\{ p \log |\, \mathbf{\Sigma}^{-1} \,| + n \sum_{k=1}^K \log |\, \mathbf{\Delta}_k^{-1} \,| \\ & - \sum_{k=1}^K \text{tr}\left(\mathbf{\Sigma}^{-1} \, \mathbf{X}_k \, \mathbf{\Delta}_k^{-1} \, \mathbf{X}_k^T\right) \\ & - \lambda_{\mathbf{\Sigma}} \|\, \mathbf{\Sigma}^{-1} \,\|_F^2 - \sum_{k=1}^K \lambda_k \|\, \mathbf{\Delta}_k^{-1} \,\|_F^2 \Big\}. \end{aligned} \tag{21}$$

The gradient equations corresponding to (21) are given by

$$p\hat{\mathbf{\Sigma}} - \sum_{k=1}^{K} \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T - 2\lambda_{\mathbf{\Sigma}} \hat{\mathbf{\Sigma}}^{-1} = 0, \tag{22}$$

$$n\hat{\mathbf{\Delta}}_k - \mathbf{X}_k^T \, \hat{\mathbf{\Sigma}}^{-1} \, \mathbf{X}_k - 2\lambda_k \hat{\mathbf{\Delta}}_k^{-1} = 0 \qquad \forall\, k = 1, \ldots, K. \tag{23}$$

Lemma 13 *If gradient equations* (22) *and* (23) *are satisfied, then*

$$\hat{\mathbf{\Sigma}} = \left(\frac{2\lambda_{\mathbf{\Sigma}}}{p} \, \mathbf{I} + \frac{1}{4p^2} \left(\sum_{k=1}^{K} \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T \right)^2 \right)^{\frac{1}{2}} + \frac{1}{2p} \sum_{k=1}^{K} \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T \tag{24}$$

$$\hat{\mathbf{\Delta}}_k = \left(\frac{2\lambda_k}{n} \, \mathbf{I} + \frac{1}{4n^2} \left(\mathbf{X}_k^T \, \hat{\mathbf{\Sigma}}^{-1} \, \mathbf{X}_k \right)^2 \right)^{\frac{1}{2}} + \frac{1}{2n} \, \mathbf{X}_k^T \, \hat{\mathbf{\Sigma}}^{-1} \, \mathbf{X}_k \qquad \forall\, k = 1, \ldots, K \tag{25}$$

Proof Define $\hat{\mathbf{S}}_{\mathbf{\Sigma}} = \sum_{k=1}^{K} \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T$, and right multiply (22) by $\hat{\mathbf{\Sigma}}$ to obtain

$$p\hat{\mathbf{\Sigma}}^2 - \hat{\mathbf{S}}_{\mathbf{\Sigma}} \hat{\mathbf{\Sigma}} - 2\lambda_{\mathbf{\Sigma}} \, \mathbf{I} = \mathbf{0} \tag{26}$$

$$\implies \hat{\mathbf{\Sigma}}^2 - \frac{1}{p} \hat{\mathbf{S}}_{\mathbf{\Sigma}} \hat{\mathbf{\Sigma}} + \frac{1}{4p^2} \hat{\mathbf{S}}_{\mathbf{\Sigma}}^2 = \frac{2\lambda_{\mathbf{\Sigma}}}{p} \, \mathbf{I} + \frac{1}{4p^2} \hat{\mathbf{S}}_{\mathbf{\Sigma}}^2. \tag{27}$$

On the other hand, we can multiply (22) by $\hat{\mathbf{\Sigma}}$ on the left to obtain

$$p\hat{\mathbf{\Sigma}}^2 - \hat{\mathbf{\Sigma}} \hat{\mathbf{S}}_{\mathbf{\Sigma}} - 2\lambda_{\mathbf{\Sigma}} \, \mathbf{I} = \mathbf{0} \tag{28}$$

$$\implies \hat{\mathbf{\Sigma}}^2 - \frac{1}{p} \hat{\mathbf{\Sigma}} \hat{\mathbf{S}}_{\mathbf{\Sigma}} + \frac{1}{4p^2} \hat{\mathbf{S}}_{\mathbf{\Sigma}}^2 = \frac{2\lambda_{\mathbf{\Sigma}}}{p} \, \mathbf{I} + \frac{1}{4p^2} \hat{\mathbf{S}}_{\mathbf{\Sigma}}^2. \tag{29}$$

Adding (27) and (29) and then dividing by 2 gives

$$\hat{\mathbf{\Sigma}}^2 - \frac{1}{2p} \hat{\mathbf{S}}_{\mathbf{\Sigma}} \hat{\mathbf{\Sigma}} - \frac{1}{2p} \hat{\mathbf{\Sigma}} \hat{\mathbf{S}}_{\mathbf{\Sigma}} + \frac{1}{4p^2} \hat{\mathbf{S}}_{\mathbf{\Sigma}}^2 = \frac{2\lambda_{\mathbf{\Sigma}}}{p} \, \mathbf{I} + \frac{1}{4p^2} \hat{\mathbf{S}}_{\mathbf{\Sigma}}^2$$

$$\implies \left(\hat{\mathbf{\Sigma}} - \frac{1}{2p} \hat{\mathbf{S}}_{\mathbf{\Sigma}} \right)^2 = \frac{2\lambda_{\mathbf{\Sigma}}}{p} \, \mathbf{I} + \frac{1}{4p^2} \hat{\mathbf{S}}_{\mathbf{\Sigma}}^2.$$

Since $\frac{2\lambda_{\mathbf{\Sigma}}}{p} \, \mathbf{I} + \frac{1}{4p^2} \hat{\mathbf{S}}_{\mathbf{\Sigma}}^2$ is positive definite, its has a unique square root. Thus,

$$\hat{\mathbf{\Sigma}} = \left(\frac{2\lambda_{\mathbf{\Sigma}}}{p} \, \mathbf{I} + \frac{1}{4p^2} \hat{\mathbf{S}}_{\mathbf{\Sigma}}^2 \right)^{1/2} + \frac{1}{2p} \hat{\mathbf{S}}_{\mathbf{\Sigma}}.$$

We can similarly rearrange (23) and complete the square to obtain (25). ■

We now have the machinery to prove Proposition 14, which gives us the form of each update in the additive Frobenius Flip-Flop algorithm.

Proposition 14 *$\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}_1, \ldots, \hat{\mathbf{\Delta}}_K$ are solutions to the gradient equations in* (22) *and* (23) *if and only if*

$$\hat{\mathbf{\Sigma}} = \mathbf{U} \left[\frac{1}{2p} \left(\mathbf{\Gamma} + \left(\mathbf{\Gamma}^2 + 8\lambda_{\mathbf{\Sigma}} p \, \mathbf{I} \right)^{\frac{1}{2}} \right) \right] \mathbf{U}^T \tag{30}$$

$$\text{and } \; \hat{\mathbf{\Delta}}_k = \mathbf{V}_k \left[\frac{1}{2n} \left(\mathbf{\Phi}_k + \left(\mathbf{\Phi}_k^2 + 8\lambda_k n \, \mathbf{I} \right)^{\frac{1}{2}} \right) \right] \mathbf{V}_k^T \qquad \forall \, k = 1, \ldots, K, \tag{31}$$

where $\mathbf{U}, \mathbf{V}_k, \mathbf{\Gamma}$, and $\mathbf{\Phi}_k$ are defined by the eigendecompositions $\sum_{k=1}^K \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T = \mathbf{U} \, \mathbf{\Gamma} \, \mathbf{U}^T$ and $\mathbf{X}_k^T \, \hat{\mathbf{\Sigma}}^{-1} \, \mathbf{X}_k = \mathbf{V}_k \, \mathbf{\Phi}_k \, \mathbf{V}_k^T$.

Proof ($\Rightarrow$) Suppose that $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}_1, \ldots, \hat{\mathbf{\Delta}}_K$ are solutions to the gradient equations in (22) and (23). We will first show that the eigenvectors of $\hat{\mathbf{\Sigma}}$ and $\sum_{k=1}^K \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T$ are equivalent.

Let $\mathbf{u}$ be an eigenvector of $\hat{\mathbf{\Sigma}}$ with the corresponding eigenvalue ϕ. Then, by (22),

$$\sum_{k=1}^K \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T \, \mathbf{u} = (p\hat{\mathbf{\Sigma}} - 2\lambda_{\mathbf{\Sigma}} \hat{\mathbf{\Sigma}}^{-1}) \, \mathbf{u} = (p\phi - 2\lambda_{\mathbf{\Sigma}} \phi^{-1}) \, \mathbf{u} \, .$$

Therefore, $\mathbf{u}$ is an eigenvector of $\sum_{k=1}^K \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T$ with the eigenvalue $p\phi - 2\lambda_{\mathbf{\Sigma}}\phi^{-1}$.

Conversely, suppose $\mathbf{u}$ is an eigenvector of $\sum_{k=1}^K \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T$ with eigenvalue γ. Then

$$\left(\frac{2\lambda_{\mathbf{\Sigma}}}{p} \, \mathbf{I} + \frac{1}{4p^2} \left(\sum_{k=1}^K \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T \right)^2 \right) \mathbf{u} = \left(\frac{2\lambda_{\mathbf{\Sigma}}}{p} + \frac{1}{4p^2} \gamma^2 \right) \mathbf{u} \, .$$

This implies that

$$\left(\frac{2\lambda_{\mathbf{\Sigma}}}{p} \, \mathbf{I} + \frac{1}{4p^2} \left(\sum_{k=1}^K \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T \right)^2 \right)^{\frac{1}{2}} \mathbf{u} = \left(\frac{2\lambda_{\mathbf{\Sigma}}}{p} + \frac{1}{4p^2} \gamma^2 \right)^{\frac{1}{2}} \mathbf{u} \, . \tag{32}$$

So by Lemma 13 and (32), we have that

$$\hat{\mathbf{\Sigma}} \, \mathbf{u} = \left[\left(\frac{2\lambda_{\mathbf{\Sigma}}}{p} \, \mathbf{I} + \frac{1}{4p^2} \left(\sum_{k=1}^K \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T \right)^2 \right)^{\frac{1}{2}} + \frac{1}{2p} \sum_{k=1}^K \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T \right] \mathbf{u} \tag{33}$$

$$= \left[\left(\frac{2\lambda_{\mathbf{\Sigma}}}{p} + \frac{1}{4p^2} \gamma^2 \right)^{\frac{1}{2}} + \frac{1}{2p} \gamma \right] \mathbf{u} \tag{34}$$

$$= \left[\frac{1}{2p} \left(\gamma + \sqrt{\gamma^2 + 8\lambda_{\mathbf{\Sigma}} p} \right) \right] \mathbf{u}, \tag{35}$$

so $\mathbf{u}$ is indeed an eigenvector of $\hat{\mathbf{\Sigma}}$ with the eigenvalue $\frac{1}{2p} \left(\gamma + \sqrt{\gamma^2 + 8\lambda_{\mathbf{\Sigma}} p} \right)$.

Since the eigenvectors of $\hat{\mathbf{\Sigma}}$ and $\sum_{k=1}^K \mathbf{X}_k \, \hat{\mathbf{\Delta}}_k^{-1} \, \mathbf{X}_k^T$ are equivalent and (35) gives us the exact connection between their eigenvalues, it follows that

$$\hat{\mathbf{\Sigma}} = \mathbf{U} \left[\frac{1}{2p} \left(\mathbf{\Gamma} + \left(\mathbf{\Gamma}^2 + 8\lambda_{\mathbf{\Sigma}} p \, \mathbf{I} \right)^{\frac{1}{2}} \right) \right] \mathbf{U}^T,$$

where $\mathbf{U}, \boldsymbol{\Gamma}$ are defined by the eigendecomposition $\sum_{k=1}^{K} \mathbf{X}_k \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \mathbf{X}_k^T = \mathbf{U} \, \boldsymbol{\Gamma} \, \mathbf{U}^T$.

This same logic can be used to show that for each $k = 1, \ldots, K$,

$$\hat{\boldsymbol{\Delta}}_k = \mathbf{V}_k \left[\frac{1}{2n} \left(\boldsymbol{\Phi}_k + \left(\boldsymbol{\Phi}_k^2 + 8\lambda_k n \, \mathbf{I} \right)^{\frac{1}{2}} \right) \right] \mathbf{V}_k^T,$$

where $\mathbf{V}_k, \boldsymbol{\Phi}_k$ are defined by the eigendecomposition $\mathbf{X}_k^T \, \hat{\boldsymbol{\Sigma}}^{-1} \, \mathbf{X}_k = \mathbf{V}_k \, \boldsymbol{\Phi}_k \, \mathbf{V}_k^T$. We omit the proof since it uses the same argument as above.

($\Leftarrow$) Now suppose that $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Delta}}_1, \ldots, \hat{\boldsymbol{\Delta}}_K$ satisfy equations (30) and (31).

Since we know $\left[\frac{1}{2p} \left(\gamma + \sqrt{\gamma^2 + 8\lambda_{\boldsymbol{\Sigma}} p} \right) \right]^{-1} = \frac{1}{4\lambda_{\boldsymbol{\Sigma}}} \left(\sqrt{\gamma^2 + 8\lambda_{\boldsymbol{\Sigma}} p} - \gamma \right)$, it follows that $\left[\frac{1}{2p} \left(\boldsymbol{\Gamma} + \left(\boldsymbol{\Gamma}^2 + 8\lambda_{\boldsymbol{\Sigma}} p \, \mathbf{I} \right)^{\frac{1}{2}} \right) \right]^{-1} = \frac{1}{4\lambda_{\boldsymbol{\Sigma}}} \left(\left(\boldsymbol{\Gamma}^2 + 8\lambda_{\boldsymbol{\Sigma}} p \, \mathbf{I} \right)^{\frac{1}{2}} - \boldsymbol{\Gamma} \right)$. Therefore,

$$\begin{aligned} p\hat{\boldsymbol{\Sigma}} - \sum_{k=1}^{K} \mathbf{X}_k \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \mathbf{X}_k^T - 2\lambda_{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Sigma}}^{-1} &= p \, \mathbf{U} \left[\frac{1}{2p} \left(\boldsymbol{\Gamma} + \left(\boldsymbol{\Gamma}^2 + 8\lambda_{\boldsymbol{\Sigma}} p \, \mathbf{I} \right)^{\frac{1}{2}} \right) \right] \mathbf{U}^T - \mathbf{U} \, \boldsymbol{\Gamma} \, \mathbf{U}^T \\ &\quad - 2\lambda_{\boldsymbol{\Sigma}} \, \mathbf{U} \left[\frac{1}{2p} \left(\boldsymbol{\Gamma} + \left(\boldsymbol{\Gamma}^2 + 8\lambda_{\boldsymbol{\Sigma}} p \, \mathbf{I} \right)^{\frac{1}{2}} \right) \right]^{-1} \mathbf{U}^T \\ &= \mathbf{U} \left[\frac{1}{2} \left(\boldsymbol{\Gamma} + \left(\boldsymbol{\Gamma}^2 + 8\lambda_{\boldsymbol{\Sigma}} p \, \mathbf{I} \right)^{\frac{1}{2}} \right) \right] \mathbf{U}^T - \mathbf{U} \, \boldsymbol{\Gamma} \, \mathbf{U}^T \\ &\quad - \mathbf{U} \left[\frac{1}{2} \left(\left(\boldsymbol{\Gamma}^2 + 8\lambda_{\boldsymbol{\Sigma}} p \, \mathbf{I} \right)^{\frac{1}{2}} - \boldsymbol{\Gamma} \right) \right] \mathbf{U}^T \\ &= 0. \end{aligned}$$

Similarly, one can substitute in (31) and follow the same argument to show that (23) is also satisfied. ■

As a consequence of Proposition 14, each update step in the additive Frobenius Flip-Flop algorithm can be solved by taking a full eigendecomposition and then regularizing the eigenvalues. This algorithm is given in Algorithm 4.

Algorithm 4 Flip-Flop Algorithm for Additive Frobenius Penalized iPCA Estimators

```
Center the columns of $\mathbf{X}_1, \ldots, \mathbf{X}_K$, and initialize $\hat{\boldsymbol{\Sigma}}$, $\hat{\boldsymbol{\Delta}}_1, \ldots, \hat{\boldsymbol{\Delta}}_K$ to be positive definite.
while not converged do
    Take eigendecomposition: $\sum_{k=1}^{K} \mathbf{X}_k \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \mathbf{X}_k^T = \mathbf{U} \, \boldsymbol{\Gamma} \, \mathbf{U}^T$
    Regularize eigenvalues: $\Phi_{ii} = \frac{1}{2p} \left( \Gamma_{ii} + \sqrt{\Gamma_{ii}^2 + 8\lambda_{\boldsymbol{\Sigma}} p} \right)$        } Update $\boldsymbol{\Sigma}$
    Update $\hat{\boldsymbol{\Sigma}}^{-1} = \mathbf{U} \, \boldsymbol{\Phi}^{-1} \, \mathbf{U}^T$
    for $k = 1, \ldots, K$ do
        Take eigendecomposition: $\mathbf{X}_k^T \, \hat{\boldsymbol{\Sigma}}^{-1} \, \mathbf{X}_k = \mathbf{V} \, \boldsymbol{\Phi} \, \mathbf{V}^T$
        Regularize eigenvalues: $\Gamma_{ii} = \frac{1}{2n} \left( \Phi_{ii} + \sqrt{\Phi_{ii}^2 + 8\lambda_k n} \right)$        } Update $\boldsymbol{\Delta}_k$
        Update $\hat{\boldsymbol{\Delta}}_k^{-1} = \mathbf{V} \, \boldsymbol{\Gamma}^{-1} \, \mathbf{V}^T$
```

C.2.2 Multiplicative Frobenius Penalized Flip-Flop Estimator

The derivation of the multiplicative Frobenius Flip-Flop algorithm is very similar to the previous derivation with the additive Frobenius penalty. Thus, we omit most of the details and simply provide a sketch of the derivation.

Recall that the multiplicative Frobenius penalized estimator solves

$$\begin{aligned}\hat{\boldsymbol{\Sigma}}^{-1}, \hat{\boldsymbol{\Delta}}_1^{-1}, \ldots, \hat{\boldsymbol{\Delta}}_K^{-1} = \underset{\substack{\boldsymbol{\Sigma}^{-1} \succ 0 \\ \boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_K^{-1} \succ 0}}{\operatorname{argmax}} \Big\{ & p \log |\, \boldsymbol{\Sigma}^{-1} \,| + n \sum_{k=1}^K \log |\, \boldsymbol{\Delta}_k^{-1} \,| \\ & - \sum_{k=1}^K \operatorname{tr}\left(\boldsymbol{\Sigma}^{-1} \, \mathbf{X}_k \, \boldsymbol{\Delta}_k^{-1} \, \mathbf{X}_k^T \right) \\ & - \| \, \boldsymbol{\Sigma}^{-1} \, \|_F^2 \sum_{k=1}^K \lambda_k \| \, \boldsymbol{\Delta}_k^{-1} \, \|_F^2 \Big\}, \end{aligned} \tag{36}$$

for which the gradient equations are

$$p \hat{\boldsymbol{\Sigma}} - \sum_{k=1}^K \mathbf{X}_k \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \mathbf{X}_k^T - 2 \hat{\boldsymbol{\Sigma}}^{-1} \sum_{k=1}^K \lambda_k \| \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \|_F^2 = 0, \tag{37}$$

$$n \hat{\boldsymbol{\Delta}}_k - \mathbf{X}_k^T \, \hat{\boldsymbol{\Sigma}}^{-1} \, \mathbf{X}_k - 2 \lambda_k \hat{\boldsymbol{\Delta}}_k^{-1} \| \, \hat{\boldsymbol{\Sigma}}^{-1} \, \|_F^2 = 0 \qquad \forall \, k = 1, \ldots, K. \tag{38}$$

Assuming these gradient equations (37) and (38) are satisfied, we can write

$$\hat{\boldsymbol{\Sigma}} = \left(\frac{2 \sum_{k=1}^K \lambda_k || \hat{\boldsymbol{\Delta}}_k^{-1} ||_F^2}{p} \, \mathbf{I} + \frac{1}{4p^2} \left(\sum_{k=1}^K \mathbf{X}_k \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \mathbf{X}_k^T \right)^2 \right)^{\frac{1}{2}} + \frac{1}{2p} \sum_{k=1}^K \mathbf{X}_k \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \mathbf{X}_k^T,$$

$$\hat{\boldsymbol{\Delta}}_k = \left(\frac{2 \lambda_k || \hat{\boldsymbol{\Sigma}}^{-1} ||_F^2}{n} \, \mathbf{I} + \frac{1}{4n^2} \left(\mathbf{X}_k^T \, \hat{\boldsymbol{\Sigma}}^{-1} \, \mathbf{X}_k \right)^2 \right)^{\frac{1}{2}} + \frac{1}{2n} \, \mathbf{X}_k^T \, \hat{\boldsymbol{\Sigma}}^{-1} \, \mathbf{X}_k \qquad \forall \, k = 1, \ldots, K.$$

We then can follow the same argument in Proposition 14 to show that $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Delta}}_1, \ldots, \hat{\boldsymbol{\Delta}}_K$ are solutions to the multiplicative Frobenius gradient equations (37) and (38) if and only if

$$\hat{\boldsymbol{\Sigma}} = \mathbf{U} \left[\frac{1}{2p} \left(\boldsymbol{\Gamma} + \left(\boldsymbol{\Gamma}^2 + 8p \sum_{k=1}^K \lambda_k || \hat{\boldsymbol{\Delta}}_k^{-1} ||_F^2 \, \mathbf{I} \right)^{\frac{1}{2}} \right) \right] \mathbf{U}^T \tag{39}$$

$$\text{and} \;\; \hat{\boldsymbol{\Delta}}_k = \mathbf{V}_k \left[\frac{1}{2n} \left(\boldsymbol{\Phi}_k + \left(\boldsymbol{\Phi}_k^2 + 8n\lambda_k || \hat{\boldsymbol{\Sigma}}^{-1} ||_F^2 \, \mathbf{I} \right)^{\frac{1}{2}} \right) \right] \mathbf{V}_k^T \qquad \forall \, k = 1, \ldots, K, \tag{40}$$

where $\mathbf{U}, \mathbf{V}_k, \boldsymbol{\Gamma}$, and $\boldsymbol{\Phi}_k$ are defined by the eigendecompositions $\sum_{k=1}^K \mathbf{X}_k \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \mathbf{X}_k^T = \mathbf{U} \, \boldsymbol{\Gamma} \, \mathbf{U}^T$ and $\mathbf{X}_k^T \, \hat{\boldsymbol{\Sigma}}^{-1} \, \mathbf{X}_k = \mathbf{V}_k \, \boldsymbol{\Phi}_k \, \mathbf{V}_k^T$. This gives rise to the Flip Flop algorithm for solving the multiplicative Frobenius penalized estimators, as given in Algorithm 1.

C.2.3 Additive L_1 Penalized Flip-Flop Estimator

If the inverse covariance matrices are known to have a sparse underlying structure, then we can apply an L_1 penalty, rather than the Frobenius penalty, to induce this sparsity. Note that while it is possible to use a multiplicative L_1 penalty, the multiplicative L_1 penalty is not geodesically convex and is not separable (as defined in Tseng, 2001). Thus, we primarily consider the additive L_1 penalized iPCA estimator:

$$\begin{aligned}\hat{\mathbf{\Sigma}}^{-1}, \hat{\mathbf{\Delta}}_1^{-1}, \dots, \hat{\mathbf{\Delta}}_K^{-1} = \underset{\substack{\mathbf{\Sigma}^{-1}\succ 0\\ \mathbf{\Delta}_1^{-1},\dots,\mathbf{\Delta}_K^{-1}\succ 0}}{\operatorname{argmax}} \quad & \Big\{ p\log|\,\mathbf{\Sigma}^{-1}\,| + n\sum_{k=1}^{K} \log|\,\mathbf{\Delta}_k^{-1}\,| \\ & - \sum_{k=1}^{K} \operatorname{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T\right) \\ & - \lambda_{\mathbf{\Sigma}}\|\,\mathbf{\Sigma}^{-1}\,\|_{1,\text{off}} - \sum_{k=1}^{K} \lambda_k \|\,\mathbf{\Delta}_k^{-1}\,\|_{1,\text{off}}\Big\}. \end{aligned} \tag{41}$$

Note that $\|\cdot\|_{1,\text{off}}$ penalizes the off-diagonal entries (i.e. $\|\,\mathbf{A}\,\|_{1,\text{off}} = \sum_{i\neq j}|a_{ij}|$), but it is also possible to use the ordinary L_1 norm $\|\cdot\|_1$.

For fixed $\hat{\mathbf{\Delta}}_1, \dots, \hat{\mathbf{\Delta}}_K$, the Flip-Flop update for $\hat{\mathbf{\Sigma}}$ is seen to be

$$\hat{\mathbf{\Sigma}}^{-1} = \underset{\mathbf{\Sigma}^{-1}\succ 0}{\operatorname{argmin}} \; -p\log|\,\mathbf{\Sigma}^{-1}\,| + \operatorname{tr}\left(\mathbf{\Sigma}^{-1}\left(\sum_{k=1}^{K}\mathbf{X}_k\,\hat{\mathbf{\Delta}}_k^{-1}\,\mathbf{X}_k^T\right)\right) + \lambda_{\mathbf{\Sigma}}\|\,\mathbf{\Sigma}^{-1}\,\|_{1,\text{off}},$$

and similarly for fixed $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Delta}}_j$, $j \neq k$, the update for $\hat{\mathbf{\Delta}}_k$ is

$$\hat{\mathbf{\Delta}}_k^{-1} = \underset{\mathbf{\Delta}_k^{-1}\succ 0}{\operatorname{argmin}} \; -n\log|\,\mathbf{\Delta}_k^{-1}\,| + \operatorname{tr}\left(\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T\,\hat{\mathbf{\Sigma}}^{-1}\,\mathbf{X}_k\right) + \lambda_k\|\,\mathbf{\Delta}_k^{-1}\,\|_{1,\text{off}}.$$

Both of which can be solved via the graphical lasso algorithm (Hsieh et al., 2011). Plugging in these updates to the framework laid out in Algorithm 3, we give the additive L_1 Flip-Flop algorithm in Algorithm 5.

Algorithm 5 Flip-Flop Algorithm for Additive L_1 Penalized iPCA Covariance Estimators

```
Center the columns of $\mathbf{X}_1, \dots, \mathbf{X}_K$, and initialize $\hat{\mathbf{\Sigma}}$, $\hat{\mathbf{\Delta}}_1, \dots, \hat{\mathbf{\Delta}}_K$ to be positive definite.
while not converged do
    Put $\mathbf{A} = \frac{1}{p}\sum_{k=1}^{K}\mathbf{X}_k\,\mathbf{\Delta}_k^{-1}\,\mathbf{X}_k^T$                                        } Update $\mathbf{\Sigma}$
    Apply graphical lasso:
        $\hat{\mathbf{\Sigma}}^{-1} = \underset{\mathbf{\Sigma}^{-1}}{\operatorname{argmin}} \; -\log|\,\mathbf{\Sigma}^{-1}\,| + \operatorname{tr}(\mathbf{A}\,\mathbf{\Sigma}^{-1}) + \frac{\lambda_{\mathbf{\Sigma}}}{p}\|\,\mathbf{\Sigma}^{-1}\,\|_{1,\text{off}}$
    for $k = 1, \dots, K$ do
        Put $\mathbf{A}_k := \frac{1}{n}\mathbf{X}_k^T\,\mathbf{\Sigma}^{-1}\,\mathbf{X}_k$                                                } Update $\mathbf{\Delta}_k$
        Apply graphical lasso:
            $\hat{\mathbf{\Delta}}_k^{-1} = \underset{\mathbf{\Delta}_k^{-1}}{\operatorname{argmin}} \; -\log|\,\mathbf{\Delta}_k^{-1}\,| + \operatorname{tr}(\mathbf{A}_k\,\mathbf{\Delta}_k^{-1}) + \frac{\lambda_k}{n}\|\,\mathbf{\Delta}_k^{-1}\,\|_{1,\text{off}}$
```

Appendix D. Geodesic Convexity and the Multiplciative Frobenius iPCA Estimator

Because of the central role that geodesic convexity plays in the multiplicative Frobenius iPCA estimator, we give a self-contained introduction to geodesic convexity in Appendix D.1.

This review provides the necessary concepts and tools to prove the theorems in Appendix D.2. For a more comprehensive review, we refer to Rapcsák (1991) and Vishnoi (2018).

D.1 Introduction to Geodesic Convexity

Convex optimization problems arise frequently in a variety of machine learning applications such as regression, matrix completion, and clustering, to name a few examples. Beyond its widespread applications, convex problems are well-understood theoretically and can be reliably solved in polynomial-time. As a result, machine learning tasks are often formulated as convex problems in Euclidean space to guarantee fast convergence to a global solution. Convexity, however, is not limited to Euclidean spaces. Many tools which we know and love from convex optimization can be extended to geodesic convexity (g-convexity) on Riemannian manifolds. In this general Riemannian setting, there are several applications in which we have g-convexity but not convexity (Zhang and Sra, 2016).

Before formally defining geodesic convexity, we first recall some useful concepts from metric geometry. A *metric space* is a pair (X, d) of a set X and a distance function d that satisfies positivity, symmetry, and the triangle inequality. A *path* γ is a continuous mapping from $[0, 1]$ to X, and the *length* ℓ of a path γ is defined as $\ell(\gamma) := \sup\{\sum_{i=1}^{n} d(\gamma(t_{i-1}), \gamma(t_i)) : 0 = t_0 < \ldots < t_n = 1,\ n \in \mathbb{N}\}$. A metric space is a *length space* if $d(x, y) = \inf \ell(\gamma)$ where the infimum is taken over all paths $\gamma : [0, 1] \to X$ joining x and y.

Definition 15 *Let (X, d) be a length space. A path $\gamma : [0, 1] \to X$ is a* ***geodesic*** *if for every $t \in [0, 1]$ there exists an interval $[a, b] \subset [0, 1]$ which contains a neighborhood of t and $\gamma|_{[a,b]}$ is a shortest path from $\gamma(a)$ to $\gamma(b)$. Put simply, a geodesic is a path which locally minimizes length.*

Note that geodesics minimize length locally, but not globally. A canonical example of geodesics are the great circles on a sphere.

This concept of a geodesic generalizes the notion of a line in Euclidean space to general (nonlinear) length spaces. By replacing lines by geodesics in the definition of convexity, we can extend convexity to g-convexity in a straightforward manner.

Definition 16 *Let $\mathcal{M}$ be a Riemannian manifold. A function $f : \mathcal{M} \to \mathbb{R}$ is* ***geodesically convex*** *if for any $x, y \in \mathcal{M}$, geodesic γ such that $\gamma(0) = x$ and $\gamma(1) = y$, and $t \in [0, 1]$, it holds that*

$$f(\gamma(t)) \leq (1 - t)f(x) + tf(y).$$

The only caveat is that the underlying space must be a Riemannian manifold. To prevent a long winded detour into the details of Riemannian geometry, we avoid the full technical definition and simply think of a Riemannian manifold as a real differentiable manifold equipped with the notion of an inner product. We need the structure of a Riemannian manifold in order to meaningfully perform algebraic operations in our space.

In summary, by extending the notion of a line to a geodesic, we can easily translate the notion of convexity on a Euclidean space to geodesic convexity on a Riemannian manifold. We next give a series of known properties regarding geodesic convexity that will be of use in the following proofs (Wiesel, 2012; Vishnoi, 2018). Many of these properties are analogous to the familiar convex setting.

Theorem 17 *Any local minima of a geodesically convex function is a global minima.*

Proposition 18 *The following operations preserve geodesic convexity:*

(i) (Addition) If f and g are g-convex functions, then $f + g$ is g-convex.

(ii) (Kronecker Products) Suppose f is a real-valued, g-convex function on $\mathbb{S}^d_{++}$, and $\mathbf{Q}_j \in \mathbb{S}^{d_j}_{++}$ for each $j = 1, \dots, J$ such that $\prod_{j=1}^J d_j = d$. Then

$$g(\mathbf{Q}_1, \dots, \mathbf{Q}_K) = f(\mathbf{Q}_1 \otimes \cdots \otimes \mathbf{Q}_K)$$

is jointly g-convex in $\{\mathbf{Q}_j\}_{j=1}^J \in \mathbb{S}^{d_1}_{++} \times \cdots \times \mathbb{S}^{d_K}_{++}$.

The proof of Proposition 18(i) is straightforward from the definition of g-convex functions, and Proposition 18(ii) is proved in Wiesel (2012).

Remark 19 *(Example 4.9 in Vishnoi, 2018) Consider the manifold of positive definite matrices $\mathbb{S}^n_{++}$. For $\mathbf{Q}_0, \mathbf{Q}_1 \in \mathbb{S}^n_{++}$, the geodesic joining $\mathbf{Q}_0$ to $\mathbf{Q}_1$ can be parameterized as*

$$\mathbf{Q}_t = \mathbf{Q}_0^{\frac{1}{2}} \left(\mathbf{Q}_0^{-\frac{1}{2}} \mathbf{Q}_1 \mathbf{Q}_0^{-\frac{1}{2}} \right)^t \mathbf{Q}_0^{\frac{1}{2}} \qquad \forall\, t \in [0, 1]. \tag{42}$$

D.2 Global Convergence of the Multiplicative Frobenius iPCA Estimator

We now have the tools to start proving global convergence of the multiplicative iPCA estimator. The roadmap of this proofs is as follows. First, we prove that the negative log-likelihood is g-convex. Then, we prove that the multiplicative Frobenius penalty is g-convex. Since the sum of g-convex functions is g-convex, this implies that the multiplicative iPCA estimator is a g-convex optimization problem. Furthermore, since the Flip-Flop algorithm was proven to converge in Theorem 7, it follows that the multiplicative Frobenius iPCA estimator converges to the global solution as a consequence of geodesic convexity!

Without loss of generality, we assume that each data set $\mathbf{X}_k$ has been column-centered for the remainder of this section.

Lemma 20 *The negative log-likelihood of our model,* (12)*, is jointly geodesically convex in $\mathbf{\Sigma}^{-1}$, $\mathbf{\Delta}_1^{-1}, \dots \mathbf{\Delta}_K^{-1}$ with respect to $\mathbb{S}^n_{++} \times \mathbb{S}^{p_1}_{++} \times \cdots \times \mathbb{S}^{p_K}_{++}$.*

Proof Let $\tilde{\mathbf{X}} = [\mathbf{X}_1, \dots, \mathbf{X}_K] \in \mathbb{R}^{n \times p}$ and define

$$\tilde{\mathbf{\Delta}} = \begin{bmatrix} \mathbf{\Delta}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{\Delta}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Delta}_K \end{bmatrix}$$

Since $\tilde{\mathbf{\Delta}}$ is a block diagonal matrix, then $|\tilde{\mathbf{\Delta}}| = \prod_{k=1}^K |\, \mathbf{\Delta}_k \,|$ and $|\tilde{\mathbf{\Delta}}^{-1}| = \prod_{k=1}^K |\, \mathbf{\Delta}_k^{-1} \,|$. This implies that

$$\sum_{k=1}^K \log |\, \mathbf{\Delta}_k^{-1} \,| = \log(\prod_{k=1}^K |\, \mathbf{\Delta}_k^{-1} \,|) = \log |\tilde{\mathbf{\Delta}}^{-1}|. \tag{43}$$

Also, using this new parameterization,

$$\sum_{k=1}^{K} \operatorname{tr}(\mathbf{X}_k\, \boldsymbol{\Delta}_k^{-1}\, \mathbf{X}_k^T\, \boldsymbol{\Sigma}^{-1}) = \operatorname{tr}(\tilde{\mathbf{X}}\tilde{\boldsymbol{\Delta}}^{-1}\tilde{\mathbf{X}}^T\, \boldsymbol{\Sigma}^{-1}) = \operatorname{vec}(\tilde{\mathbf{X}})^T(\tilde{\boldsymbol{\Delta}}^{-1} \otimes \boldsymbol{\Sigma}^{-1}) \operatorname{vec}(\tilde{\mathbf{X}}). \qquad (44)$$

Using (43) and (44), we rewrite the negative log-likelihood as

$$\begin{aligned}
-\ell(\boldsymbol{\Sigma}^{-1}, \boldsymbol{\Delta}_1^{-1}, \dots, \boldsymbol{\Delta}_K^{-1}) &\propto -p \log |\, \boldsymbol{\Sigma}^{-1} \,| - n \sum_{k=1}^{K} \log |\, \boldsymbol{\Delta}_k^{-1} \,| + \sum_{k=1}^{K} \operatorname{tr}(\mathbf{X}_k\, \boldsymbol{\Delta}_k^{-1}\, \mathbf{X}_k^T\, \boldsymbol{\Sigma}^{-1}) \\
&= -p \log |\, \boldsymbol{\Sigma}^{-1} \,| - n \log |\tilde{\boldsymbol{\Delta}}^{-1}| + \operatorname{vec}(\tilde{\mathbf{X}})^T(\tilde{\boldsymbol{\Delta}}^{-1} \otimes \boldsymbol{\Sigma}^{-1}) \operatorname{vec}(\tilde{\mathbf{X}}) \\
&= -\log(|\, \boldsymbol{\Sigma}^{-1} \,|^p |\tilde{\boldsymbol{\Delta}}^{-1}|^n) + \operatorname{vec}(\tilde{\mathbf{X}})^T(\tilde{\boldsymbol{\Delta}}^{-1} \otimes \boldsymbol{\Sigma}^{-1}) \operatorname{vec}(\tilde{\mathbf{X}}) \\
&= \log(|\tilde{\boldsymbol{\Delta}}^{-1} \otimes \boldsymbol{\Sigma}^{-1} \,|^{-1}) + \operatorname{vec}(\tilde{\mathbf{X}})^T(\tilde{\boldsymbol{\Delta}}^{-1} \otimes \boldsymbol{\Sigma}^{-1}) \operatorname{vec}(\tilde{\mathbf{X}}).
\end{aligned}$$

Next, consider the manifold $\mathbb{S}^{np}_{++}$, and let the function $f : \mathbb{S}^{np}_{++} \to \mathbb{R}$ be given by

$$f(\mathbf{Q}) = \log(|\, \mathbf{Q} \,|^{-1}) + \operatorname{vec}(\tilde{\mathbf{X}})^T\, \mathbf{Q} \operatorname{vec}(\tilde{\mathbf{X}}).$$

Since f is the sum of g-convex functions on $\mathbb{S}^{np}_{++}$ (Wiesel, 2012), and $-\ell(\boldsymbol{\Sigma}^{-1}, \tilde{\boldsymbol{\Delta}}^{-1}) = f(\tilde{\boldsymbol{\Delta}}^{-1} \otimes \boldsymbol{\Sigma}^{-1})$, then by Proposition 18, the negative log-likelihood is jointly geodesically convex in $\boldsymbol{\Sigma}^{-1}, \boldsymbol{\Delta}_1^{-1}, \dots, \boldsymbol{\Delta}_K^{-1}$ with respect to $\mathbb{S}^{n}_{++} \times \mathbb{S}^{p_1}_{++} \times \cdots \times \mathbb{S}^{p_K}_{++}$. ■

Corollary 21 *Suppose the unpenalized log-likelihood in* (20) *is bounded. If the Flip-Flop estimators for the unpenalized log-likelihood exist, then they converge to the global solution of* (20).

Proof

Recall that the Flip-Flop algorithm yields the iterates:

1. $\hat{\boldsymbol{\Sigma}} = \frac{1}{p} \sum_{k=1}^{K} \mathbf{X}_k\, \hat{\boldsymbol{\Delta}}_k^{-1}\, \mathbf{X}_k^T = \operatorname{argmax}_{\,\boldsymbol{\Sigma}}\, \ell(\boldsymbol{\Sigma}, \hat{\boldsymbol{\Delta}}_1, \dots, \hat{\boldsymbol{\Delta}}_K)$
2. For each $k = 1, \dots, K$,
   $\hat{\boldsymbol{\Delta}}_k = \frac{1}{n}\, \mathbf{X}_k^T (\hat{\boldsymbol{\Sigma}})^{-1}\, \mathbf{X}_k = \operatorname{argmax}_{\,\boldsymbol{\Delta}_k}\, \ell(\hat{\boldsymbol{\Sigma}}, \hat{\boldsymbol{\Delta}}_1, \dots, \hat{\boldsymbol{\Delta}}_{k-1}, \boldsymbol{\Delta}_k, \hat{\boldsymbol{\Delta}}_{k+1}, \dots, \hat{\boldsymbol{\Delta}}_K)$

Thus, each update of the Flip-Flop algorithm monotonically increases the log-likelihood ℓ. Assuming that the MLEs exist and are bounded, this Flip-Flop algorithm converges to a local maximum. Furthermore, since ℓ is jointly geodesically concave in $\boldsymbol{\Sigma}^{-1}$, $\boldsymbol{\Delta}_1^{-1}, \dots \boldsymbol{\Delta}_K^{-1}$, then all local maxima are global maxima so that the Flip-Flop estimators for the unpenalized log-likelihood converge to the global solution. ■

Lemma 22 *The multiplicative Frobenius norm penalty,* $\|\, \boldsymbol{\Sigma}^{-1} \,\|_F^2 \sum_{k=1}^{K} \lambda_k \|\, \boldsymbol{\Delta}_k^{-1} \,\|_F^2$*, is jointly geodesically convex in* $\boldsymbol{\Sigma}^{-1}$*,* $\boldsymbol{\Delta}_1^{-1}, \dots, \boldsymbol{\Delta}_K^{-1}$ *with respect to* $\mathbb{S}^{n}_{++} \times \mathbb{S}^{p_1}_{++} \times \cdots \times \mathbb{S}^{p_K}_{++}$.

Proof Since $\operatorname{tr}(\mathbf{A} \otimes \mathbf{B}) = \operatorname{tr}(\mathbf{A})\operatorname{tr}(\mathbf{B})$ and $\| \mathbf{A} \|_F^2 = \operatorname{tr}(\mathbf{A}^T \mathbf{A})$, then

$$\begin{aligned}
P(\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}) &:= \| \mathbf{\Sigma}^{-1} \|_F^2 \sum_{k=1}^{K} \lambda_k \| \mathbf{\Delta}_K^{-1} \|_F^2 && (45)\\
&= \sum_{k=1}^{K} \lambda_k \operatorname{tr}(\mathbf{\Sigma}^{-2}) \operatorname{tr}(\mathbf{\Delta}_k^{-2}) && (46)\\
&= \sum_{k=1}^{K} \lambda_k \operatorname{tr}(\mathbf{\Delta}_k^{-2} \otimes \mathbf{\Sigma}^{-2}) && (47)\\
&= \sum_{k=1}^{K} \lambda_k \| \mathbf{\Delta}_k^{-1} \otimes \mathbf{\Sigma}^{-1} \|_F^2 && (48)\\
&= \sum_{k=1}^{K} \| (\sqrt{\lambda_k}\, \mathbf{\Delta}_k^{-1}) \otimes \mathbf{\Sigma}^{-1} \|_F^2 && (49)\\
&= \| \bar{\mathbf{\Delta}}^{-1} \otimes \mathbf{\Sigma}^{-1} \|_F^2, && (50)
\end{aligned}$$

where

$$\bar{\mathbf{\Delta}} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} \mathbf{\Delta}_1 & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} \mathbf{\Delta}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sqrt{\lambda_K}} \mathbf{\Delta}_K \end{bmatrix}.$$

We will next show that the function $f : \mathbb{S}_{++}^{np} \to \mathbb{R}$ defined by

$$f(\mathbf{Q}^\alpha) := \| \mathbf{Q}^\alpha \|_F^2 = \operatorname{tr}((\mathbf{Q}^\alpha)^T \mathbf{Q}^\alpha) = \operatorname{tr}((\mathbf{Q}^\alpha)^2), \qquad \alpha \in \{\pm 1\} \tag{51}$$

is geodesically convex in $\mathbf{Q} \in \mathbb{S}_{++}^{np}$. That is, let $\mathbf{Q}_0, \mathbf{Q}_1 \in \mathbb{S}_{++}^{np}$ be given, and let $\mathbf{Q}_t$ be the geodesic between $\mathbf{Q}_0$ and $\mathbf{Q}_1$ as given in (42). We want to show that $f(\mathbf{Q}_t^\alpha)$ is a convex function with respect to t.

So consider the eigendecomposition $\mathbf{Q}_0^{-\frac{1}{2}} \mathbf{Q}_1 \mathbf{Q}_0^{-\frac{1}{2}} = \mathbf{U} \mathbf{D} \mathbf{U}^T$, where $\mathbf{U}$ is an orthogonal matrix and $\mathbf{D}$ is a diagonal matrix with diagonal entries d_i. Then for all $t \in [0, 1]$, it follows from (51) that

$$\begin{aligned}
f(\mathbf{Q}_t^\alpha) &= \operatorname{tr}(\mathbf{Q}_t^\alpha \mathbf{Q}_t^\alpha) && (52)\\
&= \operatorname{tr}(\mathbf{Q}_0^{\frac{\alpha}{2}} (\mathbf{Q}_0^{-\frac{1}{2}} \mathbf{Q}_1 \mathbf{Q}_0^{-\frac{1}{2}})^{\alpha t} \mathbf{Q}_0^{\alpha} (\mathbf{Q}_0^{-\frac{1}{2}} \mathbf{Q}_1 \mathbf{Q}_0^{-\frac{1}{2}})^{\alpha t} \mathbf{Q}_0^{\frac{\alpha}{2}}) && (53)\\
&= \operatorname{tr}(\mathbf{Q}_0^{\frac{\alpha}{2}} (\mathbf{U} \mathbf{D} \mathbf{U}^T)^{\alpha t} \mathbf{Q}_0^{\alpha} (\mathbf{U} \mathbf{D} \mathbf{U}^T)^{\alpha t} \mathbf{Q}_0^{\frac{\alpha}{2}}) && (54)\\
&= \operatorname{tr}(\mathbf{Q}_0^{\frac{\alpha}{2}} \mathbf{U} \mathbf{D}^{\alpha t} \mathbf{U}^T \mathbf{Q}_0^{\alpha} \mathbf{U} \mathbf{D}^{\alpha t} \mathbf{U}^T \mathbf{Q}_0^{\frac{\alpha}{2}}) && (55)\\
&= \operatorname{tr}(\mathbf{D}^{\alpha t} \mathbf{U}^T \mathbf{Q}_0^{\alpha} \mathbf{U} \mathbf{D}^{\alpha t} \mathbf{U}^T \mathbf{Q}_0^{\alpha} \mathbf{U}) && (56)\\
&= \operatorname{tr}(\mathbf{D}^{\alpha t} \mathbf{A} \mathbf{D}^{\alpha t} \mathbf{A}), && (57)
\end{aligned}$$

where $\mathbf{A} := \mathbf{U}^T \mathbf{Q}_0^{\alpha} \mathbf{U}$.

Because $\mathbf{A}$ is symmetric and $(\mathbf{D}^{\alpha t}\,\mathbf{A})_{ij} = d_i^{\alpha t} a_{ij}$, then

$$(\mathbf{D}^{\alpha t}\,\mathbf{A}\,\mathbf{D}^{\alpha t}\,\mathbf{A})_{ij} = \sum_{l=1}^{np}((\mathbf{D}^{\alpha t}\,\mathbf{A})_{il}(\mathbf{D}^{\alpha t}\,\mathbf{A})_{lj}) = \sum_{l=1}^{np}(d_i^{\alpha t} a_{il} d_l^{\alpha t} a_{lj}) = \sum_{l=1}^{np}(d_i^{\alpha t} a_{il} d_l^{\alpha t} a_{jl}).$$

Plugging this into (57) gives

$$f(\mathbf{Q}_t^{\alpha}) = \sum_{i=1}^{np}(\mathbf{D}^{\alpha t}\,\mathbf{A}\,\mathbf{D}^{\alpha t}\,\mathbf{A})_{ii} = \sum_{i=1}^{np}\sum_{l=1}^{np}(d_i^{\alpha t} a_{il} d_l^{\alpha t} a_{il}) = \sum_{i=1}^{np}\sum_{l=1}^{np}(a_{il}^2 (d_i d_l)^{\alpha t}).$$

Note that since $\mathbf{Q}_0$ and $\mathbf{Q}_1$ are positive definite, then $\mathbf{Q}_0^{-\frac{1}{2}}\,\mathbf{Q}_1\,\mathbf{Q}_0^{-\frac{1}{2}}$ is positive definite. Thus, $d_i > 0$ for each $i = 1, \ldots, np$, and because $d_i > 0$ for each $i = 1, \ldots, np$, $f(\mathbf{Q}_t^{\alpha}) = \sum_{i=1}^{np}\sum_{l=1}^{np}(a_{il}^2 (d_i d_l)^{\alpha t})$ is a convex function in t. This proves f is g-convex in $\mathbf{Q} \in \mathbb{S}_{++}^{np}$.

Since f is g-convex in $\mathbf{Q} \in \mathbb{S}_{++}^{np}$ and $P(\mathbf{\Sigma}^{-1}, \bar{\mathbf{\Delta}}^{-1}) = f(\bar{\mathbf{\Delta}}^{-1} \otimes \mathbf{\Sigma}^{-1})$ from (50), then the multiplicative Frobenius norm penalty P is g-convex in $\mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}$ by Proposition 18(ii). ■

Theorem 8 *The multiplicative Frobenius iPCA estimator is jointly geodesically convex in* $\mathbf{\Sigma}^{-1}$ *and* $\mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}$. *Because of this, the Flip-Flop algorithm for the multiplicative Frobenius iPCA estimator given in Algorithm 1 converges to the global solution.*

Proof From Lemma 20 and Lemma 22, we see that the objective function corresponding to the multiplicative Frobenius penalized estimator in (36) is the sum of jointly g-convex functions. Thus, the multiplicative Frobenius penalized estimator is also jointly geodesically convex in $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}_1^{-1}, \ldots, \mathbf{\Delta}_K^{-1}$.

Recall we have already proved that Algorithm 1 converges to a stationary point in Theorem 7. Since the multiplicative Frobenius iPCA estimator is geodesically convex, then all local optima are global optima, and the Flip-Flop algorithm for the multiplicative Frobenius iPCA estimator converges to the global solution of (36). ■

Remark 23 *The multiplicative Frobenius iPCA estimator is unique in the sense that the Kronecker production solution* $\hat{\mathbf{\Sigma}} \otimes \hat{\mathbf{\Delta}}_k$ *is unique.*

Appendix E. Equivalence between PCA and iPCA with Frobenius Penalties when $K = 1$

One major reason for using the Frobenius penalties is that in the case where we observe only one data set, iPCA with the Frobenius penalties and the classical PCA are equivalent in the sense that the PC scores and loadings are the same. Throughout this section, we will assume $K = 1$, and we let the SVD of $\mathbf{X}$ (which has been column-centered) be given by $\mathbf{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{D} \in \mathbb{R}^{n \times p}$ with the diagonal elements d_i, $\mathbf{V} \in \mathbb{R}^{p \times p}$, and $r = \text{rank}(\mathbf{X}) < \min\{n, p\}$. Without loss of generality, suppose also that $p \geq n$. In PCA, it is well known that the PC scores are given by the columns of $\mathbf{U}$ and the PC loadings are given by the columns of $\mathbf{V}$. We will show that iPCA with the Frobenius penalties also yield the same PC scores $\mathbf{U}$ and loadings $\mathbf{V}$.

Let us first consider iPCA with the additive Frobenius penalty $\lambda_{\boldsymbol{\Sigma}} \| \boldsymbol{\Sigma}^{-1} \|_F^2 + \lambda_{\boldsymbol{\Delta}} \| \boldsymbol{\Delta}^{-1} \|_F^2$. By Theorem 1 in Allen and Tibshirani (2010), there is a unique global solution maximizing the matrix-variate normal log-likelihood with additive Frobenius penalties (21) when $K = 1$. This global solution is given by

$$\hat{\boldsymbol{\Sigma}} = \mathbf{U}\,\boldsymbol{\beta}\,\mathbf{U}^T \quad \text{and} \quad \hat{\boldsymbol{\Delta}} = \mathbf{V}\,\boldsymbol{\theta}\,\mathbf{V}^T,$$

where $\boldsymbol{\beta} = \text{diag}(\beta_1, \ldots, \beta_n)$ and $\boldsymbol{\theta} = \text{diag}(\theta_1, \ldots, \theta_p)$ are defined as follows:

$$\beta_i = \begin{cases} \dfrac{d_i^2 \theta_i}{n\theta_i^2 - 2\lambda_{\boldsymbol{\Delta}}}, & i = 1, \ldots, r \\ \sqrt{\dfrac{2\lambda_{\boldsymbol{\Sigma}}}{p}}, & i = r+1, \ldots, n \end{cases}$$

$$\theta_i = \begin{cases} \sqrt{\dfrac{-c_{2,i} - \sqrt{c_{2,i}^2 - 4c_1 c_{3,i}}}{2c_1}}, & i = 1, \ldots, r \\ \sqrt{\dfrac{2\lambda_{\boldsymbol{\Delta}}}{n}}, & i = r+1, \ldots, p, \end{cases}$$

with coefficients

$$\begin{aligned} c_1 &= -2\lambda_{\boldsymbol{\Sigma}} n^2, \\ c_{2,i} &= d_i^4(p-n) + 8n\lambda_{\boldsymbol{\Sigma}}\lambda_{\boldsymbol{\Delta}}, \\ c_{3,i} &= 2\lambda_{\boldsymbol{\Delta}}(d_i^4 - 4\lambda_{\boldsymbol{\Sigma}}\lambda_{\boldsymbol{\Delta}}). \end{aligned}$$

Since the PC scores and loadings from iPCA are the eigenvectors of $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Delta}}$, respectively, the above result shows that the PC scores and loadings from iPCA with the additive Frobenius penalty are precisely $\mathbf{U}$ and $\mathbf{V}$, as in PCA.

We next investigate the equivalence of iPCA with the multiplicative Frobenius penalty $\lambda \| \boldsymbol{\Sigma}^{-1} \|_F^2 \| \boldsymbol{\Delta}^{-1} \|_F^2$ and PCA when $K = 1$. While we can follow a similar argument as Allen and Tibshirani (2010), the proof is complicated by the non-separable penalty terms. That is, each term in the multiplicative penalty depends on both $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Delta}}$. We will return to this complication in the proof of the following theorem.

Theorem 24 *In the case when $K = 1$, the solution to iPCA with the multiplicative Frobenius penalty* (36) *is given by $\hat{\boldsymbol{\Sigma}} = \mathbf{U}\,\boldsymbol{\beta}\,\mathbf{U}^T$ and $\hat{\boldsymbol{\Delta}} = \mathbf{V}\,\boldsymbol{\theta}\,\mathbf{V}^T$, where $\boldsymbol{\beta} = \text{diag}(\beta_1, \ldots, \beta_n)$*

and $\boldsymbol{\theta} = \operatorname{diag}(\theta_1, \ldots, \theta_p)$ *satisfy the system*

$$\begin{cases} \theta_i = \sqrt{\dfrac{-c_{2,i}(T) - \sqrt{c_{2,i}^2(T) - 4c_1(T)c_{3,i}(T)}}{2c_1(T)}}, & i = 1, \ldots, r \\ \theta_i = \sqrt{\dfrac{2\lambda}{n}}, & i = r+1, \ldots, p \\ \beta_i = \dfrac{d_i^2 \theta_i}{n\theta_i^2 - 2\lambda}, & i = 1, \ldots, r \\ \beta_i = \sqrt{\dfrac{2\lambda T}{p}}, & i = r+1, \ldots, n \\ T = \displaystyle\sum_{k=1}^{r} \theta_k^{-2} + (p-r)\dfrac{n}{2\lambda}, & \end{cases}$$

with $c_1(T) = -2\lambda n^2 T$*,* $c_{2,i}(T) = d_i^4(p-n) + 8n\lambda^2 T$*, and* $c_{3,i}(T) = 2\lambda(d_i^4 - 4\lambda^2 T)$*. This solution exists and is unique (up to a scaling factor).*

Proof As seen previously, the gradient equations from the matrix-variate normal log-likelihood with the multiplicative Frobenius penalty are

$$p\hat{\boldsymbol{\Sigma}} - 2\lambda\hat{\boldsymbol{\Sigma}}^{-1} \| \hat{\boldsymbol{\Delta}}^{-1} \|_F^2 = \mathbf{X}\,\hat{\boldsymbol{\Delta}}^{-1}\,\mathbf{X}^T$$
$$n\hat{\boldsymbol{\Delta}} - 2\lambda\hat{\boldsymbol{\Delta}}^{-1} \| \hat{\boldsymbol{\Sigma}}^{-1} \|_F^2 = \mathbf{X}^T\,\hat{\boldsymbol{\Sigma}}^{-1}\,\mathbf{X}\,.$$

Thus, the eigenvectors of $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Delta}}$ are equal to their respective quadratic forms. It follows that there is only one solution for the eigenvectors—namely, the left and right singular vectors of $\mathbf{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$. [Note: the last $n-r$ eigenvectors of $\mathbf{U}$ and the last $p-r$ eigenvectors of $\mathbf{V}$ are not unique.]

Put $\hat{\boldsymbol{\Sigma}} = \mathbf{U}\,\boldsymbol{\beta}\,\mathbf{U}^T$ and $\hat{\boldsymbol{\Delta}} = \mathbf{V}\,\boldsymbol{\theta}\,\mathbf{V}^T$, where $\boldsymbol{\beta} = \operatorname{diag}(\beta_1, \ldots, \beta_n)$ and $\boldsymbol{\theta} = \operatorname{diag}(\theta_1, \ldots, \theta_p)$. Note that we have the implicit constraints $\beta_i > 0$ and $\theta_i > 0$ for each i due to the positive definiteness of the covariances. Plugging in these decompositions and the SVD of $\mathbf{X}$ into the gradient equations gives

$$p\,\mathbf{U}\,\boldsymbol{\beta}\,\mathbf{U}^T - 2\lambda\,\mathbf{U}\,\boldsymbol{\beta}^{-1}\,\mathbf{U}^T \| \mathbf{V}\,\boldsymbol{\theta}^{-1}\,\mathbf{V}^T \|_F^2 = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T\,\mathbf{V}\,\boldsymbol{\theta}^{-1}\,\mathbf{V}^T\,\mathbf{V}\,\mathbf{D}^T\,\mathbf{U}^T$$
$$n\,\mathbf{V}\,\boldsymbol{\theta}\,\mathbf{V}^T - 2\lambda\,\mathbf{V}\,\boldsymbol{\theta}^{-1}\,\mathbf{V}^T \| \mathbf{U}\,\boldsymbol{\beta}^{-1}\,\mathbf{U}^T \|_F^2 = \mathbf{V}\,\mathbf{D}^T\,\mathbf{U}^T\,\mathbf{U}\,\boldsymbol{\beta}^{-1}\,\mathbf{U}^T\,\mathbf{U}\,\mathbf{D}\,\mathbf{V}^T,$$

or equivalently, using the orthogonality of $\mathbf{U}$ and $\mathbf{V}$,

$$p\boldsymbol{\beta} - 2\lambda\boldsymbol{\beta}^{-1} \| \boldsymbol{\theta}^{-1} \|_F^2 = \mathbf{D}\,\boldsymbol{\theta}^{-1}\,\mathbf{D}^T$$
$$n\boldsymbol{\theta} - 2\lambda\boldsymbol{\theta}^{-1} \| \boldsymbol{\beta}^{-1} \|_F^2 = \mathbf{D}^T\,\boldsymbol{\beta}^{-1}\,\mathbf{D}\,.$$

We can write this element-wise as the following system of equations:

$$p\beta_i - 2\lambda\beta_i^{-1}\sum_{k=1}^{p}\theta_k^{-2} = d_i^2\theta_i^{-1}, \qquad i = 1,\dots,r \tag{58}$$

$$p\beta_i - 2\lambda\beta_i^{-1}\sum_{k=1}^{p}\theta_k^{-2} = 0, \qquad i = r+1,\dots,n \tag{59}$$

$$n\theta_i - 2\lambda\theta_i^{-1}\sum_{k=1}^{n}\beta_k^{-2} = d_i^2\beta_i^{-1}, \qquad i = 1,\dots,r \tag{60}$$

$$n\theta_i - 2\lambda\theta_i^{-1}\sum_{k=1}^{n}\beta_k^{-2} = 0, \qquad i = r+1,\dots,p. \tag{61}$$

Now, to simplify this system of equations, we notice that if $(\boldsymbol{\beta},\boldsymbol{\theta})$ solve this system, then for any positive scalar c, $(c\boldsymbol{\beta}, c^{-1}\boldsymbol{\theta})$ is also a solution. Without loss of generality, we may thus assume that $\boldsymbol{\beta}$ is normalized so that $\sum_{k=1}^{n}\beta_k^{-2} = 1$.

It immediately follows from (61) that $\theta_i = \sqrt{\frac{2\lambda}{n}}$ for $i = r+1,\dots,p$.

On the other hand, for $i = 1,\dots,r$, (60) gives us that

$$\begin{aligned} n\theta_i^2\beta_i - 2\lambda\beta_i &= d_i^2\theta_i \\ \implies \beta_i &= \frac{d_i^2\theta_i}{n\theta_i^2 - 2\lambda}. \end{aligned}$$

Substituting this equation for β_i into (58) then yields

$$\frac{pd_i^2\theta_i}{n\theta_i^2 - 2\lambda} - \frac{2\lambda(n\theta_i^2 - 2\lambda)}{d_i^2\theta_i}\sum_{k=1}^{p}\theta_k^{-2} = \frac{d_i^2}{\theta_i}$$

Finding a common denominator and expanding all terms yields

$$[-2\lambda n^2 T]\theta_i^4 + [d_i^4(p-n) + 8n\lambda^2 T]\theta_i^2 + [2\lambda(d_i^4 - 4\lambda^2 T)] = 0,$$

where $T = \sum_{k=1}^{p}\theta_k^{-2} = \sum_{k=1}^{r}\theta_k^{-2} + (p-r)\frac{n}{2\lambda}$. For the sake of notation, let us define $c_1(T) = -2\lambda n^2 T$, $c_{2,i}(T) = d_i^4(p-n) + 8n\lambda^2 T$, and $c_{3,i}(T) = 2\lambda(d_i^4 - 4\lambda^2 T)$.

Summarizing what we have done so far, $\boldsymbol{\beta} \succ 0$ and $\boldsymbol{\theta} \succ 0$ must satisfy:

$$\beta_i = \frac{d_i^2\theta_i}{n\theta_i^2 - 2\lambda}, \qquad i = 1,\dots,r \tag{62}$$

$$\beta_i = \sqrt{\frac{2\lambda T}{p}}, \qquad i = r+1,\dots,n \tag{63}$$

$$0 = c_1(T)\theta_i^4 + c_{2,i}(T)\theta_i^2 + c_{3,i}(T), \qquad i = 1,\dots,r \tag{64}$$

$$\theta_i = \sqrt{\frac{2\lambda}{n}}, \qquad i = r+1,\dots,p \tag{65}$$

$$T = \sum_{k=1}^{r}\theta_k^{-2} + (p-r)\frac{n}{2\lambda}. \tag{66}$$

Now, from the quartic polynomial in (64), the four possible roots are

$$\theta_i = \pm\sqrt{\frac{-c_{2,i}(T) \pm \sqrt{c_{2,i}^2(T) - 4c_1(T)c_{3,i}(T)}}{2c_1(T)}}.$$

In any case, $c_1(T) < 0$, $c_{2,i}(T) > 0$, and

$$c_{2,i}^2(T) - 4c_1(T)c_{3,i}(T) = d_i^8(p-n)^2 + 16\lambda^2 npd_i^4 T > 0,$$

so that

$$\theta_i = \sqrt{\frac{-c_{2,i}(T) - \sqrt{c_{2,i}^2(T) - 4c_1(T)c_{3,i}(T)}}{2c_1(T)}}. \tag{67}$$

is always a real positive root. Furthermore, if θ_i is given by (67) for each $i = 1, \dots, r$, then

$$\begin{aligned} n\theta_i^2 - 2\lambda &= \frac{1}{-4\lambda nT}\left[-d_i^4(p-n) - 8n\lambda^2 T - \sqrt{d_i^8(p-n)^2 + 16\lambda^2 npd_i^4 T}\right] - 2\lambda \\ &= 2\lambda + \frac{1}{4\lambda nT}\left[d_i^4(p-n) + \sqrt{d_i^8(p-n)^2 + 16\lambda^2 npd_i^4 T}\right] - 2\lambda \\ &\geq 0. \end{aligned}$$

Thus, the corresponding β_i, which we have already shown to be given by $\beta_i = \frac{d_i^2\theta_i}{n\theta_i^2 - 2\lambda}$, is also positive. This shows that (67) yields a feasible solution for our system of equations (62)-(66). We claim that this is the only choice of θ_i, which yields a feasible solution. To see this, we first immediately eliminate the two negative square root solutions due to the positivity constraint on θ_i. We next divide the argument into three cases.

Case 1: If $d_i^4 - 4\lambda^2 T = 0$, then $c_{3,1}(T) = 0$ and so

$$\theta_i = \sqrt{\frac{-c_{2,i}(T) \pm c_{2,i}(T)}{2c_1(T)}},$$

which equals 0 if we take the positive sign. Thus, in this case, there is only one feasible root by choosing the negative sign.

Case 2: If $d_i^4 - 4\lambda^2 T > 0$, then $c_{3,i}(T) > 0$. Additionally, $c_1(T) < 0$ and $c_{2,i}(T) > 0$, so by Descartes' rule of signs, there is at most one real positive solution to (64). As shown earlier, the root given in (67) is a real positive solution. This must be the unique real positive root by Descartes'.

Case 3: If $d_i^4 - 4\lambda^2 T < 0$, then $c_{3,i}(T) < 0$, so by Descartes' rule of signs, there are at most two real positive solutions to (64). We have already found one real positive root, given by (67). Suppose also that the other possible root

$$\theta_i = \sqrt{\frac{-c_{2,i}(T) + \sqrt{c_{2,i}^2(T) - 4c_1(T)c_{3,i}(T)}}{2c_1(T)}}. \tag{68}$$

is a real positive root. Since the corresponding β_i must also be positive (in order to be feasible) and we have already shown that $\beta_i = \frac{d_i^2\theta_i}{n\theta_i^2 - 2\lambda}$, it follows that the denominator $n\theta_i^2 - 2\lambda$ must be positive. However, substituting (68) into this denominator gives

$$\begin{aligned}
n\theta_i^2 - 2\lambda &= \frac{1}{-4\lambda nT}\left[-d_i^4(p-n) - 8n\lambda^2 T + \sqrt{d_i^8(p-n)^2 + 16\lambda^2 npd_i^4 T}\right] - 2\lambda \\
&= 2\lambda + \frac{1}{4\lambda nT}\left[d_i^4(p-n) - \sqrt{d_i^8(p-n)^2 + 16\lambda^2 npd_i^4 T}\right] - 2\lambda \\
&\leq 0.
\end{aligned}$$

This contradicts the positivity of β_i and hence is not a feasible solution.

Thus, in any case, we have shown that there is only one feasible root of (64), which is given by (67).

The last step of this proof is to show that there exists a T which solves our system of equations (62)-(66). In particular, we can substitute (67) into (66) to see that T must satisfy

$$\begin{aligned}
T &= \sum_{k=1}^{r} \frac{2c_1(T)}{-c_{2,i}(T) - \sqrt{c_{2,i}^2(T) - 4c_1(T)c_{3,i}(T)}} + (p-r)\frac{n}{2\lambda} && (69) \\
&= \sum_{k=1}^{r} \frac{4\lambda n^2 T}{d_i^4(p-n) + 8n\lambda^2 T + \sqrt{d_i^8(p-n)^2 + 16\lambda^2 npd_i^4 T}} + (p-r)\frac{n}{2\lambda}. && (70)
\end{aligned}$$

Let $f(T)$ denote the right hand side of the equation in (70). It can be shown that $f'(T) > 0$ for all $T \geq 0$. Also, when $T = 0$, we have that $f(T) = (p-r)\frac{n}{2\lambda} > 0 = T$, and as $T \to \infty$, we have $f(T) \to \frac{np}{2\lambda} < \infty$. Together, these observations imply that there exists a unique solution to the equation $T = f(T)$ for some $T > 0$. Thus, there exists a unique feasible T, and hence also $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, to our system of equations (62)-(66). ■

As a direct consequence, since the PC scores and loadings from iPCA are the eigenvectors of $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Delta}}$, respectively, Theorem 24 shows that when $K = 1$, the PC scores and loadings from iPCA with the multiplicative Frobenius penalty are precisely $\mathbf{U}$ and $\mathbf{V}$, as in PCA. In other words, iPCA with the additive or multiplicative Frobenius penalty is a proper generalization of PCA to multiple data sets.

Appendix F. Subspace Consistency of the Additive L_1 Correlation iPCA Estimator

While the multiplicative Frobenius estimator appears superior from an optimization point-of-view, we will show in this section that the additive L_1 correlation estimator satisfies one of the first statistical guarantees in the data integration context. Specifically, our primary objective will be to prove that the additive L_1 correlation estimator after one Flip-Flop iteration, outlined in Algorithm 6, is a consistent estimator of the true joint covariance matrix $\boldsymbol{\Sigma}$ and hence also consistently estimates the underlying joint subspace.

As mentioned in the main text, the additive L_1 correlation estimator applies the L_1 penalty to the correlation matrix, rather than the usual covariance matrix, and has been adopted previously in Zhou (2014a) and Rothman et al. (2008) for non-integrated data. In this non-integrated setting, Zhou (2014a) derived convergence rates for the one-step version of Algorithm 6, assuming only one matrix instance was observed from the matrix-variate normal distribution. Motivated by this approach, we extend the proof idea and results in Zhou (2014a), where $K = 1$, to iPCA, where we observe one matrix instance for each of the $K \geq 1$ distinct matrix-variate normal models. Our proof closely resembles Zhou (2014a) with technical challenges due to the estimation of $\mathbf{\Sigma}$ from multiple $\mathbf{\Delta}_k$'s with different p_k's. We will see later that the consistency rate obtained in our main theorem, Theorem 30, which holds for all $K \geq 1$, is equivalent to the rate given in Zhou (2014a) when $K = 1$.

Note however that this subspace consistency property is an incredibly unique property of the additive L_1 correlation estimator. When trying to prove a similar result for the other proposed estimators, we run into several difficulties. For instance, if we use the additive L_1 covariance estimator, the proof of our main result in Theorem 30 no longer goes through due to an additional $\sqrt{p}$ term in the graphical lasso convergence rate (Rothman et al., 2008). More specifically, the graphical lasso convergence rate when applied to the inverse covariance matrix, known to be $O\left(\sqrt{\frac{(p+s)\log(p)}{n}}\right)$, is not fast enough to guarantee statistical convergence of the Flip-Flop algorithm. However, by applying the graphical lasso to the inverse correlation matrix, we obtain a faster rate of $O\left(\sqrt{\frac{s\log(p)}{n}}\right)$, which then enables us to prove Theorem 30. Other works, which have studied convergence rates of sparse penalties in the non-integrated setting, are also not applicable for integrated data problems. In particular, Tsiligkaridis et al. (2013) proved convergence rates for the additive L_1 covariance penalty but assumed multiple matrix observations per matrix-variate normal model. Since iPCA assumes only one matrix observation per model, the guarantees from Tsiligkaridis et al. (2013) do not hold.

The problem of proving rates of convergence for the Frobenius estimators is even more difficult than the L_1 estimators. Without imposing some additional structure on the covariance matrices, we cannot even hope to prove statistical consistency in the $p_k > n$ setting. For the L_1 penalties, it is natural to impose a sparsity constraint, but with the dense Frobenius penalties, the appropriate underlying structure of the covariance matrices is unclear. One preliminary idea is to exploit the g-convexity of the log-likelihood function and impose some additional structure based upon the associated manifold. However, we leave this for future research as it requires developing a whole new set of tools, combined with Riemannian geometry, to study statistical properties in the manifold space, rather than the usual Euclidean space.

We next collect notation. Let $\hat{\mathbf{\Sigma}}$ denote the additive L_1 correlation estimator obtained after one iteration of the while loop in Algorithm 6. Assume that for each $k = 1, \ldots, K$, the true population model is given by $\mathbf{X}_k \sim N_{n,p_k}(\mathbf{0},\ \mathbf{\Sigma} \otimes \mathbf{\Delta}_k)$, where $\mathbf{\Sigma} = (\sigma_{ij})$ and $\mathbf{\Delta}_k = (\delta_{k,ij})$. For the purpose of identifiability, define $\mathbf{\Sigma}^* = (\sigma_{*,ij}) = n\,\mathbf{\Sigma}\,/\mathrm{tr}(\mathbf{\Sigma})$ and $\mathbf{\Delta}_k^* = (\delta^*_{k,ij}) = \mathrm{tr}(\mathbf{\Sigma})\,\mathbf{\Delta}_k\ n$ so that $\mathrm{tr}(\mathbf{\Sigma}^*) = n$ and $\mathbf{\Sigma}^* \otimes \mathbf{\Delta}_k^* = \mathbf{\Sigma} \otimes \mathbf{\Delta}_k$ for each k. Let $\rho(\mathbf{\Sigma})$ and $\rho(\mathbf{\Delta}_k)$ denote the true correlation matrices corresponding to $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$, respectively. Define $\mathrm{vec}(\mathbf{A})$ to be the vectorization operator which creates a column vector from the

Algorithm 6 Flip-Flop Algorithm for Additive L_1 Correlation-Penalized iPCA Estimators

```
Center the columns of $\mathbf{X}_1, \dots, \mathbf{X}_K$, and initialize $\hat{\boldsymbol{\Sigma}}$, $\hat{\boldsymbol{\Delta}}_1, \dots, \hat{\boldsymbol{\Delta}}_K$ to be the identity matrix of the appropriate size.
while not converged do
    for $k = 1, \dots, K$ do
        Compute sample covariance matrix: $\hat{\mathbf{S}}_k = \frac{1}{n} \mathbf{X}_k^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X}_k$
        Get standard deviation estimate: $\hat{\mathbf{W}}_k = \text{diag}(\hat{\mathbf{S}}_k)^{1/2}$
        Convert to sample correlation matrix: $\hat{\mathbf{S}}_{\rho,k} = \hat{\mathbf{W}}_k^{-1} \hat{\mathbf{S}}_k \hat{\mathbf{W}}_k^{-1}$
        Apply graphical lasso to estimate correlation matrix:                          Update $\boldsymbol{\Delta}_k$
            $\hat{\boldsymbol{\Delta}}_{\rho,k}^{-1} = \underset{\boldsymbol{\Delta}_{\rho,k}^{-1}}{\text{argmin}} \; -\log|\, \boldsymbol{\Delta}_{\rho,k}^{-1} \,| + \text{tr}(\hat{\mathbf{S}}_{\rho,k} \, \boldsymbol{\Delta}_{\rho,k}^{-1}) + \lambda_k \| \, \boldsymbol{\Delta}_{\rho,k}^{-1} \, \|_{1,\text{off}}$
        Convert back to covariance estimate: $\hat{\boldsymbol{\Delta}}_k = \hat{\mathbf{W}}_k \hat{\boldsymbol{\Delta}}_{\rho,k} \hat{\mathbf{W}}_k$
    Compute sample covariance matrix: $\hat{\mathbf{S}}_{\boldsymbol{\Sigma}} = \frac{1}{p} \sum_{k=1}^K \mathbf{X}_k \, \hat{\boldsymbol{\Delta}}_k^{-1} \, \mathbf{X}_k^T$
    Get standard deviation estimate: $\hat{\mathbf{W}}_{\boldsymbol{\Sigma}} = \text{diag}(\hat{\mathbf{S}}_{\boldsymbol{\Sigma}})^{1/2}$
    Convert to sample correlation matrix: $\hat{\mathbf{S}}_{\rho,\boldsymbol{\Sigma}} = \hat{\mathbf{W}}_{\boldsymbol{\Sigma}}^{-1} \hat{\mathbf{S}}_{\boldsymbol{\Sigma}} \hat{\mathbf{W}}_{\boldsymbol{\Sigma}}^{-1}$
    Apply graphical lasso to estimate correlation matrix:                              Update $\boldsymbol{\Sigma}$
        $\hat{\boldsymbol{\Sigma}}_{\rho}^{-1} = \underset{\boldsymbol{\Sigma}_{\rho}^{-1}}{\text{argmin}} \; -\log|\, \boldsymbol{\Sigma}_{\rho}^{-1} \,| + \text{tr}(\hat{\mathbf{S}}_{\rho,\boldsymbol{\Sigma}} \, \boldsymbol{\Sigma}_{\rho}^{-1}) + \lambda_{\boldsymbol{\Sigma}} \| \, \boldsymbol{\Sigma}_{\rho}^{-1} \, \|_{1,\text{off}}$
    Convert back to covariance estimate: $\hat{\boldsymbol{\Sigma}} = \hat{\mathbf{W}}_{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Sigma}}_{\rho} \hat{\mathbf{W}}_{\boldsymbol{\Sigma}}$
```

matrix $\mathbf{A}$ by stacking the columns of $\mathbf{A}$ below one another. Then for each $k = 1, \dots, K$, put $\tilde{\mathbf{S}}_k = \text{vec}(\mathbf{X}_k)\text{vec}(\mathbf{X}_k)^T$ and $\bar{\mathbf{S}}_k = \text{vec}(\mathbf{X}_k)^T\text{vec}(\mathbf{X}_k)$. Let $\tilde{\mathbf{S}}_k^{rq}$ denote the r, q^{th} block of size $n \times n$ of $\tilde{\mathbf{S}}_k$, and let $\bar{\mathbf{S}}_k^{rq}$ denote the r, q^{th} block of size $p_k \times p_k$ of $\bar{\mathbf{S}}_k$. If $\mathbf{A}$ is a matrix, let $\| \, \mathbf{A} \, \|_2$ denote the operator norm or the maximum singular value of $\mathbf{A}$, let $\| \, \mathbf{A} \, \|_F$ denote the Frobenius norm (i.e. $\| \, \mathbf{A} \, \|_F^2 = \sum_{i,j} a_{ij}^2$), let $\| \, \mathbf{A} \, \|_{0,\text{off}}$ denote the number of non-zero non-diagonal entries in $\mathbf{A}$, let $\| \, \mathbf{A} \, \|_1 = \sum_{i,j} |a_{ij}|$, and let $\| \, \mathbf{A} \, \|_{1,\text{off}} = \sum_{i \neq j} |a_{ij}|$. Denote the stable rank of $\mathbf{A}$ by $r(\mathbf{A}) = \| \, \mathbf{A} \, \|_F^2 / \| \, \mathbf{A} \, \|_2^2$. Let us also write for a real symmetric matrix $\mathbf{A}$, $\phi_{\min}(\mathbf{A})$ to be the minimum eigenvalue of $\mathbf{A}$. Define $\sigma_{\min} = \min_i \sigma_{ii}$, $\sigma_{\max} = \max_i \sigma_{ii}$, $\delta_{k,\min} = \min_i \delta_{k,ii}$, $\delta_{k,\max} = \max_i \delta_{k,ii}$, and similarly for $\sigma_{*,\min}$, $\sigma_{*,\max}$, $\delta_{k,\min}^*$, and $\delta_{k,\max}^*$. Also write $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$. If $a = o(b)$, then $|a/b| \to 0$ as $n, p_1, \dots, p_K \to \infty$. If $a \asymp b$, then there exists positive constants c, C such that $cb \leq a \leq Cb$ as $n, p_1, \dots, p_K \to \infty$. Lastly, we also adopt the notation defined in Algorithm 6 for the remainder of this section.

To establish the consistency of $\hat{\boldsymbol{\Sigma}}$, we require the following assumptions, which are generalizations of those in (Zhou, 2014a):

(A1) Assume that $\boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\Delta}_k^{-1}$ are sparse with respect to each other's dimensions: $s_{\boldsymbol{\Sigma}} := \| \, \boldsymbol{\Sigma}^{-1} \, \|_{0,\text{off}} = o\left(\frac{p^2}{p_k \log(n \vee p_k)} \right)$ and $s_k := \| \, \boldsymbol{\Delta}_k^{-1} \, \|_{0,\text{off}} = o\left(\frac{n}{\log(n \vee p_k)} \right)$ for each $k = 1, \dots, K$.

(A2) Assume that we have uniformly bounded spectra: $0 < \phi_{\min}(\boldsymbol{\Sigma}) \leq \phi_{\max}(\boldsymbol{\Sigma}) < \infty$ and $0 < \phi_{\min}(\boldsymbol{\Delta}_k) \leq \phi_{\max}(\boldsymbol{\Delta}_k) < \infty$ for each $k = 1, \dots, K$.

(A3) Assume that the inverse correlation matrices satisfy $\| \rho(\mathbf{\Sigma})^{-1} \|_1 \asymp n$ and $\| \rho(\mathbf{\Delta}_k)^{-1} \|_1 \asymp p_k$ for each $k = 1, \ldots, K$.

(A4) Assume that K is finite and the growth rate of n and $p_1, \ldots, p_K$ satisfy $n \vee p_k = o(\exp(n \wedge p_k))$ for each $k = 1, \ldots, K$.

Remark 25 *(1) Rather than verifying the sparsity assumption of $\mathbf{\Sigma}^{-1}$ in (A1) and the growth rate (A4) for each k, it is sufficient to check that $s_{\mathbf{\Sigma}} = o\left(\frac{p^2}{\max_k p_k \log(n \vee \max_k p_k)}\right)$ and $n \vee \max_k p_k = o(\exp(n \wedge \min_k p_k))$, respectively.*

(2) (A1) implies that $\sqrt{\frac{s_k \log(n \vee p_k)}{n}} \to 0$ and $\sqrt{\frac{s_{\mathbf{\Sigma}} \log(n \vee p_k)}{p_k}} \to 0$ as $n, p_1, \ldots, p_K \to \infty$.

Under these assumptions, we prove our main statistical consistency result in Theorem 30. From this, we can easily establish subspace consistency for $\hat{\mathbf{\Sigma}}$ in Corollary 31. The overall idea of this convergence proof is to follow Algorithm 6 and sequentially bound each step in the algorithm. First, the error from line 8 of Algorithm 6 can be bounded by adapting results from Rothman et al. (2008). Then, in Lemma 28, we bound the error between $\mathbf{\Sigma}^*$ and the sample covariance estimate $\hat{\mathbf{S}}_{\mathbf{\Sigma}}$ defined in the step 9. Following the Flip-Flop algorithm, we next bound the error between $\rho(\mathbf{\Sigma})$ and the sample correlation estimate $\hat{\mathbf{S}}_{\rho,\mathbf{\Sigma}}$ from step 11 in Theorem 29. Finally, in Theorem 30, we prove the rate of convergence in the operator and Frobenius norms for $\hat{\mathbf{\Sigma}}^{-1}$ and $\hat{\mathbf{\Sigma}}$ as defined in step 13 of the algorithm. Two direct consequences of convergence in the operator norm are consistent eigenvalues and eigenvectors of $\hat{\mathbf{\Sigma}}$, which in turn yield subspace consistency.

F.1 Preliminaries

The main driver behind Lemma 28 is a large deviation inequality, namely Theorem 13.1 from Zhou (2014b). As this is an important result used multiple times in the proof of Lemma 28, we state Theorem 13.1 from Zhou (2014b) below in a slightly different form for convenience.

Theorem 26 *Assume that $n \vee p_k \geq 2$. Let $\mathbf{M}$ be an $n \times n$ matrix and $\mathbf{N}$ be a $p_k \times p_k$ matrix such that $\frac{1}{n} \| \mathbf{M} \|_F^2 < \infty$ and $\frac{1}{p_k} \| \mathbf{N} \|_F^2 < \infty$. Define $\tau_k = 2C\tilde{K}^2 \log^{1/2}(n \vee p_k)$, where $C := \frac{1}{\sqrt{c}} \vee \frac{1}{c} \vee 1$ and $\tilde{K}$ and c are the constants from Theorem 12.1 in Zhou (2014b).*

(i) If the stable ranks satisfy $r(\mathbf{\Sigma}^{1/2} \mathbf{M} \mathbf{\Sigma}^{1/2}) \geq 4 \log(n \vee p_k)$ and $r(\mathbf{\Delta}_k^{1/2} \mathbf{N} \mathbf{\Delta}_k^{1/2}) \geq 4 \log(n \vee p_k)$, then with probability $1 - \frac{3}{(n \vee p_k)^2}$, we have that

$$\| \operatorname{diag}(\mathbf{\Delta}_k)^{-1/2} \left(\frac{1}{n} \sum_{q=1}^{n} \sum_{r=1}^{n} \mathbf{M}_{qr} \, \bar{\mathbf{S}}_k^{rq} \right) \operatorname{diag}(\mathbf{\Delta}_k)^{-1/2} - \frac{\operatorname{tr}(\mathbf{\Sigma} \, \mathbf{M})}{n} \rho(\mathbf{\Delta}_k) \|_\infty \leq D_k \tau_k$$

$$\textit{and } \| \operatorname{diag}(\mathbf{\Sigma})^{-1/2} \left(\frac{1}{p} \sum_{q=1}^{p_k} \sum_{r=1}^{p_k} \mathbf{N}_{qr} \, \tilde{\mathbf{S}}_k^{rq} \right) \operatorname{diag}(\mathbf{\Sigma})^{-1/2} - \frac{\operatorname{tr}(\mathbf{\Delta}_k \, \mathbf{N})}{p} \rho(\mathbf{\Sigma}) \|_\infty \leq D_k' \tau_k,$$

where $D_k = \frac{1}{n} \| \mathbf{\Sigma}^{1/2} \mathbf{M} \mathbf{\Sigma}^{1/2} \|_F$ and $D_k' = \frac{1}{p} \| \mathbf{\Delta}_k^{1/2} \mathbf{N} \mathbf{\Delta}_k^{1/2} \|_F$.

(ii) If $n \vee p_k = o(\exp n \wedge p_k)$, then with probability $1 - \frac{3}{(n \vee p_k)^2}$, the above inequalities hold with $D_k = \frac{2}{\sqrt{n}} \| \mathbf{\Sigma} \|_2 \| \mathbf{M} \|_2$ and $D'_k = \frac{2\sqrt{p_k}}{p} \| \mathbf{\Delta}_k \|_2 \| \mathbf{N} \|_2$.

We refer to Zhou (2014b) for the proof of this theorem, but if one looks at the interior of this proof, we obtain the following useful result, presented in Corollary 27.

To simplify notation, let $\mathcal{E}_{\mathbf{\Delta}}(k, \mathbf{M})$ denote the event

$$\left\{ \| \operatorname{diag}(\mathbf{\Delta}_k)^{-1/2} \left(\frac{1}{n} \sum_{q=1}^{n} \sum_{r=1}^{n} \mathbf{M}_{qr} \, \bar{\mathbf{S}}_k^{rq} \right) \operatorname{diag}(\mathbf{\Delta}_k)^{-1/2} - \frac{\operatorname{tr}(\mathbf{\Sigma}\,\mathbf{M})}{n} \rho(\mathbf{\Delta}_k) \|_\infty \leq \frac{2}{\sqrt{n}} \| \mathbf{\Sigma} \|_2 \| \mathbf{M} \|_2 \tau_k \right\},$$

let $\mathcal{E}_{\mathbf{\Sigma}}(k, \mathbf{N})$ denote the event

$$\left\{ \| \operatorname{diag}(\mathbf{\Sigma})^{-1/2} \left(\frac{1}{p} \sum_{q=1}^{p_k} \sum_{r=1}^{p_k} \mathbf{N}_{qr} \, \tilde{\mathbf{S}}_k^{rq} \right) \operatorname{diag}(\mathbf{\Sigma})^{-1/2} - \frac{\operatorname{tr}(\mathbf{\Delta}_k \, \mathbf{N})}{p} \rho(\mathbf{\Sigma}) \|_\infty \leq \frac{2\sqrt{p_k}}{p} \| \mathbf{\Delta}_k \|_2 \| \mathbf{N} \|_2 \tau_k \right\},$$

and define for each $k = 1, \ldots, K$,

$$\begin{aligned} \nu_{n,k} &:= \frac{2}{\sqrt{n}} \, \tau_k = 4C\tilde{K}^2 \sqrt{\frac{\log(n \vee p_k)}{n}} \\ \nu_{p_k} &:= \frac{2\sqrt{p_k}}{p} \, \tau_k = 4C\tilde{K}^2 \frac{\sqrt{p_k \log(n \vee p_k)}}{p} = 4C\tilde{K}^2 \frac{p_k}{p} \sqrt{\frac{\log(n \vee p_k)}{p_k}}. \end{aligned}$$

Corollary 27 *Assume the same conditions and notation as Theorem 26. Also suppose that $n \vee p_k = o(\exp n \wedge p_k)$ Then under the event $\mathcal{E}_{\mathbf{\Delta}}(k, \mathbf{M})$, we have that*

$$\left| \left(\frac{1}{n} \sum_{q=1}^{n} \sum_{r=1}^{n} \mathbf{M}_{qr} \, \bar{\mathbf{S}}_k^{rq} \right)_{ij} - \frac{\operatorname{tr}(\mathbf{\Sigma}\,\mathbf{M})}{n} \delta_{k,ij} \right| \leq \nu_{n,k} \| \mathbf{\Sigma} \|_2 \| \mathbf{M} \|_2 \sqrt{\delta_{k,ii} \delta_{k,jj}} \qquad \forall\, i, j,$$

and under the event $\mathcal{E}_{\mathbf{\Sigma}}(k, \mathbf{N})$, we have that

$$\left| \left(\frac{1}{p} \sum_{q=1}^{p_k} \sum_{r=1}^{p_k} \mathbf{N}_{qr} \, \tilde{\mathbf{S}}_k^{rq} \right)_{ij} - \frac{\operatorname{tr}(\mathbf{\Delta}_k \, \mathbf{N})}{p} \sigma_{ij} \right| \leq \nu_{p_k} \| \mathbf{\Delta}_k \|_2 \| \mathbf{N} \|_2 \sqrt{\sigma_{ii} \sigma_{jj}} \qquad \forall\, i, j.$$

We now have the necessary large deviation inequalities to prove Lemma 28, a generalization of Lemma 6.1 from Zhou (2014a). Though the proof of Lemma 28 closely resembles that of Lemma 6.1 from Zhou (2014a), modifications must be made as iPCA considers multiple distinct matrix-variate normal models while Zhou (2014a) considers only one matrix-variate normal model. For clarity, we give our proof in its entirety and refer to results in Zhou (2014a) when necessary.

Lemma 28 *Suppose that (A1)-(A4) hold. Let $\hat{\mathbf{\Delta}}_{\rho,k}$ and $\hat{\mathbf{\Delta}}_k$ be obtained as in steps 7 and 8 in Algorithm 6, where we choose*

$$\lambda_k = \frac{2\alpha_k}{\epsilon(1 - \alpha_k)} \geq \frac{3\alpha_k}{1 - \alpha_k} \quad \text{for } \ \alpha_k = A \, \nu_{n,k} \text{ where } A = \frac{\sqrt{n} \| \mathbf{\Sigma} \|_F}{\operatorname{tr}(\mathbf{\Sigma})} \tag{71}$$

and $\epsilon \in (0, 2/3)$. *Then on event* $\mathcal{E}^*$, *we have for* $\hat{\mathbf{S}}_{\mathbf{\Sigma}}$ *defined in step 9 of Algorithm 6,*

$$\begin{aligned}\left|(\hat{\mathbf{S}}_{\mathbf{\Sigma}} - \mathbf{\Sigma}^*)_{ij}\right| &\leq \sum_{k=1}^{K} \frac{p_k}{p}\left[4C\tilde{K}^2\sqrt{\sigma_{*,ii}\sigma_{*,jj}}\sqrt{\frac{\log(n \vee p_k)}{p_k}}(1+o(1))\right] + \sum_{k=1}^{K} |\sigma_{*,ij}|\tilde{\mu}_k \\ &= \sum_{k=1}^{K}\left[\sqrt{\sigma_{*,ii}\sigma_{*,jj}}\,\nu_{p_k}(1+o(1)) + |\sigma_{*,ij}|\tilde{\mu}_k\right],\end{aligned}$$

where

$$\begin{aligned}\tilde{\mu}_k &:= \lambda_k \frac{\|\,\hat{\mathbf{\Delta}}_{\rho,k}^{-1}\,\|_{1,\mathit{off}}}{p} + \frac{\alpha_k}{1-\alpha_k}\frac{\|\,\hat{\mathbf{\Delta}}_{\rho,k}^{-1}\,\|_1}{p} \leq \mu_k, \\ \mu_k &:= \lambda_k \frac{\|\,\rho(\mathbf{\Delta}_k)^{-1}\,\|_{1,\mathit{off}}}{p} + \frac{\alpha_k}{1-\alpha_k}\frac{\|\,\rho(\mathbf{\Delta}_k)^{-1}\,\|_1}{p} + o(\lambda_k).\end{aligned}$$

Moreover, $\mathbb{P}(\mathcal{E}^*) \geq 1 - \sum_{k=1}^{K}\frac{8}{(n \vee p_k)^2}$.
To put simply, with high probability,

$$\begin{aligned}\|\,\hat{\mathbf{S}}_{\mathbf{\Sigma}} - \mathbf{\Sigma}^*\,\|_\infty &\leq \sum_{k=1}^{K}\frac{p_k}{p}\left[4C\tilde{K}^2\sigma_{*,\max}\sqrt{\frac{\log(n \vee p_k)}{p_k}}(1+o(1))\right] + \sum_{k=1}^{K}\sigma_{*,\max}\mu_k \\ &= \sigma_{*,\max}\sum_{k=1}^{K}\left[\nu_{p_k}(1+o(1)) + \mu_k\right].\end{aligned}$$

Proof

For each $k = 1, \dots, K$, define

$$\begin{aligned}\mathbf{R}_{\mathbf{\Sigma},k} &:= [\delta_{k,11}\mathrm{vec}(\mathbf{\Sigma}) \ \dots \ \delta_{k,1p_k}\mathrm{vec}(\mathbf{\Sigma}) \ \dots \ \delta_{k,p_kp_k}\mathrm{vec}(\mathbf{\Sigma})] \equiv \mathrm{vec}(\mathbf{\Sigma}) \otimes \mathrm{vec}(\mathbf{\Delta}_k)^T \\ \hat{\mathbf{R}}_{\mathbf{\Sigma},k} &:= \left[\mathrm{vec}(\tilde{\mathbf{S}}_k^{11}) \ \dots \ \mathrm{vec}(\tilde{\mathbf{S}}_k^{1p_k}) \ \dots \ \mathrm{vec}(\tilde{\mathbf{S}}_k^{p_kp_k})\right].\end{aligned}$$

Then one can verify as in Zhou (2014a) that

$$\mathrm{vec}(\mathbf{\Sigma}^*) = \frac{1}{p}\sum_{k=1}^{K}\mathbf{R}_{\mathbf{\Sigma},k}\,\mathrm{vec}((\mathbf{\Delta}_k^*)^{-1}) \quad \text{and} \quad \mathrm{vec}(\hat{\mathbf{S}}_{\mathbf{\Sigma}}) = \frac{1}{p}\sum_{k=1}^{K}\hat{\mathbf{R}}_{\mathbf{\Sigma},k}\mathrm{vec}(\hat{\mathbf{\Delta}}_k^{-1}). \tag{72}$$

The equalities from (72) thus yield

$$\begin{aligned}\mathrm{vec}(\hat{\mathbf{S}}_{\mathbf{\Sigma}} - \mathbf{\Sigma}^*) &= \frac{1}{p}\sum_{k=1}^{K}\hat{\mathbf{R}}_{\mathbf{\Sigma},k}\mathrm{vec}(\hat{\mathbf{\Delta}}_k^{-1}) - \frac{1}{p}\sum_{k=1}^{K}\mathbf{R}_{\mathbf{\Sigma},k}\,\mathrm{vec}((\mathbf{\Delta}_k^*)^{-1}) && (73) \\ &= \frac{1}{p}\sum_{k=1}^{K}\hat{\mathbf{R}}_{\mathbf{\Sigma},k}\mathrm{vec}(\hat{\mathbf{\Delta}}_k^{-1}) - \frac{1}{p}\sum_{k=1}^{K}\mathbf{R}_{\mathbf{\Sigma},k}\,\mathrm{vec}((\mathbf{\Delta}_k^*)^{-1}) \\ &\quad + \frac{1}{p}\sum_{k=1}^{K}\hat{\mathbf{R}}_{\mathbf{\Sigma},k}\mathrm{vec}((\mathbf{\Delta}_k^*)^{-1}) - \frac{1}{p}\sum_{k=1}^{K}\hat{\mathbf{R}}_{\mathbf{\Sigma},k}\mathrm{vec}((\mathbf{\Delta}_k^*)^{-1}) && (74) \\ &= \frac{1}{p}\sum_{k=1}^{K}\left[(\hat{\mathbf{R}}_{\mathbf{\Sigma},k} - \mathbf{R}_{\mathbf{\Sigma},k})\mathrm{vec}((\mathbf{\Delta}_k^*)^{-1}) + \hat{\mathbf{R}}_{\mathbf{\Sigma},k}\mathrm{vec}(\hat{\mathbf{\Delta}}_k^{-1} - (\mathbf{\Delta}_k^*)^{-1})\right]. && (75)\end{aligned}$$

For each $k = 1, \dots, K$, define $\mathbf{\Theta}_k := \hat{\mathbf{\Delta}}_k - \mathbf{\Delta}_k^*$ and $\tilde{\mathbf{\Theta}}_k := \hat{\mathbf{\Delta}}_k^{-1} - (\mathbf{\Delta}_k^*)^{-1}$, and notice that $\tilde{\mathbf{\Theta}}_k = -\hat{\mathbf{\Delta}}_k^{-1}\mathbf{\Theta}_k(\mathbf{\Delta}_k^*)^{-1}$. Then

$$\begin{aligned}
\hat{\mathbf{R}}_{\mathbf{\Sigma},k}\mathrm{vec}(\tilde{\mathbf{\Theta}}_k) &= \hat{\mathbf{R}}_{\mathbf{\Sigma},k}\mathrm{vec}(\tilde{\mathbf{\Theta}}_k) + \mathbf{R}_{\mathbf{\Sigma},k}\,\mathrm{vec}(\tilde{\mathbf{\Theta}}_k) - \mathbf{R}_{\mathbf{\Sigma},k}\,\mathrm{vec}(\tilde{\mathbf{\Theta}}_k) && (76)\\
&= \mathbf{R}_{\mathbf{\Sigma},k}\,\mathrm{vec}(\tilde{\mathbf{\Theta}}_k) + (\hat{\mathbf{R}}_{\mathbf{\Sigma},k} - \mathbf{R}_{\mathbf{\Sigma},k})\mathrm{vec}(\tilde{\mathbf{\Theta}}_k) && (77)\\
&= \mathbf{R}_{\mathbf{\Sigma},k}\,\mathrm{vec}(-\hat{\mathbf{\Delta}}_k^{-1}\mathbf{\Theta}_k(\mathbf{\Delta}_k^*)^{-1}) + (\hat{\mathbf{R}}_{\mathbf{\Sigma},k} - \mathbf{R}_{\mathbf{\Sigma},k})\mathrm{vec}(-\hat{\mathbf{\Delta}}_k^{-1}\mathbf{\Theta}_k(\mathbf{\Delta}_k^*)^{-1}). && (78)
\end{aligned}$$

Putting (75) and (78) together gives

$$\begin{aligned}
\mathrm{vec}(\hat{\mathbf{S}}_{\mathbf{\Sigma}} - \mathbf{\Sigma}^*) = \sum_{k=1}^{K} \Big[&\frac{1}{p}(\hat{\mathbf{R}}_{\mathbf{\Sigma},k} - \mathbf{R}_{\mathbf{\Sigma},k})\mathrm{vec}((\mathbf{\Delta}_k^*)^{-1}) + \frac{1}{p}\,\mathbf{R}_{\mathbf{\Sigma},k}\,\mathrm{vec}(-\hat{\mathbf{\Delta}}_k^{-1}\mathbf{\Theta}_k(\mathbf{\Delta}_k^*)^{-1})\\
&+ \frac{1}{p}(\hat{\mathbf{R}}_{\mathbf{\Sigma},k} - \mathbf{R}_{\mathbf{\Sigma},k})\mathrm{vec}(-\hat{\mathbf{\Delta}}_k^{-1}\mathbf{\Theta}_k(\mathbf{\Delta}_k^*)^{-1})\Big]\\
=: \sum_{k=1}^{K} &(\mathbf{U}_{1,k} + \mathbf{U}_{2,k} + \mathbf{U}_{3,k}),
\end{aligned}$$

where the matrix correspondent for each of the above terms will be denoted by $\mathbf{M}_{1,k}$, $\mathbf{M}_{2,k}$, and $\mathbf{M}_{3,k}$, respectively. We will proceed to bound each of the terms separately.

In order to bound $\mathbf{U}_{1,k}$, notice that by definition of $\mathbf{R}_{\mathbf{\Sigma},k}$ and $\hat{\mathbf{R}}_{\mathbf{\Sigma},k}$,

$$\begin{aligned}
\mathbf{U}_{1,k} &= \frac{1}{p}(\hat{\mathbf{R}}_{\mathbf{\Sigma},k} - \mathbf{R}_{\mathbf{\Sigma},k})\mathrm{vec}((\mathbf{\Delta}_k^*)^{-1}) && (79)\\
&= \frac{1}{p}\sum_{q=1}^{p_k}\sum_{r=1}^{p_k}\mathrm{vec}(\tilde{\mathbf{S}}_k^{qr} - \delta_{k,qr}\,\mathbf{\Sigma})[(\mathbf{\Delta}_k^*)^{-1}]_{qr} && (80)\\
&= \frac{1}{p}\sum_{q=1}^{p_k}\sum_{r=1}^{p_k}\mathrm{vec}(\tilde{\mathbf{S}}_k^{qr})[(\mathbf{\Delta}_k^*)^{-1}]_{qr} - \frac{\mathrm{tr}(\mathbf{\Delta}_k(\mathbf{\Delta}_k^*)^{-1})}{p}\mathrm{vec}(\mathbf{\Sigma}) && (81)\\
&= \frac{1}{p}\sum_{q=1}^{p_k}\sum_{r=1}^{p_k}\mathrm{vec}(\tilde{\mathbf{S}}_k^{qr})[(\mathbf{\Delta}_k^*)^{-1}]_{qr} - \frac{p_k}{p}\mathrm{vec}(\mathbf{\Sigma}^*) && (82)\\
\implies \mathbf{M}_{1,k} &= \frac{1}{p}\sum_{q=1}^{p_k}\sum_{r=1}^{p_k}\tilde{\mathbf{S}}_k^{qr}[(\mathbf{\Delta}_k^*)^{-1}]_{qr} - \frac{p_k}{p}\,\mathbf{\Sigma}^*\,. && (83)
\end{aligned}$$

Define $\mathcal{E}_{0,k}$ to be the event $\mathcal{E}_{\mathbf{\Delta}}(k, (\mathbf{\Sigma}^*)^{-1}) \cap \mathcal{E}_{\mathbf{\Sigma}}(k, (\mathbf{\Delta}_k^*)^{-1})$. Then by Corollary 27, under the event $\mathcal{E}_{0,k}$, we have $|(\mathbf{M}_{1,k})_{ij}| \le \nu_{p_k}\sqrt{\sigma_{*,ii}\sigma_{*,jj}}$. Moreover, by Theorem 26, $\mathbb{P}(\mathcal{E}_{0,k}) \ge 1 - \frac{3}{(n \vee p_k)^2}$.

Next, we will bound the second term $\mathbf{U}_{2,k}$. As in Zhou (2014b), we can write

$$\begin{aligned}
\mathbf{U}_{2,k} &= \frac{1}{p}\,\mathbf{R}_{\mathbf{\Sigma},k}\,\mathrm{vec}(-\hat{\mathbf{\Delta}}_k^{-1}\mathbf{\Theta}_k(\mathbf{\Delta}_k^*)^{-1}) = \frac{1}{p}\mathrm{tr}(-\hat{\mathbf{\Delta}}_k^{-1}\mathbf{\Theta}_k)\mathrm{vec}(\mathbf{\Sigma}^*),\\
\text{and }\ \mathbf{M}_{2,k} &= \frac{1}{p}\mathrm{tr}(-\hat{\mathbf{\Delta}}_k^{-1}\mathbf{\Theta}_k)\,\mathbf{\Sigma}^*\,.
\end{aligned}$$

By Claim 17.3 in Zhou (2014b), it follows that that under event $\mathcal{X}_{0,k} := \mathcal{E}_{\mathbf{\Sigma}}(k, \mathbf{I}) \cap \mathcal{E}_{\mathbf{\Delta}}(k, \mathbf{I})$,

$$\lambda_k \| \hat{\mathbf{\Delta}}_{\rho,k}^{-1} \|_{1,\text{off}} - \frac{\alpha_k}{1-\alpha_k} \| \hat{\mathbf{\Delta}}_{\rho,k}^{-1} \|_1 \leq \text{tr}(\mathbf{\Theta}_k \hat{\mathbf{\Delta}}_k^{-1}) \leq \lambda_k \| \hat{\mathbf{\Delta}}_{\rho,k}^{-1} \|_{1,\text{off}} + \frac{\alpha_k}{1-\alpha_k} \| \hat{\mathbf{\Delta}}_{\rho,k}^{-1} \|_1$$

Thus,

$$|\text{tr}(-\mathbf{\Theta}_k \hat{\mathbf{\Delta}}_k^{-1})| \leq \lambda_k \| \hat{\mathbf{\Delta}}_{\rho,k}^{-1} \|_{1,\text{off}} + \frac{\alpha_k}{1-\alpha_k} \| \hat{\mathbf{\Delta}}_{\rho,k}^{-1} \|_1,$$

which implies that on $\mathcal{X}_{0,k}$,

$$|(\mathbf{M}_{2,k})_{ij}| \leq \frac{|\sigma_{*,ij}|}{p} \left(\lambda_k \| \hat{\mathbf{\Delta}}_{\rho,k}^{-1} \|_{1,\text{off}} + \frac{\alpha_k}{1-\alpha_k} \| \hat{\mathbf{\Delta}}_{\rho,k}^{-1} \|_1 \right) = |\sigma_{*,ij}| \tilde{\mu}_k.$$

Additionally, by Theorem 26, $\mathbb{P}(\mathcal{X}_{0,k}) \geq 1 - \frac{3}{(n \vee p_k)^2}$.

To bound the final term $\mathbf{U}_{3,k}$, we follow the same logic as (79)-(82) to obtain

$$\begin{aligned} \mathbf{U}_{3,k} &= \frac{1}{p} (\hat{\mathbf{R}}_{\mathbf{\Sigma},k} - \mathbf{R}_{\mathbf{\Sigma},k}) \text{vec}(\tilde{\mathbf{\Theta}}_k) \\ &= \frac{1}{p} \sum_{q=1}^{p_k} \sum_{r=1}^{p_k} \text{vec}(\tilde{\mathbf{S}}_k^{qr}) [\tilde{\mathbf{\Theta}}_k]_{qr} - \frac{\text{tr}(\mathbf{\Delta}_k \, \tilde{\mathbf{\Theta}}_k)}{p} \text{vec}(\mathbf{\Sigma}), \\ \text{and } \ \mathbf{M}_{3,k} &= \frac{1}{p} \sum_{q=1}^{p_k} \sum_{r=1}^{p_k} \tilde{\mathbf{S}}_k^{qr} [\tilde{\mathbf{\Theta}}_k]_{qr} - \frac{\text{tr}(\mathbf{\Delta}_k \, \tilde{\mathbf{\Theta}}_k)}{p} \, \mathbf{\Sigma} \, . \end{aligned}$$

Define $\mathcal{E}_{1,k} := \mathcal{E}_{\mathbf{\Sigma}}(k, \tilde{\mathbf{\Theta}}_k)$. By the proof of Theorem 26, $\mathbb{P}(\mathcal{E}_{1,k} | \mathcal{X}_{0,k}) \geq 1 - \frac{2}{(n \vee p_k)^2}$. On the other hand, under the event $\mathcal{E}_{1,k}$, Corollary 27 gives $|(\mathbf{M}_{3,k})_{ij}| \leq \nu_{p_k} \| \mathbf{\Delta}_k \|_2 \| \tilde{\mathbf{\Theta}}_k \|_2 \sqrt{\sigma_{ii} \sigma_{jj}}$.

We next bound $\| \tilde{\mathbf{\Theta}}_k \|_2$ using Corollary 10.1 from Zhou (2014b), so assuming that $\mathcal{X}_{0,k}$ holds, then

$$\| \tilde{\mathbf{\Theta}}_k \|_2 \leq C' \lambda_k \frac{\sqrt{s_k \vee 1}}{\delta_{k,\min}^* \phi_{\min}^2(\rho(\mathbf{\Delta}_k))}.$$

In summary, under the event $\mathcal{E}^* := \cap_{k=1}^K (\mathcal{E}_{0,k} \cap \mathcal{E}_{1,k} \cap \mathcal{X}_{0,k})$, we have that

$$\begin{aligned} |(\hat{\mathbf{S}}_{\mathbf{\Sigma}} - \mathbf{\Sigma}^*)_{ij}| &\leq \sum_{k=1}^K (|(\mathbf{M}_{1,k})_{ij}| + |(\mathbf{M}_{2,k})_{ij}| + |(\mathbf{M}_{3,k})_{ij}|) \\ &\leq \sum_{k=1}^K \Bigg[\nu_{p_k} \sqrt{\sigma_{*,ii} \sigma_{*,jj}} + |\sigma_{*,ij}| \tilde{\mu}_k \\ &\qquad + \nu_{p_k} \| \mathbf{\Delta}_k \|_2 \left(C' \lambda_k \frac{\sqrt{s_k \vee 1}}{\delta_{k,\min}^* \phi_{\min}^2(\rho(\mathbf{\Delta}_k))} \right) \sqrt{\sigma_{ii} \sigma_{jj}} \Bigg] \\ &= \sum_{k=1}^K \Bigg[\nu_{p_k} \sqrt{\sigma_{*,ii} \sigma_{*,jj}} + |\sigma_{*,ij}| \tilde{\mu}_k \end{aligned}$$

$$+ \nu_{p_k} \| \mathbf{\Delta}_k \|_2 \left(C' \lambda_k \frac{\sqrt{s_k \vee 1}}{\delta_{k,\min} \phi^2_{\min}(\rho(\mathbf{\Delta}_k))} \right) \sqrt{\sigma_{*,ii}\sigma_{*,jj}} \Bigg]$$
$$\leq \sum_{k=1}^{K} \left[\nu_{p_k} \sqrt{\sigma_{*,ii}\sigma_{*,jj}} (1 + o(1)) + |\sigma_{*,ij}| \tilde{\mu}_k \right]$$

under the assumptions.

Furthermore, applying the union bound implies that $\mathbb{P}(\mathcal{E}^*) \geq 1 - \sum_{k=1}^{K} \frac{8}{(n \vee p_k)^2}$. Assuming that event $\mathcal{X}_{0,k}$ holds, $\tilde{\mu}_k \leq \mu_k$ is a consequence of Corollary 17.4 from Zhou (2014b), and this concludes the proof. ■

It is important to point out that in our choice of λ_k in (71), A can be considered a constant under the bounded spectrum assumption (A2). In addition, (A1) implies that $\sqrt{\frac{\log(n \vee p_k)}{n}} \to 0$ as $n, p_k \to \infty$. Therefore, since λ_k is on the order of $A\sqrt{\frac{\log(n \vee p_k)}{n}}$, $\lambda_k \to 0$ as $n, p_k \to \infty$ (for $k = 1, \dots, K$).

Next, stepping through Algorithm 6, we bound the error between the correlation estimate $\hat{\mathbf{S}}_{\rho,\mathbf{\Sigma}}$ and the true correlation matrix $\rho(\mathbf{\Sigma})$.

Theorem 29 *Suppose the conditions in Lemma 28 hold. Define* $\tilde{\eta}_k := \nu_{p_k}(1 + o(1)) + \tilde{\mu}_k$. *Then under event* $\mathcal{E}^*$, *we have for* $\hat{\mathbf{S}}_{\rho,\mathbf{\Sigma}}$ *defined in step 11 in Algorithm 6 and* $i \neq j$,

$$\left| \left(\hat{\mathbf{S}}_{\rho,\mathbf{\Sigma}} - \rho(\mathbf{\Sigma}) \right)_{ij} \right| \leq \sum_{k=1}^{K} \frac{p_k}{p} \left[\frac{4C\tilde{K}^2 \sqrt{\frac{\log(n \vee p_k)}{p_k}} (1 + o(1))(1 + |\rho(\mathbf{\Sigma})_{ij}|)}{1 - \sum_{k=1}^{K} \tilde{\eta}_k} \right.$$
$$\left. + \frac{2|\rho(\mathbf{\Sigma})_{ij}| \sum_{k=1}^{K} \tilde{\mu}_k}{1 - \sum_{k=1}^{K} \tilde{\eta}_k} \right] \tag{84}$$
$$= \sum_{k=1}^{K} \left[\frac{\nu_{p_k}(1 + o(1))(1 + |\rho(\mathbf{\Sigma})_{ij}|)}{1 - \sum_{k=1}^{K} \tilde{\eta}_k} + \frac{2|\rho(\mathbf{\Sigma})_{ij}| \tilde{\mu}_k}{1 - \sum_{k=1}^{K} \tilde{\eta}_k} \right] \tag{85}$$
$$\leq \frac{2\sum_{k=1}^{K} \eta_k}{1 - \sum_{k=1}^{K} \eta_k}, \qquad \textit{where } \eta_k := \nu_{p_k}(1 + o(1)) + \mu_k. \tag{86}$$

Proof Because $\tilde{\mu}_k \leq \mu_k$ by Lemma 28 and $|\rho(\mathbf{\Sigma})_{ij}| \leq 1$ for all i, j, it is clear that

$$\sum_{k=1}^{K} \left[\frac{\nu_{p_k}(1 + o(1))(1 + |\rho(\mathbf{\Sigma})_{ij}|)}{1 - \sum_{k=1}^{K} \tilde{\eta}_k} + \frac{2|\rho(\mathbf{\Sigma})_{ij}| \tilde{\mu}_k}{1 - \sum_{k=1}^{K} \tilde{\eta}_k} \right] \leq \frac{2\sum_{k=1}^{K} \tilde{\eta}_k}{1 - \sum_{k=1}^{K} \tilde{\eta}_k} \leq \frac{2\sum_{k=1}^{K} \eta_k}{1 - \sum_{k=1}^{K} \eta_k}.$$

Therefore, it suffices to show (84).

Assume throughout this proof that event $\mathcal{E}^*$ holds. Then by Lemma 28,

$$\left| \frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ii}}{\sigma_{*,ii}} - 1 \right| \leq \sum_{k=1}^{K} \left[\nu_{p_k}(1 + o(1)) + \tilde{\mu}_k \right] = \sum_{k=1}^{K} \tilde{\eta}_k,$$

which implies

$$\frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ii}}{\sigma_{*,ii}} \geq 1 - \sum_{k=1}^{K} \tilde{\eta}_k \quad \implies \quad \sqrt{\frac{\sigma_{*,ii}}{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ii}}} \leq \sqrt{\frac{1}{1 - \sum_{k=1}^{K} \tilde{\eta}_k}} \tag{87}$$

for all i. On the other hand, for $i \neq j$, Lemma 28 gives

$$\begin{aligned}
\left| \left(\frac{\hat{\mathbf{S}}_{\mathbf{\Sigma}}}{\sqrt{\sigma_{*,ii}\sigma_{*,jj}}} - \rho(\mathbf{\Sigma}) \right)_{ij} \right| &= \left| \left(\frac{\hat{\mathbf{S}}_{\mathbf{\Sigma}}}{\sqrt{\sigma_{*,ii}\sigma_{*,jj}}} - \frac{\mathbf{\Sigma}^*}{\sqrt{\sigma_{*,ii}\sigma_{*,jj}}} \right)_{ij} \right| & (88)\\
&\leq \sum_{k=1}^{K} \left(\nu_{p_k}(1 + o(1)) + \frac{|\sigma_{*,ij}|}{\sqrt{\sigma_{*,ii}\sigma_{*,jj}}} \tilde{\mu}_k \right) & (89)\\
&= \sum_{k=1}^{K} \left(\nu_{p_k}(1 + o(1)) + |\rho(\mathbf{\Sigma})_{ij}| \tilde{\mu}_k \right), & (90)
\end{aligned}$$

Thus, for $i \neq j$,

$$\begin{aligned}
\left| \left(\hat{\mathbf{S}}_{\rho,\mathbf{\Sigma}} - \rho(\mathbf{\Sigma}) \right)_{ij} \right| &= \left| \frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ij}}{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ii}^{1/2} [\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{jj}^{1/2}} - \rho(\mathbf{\Sigma})_{ij} \right| & (91)\\
&= \left| \frac{\frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ij}}{\sqrt{\sigma_{*,ii}\sigma_{*,jj}}}}{\frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ii}^{1/2}}{\sqrt{\sigma_{*,ii}}} \frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{jj}^{1/2}}{\sqrt{\sigma_{*,jj}}}} - \rho(\mathbf{\Sigma})_{ij} \right| & (92)\\
&\leq \left| \frac{\frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ij}}{\sqrt{\sigma_{*,ii}\sigma_{*,jj}}} - \rho(\mathbf{\Sigma})_{ij}}{\frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ii}^{1/2}}{\sqrt{\sigma_{*,ii}}} \frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{jj}^{1/2}}{\sqrt{\sigma_{*,jj}}}} \right| + \left| \frac{\rho(\mathbf{\Sigma})_{ij}}{\frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ii}^{1/2}}{\sqrt{\sigma_{*,ii}}} \frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{jj}^{1/2}}{\sqrt{\sigma_{*,jj}}}} - \rho(\mathbf{\Sigma})_{ij} \right| & (93)\\
&= \left| \frac{\sqrt{\sigma_{*,ii}}}{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ii}^{1/2}} \frac{\sqrt{\sigma_{*,jj}}}{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{jj}^{1/2}} \right| \left| \frac{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ij}}{\sqrt{\sigma_{*,ii}\sigma_{*,jj}}} - \rho(\mathbf{\Sigma})_{ij} \right| \\
&\qquad + |\rho(\mathbf{\Sigma})_{ij}| \left| \frac{\sqrt{\sigma_{*,ii}}}{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{ii}^{1/2}} \frac{\sqrt{\sigma_{*,jj}}}{[\hat{\mathbf{S}}_{\mathbf{\Sigma}}]_{jj}^{1/2}} - 1 \right| & (94)\\
&\leq \frac{1}{1 - \sum_{k=1}^{K} \tilde{\eta}_k} \left(\sum_{k=1}^{K} \left(\nu_{p_k}(1 + o(1)) + |\rho(\mathbf{\Sigma})_{ij}| \tilde{\mu}_k \right) \right) \\
&\qquad + |\rho(\mathbf{\Sigma})_{ij}| \left| \frac{1}{1 - \sum_{k=1}^{K} \tilde{\eta}_k} - 1 \right| & (95)\\
&= \sum_{k=1}^{K} \frac{\nu_{p_k}(1 + o(1))(1 + |\rho(\mathbf{\Sigma})_{ij}|) + 2|\rho(\mathbf{\Sigma})_{ij}| \tilde{\mu}_k}{1 - \sum_{k=1}^{K} \tilde{\eta}_k}. & (96)
\end{aligned}$$

as desired. Note that (95) follows from (87) and (90). ■

F.2 Main Result

We can now build on top of Theorem 29 and existing results to prove our main convergence result in Theorem 30, which is a generalization of Corollary 17.2 from Zhou (2014b). Indeed, when $K = 1$, Theorem 30 gives the same rates as Zhou (2014a). Moreover, as a direct consequence of Theorem 30, we obtain Corollary 31, which establishes bounds on the eigenvectors and eigenvalues of the additive L_1 correlation iPCA estimator, thereby giving the desired subspace consistency.

Theorem 30 *Suppose that (A1)-(A4) hold and that $\sqrt{n} \geq \frac{p}{\sqrt{p_k}}$ for each $k = 1, \dots, K$. Assume also that $\eta \leq 1/4$, and $\lambda_{\mathbf{\Sigma}}$ is chosen to be*

$$\lambda_{\mathbf{\Sigma}} = \frac{2\sum_{k=1}^K \tilde{\eta}_k}{\epsilon_1(1 - \sum_{k=1}^K \tilde{\eta}_k)}, \qquad \textit{for some } \epsilon_1 \in (0,1). \tag{97}$$

Then on the event $\mathcal{E}^*$,

$$\| \hat{\mathbf{\Sigma}} - \mathbf{\Sigma}^* \|_2 \leq 2\tilde{C}\lambda_{\mathbf{\Sigma}}\sigma_{*,\max}\kappa(\rho(\mathbf{\Sigma}))^2\sqrt{s_{\mathbf{\Sigma}} \vee 1}, \tag{98}$$

$$\| \hat{\mathbf{\Sigma}} - \mathbf{\Sigma}^* \|_F \leq 2\tilde{C}\lambda_{\mathbf{\Sigma}}\sigma_{*,\max}\kappa(\rho(\mathbf{\Sigma}))^2\sqrt{s_{\mathbf{\Sigma}} \vee n}, \tag{99}$$

$$\| \hat{\mathbf{\Sigma}}^{-1} - (\mathbf{\Sigma}^*)^{-1} \|_2 \leq \frac{\tilde{C}\lambda_{\mathbf{\Sigma}}\sqrt{s_{\mathbf{\Sigma}} \vee 1}}{\sigma_{*,\min}\phi^2_{\min}(\rho(\mathbf{\Sigma}))}, \tag{100}$$

$$\| \hat{\mathbf{\Sigma}}^{-1} - (\mathbf{\Sigma}^*)^{-1} \|_2 \leq \frac{\tilde{C}\lambda_{\mathbf{\Sigma}}\sqrt{s_{\mathbf{\Sigma}} \vee n}}{\sigma_{*,\min}\phi^2_{\min}(\rho(\mathbf{\Sigma}))} \tag{101}$$

for some constant $\tilde{C}$.

Proof Assume that $\mathcal{E}^*$ holds throughout this proof.

Next, recall that from Theorem 29,

$$\max_{i\neq j}\left|\left(\hat{\mathbf{S}}_{\rho,\mathbf{\Sigma}} - \rho(\mathbf{\Sigma})\right)_{ij}\right| \leq \omega := \frac{2\sum_{k=1}^K \tilde{\eta}_k}{1 - \sum_{k=1}^K \tilde{\eta}_k} \leq \frac{2\sum_{k=1}^K \eta_k}{1 - \sum_{k=1}^K \eta_k}.$$

By defining $C_{p,k} := \frac{2}{\epsilon}\frac{\| \rho(\mathbf{\Delta}_k)^{-1} \|_{1,\text{off}}}{p} + \frac{\| \rho(\mathbf{\Delta}_k)^{-1} \|_1}{p}$, it follows that $\eta_k = \left(\nu_{p_k} + \frac{\alpha_k}{1-\alpha_k}C_{p,k}\right)(1 + o(1))$. Then because $C_{p,k} \asymp 1$ under (A3) and $\nu_{p_k} \to 0$ as $n, p_k \to \infty$ under (A1) and (A2),

$$\frac{2\sum_{k=1}^K \eta_k}{1 - \sum_{k=1}^K \eta_k} \asymp \sum_{k=1}^K \nu_{p_k} \to 0 \quad \text{as } n, p_k \to \infty.$$

Moreover, under (A1), we have $\omega\sqrt{s_{\mathbf{\Sigma}} \vee 1} = o(1)$. Thus, we can apply Theorem 4.5 from Zhou (2014a) to obtain for some constant $\tilde{C}$,

$$\| \hat{\mathbf{\Sigma}}_\rho - \rho(\mathbf{\Sigma}) \|_2 \leq \| \hat{\mathbf{\Sigma}}_\rho - \rho(\mathbf{\Sigma}) \|_F \leq \tilde{C}\kappa(\rho(\mathbf{\Sigma}))^2\lambda_{\mathbf{\Sigma}}\sqrt{s_{\mathbf{\Sigma}} \vee 1}$$

$$\text{and } \| \hat{\mathbf{\Sigma}}_\rho^{-1} - \rho(\mathbf{\Sigma})^{-1} \|_2 \leq \| \hat{\mathbf{\Sigma}}_\rho^{-1} - \rho(\mathbf{\Sigma})^{-1} \|_F \leq \tilde{C}\lambda_{\mathbf{\Sigma}}\frac{\sqrt{s_{\mathbf{\Sigma}} \vee 1}}{2\phi^2_{\min}(\rho(\mathbf{\Sigma}))}.$$

Now since we have a bound on the correlation estimates, the next step is to consider the covariance estimates. Let us define $\mathbf{W}_{\boldsymbol{\Sigma}} = \text{diag}(\boldsymbol{\Sigma}^*)^{1/2} = \sqrt{\frac{n}{\text{tr}(\boldsymbol{\Sigma})}}\text{diag}(\boldsymbol{\Sigma})^{1/2}$. By Lemma 28,

$$|\hat{\mathbf{W}}^2_{\boldsymbol{\Sigma},ii} - \mathbf{W}^2_{\boldsymbol{\Sigma},ii}| = |\hat{\mathbf{W}}^2_{\boldsymbol{\Sigma},ii} - \sigma_{*,ii}| \le \sigma_{*,ii} \sum_{k=1}^K \tilde{\eta}_k \quad \forall\, i.$$

Therefore,

$$\begin{align} \|\, \hat{\mathbf{W}}_{\boldsymbol{\Sigma}} - \mathbf{W}_{\boldsymbol{\Sigma}} \,\|_2 &\le \sqrt{\sigma_{*,\max}} \left[\left(\sqrt{1 + \sum_{k=1}^K \tilde{\eta}_k} - 1 \right) \vee \left(1 - \sqrt{1 - \sum_{k=1}^K \tilde{\eta}_k} \right) \right] \tag{102} \\ &\le \sqrt{\sigma_{*,\max}} \sum_{k=1}^K \tilde{\eta}_k \tag{103} \\ \text{and } \|\, \hat{\mathbf{W}}^{-1}_{\boldsymbol{\Sigma}} - \mathbf{W}^{-1}_{\boldsymbol{\Sigma}} \,\|_2 &\le \frac{1}{\sqrt{\sigma_{*,\max}}} \left[\frac{\sqrt{1 + \sum_{k=1}^K \tilde{\eta}_k} - 1}{\sqrt{1 + \sum_{k=1}^K \tilde{\eta}_k}} \vee \frac{1 - \sqrt{1 - \sum_{k=1}^K \tilde{\eta}_k}}{\sqrt{1 - \sum_{k=1}^K \tilde{\eta}_k}} \right] \tag{104} \\ &\le \frac{1}{\sqrt{\sigma_{*,\max}}} \frac{\sum_{k=1}^K \tilde{\eta}_k}{\sqrt{1 - \sum_{k=1}^K \tilde{\eta}_k}}. \tag{105} \end{align}$$

Using Proposition 15.2 in Zhou (2014b), (103), and (105), we obtain

$$\begin{aligned} \|\, \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^* \,\|_2 &= \|\, \hat{\mathbf{W}}_{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Sigma}}_\rho \hat{\mathbf{W}}_{\boldsymbol{\Sigma}} - \mathbf{W}_{\boldsymbol{\Sigma}}\, \rho(\boldsymbol{\Sigma})\, \mathbf{W}_{\boldsymbol{\Sigma}} \,\|_2 \\ &\le \Big(\|\, \hat{\mathbf{W}}_{\boldsymbol{\Sigma}} - \mathbf{W}_{\boldsymbol{\Sigma}} \,\|_2 + \|\, \mathbf{W}_{\boldsymbol{\Sigma}} \,\|_2 \Big)^2 \,\|\, \hat{\boldsymbol{\Sigma}}_\rho - \rho(\boldsymbol{\Sigma}) \,\|_2 \\ &\quad + \|\, \hat{\mathbf{W}}_{\boldsymbol{\Sigma}} - \mathbf{W}_{\boldsymbol{\Sigma}} \,\|_2 \Big(\|\, \hat{\mathbf{W}}_{\boldsymbol{\Sigma}} - \mathbf{W}_{\boldsymbol{\Sigma}} \,\|_2 + 2 \Big) \,\|\, \rho(\boldsymbol{\Sigma}) \,\|_2 \\ &\le \Big(\tilde{C} \kappa(\rho(\boldsymbol{\Sigma}))^2 \lambda_{\boldsymbol{\Sigma}} \sqrt{s_{\boldsymbol{\Sigma}} \vee 1} \Big)\, \sigma_{*,\max} \left(1 + \sum_{k=1}^K \tilde{\eta}_k \right)^2 \\ &\quad + \sigma_{*,\max} \sum_{k=1}^K \tilde{\eta}_k \left(\sum_{k=1}^K \tilde{\eta}_k + 2 \right) \|\, \rho(\boldsymbol{\Sigma}) \,\|_2. \end{aligned}$$

And because $\lambda_{\boldsymbol{\Sigma}}$ was chosen to satisfy $\sum_{k=1}^K \tilde{\eta}_k < \lambda_{\boldsymbol{\Sigma}}(1 - \sum_{k=1}^K \tilde{\eta}_k)/2$ where $\sum_{k=1}^K \tilde{\eta}_k \le \sum_{k=1}^K \eta_k \le 1/4$, we have

$$\begin{aligned} \|\, \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^* \,\|_2 &\le \Big(\tilde{C} \kappa(\rho(\boldsymbol{\Sigma}))^2 \lambda_{\boldsymbol{\Sigma}} \sqrt{s_{\boldsymbol{\Sigma}} \vee 1} \Big)\, \sigma_{*,\max} \left(1 + \sum_{k=1}^K \tilde{\eta}_k \right)^2 \\ &\quad + \sigma_{*,\max} \frac{\lambda_{\boldsymbol{\Sigma}}}{2} \left(1 - \sum_{k=1}^K \tilde{\eta}_k \right) \left(\sum_{k=1}^K \tilde{\eta}_k + 2 \right) \|\, \rho(\boldsymbol{\Sigma}) \,\|_2 \\ &\le 2 \tilde{C} \kappa(\rho(\boldsymbol{\Sigma}))^2 \sigma_{*,\max} \lambda_{\boldsymbol{\Sigma}} \sqrt{s_{\boldsymbol{\Sigma}} \vee 1}. \end{aligned}$$

We can also bound the error on the Frobenius norm similarly. Using Proposition 15.2 in Zhou (2014b), (103), (105), $\sum_{k=1}^K \tilde{\eta}_k < \lambda_{\mathbf{\Sigma}}(1-\sum_{k=1}^K \tilde{\eta}_k)/2$, and $\sum_{k=1}^K \tilde{\eta}_k \leq \sum_{k=1}^K \eta_k \leq 1/4$ we see that

$$\begin{aligned}\| \hat{\mathbf{\Sigma}} - \mathbf{\Sigma}^* \|_F &\leq \Big(\| \hat{\mathbf{W}}_{\mathbf{\Sigma}} - \mathbf{W}_{\mathbf{\Sigma}} \|_2 + \| \mathbf{W}_{\mathbf{\Sigma}} \|_2\Big)^2 \| \hat{\mathbf{\Sigma}}_\rho - \rho(\mathbf{\Sigma}) \|_F \\ &\quad + \| \hat{\mathbf{W}}_{\mathbf{\Sigma}} - \mathbf{W}_{\mathbf{\Sigma}} \|_2 \Big(\| \hat{\mathbf{W}}_{\mathbf{\Sigma}} - \mathbf{W}_{\mathbf{\Sigma}} \|_2 + 2\Big) \| \rho(\mathbf{\Sigma}) \|_F \\ &\leq \Big(\tilde{C}\kappa(\rho(\mathbf{\Sigma}))^2 \lambda_{\mathbf{\Sigma}} \sqrt{s_{\mathbf{\Sigma}} \vee 1}\Big)\, \sigma_{*,\max} \left(1 + \sum_{k=1}^K \tilde{\eta}_k\right)^2 \\ &\quad + \sigma_{*,\max} \sum_{k=1}^K \tilde{\eta}_k \left(\sum_{k=1}^K \tilde{\eta}_k + 2\right) \sqrt{n} \| \rho(\mathbf{\Sigma}) \|_2 \\ &\leq 2\tilde{C}\kappa(\rho(\mathbf{\Sigma}))^2 \sigma_{*,\max} \lambda_{\mathbf{\Sigma}} \sqrt{s_{\mathbf{\Sigma}} \vee n}.\end{aligned}$$

The same logic can be used to prove (100) and (101), so we omit the details. ■

To summarize the convergence results from Theorem 30, if we set

$$\lambda_k = \frac{2\alpha_k}{\epsilon(1-\alpha_k)} \asymp \sqrt{\frac{\log(n \vee p_k)}{n}} \quad \forall\, k = 1, \dots, K, \tag{106}$$

$$\text{and } \lambda_{\mathbf{\Sigma}} = \frac{2\sum_{k=1}^K \tilde{\eta}_k}{\epsilon_1(1-\sum_{k=1}^K \tilde{\eta}_k)} \asymp \sum_{k=1}^K \frac{p_k}{p} \sqrt{\frac{\log(n \vee p_k)}{p_k}}, \tag{107}$$

then according to Theorem 30, with probability $1 - \sum_{k=1}^K \frac{8}{(n \vee p_k)^2}$,

$$\| \hat{\mathbf{\Sigma}} - \mathbf{\Sigma}^* \|_2 = O\left(\sum_{k=1}^K \frac{p_k}{p} \sqrt{\frac{(s_{\mathbf{\Sigma}} \vee 1)\log(n \vee p_k)}{p_k}}\right), \tag{108}$$

$$\| \hat{\mathbf{\Sigma}}^{-1} - (\mathbf{\Sigma}^{-1})^* \|_2 = O\left(\sum_{k=1}^K \frac{p_k}{p} \sqrt{\frac{(s_{\mathbf{\Sigma}} \vee 1)\log(n \vee p_k)}{p_k}}\right), \tag{109}$$

$$\| \hat{\mathbf{\Sigma}} - \mathbf{\Sigma}^* \|_F = O\left(\sum_{k=1}^K \frac{p_k}{p} \sqrt{\frac{(s_{\mathbf{\Sigma}} \vee n)\log(n \vee p_k)}{p_k}}\right), \tag{110}$$

$$\| \hat{\mathbf{\Sigma}}^{-1} - (\mathbf{\Sigma}^{-1})^* \|_F = O\left(\sum_{k=1}^K \frac{p_k}{p} \sqrt{\frac{(s_{\mathbf{\Sigma}} \vee n)\log(n \vee p_k)}{p_k}}\right). \tag{111}$$

Subspace consistency for the additive L_1 correlation estimator, $\hat{\mathbf{\Sigma}}$, follows as a direct corollary of Theorem 30.

Corollary 31 *Let $\mathbf{u}_i$ denote the eigenvector of $\mathbf{\Sigma}^*$ corresponding to the eigenvalue ϕ_i and $\hat{\mathbf{u}}_i$ be the eigenvector of $\hat{\mathbf{\Sigma}}$ with eigenvalue $\hat{\phi}_i$, where the eigenvalues are sorted in descending order. Under the conditions stated in Theorem 30, we have the following:*

(i) For each i,

$$|\hat{\phi}_i - \phi_i| = O\left(\sum_{k=1}^{K} \frac{p_k}{p} \sqrt{\frac{(s_{\boldsymbol{\Sigma}} \vee 1)\log(n \vee p_k)}{p_k}}\right).$$

(ii) For each i such that $\min(\phi_{i-1} - \phi_i,\, \phi_i - \phi_{i+1}) \neq 0$*,*

$$\| \hat{\mathbf{u}}_i - \mathbf{u}_i \|_2 = O\left(\sum_{k=1}^{K} \frac{p_k}{p} \sqrt{\frac{(s_{\boldsymbol{\Sigma}} \vee 1)\log(n \vee p_k)}{p_k}}\right).$$

Thus, the eigenvectors and eigenvalues of the one-step additive L_1 correlation estimator, $\hat{\boldsymbol{\Sigma}}$, are consistent.

Proof (i) By Weyl's theorem (Horn and Johnson, 2012), we have for each i,

$$|\hat{\phi}_i - \phi_i| \leq \| \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^* \|_2.$$

Thus, it follows from Theorem 30 that

$$|\hat{\phi}_i - \phi_i| = O\left(\sum_{k=1}^{K} \frac{p_k}{p} \sqrt{\frac{(s_{\boldsymbol{\Sigma}} \vee 1)\log(n \vee p_k)}{p_k}}\right).$$

(ii) By a variant of the Davis-Kahan $\sin\theta$ theorem given in Yu et al. (2015), we have for each i,

$$\| \hat{\mathbf{u}}_i - \mathbf{u}_i \|_2 \leq \frac{2^{3/2} \| \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^* \|_2}{\min(\phi_{i-1} - \phi_i,\, \phi_i - \phi_{i+1})},$$

assuming that $\min(\phi_{i-1} - \phi_i,\, \phi_i - \phi_{i+1}) \neq 0$ and $\hat{\mathbf{v}}^T \mathbf{v} \geq 0$. Note however that if $\hat{\mathbf{u}}_i^T \mathbf{u}_i < 0$, we can simply take the negative of $\hat{\mathbf{u}}_i$ and apply the theorem to $-\hat{\mathbf{u}}_i$ and $\mathbf{u}_i$. Thus, for each i such that $\min(\phi_{i-1} - \phi_i,\, \phi_i - \phi_{i+1}) \neq 0$, it follows from Theorem 30 that

$$\| \hat{\mathbf{u}}_i - \mathbf{u}_i \|_2 = O\left(\sum_{k=1}^{K} \frac{p_k}{p} \sqrt{\frac{(s_{\boldsymbol{\Sigma}} \vee 1)\log(n \vee p_k)}{p_k}}\right).$$

Since $\sum_{k=1}^{K} \frac{p_k}{p} \sqrt{\frac{(s_{\boldsymbol{\Sigma}} \vee 1)\log(n \vee p_k)}{p_k}}$ converges to 0 as $n, p_1, \ldots, p_K \to \infty$ under (A1), this gives us the consistency of the eigenvalues and eigenvectors of $\hat{\boldsymbol{\Sigma}}$. ■

Note though that the consistency statements above assume that $\sqrt{n} \geq \frac{p}{\sqrt{p_k}}$ $\forall\, k = 1, \ldots, K$ (i.e. the "large n" setting). If instead $\sqrt{n} < \frac{p}{\sqrt{p_k}}$ $\forall\, k = 1, \ldots, K$ (i.e. the "large p" setting), then we can modify Algorithm 6 to first initialize an estimate of $\boldsymbol{\Sigma}$ assuming $\hat{\boldsymbol{\Delta}}_k = \mathbf{I}$ for each k. Call this initial estimate $\hat{\boldsymbol{\Sigma}}^1$. Then estimate $\boldsymbol{\Delta}_k$ given $\hat{\boldsymbol{\Sigma}}^1$. Call this estimate $\hat{\boldsymbol{\Delta}}_k$. Lastly, obtain the final estimate of $\boldsymbol{\Sigma}$ given $\hat{\boldsymbol{\Delta}}_1, \ldots, \hat{\boldsymbol{\Delta}}_K$. Denote this final estimate of $\boldsymbol{\Sigma}$ by $\hat{\boldsymbol{\Sigma}}$. Similar convergence rates can be obtained for the "large p" setting by using this modified algorithm. Namely, if the penalty parameters λ_k and $\lambda_{\mathbf{\Sigma}}$ are chosen on the order of (106) and (107), we can obtain the same rates as (108)-(111). The only additional assumption required here is a bound on $|\rho(\mathbf{\Sigma})_{ij}|$, namely, $|\rho(\mathbf{\Sigma})_{ij}| = O(\frac{\sqrt{np_k}}{p})$ for each $k = 1, \dots, K$ and $i \neq j$. We omit this proof as it is a repetition of previous arguments with slight differences. A more thorough discussion of this scenario is presented in Zhou (2014a).

Thus, in either the large n or large p case, we have a variant of the additive L_1 correlation Flip-Flop algorithm such that under certain assumptions, the estimate of $\mathbf{\Sigma}$ converges at a rate of $O\left(\sum_{k=1}^{K} \frac{p_k}{p} \sqrt{\frac{(s_{\mathbf{\Sigma}} \vee 1)\log(n \vee p_k)}{p_k}}\right)$ in the operator norm. As a result of convergence in the operator norm, we obtain consistency of the eigenvalues and eigenvectors of $\hat{\mathbf{\Sigma}}$ (see Corollary 31). Since eigenvectors of $\hat{\mathbf{\Sigma}}$ define the estimated iPCA subspace, this in turn implies subspace consistency of the one-step additive L_1 correlation estimator.

Appendix G. Selecting Penalty Parameters

Our missing data imputation framework for selecting penalty parameters is given in Algorithm 7. In the subsequent sections, we discuss algorithms to impute the missing data in step 4 of this algorithm. As one option, one could use the multi-cycle expectation-conditional maximization (MCECM) (Meng and Rubin, 1993) algorithm, which iterates between taking conditional expectations in the E-step and maximizing with respect to one variable at a time in the M-step. We derive the full MCECM algorithm for iPCA in Appendix G.1. This method generalizes the MCECM imputation method proposed in Allen and Tibshirani (2010), which only considered the $K = 1$ case. However, as Allen and Tibshirani (2010) pointed out, the full MCECM algorithm is computationally expensive, so in practice, we also advocate using a faster one-step approximation to the MCECM algorithm, which we discuss in Appendix G.2.

Algorithm 7 Selecting Penalty Parameters via Missing Imputation Framework

```
Given: data $\mathbf{X}_1, \dots, \mathbf{X}_K$, space of penalty parameters $\Lambda$, type of penalized iPCA estimator
for $k = 1, \dots, K$ do
    Randomly leave out 5% of the elements in $\mathbf{X}_k$; denote these scattered missing elements by $\mathbf{X}_k^m$
for $\lambda$ in $\Lambda$ do
    Impute missing values (preferably by Algorithm 9); denote these imputed values by $\hat{\mathbf{X}}_k^m$
Select $\lambda$ which minimizes $\sum_{k=1}^{K} \frac{\| \hat{\mathbf{X}}_k^m - \mathbf{X}_k^m \|_F^2}{\| \mathbf{X}_k^m - \bar{\mathbf{X}}_k^m \|_F^2}$, where $\bar{\mathbf{X}}_k^m$ are the values of the column mean matrix $\bar{\mathbf{X}}_k$ at the missing indicies.
```

Following the notation in Allen and Tibshirani (2010), we write $\mathbf{X}^o = (\mathbf{X}_1^o, \dots, \mathbf{X}_K^o)$ to denote the totality of observed entries of $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_K)$, and $\mathbf{X}^m = (\mathbf{X}_1^m, \dots, \mathbf{X}_K^m)$ to denote the missing entries of $\mathbf{X}$. Also define $\mathbf{\Theta} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K, \mathbf{\Sigma}^{-1}, \mathbf{\Delta}_1^{-1}, \dots, \mathbf{\Delta}_K^{-1})$, and let

$\mathbf{\Theta}'$ be the current estimates of $\mathbf{\Theta}$. Note that the MCECM and its one-step approximation for iPCA (where $K \geq 1$) are generalizations of the imputation algorithms from Allen and Tibshirani (2010) (where $K = 1$).

G.1 Multi-Cycle Expectation-Conditional Maximization Algorithm

For concreteness, we will work with the multiplicative Frobenius penalty. The other penalized methods are very similar. We will proceed to derive the E-steps and M-steps with respect to each variable for the MCECM algorithm. Note here that we will estimate the column means $\boldsymbol{\mu}_k$ adaptively within the MCECM algorithm, rather than fixing them a priori. We find that this strategy works better in practice.

So first, in order to compute the E-steps, note that the Q function is

$$\begin{aligned}
Q(\mathbf{\Theta};\mathbf{\Theta}') &:= \mathbb{E}_{\mathbf{X}^m \,|\, \mathbf{X}^o, \mathbf{\Theta}'}[\ell(\mathbf{\Theta}|\, \mathbf{X})] \\
&= p \log |\, \mathbf{\Sigma}^{-1} \,| + n \sum_{k=1}^{K} \log |\, \mathbf{\Delta}_k^{-1} \,| \\
&\quad - \mathbb{E}_{\mathbf{X}^m \,|\, \mathbf{X}^o, \mathbf{\Theta}'} \left[\sum_{k=1}^{K} \operatorname{tr}\left(\mathbf{\Sigma}^{-1} \left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right) \mathbf{\Delta}_k^{-1} \left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right)^T \right) \right] \\
&\quad - \|\, \mathbf{\Sigma}^{-1} \,\|_F^2 \sum_{k=1}^{K} \rho_k \|\, \mathbf{\Delta}_k^{-1} \,\|_F^2
\end{aligned}$$

Thus, for the E-step with respect to $\mathbf{\Sigma}$, we use linearity of the expectation and trace operators to obtain

$$\begin{aligned}
&\mathbb{E}_{\mathbf{X}^m \,|\, \mathbf{X}^o, \mathbf{\Theta}'} \left[\sum_{k=1}^{K} \operatorname{tr}\left(\mathbf{\Sigma}^{-1} \left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right) \mathbf{\Delta}_k^{-1} \left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right)^T \right) \right] \\
&\qquad = \operatorname{tr}\left(\mathbf{\Sigma}^{-1} \sum_{k=1}^{K} \mathbb{E}_{\mathbf{X}^m \,|\, \mathbf{X}^o, \mathbf{\Theta}'} \left[\left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right) \mathbf{\Delta}_k^{-1} \left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right)^T \right] \right).
\end{aligned}$$

Using the notation and proof of Proposition 3 in Allen and Tibshirani (2010), the conditional expectation reduces to

$$\mathbb{E}_{\mathbf{X}^m \,|\, \mathbf{X}^o, \mathbf{\Theta}'} \left[\left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right) \mathbf{\Delta}_k^{-1} \left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right)^T \right] = \sum_{k=1}^{K} \left(\hat{\mathbf{X}}_k \mathbf{\Delta}_k^{-1} \hat{\mathbf{X}}_k^T + F_k(\mathbf{\Delta}_k^{-1}) \right). \quad (112)$$

For the E-Step with respect to $\mathbf{\Delta}_k$, a similar argument shows that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{X}^m \,|\, \mathbf{X}^o, \mathbf{\Theta}'} \left[\sum_{k=1}^{K} \operatorname{tr}\left(\mathbf{\Sigma}^{-1} \left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right) \mathbf{\Delta}_k^{-1} \left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right)^T \right) \right] \\
&\qquad = \sum_{k=1}^{K} \operatorname{tr}\left(\mathbb{E}_{\mathbf{X}^m \,|\, \mathbf{X}^o, \mathbf{\Theta}'} \left[\left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right)^T \mathbf{\Sigma}^{-1} \left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T \right) \right] \mathbf{\Delta}_k^{-1} \right),
\end{aligned}$$

and again from Allen and Tibshirani (2010), we have that

$$\mathbb{E}_{\mathbf{X}^m \,|\, \mathbf{X}^o, \boldsymbol{\Theta}'}\left[\left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T\right)^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{X}_k - \mathbf{1}_n \boldsymbol{\mu}_k^T\right)\right] = \hat{\mathbf{X}}_k^T \boldsymbol{\Sigma}^{-1}\hat{\mathbf{X}}_k + G(\boldsymbol{\Sigma}^{-1}). \tag{113}$$

We next plug (112) and (113) back into the Q function and take partial derivatives to compute the M steps.

For the M-step with respect to $\boldsymbol{\Sigma}$, we have that

$$\frac{\partial Q}{\partial \boldsymbol{\Sigma}^{-1}} = p\,\boldsymbol{\Sigma} - \sum_{k=1}^{K}\left(\hat{\mathbf{X}}_k \boldsymbol{\Delta}_k^{-1}\hat{\mathbf{X}}_k^T + F_k(\boldsymbol{\Delta}_k^{-1})\right) - 2\,\boldsymbol{\Sigma}^{-1}\sum_{k=1}^{K}\lambda_k \| \boldsymbol{\Delta}_k^{-1} \|_F^2 = 0,$$

so given $\boldsymbol{\Delta}_k^{-1}$ from the previous iteration, we can update $\boldsymbol{\Sigma}$ via an eigendecomposition of $\sum_{k=1}^{K}\left(\hat{\mathbf{X}}_k \boldsymbol{\Delta}_k^{-1}\hat{\mathbf{X}}_k^T + F_k(\boldsymbol{\Delta}_k^{-1})\right)$. (The form of this update is analogous to the Flip-Flop updates in the multiplicative Frobenius Flip-Flop algorithm.)

Similarly, for the M-step with respect to $\boldsymbol{\Delta}_k$,

$$\frac{\partial Q}{\partial \boldsymbol{\Delta}_k^{-1}} = n\,\boldsymbol{\Delta}_k - \left(\hat{\mathbf{X}}_k^T \boldsymbol{\Sigma}^{-1}\hat{\mathbf{X}}_k + G(\boldsymbol{\Sigma}^{-1})\right) - 2\lambda_k\,\boldsymbol{\Delta}_k^{-1} \| \boldsymbol{\Sigma}^{-1} \|_F^2 = 0,$$

so given $\boldsymbol{\Sigma}^{-1}$ from the previous iteration, we can update $\boldsymbol{\Delta}_k$ by an eigendecomposition of $\hat{\mathbf{X}}_k^T \boldsymbol{\Sigma}^{-1}\hat{\mathbf{X}}_k + G(\boldsymbol{\Sigma}^{-1})$.

Putting these E-steps and M-steps together, we provide the full MCECM algorithm to impute missing values in Algorithm 8.

Algorithm 8 Full MCECM Algorithm for iPCA

```
Set $\hat{\boldsymbol{\mu}}_k$ to be the column means of $\mathbf{X}_k^o$ for each $k = 1, \dots, K$
If $x_{ij}^k$ is missing, set $x_{ij}^k = \hat{\boldsymbol{\mu}}_k^j$.                                   } Initialization
Initialize $\hat{\boldsymbol{\Sigma}}^{-1}$ and $\hat{\boldsymbol{\Delta}}_1^{-1} \dots \hat{\boldsymbol{\Delta}}_K^{-1}$ to be symmetric positive definite.
while not converged do
    Compute $\sum_{k=1}^{K}\left[\hat{\mathbf{X}}_k \hat{\boldsymbol{\Delta}}_k^{-1}\hat{\mathbf{X}}_k^T + F(\hat{\boldsymbol{\Delta}}_k^{-1})\right]$          ▷ E-Step ($\boldsymbol{\Sigma}$)
    Update $\hat{\boldsymbol{\mu}}_k$ to be the column means of $\hat{\mathbf{X}}_k$
    Take eigendecomposition: $\sum_{k=1}^{K}\left[\hat{\mathbf{X}}_k \hat{\boldsymbol{\Delta}}_k^{-1}\hat{\mathbf{X}}_k^T + F(\hat{\boldsymbol{\Delta}}_k^{-1})\right] = \mathbf{U}\,\boldsymbol{\Gamma}\,\mathbf{U}^T$
    Regularize eigenvalues: $\phi_i = \frac{1}{2p}\left(\gamma_i + \sqrt{\gamma_i^2 + 8p\sum_{k=1}^{K}\lambda_k \| \hat{\boldsymbol{\Delta}}_k^{-1} \|_F^2}\right)$   } M-Step ($\boldsymbol{\Sigma}$)
    Update $\hat{\boldsymbol{\Sigma}}^{-1} = \mathbf{U}\,\boldsymbol{\Phi}^{-1}\,\mathbf{U}^T$
    for $k = 1, \dots, K$ do
        Compute $\hat{\mathbf{X}}_k^T \hat{\boldsymbol{\Sigma}}^{-1}\hat{\mathbf{X}}_k + G_k(\hat{\boldsymbol{\Sigma}}^{-1})$          ▷ E-Step ($\boldsymbol{\Delta}_k$)
        Update $\hat{\boldsymbol{\mu}}_k$ to be the column means of $\hat{\mathbf{X}}_k$
        Take eigendecomposition: $\hat{\mathbf{X}}_k^T \hat{\boldsymbol{\Sigma}}^{-1}\hat{\mathbf{X}}_k + G_k(\hat{\boldsymbol{\Sigma}}^{-1}) = \mathbf{V}\,\boldsymbol{\Phi}\,\mathbf{V}^T$
        Regularize eigenvalues: $\gamma_i = \frac{1}{2n}\left(\phi_i + \sqrt{\phi_i^2 + 8n\lambda_k \| \hat{\boldsymbol{\Sigma}}^{-1} \|_F^2}\right)$   } M-Step ($\boldsymbol{\Delta}_k$)
        Update $\hat{\boldsymbol{\Delta}}_k^{-1} = \mathbf{V}\,\boldsymbol{\Gamma}^{-1}\,\mathbf{V}^T$
```

G.2 One-Step Approximation

Algorithm 8 is a generalization of the TRCMAimpute algorithm from Allen and Tibshirani (2010), and as discussed in Allen and Tibshirani (2010), it is computationally expensive to compute $F(\hat{\mathbf{\Delta}}_k^{-1})$ and $G(\hat{\mathbf{\Delta}}_k^{-1})$. Hence, rather than using the full MCECM algorithm to impute the missing values in step 4 of Algorithm 7, we advocate, as in Allen and Tibshirani (2010), using a one-step approximation to the MCECM algorithm. We will empirically show that the one-step approximation is both a good approximation to the full MCECM algorithm and works well in practice.

The idea behind the one-step approximation is that since the first step of the MCECM algorithm typically gives the steepest decrease in the objective function, we will quickly approximate the MCECM algorithm by first obtaining a decent initial imputation and then stopping the algorithm after one M-step and one E-step. We detail the initial imputation step, M-step, and E-step as follows.

For the initial imputation step, we impute missing values assuming $\mathbf{\Sigma} = \mathbf{I}$. If we assume $\mathbf{\Sigma} = \mathbf{I}$, then $\mathbf{X}_k \sim \mathbf{N}_{n,p_k}(\mathbf{1}_n \boldsymbol{\mu}_k^T, \mathbf{I} \otimes \mathbf{\Delta}_k)$ for each $k = 1, \dots, K$, or equivalently, $\mathbf{x}_1^k, \dots, \mathbf{x}_n^k \overset{iid}{\sim} N(\boldsymbol{\mu}_k, \mathbf{\Delta}_k)$, where x_i^k is the i^{th} row of $\mathbf{X}_k$. Since this reduces to the familiar multivariate case, we can initially impute the missing values in $\mathbf{X}_k$ using any (regularized) multivariate normal imputation method such as RCMimpute from Allen and Tibshirani (2010) for each $k = 1, \dots, K$.

Given the initial imputation for $\mathbf{X}_1, \dots, \mathbf{X}_K$ from the previous step, we next compute the M-step and update $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K, \mathbf{\Sigma}, \mathbf{\Delta}_1, \dots, \mathbf{\Delta}_K$ using the penalized Flip-Flop algorithms derived in Appendix C.

In the next and final step, we take an E-step to impute the missing values by $\hat{\mathbf{X}}_k^m = \mathbb{E}[\mathbf{X}_k^m \,|\, \mathbf{X}_k^o, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}, \hat{\mathbf{\Delta}}_k]$ for each $k = 1, \dots, K$. This can be done using the Alternating Conditional Expectations Algorithm from Allen and Tibshirani (2010), applied to each $\mathbf{X}_k$ separately. The only difference is that we specialize to the case where the mean matrix is $\mathbf{M}_k = \mathbf{1}_n \boldsymbol{\mu}_k^T$. We summarize this one-step approximation method for missing data imputation in Algorithm 9.

Algorithm 9 One-Step MCECM Approximation

```
for $k = 1, \dots, K$ do                                          ▷ Initial Imputation // E-Step
    Impute missing values in $\mathbf{X}_k$ assuming $\mathbf{\Sigma} = \mathbf{I}$; call it $\hat{\mathbf{X}}_{k,0}$
        • Use any regularized multivariate normal imputation method
Estimate $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K, \mathbf{\Sigma}, \mathbf{\Delta}_1, \dots, \mathbf{\Delta}_K$ given $\hat{\mathbf{X}}_{1,0}, \dots, \hat{\mathbf{X}}_{K,0}$ via penalized MLE Flip-Flop
algorithm                                                         ▷ M-Step
for $k = 1, \dots, K$ do                                          ▷ E-Step
    Set the missing values $\hat{\mathbf{X}}_k^m = \mathbb{E}[\mathbf{X}_k^m \,|\, \mathbf{X}_k^o, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}, \hat{\mathbf{\Delta}}_k]$ using the alternating conditional
    expectations algorithm as in Allen and Tibshirani (2010)
```

In Figure 8, we compare the numerical convergence of the one-step approximation and the full MCECM algorithm for a small simulation. From this plot, we note two important observations. First, the initialization (assuming $\mathbf{\Sigma} = \mathbf{I}$) in the one-step approximation algorithm makes a significant difference, compared to initializing missing elements to their respective column means as in the full MCECM algorithm. Second, the first update of

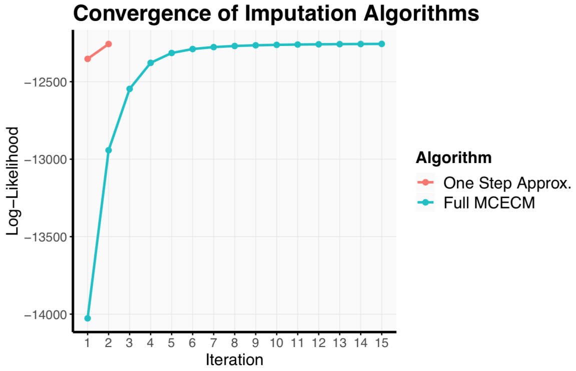

Figure 8: We use the same simulation as that in Figure 9 and randomly leave out 5% of the entries in each data set. We impute the missing values using the full MCECM and one-step approximation algorithms with the multiplicative Frobenius penalty ($\lambda = (1, 1)$), and we plot the log-likelihood value over each iterate. The log-likelihood obtained by the one-step approximation is on par with the log-likelihood after 15 iterations of the MCECM algorithm.

the MCECM algorithm results in the largest increase in the log-likelihood function. As discussed in Allen and Tibshirani (2010), these two observations motivate the one-step approximation algorithm, which takes advantage of a good initialization and one update step to main sufficient accuracy while reducing the computational workload. Figure 8 also shows that the likelihood function after a good initialization and one iteration is on par with the full MCECM algorithm after 15 iterations. For a more detailed discussion on computation and timing comparisons between the one-step approximation and the full MCECM algorithm, we refer to Allen and Tibshirani (2010). Note that though Allen and Tibshirani (2010) only treats the $K = 1$ case, the results are applicable to the $K > 1$ case due to the separability of the log-likelihood function.

Figure 9 compares the average imputation errors from the full MCECM and the one-step approximation for a small simulation. In this case, for both the one-step approximation and the full MCECM, $\lambda = (10^{-.5}, 100) \approx (0.32, 100)$ gave the lowest average imputation error, and hence, both imputation methods selected $\lambda = (10^{-.5}, 100)$ for the multiplicative Frobenius iPCA estimator. This further supports the use of the one-step algorithm as an approximation to the full MCECM in practice. Moreover, we verified that the minimum subspace recovery error of 2.00 was obtained at $\lambda^* = (0.01, 10^{1.5}) \approx (0.01, 31.62)$ and that $\lambda = (10^{-.5}, 100)$ yielded a similar subspace recovery error of 2.08. This preliminary empirical evidence leads us to believe that the one-step imputation algorithm is indeed a good approximation to the full MCECM algorithm.

Appendix H. Simulations

In order to check that the simulation results in Figure 3 are not heavily dependent on our choice of $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \mathbf{\Delta}_2, \mathbf{\Delta}_3$, we ran additional simulations, varying the dimension of the

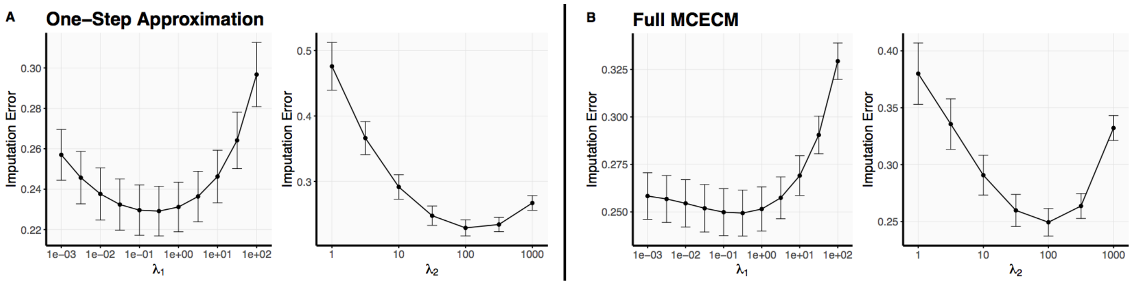

Figure 9: We simulated two coupled data matrices $\mathbf{X}_1, \mathbf{X}_2$ with $n = 50$, $p_1 = 60, p_2 = 70$ according to the iPCA model (2). Here, we took $\mathbf{\Sigma}$ to be as in the base simulation described in Section 4.1, $\mathbf{\Delta}_1$ to be an autoregressive Toeplitz matrix with the $(ij)^{th}$ entry given by $.9^{|i-j|}$, and $\mathbf{\Delta}_2$ to be a block-diagonal matrix with five equal-sized blocks. Then, we randomly removed 5% of the elements in $\mathbf{X}_1$ and $\mathbf{X}_2$ and imputed these missing values using the full MCECM and the one-step approximation algorithms with the multiplicative Frobenius penalty. We plot the average imputation error $\sum_{k=1}^{K} \| \hat{\mathbf{X}}_k^m - \mathbf{X}_k^m \|_F^2 / \| \mathbf{X}_k^m - \bar{\mathbf{X}}_k^m \|_F^2$ plus or minus one standard error, taken over 10 trials. In the left graph of each panel (A and B), λ_1 varies while λ_2 is fixed at its optimal value (i.e. $\lambda_2 = 100$), and in the right graph of each panel, λ_2 varies while λ_1 is fixed at its optimal value (i.e. $\lambda_1 = 10^{-.5}$). The minimum average imputation error is achieved at $\lambda = (10^{-.5}, 100)$ for both imputation algorithms.

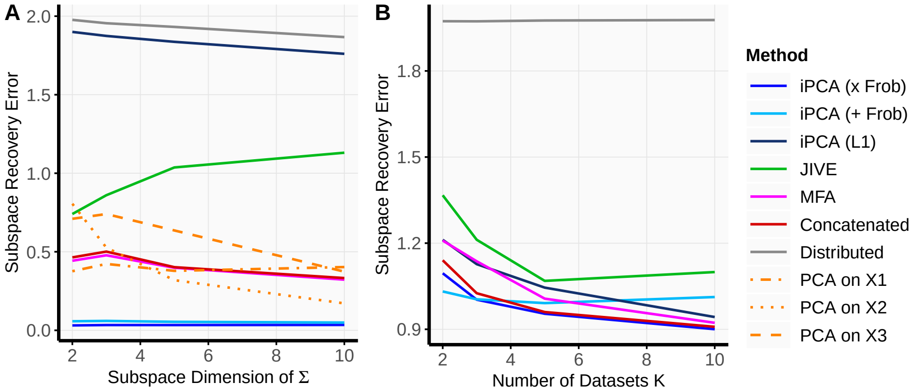

Figure 10: Additional Simulations: (A) The Frobenius iPCA estimators yield the lower subspace recovery error regardless of the subspace dimension of $\mathbf{\Sigma}$; (B) As the number of integrated data sets increases, most methods tend to do better, with the multiplicative Frobenius iPCA penalty slightly outperforming the others when $K = 10$. Note that we did not run individual PCAs in (B) because the number of data sets is changing.

true underlying subspace $\mathbf{U}$ and the number of data sets K. These simulation results are shown in Figure 10.

For the simulations in Figure 10A, we took $\mathbf{\Delta}_1, \mathbf{\Delta}_2, \mathbf{\Delta}_3$ to be the same as in the base simulation, and we put $\mathbf{\Sigma}$ to be of the form $\mathbf{U}\,\mathbf{D}\,\mathbf{U}^T$, where $\mathbf{U}$ was a random $n \times n$ orthogonal matrix, and $\mathbf{D} = \text{diag}(d_1, \ldots, d_n)$ was simulated by $d_i \sim \text{Unif}(5, 75)$ for $i = 1, \ldots, D$, and $d_i = 1$ otherwise. Here, D is the dimension of the true underlying subspace of $\mathbf{\Sigma}$ (e.g. D was taken to be 2 in the base simulation). Then, like in the base simulation, we generated $\mathbf{X}_k$ for each $k = 1, 2, 3$ by $\mathbf{X}_k \sim N(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{\Delta}_k)$, or equivalently $\mathbf{X}_k = \mathbf{\Sigma}^{1/2}\,\mathbf{\Omega}_k\,\mathbf{\Delta}_k^{1/2}$, where $\mathbf{\Omega}_k$ is an $n \times p_k$ random matrix with *i.i.d.* $N(0, 1)$ entries.

For the simulations in Figure 10B, we took $\mathbf{\Sigma}$ to be the same as in the base simulation, and we generated $\mathbf{\Delta}_k$ from a random choice among the covariance types:

(i) Autoregressive Toeplitz matrix with the $(ij)^{th}$ entry given by $\rho^{|i-j|}$, where $\rho \sim \text{Unif}(-.9, .9)$;

(ii) Block diagonal matrix with B blocks of the entries $q_1, \ldots, q_B$, where $B \sim \text{Unif}(3, 10)$ and $q_i \sim \text{Unif}(0, .9)$;

(iii) Spiked covariance matrix $\mathbf{U}\,\mathbf{D}\,\mathbf{U}^T$, where $\mathbf{U}$ is a random orthogonal matrix, $d_i \sim \text{Unif}(5, 75)$ for $i = 1, \ldots, D$, $d_i = 1$ for $i = D+1, \ldots, p_k$, and $D \sim \text{Unif}(5, 50)$;

(iv) Observed covariance matrix of real miRNA data from TCGA Ovarian Cancer. (Note that this covariance matrix can only be used for one $k \in \{1, \ldots, K\}$)

The number of features was randomly selected by $p_k \sim \text{Unif}(200, 500)$, and we ensured that $\|\,\mathbf{\Delta}_k\,\|$ was larger than that of $\mathbf{\Sigma}$ so that the individual signal was larger than the joint signal.

As conveyed in Figure 10A, the Frobenius iPCA estimators outperformed its competitors regardless of the subspace dimension of $\mathbf{\Sigma}$. It is also encouraging to see that the strong performance of the Frobenius iPCA estimators is not dependent on the cluster model for $\mathbf{\Sigma}$, which was used in the simulations in Figure 3 but not in Figure 10A. Note that since the simulated $\mathbf{\Sigma}$ was dense, the additive L_1 penalized iPCA estimator should perform poorly, as it does. We also point out that JIVE tends to do worse as the subspace dimension of $\mathbf{\Sigma}$ increases because JIVE tends to underestimate the rank of the joint variation matrix when the subspace dimension of $\mathbf{\Sigma}$ is larger. This is one of the disadvantages of matrix factorizations, as they require the rank of the factorized matrices to be chosen a priori.

In Figure 10B, we see that most of the methods perform better as the number of integrated data sets increases and that the multiplicative Frobenius iPCA estimator slightly outperforms all the other methods when $K = 10$. We speculate that the additive Frobenius iPCA estimator does worse for large values of K because the grid of possible penalty parameters was too course, and as K increase, so does the number of penalty parameters. Hence, it becomes more difficult to choose appropriate penalty parameters when K is large.

For the Laplacian simulations in Figure 4A, we generated $\mathbf{X}_k$ for each $k = 1, 2, 3$ by $\mathbf{X}_k = \mathbf{\Sigma}^{1/2}\,\mathbf{\Omega}\,\mathbf{\Delta}_k^{1/2} + \mathbf{E}_k$, where $\mathbf{\Sigma}$ and $\mathbf{\Delta}_k$ were taken as in the base simulation, and $\mathbf{E}_k$ was an $n \times p_k$ random matrix with *i.i.d.* Laplace$(0, b)$ entries.

For the simulations in Figure 4B, we simulated three coupled data sets from an instance of the JIVE model: for each $k = 1, 2, 3$, $\mathbf{X}_k = \mathbf{J}_k + \mathbf{A}_k + \mathbf{E}_k$, where $\mathbf{J} = [\mathbf{J}_1, \mathbf{J}_2, \mathbf{J}_3]$ is the joint variation matrix of rank $r = 5$, $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ are the individual variation matrices of rank $r_1 = 10$, $r_2 = 15$, $r_3 = 20$, respectively, and $\mathbf{E}_k$ are error matrices with independent entries from $N(0, \sigma^2)$. Similar to the simulations in Lock et al. (2013), we set $\mathbf{J}$ and $\mathbf{A}_k$ by $\mathbf{J} = \mathbf{U}\,\mathbf{V}_k$ and $\mathbf{A}_k = \mathbf{U}_k\,\mathbf{W}_k$, where $\mathbf{U} \in \mathbb{R}^{n \times r}$, $\mathbf{V}_k \in \mathbb{R}^{r \times p_k}$, $\mathbf{U}_k \in \mathbb{R}^{n \times r_k}$, and $\mathbf{W}_k \in \mathbb{R}^{r_k \times p_k}$. Here, $\mathbf{U}, \mathbf{U}_k, \mathbf{V}_k, \mathbf{W}_k$ are matrices whose entries are randomly generated *i.i.d.* from one of the following distributions: $N(0, 1)$, Unif$(0, 1)$, Exp(1), and the discrete random variable $\{-2, -1, 0, 1, 2\}$ with uniform probabilities. Note that under this JIVE model, the true joint covariance matrix is given by $\mathbf{\Sigma} = \mathbf{J}\mathbf{J}^T$.

In addition to the robustness simulations in Figure 4, we also ran simulations under the CMF model and simulations with uncommon row covariance matrices. When simulating from the CMF model, we generated 3 coupled data matrices with $n = 150$ and $p_1 = 300$, $p_2 = 500$, $p_3 = 400$ via the model $X_k = \mathbf{U}\,\mathbf{V}_k^T + \mathbf{E}_k$. Here, $\mathbf{U} \in \mathbb{R}^{n \times 2}$ was taken to be a random two-dimensional subspace from a cluster model with three clusters (as shown in Figure 1), each $\mathbf{V}_k \in \mathbb{R}^{p_k \times 2}$ was taken to be the two top eigenvectors from the base simulation's $\mathbf{\Delta}_k$ (e.g., $\mathbf{V}_1$ was taken to be the top two eigenvectors of the autoregressive Toeplitz matrix with entry (i, j) given by $.9^{|i-j|}$), and $\mathbf{E}_k$ is a random matrix with *i.i.d.* $N(0, \sigma^2)$ entries. We summarize the simulation results from the CMF model with increasing levels of noise σ in Figure 11. From this figure, we see that when the additive noise is small, CMF (and concatenated PCA) yield slightly lower subspace recovery errors than the multiplicative Frobenius iPCA estimator, but as σ increases, this improvement over the multiplicative Frobenius iPCA estimator diminishes.

For the simulations with uncommon row covariance matrices, we modified the base simulation so that two out of the three data sets arose from the model $\mathbf{X}_k \sim N_{n,p_k}(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{\Delta}_k)$ while the final data set arose from the model $\mathbf{X}_k \sim N_{n,p_k}(\mathbf{0}, \tilde{\mathbf{\Sigma}} \otimes \mathbf{\Delta}_k)$. Here, $\mathbf{\Sigma}$ and $\mathbf{\Delta}_1, \mathbf{\Delta}_2, \mathbf{\Delta}_3$ are as in the base simulation while $\tilde{\mathbf{\Sigma}}$ is a rank-2 spiked covariance with eigen-

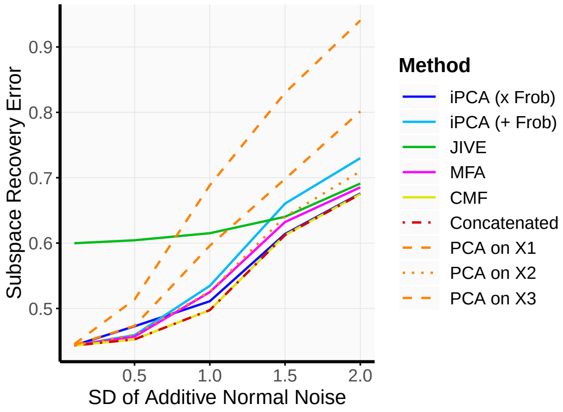

Figure 11: CMF Model Simulations: As the noise level σ of the CMF model increases, the multiplicative Frobenius iPCA estimator performs on par with CMF (and concatenated PCA).

vectors generated from *i.i.d.* random normal entries. We summarize the simulation results from this uncommon row covariance model in Figure 12. In Figure 12A, we see that regardless of which $\mathbf{X}_k$ is generated from the uncommon $\tilde{\mathbf{\Sigma}}$, the Frobenius iPCA estimators are relatively robust to the model misspecification. Figure 12B shows the difference in subspace recovery error when applying the methods to all three data sets (i.e. mixture) versus applying them to the two data sets generated by the common $\mathbf{\Sigma}$ (i.e. oracle). Here, we see that while the Frobenius iPCA estimators perform better with oracle knowledge of the two data sets generated by the common $\mathbf{\Sigma}$, the decline in performance is relatively small when adding the outlying data set, again demonstrating the robustness of the Frobenius iPCA estimators.

In the last set of simulations, we empirically study the iPCA estimators under the sparse setting. For these sparse simulations, we generated two data sets with $n = 50$ and $p_1 = 50$, $p_2 = 100$ according to the Kronecker product model $\mathbf{X}_k \sim N_{n,p_k}(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{\Delta}_k)$. Because we are interested in uncovering the low-rank sparse structure when performing dimension reduction, we generate $\mathbf{\Sigma}$ as follows: Let $\mathbf{U}$ denote the eigenvectors of a block diagonal matrix with B equally-sized blocks, and put $\mathbf{D} = \text{diag}(25, 12.5, 1, \dots, 1) \in \mathbb{R}^{n \times n}$. Then simulate $\mathbf{\Sigma}$ $= \mathbf{UDU}^T$ so that $\mathbf{\Sigma}$ is a low-rank block diagonal matrix with B blocks. To generate $\mathbf{\Delta}_1$, we use the `huge` package (Jiang et al., 2019) in `R` to obtain the sparse covariance matrix associated with multivariate normal data being generated from a sparse banded graph (with bandwidth = 4). Lastly, we took $\mathbf{\Delta}_2$ to be the block diagonal matrix (with 5 equally-sized blocks), created by taking the observed covariance matrix of miRNA data from TCGA ovarian cancer (The Cancer Genome Atlas Research Network, 2011) and zeroing out the entries off of the block diagonal. The results from this sparse simulation study as we increase the number of blocks in $\mathbf{\Sigma}$ (and hence increase the sparsity in $\mathbf{\Sigma}$) are shown in Figure 13. From this study, we see that when the amount of sparsity is relatively low, the multiplicative Frobenius iPCA estimator and the additive L_1 iPCA covariance estimator perform the best while the additive L_1 iPCA correlation estimator performs poorly. We believe that in this

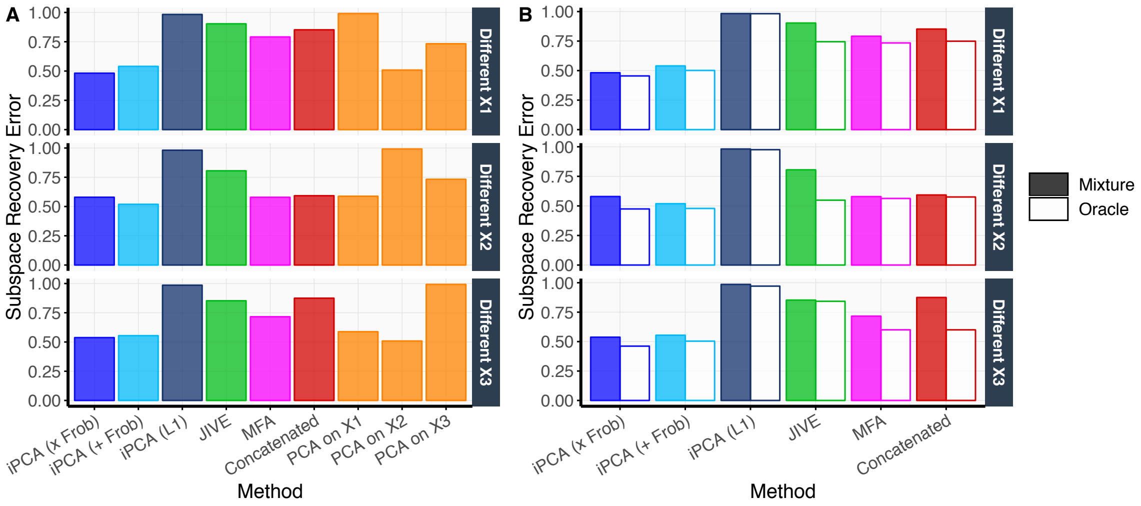

Figure 12: Uncommon Row Covariance Simulations: (A) We plot the subspace recovery errors for various method under the uncommon row covariance model. In the top panel, $\mathbf{X}_1$ was generated from the uncommon $\tilde{\mathbf{\Sigma}}$. In the middle panel, $\mathbf{X}_2$ was generated differently, and in the bottom panel, $\mathbf{X}_3$ was different. (B) We compare the subspace recovery errors from various methods when applied to the mixture of all three data sets (i.e. mixture) versus when applied to the two data sets which were generated by the common $\mathbf{\Sigma}$ (i.e. oracle).

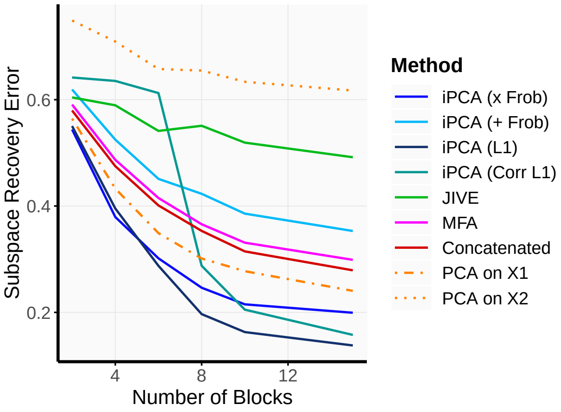

Figure 13: Sparse Simulations: As we increase the number of blocks in $\mathbf{\Sigma}$ (and hence also the sparsity of $\mathbf{\Sigma}$), the sparse iPCA estimators improve over the Frobenius iPCA estimators. However, when the sparsity level is relatively low, the multiplicative Frobenius iPCA estimator performs similarly to the additive L_1 iPCA covariance estimator.

low sparsity scenario, the additive L_1 iPCA correlation estimator struggles with choosing the appropriate penalty parameters and thus does not perform as well. However, as the sparsity level increases, both the additive L_1 iPCA covariance and correlation estimators outperform the dense Frobenius iPCA estimators.

We finally note that other metrics such as canonical angles, which also quantify the distance between subspaces, these metrics behave similarly to the subspace recovery error, so we omitted the results for brevity. Other common metrics such as $|\| \hat{\mathbf{\Sigma}} \|_2 - \| \mathbf{\Sigma} \|_2|$ or $\| \hat{\mathbf{\Sigma}} - \mathbf{\Sigma} \|_F^2$ are not appropriate for our study because we are interested in the distance between subspaces of eigenvectors, and eigenvectors are scale-invariant while these metric are not.

Appendix I. Case Study: Integrative Genomics of Alzheimer's Disease

The ROSMAP data originally contained 309 miRNAs, $41,809$ genes, and $420,132$ CpG sites, so we aggressively preprocessed the number of features in the RNASeq and methylation data sets to manageable sizes. First, we transformed the methylation data to m-values and log-transformed the RNASeq counts, as is common in most analyses for these data types. Then, we removed batch (experimental) effects from both data sets via ComBat (Johnson et al., 2007). We next filtered the features by taking those with the highest variance (top 20,000 genes for RNASeq and top 50,000 CpG sites for methylation). Then, we performed univariate filtering and kept the features with the highest association to clinician's diagnosis. This left us with $p_1 = 309$, $p_2 = 900$, $p_3 = 1250$ in the miRNA, RNASeq, and methylation data sets, respectively.

R code can be found at `https://github.com/DataSlingers/iPCA`.

References

H. Abdi, L. J. Williams, and D. Valentin. Multiple factor analysis: principal component analysis for multitable and multiblock data sets. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(2):149–179, 2013.

E. Acar, E. E. Papalexakis, G. Gürdeniz, M. A. Rasmussen, A. J. Lawaetz, M. Nilsson, and R. Bro. Structure-revealing data fusion. *BMC Bioinformatics*, 15(1):239, Jul 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-239. URL `https://doi.org/10.1186/1471-2105-15-239`.

G. I. Allen and R. Tibshirani. Transposable regularized covariance models with an application to missing data imputation. *The Annals of Applied Statistics*, 4(2):764–790, 2010.

O. Alter, P. O. Brown, and D. Botstein. Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *Proceedings of the National Academy of Sciences*, 100(6):3351–3356, 2003.

K. Benidis, Y. Sun, P. Babu, and D. P. Palomar. Orthogonal sparse PCA and covariance estimation via procrustes reformulation. *IEEE Transactions on Signal Processing*, 64(23):6211–6226, 2016. URL `http://www.danielppalomar.com/publications.html`.

O. Carrette, I. Demalte, A. Scherl, O. Yalkinoglu, G. Corthals, P. Burkhard, D. F. Hochstrasser, and J.C. Sanchez. A panel of cerebrospinal fluid potential biomarkers for the diagnosis of Alzheimer's disease. *Proteomics*, 3(8):1486–1494, 2003.

A.P. Dawid. Some matrix-variate distribution theory: notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274, 1981.

P. Dutilleul. The MLE algorithm for the matrix normal distribution. *Journal of Statistical Computation and Simulation*, 64(2):105–123, 1999.

B. Escofier and J. Pages. Multiple factor analysis. *Computational Statistics & Data Analysis*, 18(1):121–140, 1994.

I. Espuny-Camacho, A. M. Arranz, M. Fiers, A. Snellinx, K. Ando, S. Munck, J. Bonnefont, L. Lambot, N. Corthout, L. Omodho, et al. Hallmarks of Alzheimer's disease in stem-cell-derived human neurons transplanted into mouse brain. *Neuron*, 93(5):1066–1081, 2017.

J. Fan, D. Wang, K. Wang, and Z. Zhu. Distributed estimation of principal eigenspaces. *ArXiv e-prints*, February 2017.

M. Ghil and P. Malanotte-Rizzoli. Data assimilation in meteorology and oceanography. In *Advances in Geophysics*, volume 33, pages 141–266. Elsevier, 1991.

K. Greenewald and A. O. Hero. Robust kronecker product PCA for spatio-temporal covariance estimation. *IEEE Transactions on Signal Processing*, 63(23):6368–6378, 2015.

A. K. Gupta and D. K. Nagar. *Matrix variate distributions*. CRC Press, 1999.

P. Han, W. Liang, L. C. Baxter, J. Yin, Z. Tang, T. G. Beach, R. J. Caselli, E. M. Reiman, and J. Shi. Pituitary adenylate cyclase–activating polypeptide is reduced in Alzheimer disease. *Neurology*, 82(19):1724–1728, 2014.

T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh. Matrix completion and low-rank SVD via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1): 3367–3402, 2015.

R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

C. J. Hsieh, I. S. Dhillon, P. K. Ravikumar, and M. A. Sustik. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems*, pages 2330–2338, 2011.

Sijia Huang, Kumardeep Chaudhary, and Lana X Garmire. More is better: recent progress in multi-omics data integration methods. *Frontiers in genetics*, 8:84, 2017.

Haoming Jiang, Xinyu Fei, Han Liu, Kathryn Roeder, John Lafferty, Larry Wasserman, Xingguo Li, and Tuo Zhao. *huge: High-Dimensional Undirected Graph Estimation*, 2019. URL `https://CRAN.R-project.org/package=huge`. R package version 1.3.2.

W. E. Johnson, C. Li, and A. Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127, 2007.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004.

Y. Li, Z. Chen, Y. Gao, G. Pan, H. Zheng, Y. Zhang, H. Xu, G. Bu, and H. Zheng. Synaptic adhesion molecule Pcdh-γC5 mediates synaptic dysfunction in Alzheimer's disease. *Journal of Neuroscience*, pages 1051–17, 2017.

E. F. Lock, K. A. Hoadley, J. S. Marron, and A. B. Nobel. Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. *The Annals of Applied Statistics*, 7(1):523, 2013.

X. L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993. ISSN 00063444. URL `http://www.jstor.org/stable/2337198`.

S. Mostafavi, C. Gaiteri, S. E. Sullivan, C. C. White, S. Tasaki, J. Xu, M. Taga, H. U. Klein, E. Patrick, V. Komashko, et al. A molecular network of the aging human brain provides insights into the pathology and cognitive decline of Alzheimer's disease. *Nature Neuroscience*, 21(6):811–819, 2018. ISSN 1546-1726. URL `https://doi.org/10.1038/s41593-018-0154-9`.

S. P. Ponnapalli, M. A. Saunders, C. F. Van Loan, and O. Alter. A higher-order generalized singular value decomposition for comparison of global mRNA expression from multiple organisms. *PloS one*, 6(12):e28072, 2011.

T. Rapcsák. Geodesic convexity in nonlinear optimization. *Journal of Optimization Theory and Applications*, 69(1):169–183, Apr 1991. ISSN 1573-2878. doi: 10.1007/BF00940467. URL `https://doi.org/10.1007/BF00940467`.

A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008. doi: 10.1214/08-EJS176.

S. T. Shivappa, M. M. Trivedi, and B. D. Rao. Audiovisual information fusion in human-computer interfaces and intelligent environments: A survey. *Proceedings of the IEEE*, 98 (10):1692–1715, 2010.

A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 650–658. ACM, 2008.

The Cancer Genome Atlas Research Network. Integrated genomic analyses of ovarian carcinoma. *Nature*, 474(7353):609, 2011.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

T. Tsiligkaridis, A. O. Hero, and S. Zhou. On convergence of kronecker graphical lasso algorithms. *IEEE Transactions on Signal Processing*, 61:1743–1755, 2013.

N. K. Vishnoi. Geodesic Convex Optimization: Differentiation on Manifolds, Geodesics, and Convexity. *ArXiv e-prints*, June 2018.

J. A. Westerhuis, T. Kourti, and J. F. MacGregor. Analysis of multiblock and hierarchical PCA and PLS models. *Journal of Chemometrics*, 12(5):301–321, 1998.

A. Wiesel. Geodesic convexity and covariance estimation. *IEEE Transactions on Signal Processing*, 60(12):6182, 2012.

J. Yin and H. Li. Model selection and estimation in the matrix normal graphical model. *Journal of Multivariate Analysis*, 107:119–140, 2012.

Y. Yu, T. Wang, and R. J. Samworth. A useful variant of the Davis-Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015. doi: 10.1093/biomet/asv008. URL `http://dx.doi.org/10.1093/biomet/asv008`.

H. Zhang and S. Sra. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pages 1617–1638, 2016.

S. Zhou. Gemini: Graph estimation with matrix variate normal instances. *The Annals of Statistics*, 42(2):532–562, 2014a.

S. Zhou. Supplement to "gemini: Graph estimation with matrix variate normal instances". 2014b.