

ALPCAH: Subspace Learning for Sample-Wise Heteroscedastic Data

Javier Salazar Cavazos , *Graduate Student Member, IEEE*, Jeffrey A. Fessler , *Fellow, IEEE*, and Laura Balzano , *Senior Member, IEEE*

Abstract—Principal component analysis (PCA) is a key tool in the field of data dimensionality reduction. However, some applications involve heterogeneous data that vary in quality due to noise characteristics associated with each data sample. Heteroscedastic methods aim to deal with such mixed data quality. This paper develops a subspace learning method, named ALPCAH, that can estimate the sample-wise noise variances and use this information to improve the estimate of the subspace basis associated with the low-rank structure of the data. Our method makes no distributional assumptions of the low-rank component and does not assume that the noise variances are known. Further, this method uses a soft rank constraint that does not require subspace dimension to be known. Additionally, this paper develops a matrix factorized version of ALPCAH, named LR-ALPCAH, that is much faster and more memory efficient at the cost of requiring subspace dimension to be known or estimated. Simulations and real data experiments show the effectiveness of accounting for data heteroscedasticity compared to existing algorithms. Code available at <https://github.com/javiersc1/ALPCAH>.

Index Terms—Heteroscedastic data, heterogeneous data quality, subspace basis estimation, subspace learning.

I. INTRODUCTION

MANY modern data-science problems require learning an approximate signal subspace basis for some collection of data. This process is important for downstream tasks involving the subspace basis coefficients such as classification [1], regression [2], and compression [3]. More concretely, lesion detection [4], motion estimation [5], dynamic MRI reconstruction [6], and image/video denoising [7] are practical applications involving the estimation of a subspace basis. In the modern “big data” world, a significant amount of data is collected to solve problems, and this data tends to belong to a high-dimensional ambient space. However, the underlying relationships between the data features are often low dimensional so the problem shifts towards finding low-dimensional structure in the data.

Received 30 April 2024; revised 12 December 2024; accepted 22 January 2025. Date of publication 31 January 2025; date of current version 17 February 2025. This work was supported in part by NSF CAREER under Grant CCF-1845076 and in part by NSF under Grant CCF-2331590. The associate editor coordinating the review of this article and approving it for publication was Renato L. G. Cavalcante. (*Corresponding author: Javier Salazar Cavazos.*)

The authors are with ECE Department, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: javiersc@umich.edu; fessler@umich.edu; girasole@umich.edu).

Digital Object Identifier 10.1109/TSP.2025.3537867

Some applications involve heterogeneous data that vary in quality due in part to noise characteristics associated with each data sample. Some examples of heteroscedastic datasets include environmental air data [8], astronomical spectral data [9], and biological sequencing data [10]. In heteroscedastic settings, the noisier data samples can significantly corrupt conventional basis estimates [11]. Subspace learning methods like probabilistic PCA (PPCA) [12] work well in the homoscedastic setting, meaning when the data is the same quality throughout, but fail to accurately estimate bases in the heteroscedastic setting [13]. This limitation is due to implicit assumptions such as assuming that each sample’s noise distribution is the same throughout (PPCA), or in the case of the classical Robust PCA (RPCA) method [14], that there are fewer outliers than good quality data samples.

A natural approach could be to simply discard the noisiest samples to avoid this issue. This approach requires the user to know the data quality, which may be unavailable in practice. That approach also assumes that there is enough good data to estimate the basis, but it is possible that a lack of good data requires using the noisy data, especially if the subspace dimension is higher than the amount of good data. Furthermore, even the noisier samples can help improve the basis estimate if properly modeled [13], so it is preferable to use all of the data available. This paper introduces subspace learning algorithms that can estimate the sample-wise noise variances and use this information in the model to improve the estimate of the subspace basis associated with the low-rank structure of the data. See Fig. 1 for a visualization where PCA fails to account for heteroscedasticity in a simple 2D data example, but our LR-ALPCAH method more accurately finds the subspace basis.

The proposed subspace learning method, ALPCAH, first introduced in previous proceedings work [15], allows for optional use of rank knowledge via a low-rank promoting functional and makes no distributional assumptions about the low-rank component of the data, allowing it to achieve higher accuracy than current methods without knowing the noise variances. Moreover, we extend our previous proceedings work [15] by developing an alternative formulation inspired by the matrix factorization literature [16], that saves both memory and compute time at the cost of requiring the subspace dimension to be known or estimated.

The paper is organized as follows. Section II introduces the heteroscedastic problem formulation for subspace learning and

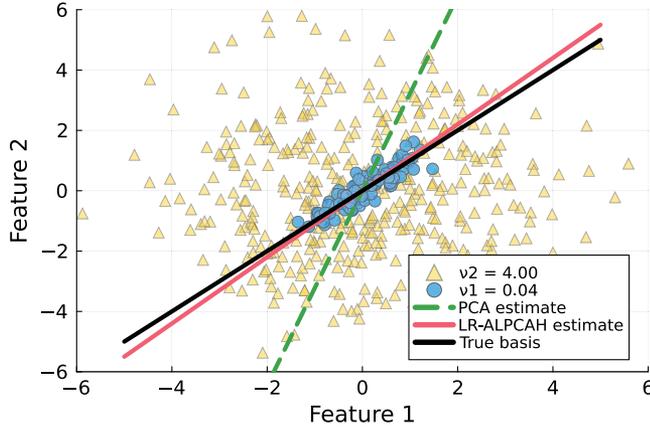


Fig. 1. 1D subspace with data consisting of two noise groups shown with circle and triangle markers.

discusses related work. Section III introduces two subspace learning methods, one with nuclear norm style low-rank regularization originally introduced in our proceedings paper [15], and an extension to a regularization-free maximum likelihood approach. Section IV covers synthetic and real data experiments that illustrate the effectiveness of these methods. Finally, Section V discusses some limitations of our methods and possible extensions.

II. PROBLEM FORMULATION & RELATED WORKS

Let $\mathbf{y}_i \in \mathbb{R}^D$ denote the data samples for index $i \in \{1, \dots, N\}$ given N total samples, and let D denote the ambient dimension. Let \mathbf{x}_i represent the low-dimensional data sample generated by $\mathbf{x}_i = \mathbf{U}\mathbf{z}_i$ where $\mathbf{U} \in \mathbb{R}^{D \times d}$ is an unknown subspace basis of dimension d and $\mathbf{z}_i \in \mathbb{R}^d$ are the corresponding basis coordinates. Collect the measurements into a matrix $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$. Then the heteroscedastic model we consider is

$$\mathbf{y}_i = \mathbf{x}_i + \boldsymbol{\epsilon}_i \quad \text{where} \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \nu_i \mathbf{I}) \quad (1)$$

assuming Gaussian noise with variance ν_i , where \mathbf{I} denotes the $D \times D$ identity matrix. We consider both the case where each data point may have its own noise variance, and cases where there are G groups of data having shared noise variance terms $\{\nu_1, \dots, \nu_G\}$. Section III proposes an optimization problem that estimates the heterogeneous noise variances $\{\nu_i\}$ and the subspace basis \mathbf{U} .

A. Heteroscedastic Impact on Subspace Quality

Before describing the methods, we illustrate how heteroscedastic data impacts the quality of the PCA subspace basis estimate $\hat{\mathbf{U}}_{:,1:d}$, the first d columns of $\hat{\mathbf{U}}$. Let $\mathbf{Y} = \mathbf{X} + \mathbf{E}$ where $\mathbf{X} \in \mathbb{R}^{D \times N}$ is a rank- d matrix and $\mathbf{E} \in \mathbb{R}^{D \times N}$ is the noise matrix where $\mathbf{E}_{:,i} \sim \mathcal{N}(\mathbf{0}, \nu_i \mathbf{I}) \quad \forall j$. Let $\mathbf{Y} = \hat{\mathbf{U}}\hat{\boldsymbol{\Sigma}}\hat{\mathbf{V}}'$ and $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}'$ denote singular value decompositions of their respective matrices, $\sigma_i(\mathbf{A})$ denotes the i th singular value of \mathbf{A} . Let $\|\mathbf{A}\|_2$ denote the spectral norm of matrix \mathbf{A} and $\|\mathbf{x}\|_2$

denote the Euclidean norm of a vector \mathbf{x} . The notation $a \lesssim b$ means $\exists k > 0$ s.t. $a \leq kb$. By Wedin-Davis-Kahan $\sin \theta$ theorem [17, p. 95], it is known that

$$\|\hat{\mathbf{U}}_{:,1:d}\hat{\mathbf{U}}'_{:,1:d} - \mathbf{U}_{:,1:d}\mathbf{U}'_{:,1:d}\|_2 \leq \frac{2\|\mathbf{E}\|_2}{\sigma_d(\mathbf{X}) - \sigma_{d+1}(\mathbf{X})}. \quad (2)$$

This inequality states that the maximum angle misalignment between the latent subspace basis $\mathbf{U}_{:,1:d}$ and the SVD-estimated subspace $\hat{\mathbf{U}}_{:,1:d}$ is bounded by the spectral norm of the noise matrix over the spectral gap in matrix \mathbf{X} . Assuming the elements in \mathbf{E} are zero mean and independent (not necessarily identically distributed) random variables it is known from [18] that, in expectation, the spectral norm of \mathbf{E} is bounded as

$$\begin{aligned} \mathbb{E}[\|\mathbf{E}\|_2] &\lesssim \max_i \sqrt{\sum_j \mathbb{E}[E_{ij}^2]} \\ &+ \max_j \sqrt{\sum_i \mathbb{E}[E_{ij}^2]} + \sqrt[4]{\sum_{i,j} \mathbb{E}[E_{ij}^4]}. \end{aligned} \quad (3)$$

Because $\mathbf{E}_{:,i} \sim \mathcal{N}(\mathbf{0}, \nu_i \mathbf{I}) \quad \forall j$ in our application, it can be verified that

$$\max_i \sqrt{\sum_j \mathbb{E}[E_{ij}^2]} = \sqrt{\nu_{\text{sum}}^{(1)}} \quad (4)$$

$$\max_j \sqrt{\sum_i \mathbb{E}[E_{ij}^2]} = \sqrt{D\nu_{\text{max}}} \quad (5)$$

$$\sqrt[4]{\sum_{i,j} \mathbb{E}[E_{ij}^4]} = \sqrt[4]{3D\nu_{\text{sum}}^{(2)}} \quad (6)$$

for $\nu_{\text{max}} = \max_i \nu_i$ and $\nu_{\text{sum}}^{(k)} = \sum_i \nu_i^k$. Let $\mathbf{C}_{\mathbf{X}}$ correspond to the covariance matrix of \mathbf{X} , i.e., $\mathbf{C}_{\mathbf{X}} = \frac{1}{N}\mathbf{X}\mathbf{X}'$. Combining these bounds with the property that $\sigma_{d+1}(\mathbf{X}) = 0$ for a rank- d matrix leads to the following result. The subspace error, or more precisely, the maximum angle separation between the true subspace basis $\mathbf{U}_{:,1:d}$ and the estimated subspace basis $\hat{\mathbf{U}}_{:,1:d}$ is bounded as follows

$$\begin{aligned} &\mathbb{E}[\|\hat{\mathbf{U}}_{:,1:d}\hat{\mathbf{U}}'_{:,1:d} - \mathbf{U}_{:,1:d}\mathbf{U}'_{:,1:d}\|_2^2] \\ &\lesssim \frac{\left(\sqrt{\nu_{\text{sum}}^{(1)}} + \sqrt{D\nu_{\text{max}}} + \sqrt[4]{3D\nu_{\text{sum}}^{(2)}}\right)^2}{N\sigma_d(\mathbf{C}_{\mathbf{X}})}. \end{aligned} \quad (7)$$

This upper bound indicates that the quality of the subspace basis estimate $\hat{\mathbf{U}}_{:,1:d}$ provided by the SVD of noisy data \mathbf{Y} , i.e., by conventional PCA, could be degraded by heteroscedastic noise. Fig. 8 in the appendix provides empirical evidence for this claim. Thus, it can be advantageous to model the heteroscedasticity and design a more robust PCA-like algorithm that mitigates some of the effects of heteroscedastic noise and achieves more accurate subspace basis estimates.

B. Other Heteroscedastic Models

This paper focuses on heteroscedastic noise across the data samples. There are other methods in the literature that explore

heteroscedasticity in different ways. For example, HeteroPCA considers heteroscedasticity across the feature space [19]. One possible application of that model is for data that consists of sensor information with multiple devices that naturally have different levels of precision and signal to noise ratio (SNR). Another heterogeneity model considers the noise to be homoscedastic and instead assume that the signal itself is heteroscedastic [20]. In that case, the power fluctuating signals are embedded in white Gaussian noise. These different models each have their own families of applications.

C. Probabilistic PCA (PPCA)

PCA methods like PPCA [12] work well in the homoscedastic setting, i.e., when the data is the same quality throughout, but fail to accurately estimate the basis when the data varies in quality, e.g., in the heteroscedastic setting [15].

Let $\mathbf{C} = \mathbf{F}\mathbf{F}' + \nu\mathbf{I}$ and observe that the model

$$\mathbf{y}_i = \mathbf{F}\mathbf{z}_i + \boldsymbol{\epsilon} \quad (8)$$

$$\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \nu\mathbf{I}), \mathbf{y}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \quad (9)$$

is similar to (1) in that we have observation data \mathbf{y}_i , unobserved variables \mathbf{z}_i , factor matrix \mathbf{F} , and noise term $\boldsymbol{\epsilon}$. Then, for a covariance-type matrix $\mathbf{C}_y = \sum_i \mathbf{y}_i \mathbf{y}_i'$ formed from data samples \mathbf{Y} , the negative log-likelihood is

$$\mathcal{L}(\mathbf{F}, \nu) = -\frac{N}{2} (d \log(2\pi) + \log(|\mathbf{C}|) + \text{Tr}(\mathbf{C}^{-1} \mathbf{C}_y)), \quad (10)$$

where $|\cdot|$ and $\text{Tr}(\cdot)$ denote matrix determinant and trace, respectively. After estimating \mathbf{F} and ν by minimizing (10), PPCA finds the subspace basis by orthogonalizing \mathbf{F} . Because $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \nu\mathbf{I})$ is identically distributed across all data samples, PPCA does not account for heterogeneous data samples.

D. Robust PCA (RPCA)

Robust PCA (RPCA) [14] decomposes the data matrix $\mathbf{Y} = \mathbf{X} + \mathbf{E}$ into a low-rank component \mathbf{X} and an outlier matrix \mathbf{E} by the following optimization problem:

$$\arg \min_{\mathbf{X}, \mathbf{E}} (\lambda \|\mathbf{X}\|_* + \|\mathbf{E}\|_{1,1}) \quad \text{s.t. } \mathbf{Y} = \mathbf{X} + \mathbf{E} \quad (11)$$

where $\|\mathbf{X}\|_* = \sum_{i=1} \sigma_i(\mathbf{X})$ and $\|\mathbf{E}\|_{1,1} = \sum_{i,j} |E_{ij}|$. RPCA finds the subspace basis by iteratively applying an SVD to \mathbf{X} to soft threshold the singular values. Here, the term $\|\mathbf{E}\|_{1,1}$ encourages sparsity and so captures noise in the data matrix by assuming there is a sparse collection of outliers. This modeling assumption may not be true in some applications; for instance, low-quality and abundant commercial sensors are often combined with fewer high-quality sensors. Ref. [15] illustrated the limitations of RPCA in the heteroscedastic regime.

E. Weighted PCA (WPCA)

Given data samples $\{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ and weights $\{w_1, \dots, w_N\}$, the weighted PCA (WPCA) approach [21] for modeling

heteroscedastic data forms the following weighted sample covariance matrix

$$\mathbf{C}_y(\mathbf{w}) = \sum_{i=1}^N w_i (\mathbf{y}_i \mathbf{y}_i'), \quad (12)$$

where a natural choice for the weights is $w_i = \nu_i^{-1}$. WPCA finds the subspace basis by orthogonalizing $\mathbf{C}_y(\mathbf{w})$ via eigenvalue decomposition (EVD). However, the noise variances may not be known, e.g., unknown origin of the dataset or unavailable data sheet for physical sensors.

F. Heteroscedastic PPCA Technique (HePPCAT)

To our knowledge, besides ALPCA, there is only one sample-based heteroscedastic PCA algorithm that estimates unknown noise variances. The Heteroscedastic Probabilistic PCA Technique (HePPCAT) [13] builds on the PPCA formulation. For $n_1 + \dots + n_G = N$ data samples from G noise groups, the model is described as

$$\mathbf{y}_{g,i} = \mathbf{F}\mathbf{z}_{g,i} + \boldsymbol{\epsilon}_{g,i} \quad i \in \{1, \dots, n_G\}, g \in \{1, \dots, G\} \quad (13)$$

for factor scores $\mathbf{z}_{g,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, noise terms $\boldsymbol{\epsilon}_{g,i} \sim \mathcal{N}(\mathbf{0}, v_g \mathbf{I})$, and points $\mathbf{y}_{g,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_g)$ where $\mathbf{C}_g = \mathbf{F}\mathbf{F}' + v_g \mathbf{I}$ for factor matrix \mathbf{F} . Then, the negative log-likelihood model to optimize is the following

$$\mathcal{L}(\mathbf{F}, \mathbf{v}) = \frac{1}{2} \sum_{g=1}^G [n_g \ln \det(\mathbf{C}_g)^{-1} - \text{Tr}\{\mathbf{Y}_{(g)}^T (\mathbf{C}_g)^{-1} \mathbf{Y}_{(g)}\}] \quad (14)$$

where $\mathbf{Y}_{(g)}$ denotes the submatrix of \mathbf{Y} that consists only of data samples belonging to the g th noise group, and $\mathbf{v} = (v_1, \dots, v_G)$ denotes the unknown noise variances for each group. Being a factor analysis method, HePPCAT makes Gaussian assumptions about the basis coefficients $\mathbf{z}_{l,i}$ that may not be a good model for some datasets. Additionally, HePPCAT requires the rank parameter d associated with the latent signal matrix \mathbf{X} to be estimated or known a priori.

III. PROPOSED SUBSPACE LEARNING METHODS

This section introduces the ALPCA formulation for subspace learning. Since nuclear norm computation is expensive for big data applications due to SVD computations, we take inspiration from the matrix factorization literature and additionally develop LR-ALPCA to be a fast and memory efficient alternative to ALPCA.

A. ALPCA

For the measurement model $\mathbf{y}_i \sim \mathcal{N}(\mathbf{x}_i, \nu_i \mathbf{I})$ in (1), the probability density function for a single data vector \mathbf{y}_i is

$$\frac{1}{\sqrt{(2\pi)^D |\nu_i \mathbf{I}|}} \exp \left[-\frac{1}{2} (\mathbf{y}_i - \mathbf{x}_i)' (\nu_i \mathbf{I})^{-1} (\mathbf{y}_i - \mathbf{x}_i) \right]. \quad (15)$$

For uncorrelated samples, after dropping constants, the joint log likelihood of all data $\{\mathbf{y}_i\}_{i=1}^N$ is the following

$$\sum_{i=1}^N -\frac{1}{2} \log |\nu_i \mathbf{I}| - \frac{1}{2} (\mathbf{y}_i - \mathbf{x}_i)' (\nu_i \mathbf{I})^{-1} (\mathbf{y}_i - \mathbf{x}_i). \quad (16)$$

Let $\mathbf{\Pi} = \text{diag}(\nu_1, \dots, \nu_N) \in \mathbb{R}^{N \times N}$ be a diagonal matrix representing the (typically unknown) noise variances. Then, the negative log likelihood in matrix form is

$$\begin{aligned} & \frac{D}{2} \log |\mathbf{\Pi}| + \frac{1}{2} \text{Tr}[(\mathbf{Y} - \mathbf{X})\mathbf{\Pi}^{-1}(\mathbf{Y} - \mathbf{X})'] \\ &= \frac{D}{2} \log |\mathbf{\Pi}| + \frac{1}{2} \|(\mathbf{Y} - \mathbf{X})\mathbf{\Pi}^{-1/2}\|_{\text{F}}^2, \end{aligned} \quad (17)$$

using trace lemmas. When both $\mathbf{\Pi}$ and \mathbf{X} are unknown, pursuing maximum-likelihood estimation with (17) would lead to degenerate solutions. Thus, regularization is necessary to promote a low-rank solution. In this work, we use a functional modified from the nuclear norm regularizer to encourage the estimate of \mathbf{X} to be low rank.

The optimization problem used by ALPCAH for the heteroscedastic model is

$$\arg \min_{\mathbf{X}, \mathbf{\Pi}} \lambda f_{\hat{d}}(\mathbf{X}) + \frac{1}{2} \|(\mathbf{Y} - \mathbf{X})\mathbf{\Pi}^{-1/2}\|_{\text{F}}^2 + \frac{D}{2} \log |\mathbf{\Pi}|, \quad (18)$$

where $f_{\hat{d}}(\mathbf{X})$ is a novel functional [22] that promotes low-rank structure in \mathbf{X} , \hat{d} is the rank parameter, and $\lambda \in \mathbb{R}^+$ is a regularization parameter. In the following, we introduce our algorithm called ALPCAH (Algorithm for Low-rank regularized PCA for Heteroscedastic data) for solving (18). Since \mathbf{X} represents the denoised data matrix, the subspace basis is calculated by SVD on the optimal solution from (18) and extracting the first \hat{d} left singular vectors so that $\hat{\mathbf{X}} = \sum_i \hat{\sigma}_i \hat{\mathbf{u}}_i \hat{\mathbf{v}}_i'$ and thus $\hat{\mathbf{U}} = [\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_{\hat{d}}]$. The low-rank promoting functional we use is the sum of the tail singular values defined as

$$f_{\hat{d}}(\mathbf{X}) \triangleq \sum_{i=\hat{d}+1}^{\min(D, N)} \sigma_i(\mathbf{X}) = \|\mathbf{X}\|_* - \|\mathbf{X}\|_{\text{Ky-Fan}(\hat{d})} \quad (19)$$

where $\|\cdot\|_*$ denotes the nuclear norm, and $\|\cdot\|_{\text{Ky-Fan}(\hat{d})}$ denotes the Ky-Fan norm [23] defined as the sum of the first \hat{d} singular values. For $\hat{d} = 0$, $f_0(\mathbf{X}) = \|\mathbf{X}\|_*$. For a general $\hat{d} > 0$, $f_{\hat{d}}(\mathbf{X})$ is a nonconvex difference of convex functions. We use the functional $f_{\hat{d}}$ instead of the nuclear norm since we empirically found that the nuclear norm tends to over shrink the singular values of \mathbf{X} in the heteroscedastic setting. Here, the rank parameter $\hat{d} \ll D$ is either known beforehand, estimated using methods like row permutations [24] or sign flips [25], or intentionally over-parameterized.

Definition 1: Let $\mathbf{A} \in \mathbb{R}^{D \times N}$ be a rank k matrix such that its decomposition is $\text{SVD}(\mathbf{A}) = \mathbf{U}_A \mathbf{D}_A \mathbf{V}'_A$ where $\mathbf{D}_A = \text{diag}(\sigma_1(\mathbf{A}), \dots, \sigma_{\min(D, N)}(\mathbf{A}))$. Let the soft thresholding operation be defined as $\mathcal{S}_\tau[x] = \text{sign}(x) \max(|x| - \tau, 0)$ for some threshold $\tau > 0$. Decompose \mathbf{D}_A such that $\mathbf{D}_A = \mathbf{D}_{A1} + \mathbf{D}_{A2} = \text{diag}(\sigma_1(\mathbf{A}), \dots, \sigma_{\hat{d}}(\mathbf{A}), 0, \dots, 0) + \text{diag}(0, \dots, 0,$

$\sigma_{\hat{d}+1}(\mathbf{A}), \dots, \sigma_N(\mathbf{A}))$. Then, the proximal map for $f_{\hat{d}}$ is the tail singular value thresholding operation [22]:

$$\text{TSVT}(\mathbf{A}, \tau, \hat{d}) \triangleq \mathbf{U}_A (\mathbf{D}_{A1} + \mathcal{S}_\tau[\mathbf{D}_{A2}]) \mathbf{V}'_A. \quad (20)$$

Although the proximal operator for $f_{\hat{d}}$ is provided in (20), it is unclear how one would apply a proximal gradient method (PGM) directly to (18) due to the product of \mathbf{X} and $\mathbf{\Pi}$. One could apply a block coordinate decent approach that alternates between updating \mathbf{X} using a PGM, and updates the diagonal elements of $\mathbf{\Pi}$ using a closed-form solution. The PGM update of \mathbf{X} could cause slow convergence because the Lipschitz constant of the gradient of the smooth term $g(\mathbf{X})$ is the reciprocal of the smallest diagonal element of $\mathbf{\Pi}$, which could be quite large, leading to small step sizes. Thus, instead we optimize (18) using the inexact augmented Lagrangian method known as the alternating direction method of multipliers (ADMM) [26] that introduces auxiliary variables to convert a complicated optimization problem into a sequence of simpler optimization problems.

Defining the auxiliary variable $\mathbf{Z} = \mathbf{Y} - \mathbf{X}$, the augmented penalty parameter $\mu \in \mathbb{R}$, and dual variable $\mathbf{\Lambda} \in \mathbb{R}^{D \times N}$, the augmented Lagrangian, as defined in [27], is

$$\begin{aligned} \mathcal{L}_\mu(\mathbf{X}, \mathbf{Z}, \mathbf{\Lambda}, \mathbf{\Pi}) &= \lambda f_{\hat{d}}(\mathbf{X}) + \frac{1}{2} \|\mathbf{Z}\mathbf{\Pi}^{-1/2}\|_{\text{F}}^2 + \frac{D}{2} \log |\mathbf{\Pi}| \\ &+ \langle \mathbf{\Lambda}, \mathbf{Y} - \mathbf{X} - \mathbf{Z} \rangle + \frac{\mu}{2} \|\mathbf{Y} - \mathbf{X} - \mathbf{Z}\|_{\text{F}}^2, \end{aligned} \quad (21)$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product between two matrices.

Performing a block Gauss-Seidel pass for each variable in (21) results in the following closed-form updates

$$\begin{aligned} \mathbf{Z}_{t+1} &= \arg \min_{\mathbf{Z}} \mathcal{L}_\mu(\mathbf{X}_t, \mathbf{Z}, \mathbf{\Lambda}_t, \mathbf{\Pi}_t) \\ &= [\mu(\mathbf{Y} - \mathbf{X}_t) + \mathbf{\Lambda}_t] (\mathbf{\Pi}_t^{-1} + \mu \mathbf{I})^{-1} \end{aligned} \quad (22)$$

$$\begin{aligned} \mathbf{X}_{t+1} &= \arg \min_{\mathbf{X}} \mathcal{L}_\mu(\mathbf{X}, \mathbf{Z}_t, \mathbf{\Lambda}_t, \mathbf{\Pi}_t) \\ &= \text{TSVT} \left(\mathbf{Y} - \mathbf{Z}_t + \frac{1}{\mu} \mathbf{\Lambda}_t, \frac{\lambda}{\mu}, \hat{d} \right) \end{aligned} \quad (23)$$

$$\mathbf{\Lambda}_{t+1} = \mathbf{\Lambda}_t + \mu (\mathbf{Y} - \mathbf{X}_t - \mathbf{Z}_t) \quad (24)$$

for current iteration pass t . Each pass is run for T total iterations. When we treat data sample \mathbf{y}_i as having its own unknown noise variance, then the variance update is

$$\mathbf{\Pi}_{t+1} = \arg \min_{\mathbf{\Pi}} \mathcal{L}_\mu(\mathbf{X}_t, \mathbf{Z}_t, \mathbf{\Lambda}_t, \mathbf{\Pi}) = \frac{1}{D} \mathbf{Z}'_t \mathbf{Z}_t \odot \mathbf{I}. \quad (25)$$

For the case when the data points have grouped noise variances, let $g \in \{1, \dots, G\}$ signify the g th noise group out of G total groups with n_g denoting the number of samples in the g th group; then the grouped noise variance update instead becomes

$$\nu_g = \frac{1}{D n_g} \|\mathbf{Z}_{(g)}\|_{\text{F}}^2 = \frac{1}{D n_g} \|\mathbf{Y}_{(g)} - \mathbf{X}_{(g)}\|_{\text{F}}^2 \quad (26)$$

where the notation $\mathbf{Y}_{(g)}$ denotes the submatrix of \mathbf{Y} that consists only of data samples belonging to the g th noise group.

1) *Convergence With Known Variance:* Consider the cost function for the case when the variances $\mathbf{\Pi}$ are known. The formulation consists of a two-block setup written as

$$\arg \min_{\mathbf{X}, \mathbf{Z}} \lambda f_{\hat{d}}(\mathbf{X}) + \frac{1}{2} \|\mathbf{Z}\mathbf{\Pi}^{-1/2}\|_{\text{F}}^2 \quad \text{s.t.} \quad \mathbf{Y} = \mathbf{X} + \mathbf{Z}. \quad (27)$$

Theorem 1: Let $\Psi(\mathbf{X}, \mathbf{Z}) = f(\mathbf{X}) + g(\mathbf{Z})$ where $f(\mathbf{X}) = \lambda f_{\hat{d}}(\mathbf{X})$ and $g(\mathbf{Z}) = \frac{1}{2} \|\mathbf{Z}\mathbf{\Pi}^{-1/2}\|_{\text{F}}^2$. Let $\nu_i \geq \epsilon > 0 \quad \forall i$. Assuming that μ in (21) satisfies $\mu > 2L_g = 2\|\mathbf{\Pi}^{-1}\|_2$, the sequence $\{(\mathbf{X}_t, \mathbf{Z}_t, \mathbf{\Lambda}_t, \mathbf{\Pi})\}_{i=1}^T$ generated by ADMM in (22) (23) (24) (25) converges to a KKT (Karush–Kuhn–Tucker) point of the augmented Lagrangian $\mathcal{L}_\mu(\mathbf{X}, \mathbf{Z}, \mathbf{\Lambda}, \mathbf{\Pi})$ with fixed $\mathbf{\Pi}$.

Proof: ADMM convergence for nonconvex problems has been explored for two-block setups [28]. The functional $f(\mathbf{X})$ is a proper, lower semi-continuous function since it is a sum of continuous functions. The function $g(\mathbf{Z})$ is a continuous differentiable function whose gradient is Lipschitz continuous with modulus of continuity $L_g = \|\mathbf{\Pi}^{-1}\|_2$. By definition $g(\mathbf{Z}) = \nu_1^{-1/2} Z_{11} + \nu_1^{-1/2} Z_{21} + \dots + \nu_N^{-1/2} Z_{DN}$. Since $g(\mathbf{Z})$ is a polynomial equation, its graph is a semi-algebraic set.

Let $\mathbf{G} = \mathbf{X}'\mathbf{X} \in \mathbb{R}^{N \times N}$. Then, by Cayley Hamilton theorem, the characteristic polynomial of \mathbf{G} is $p_G(z) = z^N + c_{n-1}(\mathbf{G})z^{N-1} + \dots + c_1(\mathbf{G})z + c_0$ for constants $c_i \in \mathbb{R}$ and polynomial degree N . Let λ denote an eigenvalue of \mathbf{G} which implies $p_G(\lambda) = 0$. Then, the set $S_G = \{\lambda \mid p_G(\lambda) = 0\}$ is semi-algebraic since it is defined by polynomial equations. Note that $\lambda_i(\mathbf{G}) = \sigma_i^2(\mathbf{X})$ since \mathbf{G} is the Gram matrix of \mathbf{X} . The set $S_X = \{\sigma \mid \sigma^2 = \lambda \in S_G, \sigma \geq 0\} = \{\sigma_1, \dots, \sigma_N\}$ is semi-algebraic as it is expressed in terms of polynomial inequalities. Expressing $h(\mathbf{X}) = \|\mathbf{X}\|_* = h(\sigma_1, \dots, \sigma_N)$, its graph $h = \{(\sigma, f(\sigma))\}$ is semi-algebraic and thus by extension so is the nuclear norm.

By Tarski-Seidenburg theorem [29, p. 345], defining the map $\Phi: \mathbb{R}^n \rightarrow \mathbb{R}^{\hat{d}}$ that retains the first \hat{d} singular values of S_X , the set $\Phi(S_X) = \{\sigma_1, \dots, \sigma_{\hat{d}}\}$ is semi-algebraic and thus so is $q(\mathbf{X}) = \|\mathbf{X}\|_{\text{Ky-Fan}(\hat{d})}$. A finite weighted sum of semi-algebraic functions is known to be semi-algebraic [30] and so $f(\mathbf{X}) = h(\mathbf{X}) - q(\mathbf{X})$ is semi-algebraic. Since the functions $f(\mathbf{X})$ and $g(\mathbf{Z})$ are lower, semi-continuous and definable on an o-minimal structure such as semi-algebraic [31], it follows that $\Psi(\mathbf{X}, \mathbf{Z}) = f(\mathbf{X}) + g(\mathbf{Z})$ is a Kurdyka-Lojasiewicz function [30]. Thus the sequence $\{(\mathbf{X}_t, \mathbf{Z}_t, \mathbf{\Lambda}_t, \mathbf{\Pi})\}_{i \in \mathbb{N}}$ converges to a KKT point by [28, Thm. 3.1]. \square

B. LR-ALPCAH

The main compute expense for the ALPCAH algorithm is the SVD operations used in every iteration of complexity $\mathcal{O}(DN \min(D, N))$. To reduce computation, we take inspiration from the matrix factorization literature [32] and factorize $\mathbf{X} \in \mathbb{R}^{D \times N} \approx \mathbf{L}\mathbf{R}'$ where $\mathbf{L} \in \mathbb{R}^{D \times \hat{d}}$ and $\mathbf{R} \in \mathbb{R}^{N \times \hat{d}}$ for some

rank estimate \hat{d} . Using the factorized form, we propose to estimate \mathbf{X} by solving for \mathbf{L} and \mathbf{R} in the following optimization problem

$$\hat{\mathbf{L}}, \hat{\mathbf{R}}, \hat{\mathbf{\Pi}} = \arg \min_{\mathbf{L}, \mathbf{R}, \mathbf{\Pi}} f(\mathbf{L}, \mathbf{R}, \mathbf{\Pi})$$

$$f(\mathbf{L}, \mathbf{R}, \mathbf{\Pi}) = \frac{1}{2} \|\mathbf{Y} - \mathbf{L}\mathbf{R}'\|_{\text{F}}^2 + \frac{D}{2} \log |\mathbf{\Pi}|. \quad (28)$$

This version is a maximum-likelihood estimator of $\mathbf{\Pi}$ and the factors \mathbf{L} and \mathbf{R} . This comes from a modified model (1) where

$$\mathbf{y}_i = \mathbf{L}\mathbf{r}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \nu_i \mathbf{I}), \quad (29)$$

where \mathbf{r}_i denotes the i th column of \mathbf{R} . We call this version LR-ALPCAH given the prevalence of $\mathbf{L}\mathbf{R}'$ notation in the matrix factorization literature. The crucial difference between ALPCAH and LR-ALPCAH is that ALPCAH uses a ‘‘soft’’ low rank constraint through the regularization penalty λ with optional usage of \hat{d} , whereas LR-ALPCAH uses a ‘‘hard’’ low rank constraint since \mathbf{L} and \mathbf{R} rigidly contain \hat{d} columns.

We solve this optimization problem using alternating minimization [33] to solve each sub-block, resulting in the following updates:

$$\mathbf{L}_{t+1} = \arg \min_{\mathbf{L}} f(\mathbf{L}, \mathbf{R}_t, \mathbf{\Pi}_t)$$

$$= \mathbf{Y}\mathbf{\Pi}_t \mathbf{R}_t (\mathbf{R}_t' \mathbf{\Pi}_t \mathbf{R}_t)^{-1} \quad (30)$$

$$\mathbf{R}_{t+1} = \arg \min_{\mathbf{R}} f(\mathbf{L}_t, \mathbf{R}, \mathbf{\Pi}_t)$$

$$= \mathbf{Y}' \mathbf{L}_t (\mathbf{L}_t' \mathbf{L}_t)^{-1} \quad (31)$$

$$\mathbf{\Pi}_{t+1} = \arg \min_{\mathbf{\Pi}} f(\mathbf{L}_t, \mathbf{R}_t, \mathbf{\Pi}) \implies$$

$$\mathbf{e}'_j \mathbf{\Pi}_{t+1} \mathbf{e}_j = D^{-1} \|(\mathbf{Y} - \mathbf{L}_t \mathbf{R}'_t) \mathbf{e}_j\|_2^2, \quad \forall j, \quad (32)$$

where \mathbf{e}_j denotes the j th standard canonical basis vector that we use to select the j th column of some matrix. The $\mathbf{\Pi}_t$ update (32) is the same as (25) in that each point is treated as having its own noise variance and both equations perform the same operation. This implementation requires less computation and memory since the matrix $\mathbf{Z}'_t \mathbf{Z}_t$ is not formed. One can substitute (32) with (26) if noise grouping is known.

Since this is a nonconvex problem, initialization will play a key role in the success of optimization. First, we initialize the \mathbf{L}_t and \mathbf{R}_t matrices with the following spectral approach:

$$\text{Spectral Init}(\mathbf{Y}) = \hat{\mathbf{U}} \hat{\boldsymbol{\Sigma}} \hat{\mathbf{V}}' \approx (\hat{\mathbf{U}}_{1:\hat{d}} \hat{\boldsymbol{\Sigma}}_{1:\hat{d}}^{1/2}) (\hat{\boldsymbol{\Sigma}}_{1:\hat{d}}^{1/2} \hat{\mathbf{V}}'_{1:\hat{d}})$$

$$= (\mathbf{L}_0) (\mathbf{R}'_0). \quad (33)$$

This initialization from the matrix factorization literature [16] [34] is a natural approach due to the Eckart-Young theorem [35] that shows \mathbf{L}_0 and \mathbf{R}_0 are the best rank-constrained matrices that solve

$$\arg \min_{\mathbf{L}, \mathbf{R}} \|\mathbf{Y} - \mathbf{L}\mathbf{R}'\|_{\text{F}} \quad \text{subject to} \quad \text{rank}(\mathbf{L}), \text{rank}(\mathbf{R}) \leq \hat{d}. \quad (34)$$

Second, we initialize the noise variances using the Euclidean norms of the columns of the residual $\mathbf{Y} - \mathbf{L}_0 \mathbf{R}'_0$ with (32).

Algorithm LR-ALPCA (github.com/javiersc1/ALPCA)
(unknown variances, unknown quality noise grouping)

Input: $\mathbf{Y} \in \mathbb{R}^{D \times N}$: data, $\hat{d} \in \mathbb{N}^*$: rank estimate
Opt: $T \in \mathbb{N}^*$: iterations, $\epsilon \in \mathbb{R}^+$: variance noise floor
Output: $\mathbf{U} \in \mathbb{R}^{D \times \hat{d}}$: subspace basis, $\mathbf{X} \in \mathbb{R}^{D \times N}$: low-rank estimated data, $\boldsymbol{\nu} \in \mathbb{R}^+$: estimated noise variances
// sample mean to de-mean data
 $\boldsymbol{\mu} \leftarrow \frac{1}{N} \mathbf{Y} \mathbf{1}$
// the method assumes linear subspaces only
 $\mathbf{Y} \leftarrow \mathbf{Y} - \mathbf{1}' \boldsymbol{\mu}$
// initialize matrices by (33)
 $\mathbf{L}_0, \mathbf{R}_0 \leftarrow \text{SPECTRALINIT}(\mathbf{Y}, \hat{d})$
// compute noise variances from residuals $\mathbf{Y} - \mathbf{L}_0 \mathbf{R}_0'$
// \mathbf{e}_j is canonical basis vector
 $\nu_0 \leftarrow \max_{j=1, \dots, N} \frac{1}{D} \|\mathbf{Y} - \mathbf{L}_0 \mathbf{R}_0'\|_{\mathbf{e}_j}^2$
 $\nu_0 \leftarrow \max(\nu_0, \epsilon)$
 $\boldsymbol{\Pi}_0^{-1} \leftarrow (1/\nu_0) \mathbf{I}$
// update $\mathbf{L}, \mathbf{R}, \boldsymbol{\Pi}$ matrices using (30) (31) (32)
for $t = 1, \dots, T$ **do**
 $\mathbf{L}_t \leftarrow \mathbf{Y} \boldsymbol{\Pi}_{t-1}^{-1} \mathbf{R}_{t-1} (\mathbf{R}_{t-1}' \boldsymbol{\Pi}_{t-1}^{-1} \mathbf{R}_{t-1})^{-1}$
 $\mathbf{R}_t \leftarrow \mathbf{Y}_{t-1}' \mathbf{L}_{t-1} (\mathbf{L}_{t-1}' \mathbf{L}_{t-1})^{-1}$
 $\nu_j \leftarrow \max(\frac{1}{D} \|\mathbf{Y} - \mathbf{L}_{t-1} \mathbf{R}_{t-1}'\|_{\mathbf{e}_j}^2, \epsilon), j = 1, \dots, N$
 $\boldsymbol{\Pi}_t^{-1} \leftarrow \text{Diagonal}(1/\nu_j)$
end for
// form subspace basis from final left factor
 $\mathbf{U} \leftarrow \text{GRAMSCHMIDT}(\mathbf{L}_T)$
// construct de-meaned low-rank estimate
 $\mathbf{X} \leftarrow \mathbf{L}_T \mathbf{R}_T'$
// add back original sample mean
 $\mathbf{X} \leftarrow \mathbf{X} + \mathbf{1}' \boldsymbol{\mu}$

Finally, we apply alternating minimization to update \mathbf{L}_t and \mathbf{R}_t matrices at current iteration t via (30) (31) (32).

Since \mathbf{L}_t is not semi-unitary but has the same range as \mathbf{U} , we apply Gram-Schmidt orthonormalization to the final \mathbf{L}_t matrix to estimate the subspace basis, as described in Alg. LR-ALPCA. The matrix inversions used in the \mathbf{L}_t and \mathbf{R}_t updates involve $d \times d$ matrices that are relatively small and thus computationally feasible for many practical problems given complexity $\mathcal{O}(\hat{d}^3)$ knowing that $\hat{d} \ll \min(D, N)$. Combining the matrix multiplications and inversions, LR-ALPCA has a per-iteration complexity of $\mathcal{O}(DN\hat{d} + \hat{d}^3)$. This is in contrast to ALPCA with per-iteration complexity $\mathcal{O}(DN \min(D, N))$ due to the SVD computations.

1) *Convergence With Unknown Variance:* Note that (28) is a nonconvex function and we apply alternating minimization, also known as block coordinate descent or block nonlinear Gauss-Seidel method, to solve the optimization problem. Given a noise variance lower bound $\epsilon > 0$, the feasible sets for $\mathbf{L}, \mathbf{R}, \boldsymbol{\Pi}$ variables are given by

$$\mathcal{S}_L = \mathbb{R}^{D \times \hat{d}}, \quad \mathcal{S}_R = \mathbb{R}^{N \times \hat{d}} \quad (35)$$

$$\mathcal{S}_{\boldsymbol{\Pi}} = \{\boldsymbol{\Pi}_{i,j} \in [\epsilon, \infty) \quad \forall i = j, 0 \text{ o.w.}\}. \quad (36)$$

Given the following optimization problem with

$$\begin{aligned} & \arg \min f(\mathbf{L}, \mathbf{R}, \boldsymbol{\Pi}) \\ & \text{subject to } \mathbf{L}, \mathbf{R}, \boldsymbol{\Pi} \in \mathcal{S} = \mathcal{S}_L \times \mathcal{S}_R \times \mathcal{S}_{\boldsymbol{\Pi}}, \end{aligned} \quad (37)$$

the following theorem establishes local convergence of $\{(\mathbf{L}_t, \mathbf{R}_t, \boldsymbol{\Pi}_t)\}_{t=1}^T$ to critical points of (37).

Theorem 2: The sequence generated by alternating minimization $\{(\mathbf{L}_t, \mathbf{R}_t, \boldsymbol{\Pi}_t)\}_{t=1}^T$ in Alg. LR-ALPCA has limit points that are critical points of (37).

Proof: The cost function f is a continuously differentiable function by inspection of both terms. The feasible sets $\mathcal{S}_L, \mathcal{S}_R$ are trivially nonempty, closed, and convex sets by definition. Moreover, $\boldsymbol{\Pi} \in \mathcal{S}_{\boldsymbol{\Pi}}$ since it is an enforced constraint of the optimization in (37). The function f is componentwise strictly quasiconvex with respect to the two blocks \mathbf{L} and \mathbf{R} . This is because $f(\mathbf{L}, \mathbf{R}, \boldsymbol{\Pi})$ w.r.t. \mathbf{L} and $f(\mathbf{L}, \mathbf{R}, \boldsymbol{\Pi})$ w.r.t. \mathbf{R} are convex terms it follows that they are pseudoconvex functions [36] and this implies they are also strict quasiconvex functions [36]. It then follows from [37, Prop. 5] that the sequence generated by alternating minimization $\{(\mathbf{L}_t, \mathbf{R}_t, \boldsymbol{\Pi}_t)\}_{t=1}^T$ converges to limit points that are also critical points of (37). \square

IV. EMPIRICAL RESULTS & DISCUSSION

This section summarizes synthetic and real data experiments, including astronomy spectra and RNA sequencing data, that explore various aspects of subspace learning from heteroscedastic data.

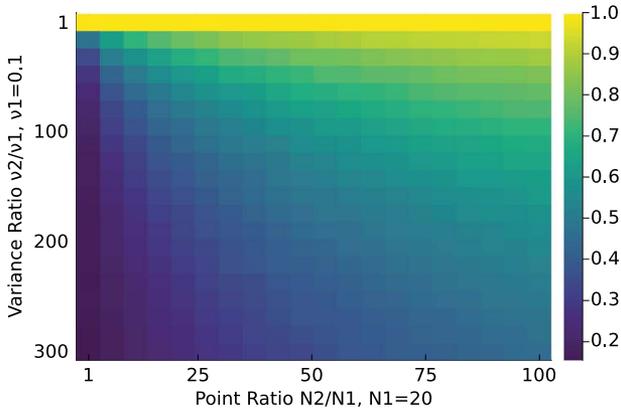
A. Synthetic Experiments

This section uses synthetic data to compare LR-ALPCA with other methods. We begin by describing the experimental setup, followed by an investigation of PCA, and after that compare to RPCA, HePPCAT, and WPCA.

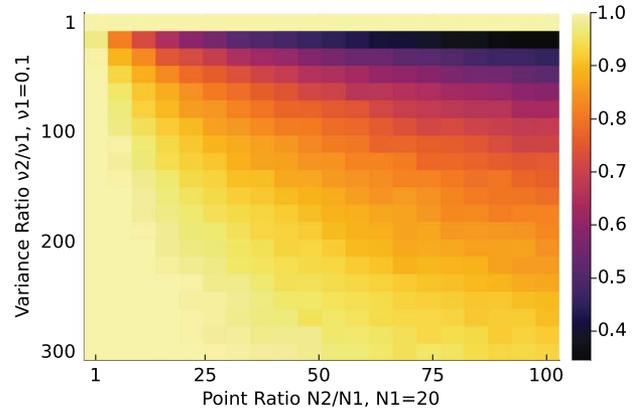
a) *Experimental Setup:* We consider two groups of data, one with fixed quality, meaning fixed size and additive noise variance, and one whose parameters we vary. Let $\mathbf{y}_i \in \mathbb{R}^{100}$ be $D = 100$ dimensional ambient-space data. Let $\mathbf{U} \in \mathbb{R}^{100 \times 5}$ denote a basis for a $d = 5$ dimensional subspace generated by random uniform matrices such that $\mathbf{U} \boldsymbol{\Sigma} \mathbf{V}' = \text{svd}(\mathbf{A})$, where $A_{i,j} \sim \mathcal{U}[0, 1]$. We use the compact SVD here. The low-rank data is simulated as $\mathbf{x}_i = \mathbf{U} \mathbf{z}_i$ where the coordinates $\mathbf{z}_i \in \mathbb{R}^5$ were generated from $\mathcal{U}[-100, 100]$ for each element. Then, we generated $\mathbf{y}_i = \mathbf{U} \mathbf{z}_i + \boldsymbol{\epsilon}_i$ where $\boldsymbol{\epsilon}_i \in \mathbb{R}^{100}$ was drawn from $\mathcal{N}(\mathbf{0}, \nu_i \mathbf{I})$. The error metric used is subspace affinity error (SAE) that compares the difference in projection matrices

$$\text{SAE}(\mathbf{U}, \hat{\mathbf{U}}) = \|\mathbf{U} \mathbf{U}' - \hat{\mathbf{U}} \hat{\mathbf{U}}'\|_{\mathbb{F}} / \|\mathbf{U} \mathbf{U}'\|_{\mathbb{F}} \quad (38)$$

so that a low error signifies a closer estimate of the true subspace. This metric is also known as normalized chordal distance [38]. In summary, the noisy data $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$ is generated accordingly, an estimate $\hat{\mathbf{X}}$ is generated from (1), the subspace basis is calculated by $\hat{\mathbf{X}} = \sum_i \hat{\sigma}_i \hat{\mathbf{u}}_i \hat{\mathbf{v}}_i' \implies \hat{\mathbf{U}} = [\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_d]$, and we report the subspace affinity error.



(a) Ratio of subspace affinity errors LR-ALPCA/PCA (known variance, no cross-validation required)



(b) Ratio of subspace affinity errors LR-ALPCA/PCA-GOOD (PCA using good data only and LR-ALPCA using all of the data)

Fig. 2. Subspace affinity error $\|UU' - \hat{U}\hat{U}'\|_F / \|UU'\|_F$ performance of LR-ALPCA compared to PCA.

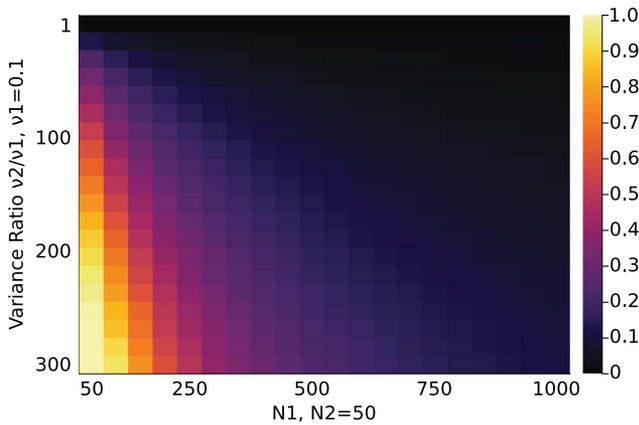


Fig. 3. Absolute difference of LR-ALPCA subspace error subtracted from PCA while good data amount varies.

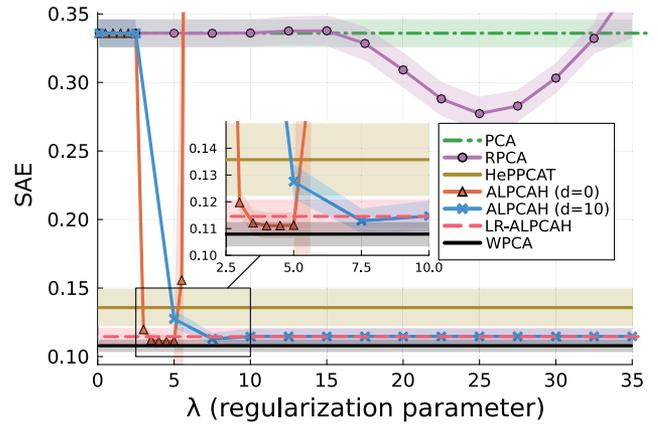


Fig. 4. Absolute subspace quality performance of ALPCA compared against other methods. Zoomed-in areas shown within plots for better visibility for certain λ ranges.

b) Subspace Basis Estimation (LR-ALPCA vs. PCA):

We explored the effects of data quality and data quantity on the heteroscedastic subspace basis estimates in different situations. For the heatmaps in Fig. 2, we focused on comparing LR-ALPCA with PCA only to discuss this method in the general context of subspace learning. In Fig. 2, each pixel represents the ratio $SAE(U, \hat{U}_{LR-ALPCA}) / SAE(U, \hat{U}_{PCA})$. A value close to 1 implies LR-ALPCA did not perform much better than the other method, whereas a ratio closer to 0 implies LR-ALPCA performed relatively well. The average SAE ratio of 50 trials is used, where each trial has different noise, basis coefficients, and subspace basis realizations. The noise variance for group 1 is fixed to $\nu_1 = 0.1$ with $N_1 = 20$ point samples. We varied group 2 point samples N_2 and noise variances ν_2 , as illustrated in the x-axis and y-axis respectively for the heatmaps shown.

Fig. 2(a) compares LR-ALPCA against PCA in the situation where noise variances are known. In this case, LR-ALPCA performs well relative to PCA in noisy situations and can improve estimation, especially in extreme heteroscedastic regions. From the bottom left corner and moving rightwards, the estimation error worsened as the number of noisy points

increased. To clarify, LR-ALPCA never performed worse than PCA, only that the advantage gap decreased as more noisy samples were added. This means that, in LR-ALPCA, the noisy points may have contributed too much to the estimation process when the good quality data should have more influence in the process. For these results, we used the inverse noise variances as the weighing scheme as this is a natural choice that arises from the Gaussian likelihood. However, finding the optimal scheme to mitigate this worsening effect is a topic of future work.

Fig. 2(b) is similar to Fig. 2(a) but only using the high quality points for PCA specifically, whereas LR-ALPCA used all of the data. One can see that even when there was enough good data, there was still an improvement relative to applying PCA to just the good data alone. The improvement increased as more noisy points were added. Thus, it is beneficial to collect and use all of the data, since the noisy points offer meaningful information that can improve the estimate of the basis versus using good data alone, especially in data-constrained situations.

c) Effects of Good Data: Fig. 3 explores how the number of good data samples affects subspace learning quality.

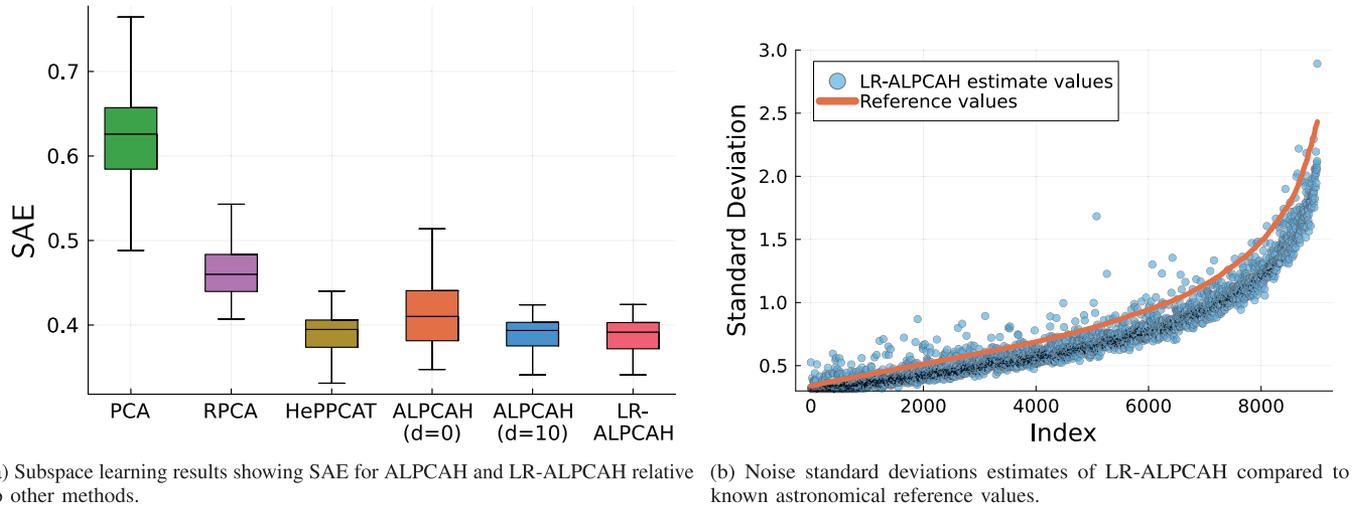


Fig. 5. Experimental results of quasar flux data for subspace learning and noise sample estimation.

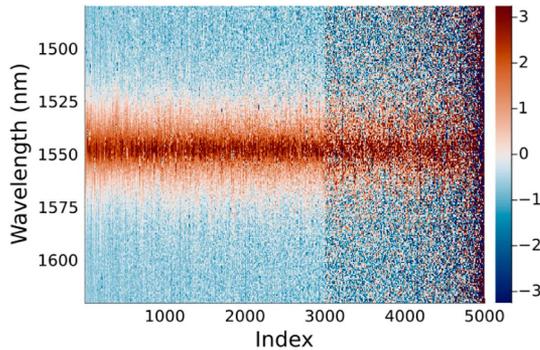


Fig. 6. Sample data matrix of quasar flux measurements across wavelengths for each (column-wise) sample.

We fixed $N_2 = 50$ and varied N_1 while keeping $\nu_1 = 0.1$ and varying ν_2 . This figure plots the difference $\text{SAE}(\mathbf{U}, \hat{\mathbf{U}}_{\text{PCA}}) - \text{SAE}(\mathbf{U}, \hat{\mathbf{U}}_{\text{LR-ALPCA}})$ to see when it is advantageous to use LR-ALPCA instead of PCA. In the absolute sense, both methods performed similarly when good data is abundant. However, when good data was more limited, there were larger differences in subspace quality, meaning it is more advantageous to use LR-ALPCA.

d) Absolute Subspace Error: In this section, we discuss absolute error of the algorithms in the unknown noise variance setting without group knowledge. For Fig. 4, we fixed $N_1 = 50$, $N_2 = 450$ that have noise variances $\nu_1 = 0.25$, $\nu_2 = 100$. The regularization parameter λ is varied (ALPCA & RPCA only) and subspace dimension is $d = 10$. As before, we use the subspace affinity error $\text{SAE}(\mathbf{U}, \hat{\mathbf{U}})$. The average error is plotted out of 50 trials with standard deviation bounds for each λ value. Fig. 4 represents the unknown variance case but we again use WPCA with weights $w_i = \nu_i^{-1}$, a known variance method, to illustrate the lowest possible affinity error if one hypothetically knew the noise variances.

In Fig. 4, when using rank knowledge, ALPCA ($\hat{d} = 10$) approaches the error of the other methods as λ grows. When

not using rank knowledge, for ALPCA ($\hat{d} = 0$), the method can perform just as well but requires cross-validation to find an ideal λ range. Both ALPCA ($\hat{d} = 10$) and LR-ALPCA achieved lower error than HePPCAT, likely because there are no distributional assumptions on the basis coefficients with ALPCA/LR-ALPCA. The RPCA method did not perform well in these experiments, likely because of the model mismatch between the heteroscedastic data and the outlier assumption of RPCA. Excluding the case when rank knowledge is not known, ALPCA ($\hat{d} = 0$), the regularization parameter appears to be robust to this landscape of different variance and point ratios.

B. Real Data Experiments

1) Astronomy Spectra Data: We investigated quasar spectra data from the Sloan Digital Sky Survey (SDSS) Data Release 16 [39] using its DR16Q quasar catalog [40]. Each quasar has a vector of flux measurements across wavelengths that describes the intensity of observing that particular wavelength. In this dataset, the noise is heteroscedastic across the sample space (quasars) and feature space (wavelength), but we focused on a subset of data that is homoscedastic across wavelengths and heteroscedastic across quasars. The noise for each quasar is known given the measurement devices used for data collection [39], but we performed estimation as if the variances were unknown so that we could compare the estimated values to the reference values. We preprocessed the data (filtering, interpolation, centering, and normalization) based on supplementary material 5 of [41]. We formed a training dataset based on the 1000 smallest variance quasar flux samples and performed PCA to get a “ground-truth” measurement of the subspace basis using $\hat{d} = 5$ as the rank parameter estimated from SignFlipPA [25]. We formed the test dataset by excluding the 1000 samples used during training and combining 9000 samples of various noise quality, leading to heteroscedasticity across samples as shown in Fig. 6. This figure shows only the 3000 lowest noise variance

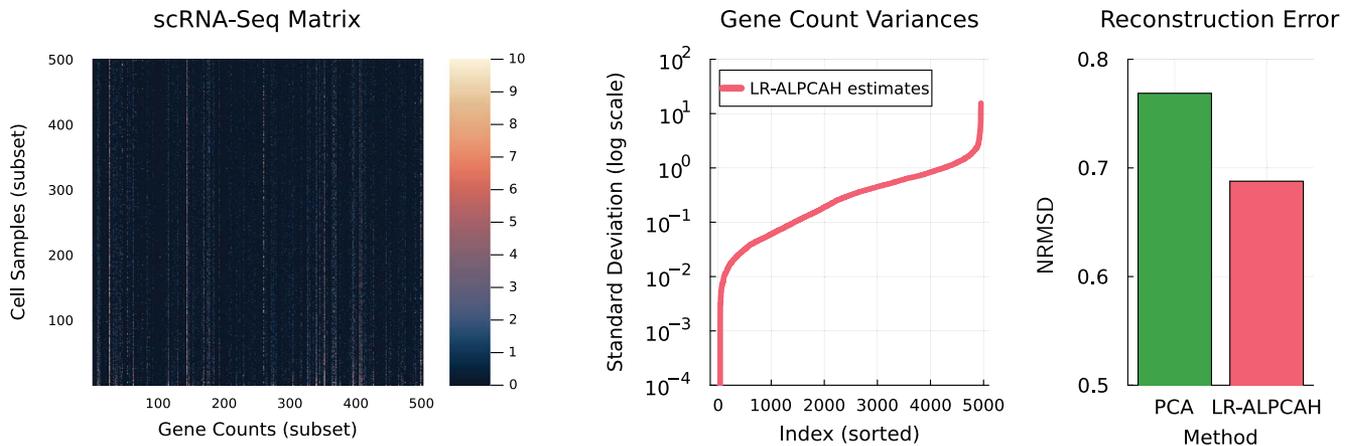


Fig. 7. Biological scRNA-seq data results.

TABLE I
SUBSPACE LEARNING RESULTS
ON QUASAR FLUX DATA

	Time (ms)	Memory (MiB)	Mean SAE
Homoscedastic Reference			
PCA	32.4	7.3	0.65
Heteroscedastic Methods			
RPCA (classical)	5091.8	7977.5	0.46
HePPCAT	1339.1	5731.6	0.39
ALPCA ($\hat{d}=0$)	4339.3	3838.6	0.41
ALPCA ($\hat{d}=10$)	4339.9	3838.8	0.39
LR-ALPCA	153.5	459.0	0.38

data samples along with 2000 noisier samples to illustrate the differences in data quality.

For our subspace quality experiment, we report SAE using the “ground-truth” basis over 100 trials for various methods. We ran each applicable subspace learning algorithm for 100 iterations to ensure convergence. In Fig. 5(a), RPCA seems to perform slightly worse than the other methods, indicating a model mismatch between outliers and heteroscedastic data. Moreover, it seems that LR-ALPCA and ALPCA performed equally well as HePPCAT in this specific real data example. All methods performed better than PCA, indicating a mismatch between the homoscedastic assumption of PCA and the heteroscedastic data. Additionally, we examined the computational time and memory requirements for these methods on this test dataset. Table I shows that the proposed LR-ALPCA method is both extremely fast and memory efficient relative to the other heteroscedastic methods as shown in bold. Since we have reference noise variance values, we also examined how the estimated noise variance values compared to the reference values. Fig. 5(b) sorts the data based on the reference variance values and plots the ALPCA estimates. ALPCA estimates generally tracked the global trend found in the reference values but there are minor variations among adjacent points.

2) *Biological scRNA-Seq Data*: This section applies PCA and LR-ALPCA to real data from single-cell RNA-sequencing data (scRNA-seq) from [42]. This sequencing

technology is useful for quantifying the transcriptome of individual cells [43]. The data is high dimensional since thousands of genes are counted for thousands of cell samples, which produces challenges for data analysis. PCA methods are useful for scRNA-seq data to perform gene variation analysis and clustering in low-dimensional spaces to study gene groups [44]. Heterogeneous noise may occur among both cells and genes [25], which prompts further investigation into heteroscedastic-aware PCA methods on scRNA-seq data. The data matrix consists of 10,000 cells by 5,000 genes. We preprocessed the data by subtracting the mean and replacing the missing values with zeros. Since the noise variances are unknown in this application, we cannot have a “ground truth” subspace to compare against. Instead, we separate the data into train and test, and calculate the NRMSD to compare reconstruction quality, i.e.,

$$\text{NRMSD} = \|\mathbf{Y}_{\text{test}} - \mathbf{U}_{\text{train}} \mathbf{U}_{\text{train}}^T \mathbf{Y}_{\text{test}}\|_F / \|\mathbf{Y}_{\text{test}}\|_F. \quad (39)$$

The subspace basis was learned on the training data with PCA or LR-ALPCA and the test data was used to assess reconstruction quality by projecting test data onto the subspace basis and using the basis coefficients to return to the ambient space. In this experiment, SignFlipPA was used to determine an appropriate rank [25]. Fig. 7 shows a subset of the data matrix. Here, the color map is clipped to 10 to better visualize the matrix as most gene counts are sparse. The middle plot shows sorted noise variances estimated by LR-ALPCA indicating some potential heterogeneity by one or two orders of magnitude. The right plot shows that LR-ALPCA has a better reconstruction quality since it has ~ 0.1 lower NRMSD than PCA. The difference between PCA and LR-ALPCA is more modest with this dataset. Possibly, the results could be improved further by developing a method that handles heteroscedasticity across both the samples and features, as this data is doubly heteroscedastic. Moreover, real scRNA-seq data has additional challenges such as dependent noise that is not modeled by our method. However, preliminary results indicate that LR-ALPCA is a promising approach and further investigation into addressing model assumptions is an interesting direction of future work.

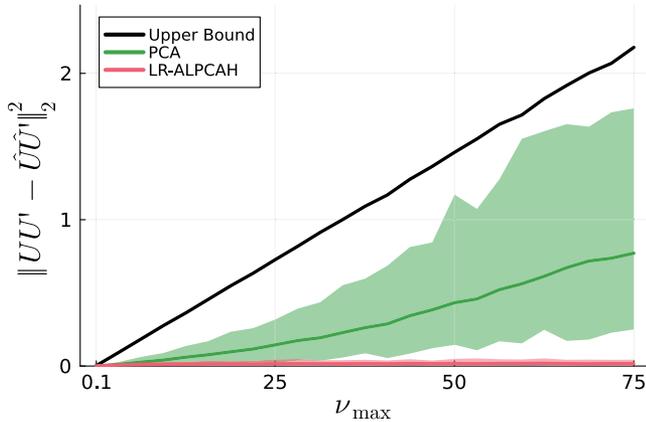


Fig. 8. Experimental verification of heteroscedastic impact on PCA upper bound (7).

V. CONCLUSION

This paper proposed two subspace learning algorithms that are robust to heteroscedasticity by jointly learning the noise variances and subspace bases. While LR-ALPCA is memory efficient and fast, its application is limited to sample-wise heteroscedasticity. It would be interesting to generalize this work to be doubly heteroscedastic, where the features themselves also have different noise variances. Applications such as biological sequencing [45] and photon imaging [46] could benefit from such an extension. In the scRNA-seq application, we computed missing entries as zeros, which is a natural choice for low-rank models. However, others have worked on adapting PCA methods for missing data [47] so such an approach could be beneficial given the higher than expected NRMSD with LR-ALPCA. This generalization is nontrivial so it is left for future work. Additionally, our model and the comparison methods are limited to the subspace setting, but some applications like resting-state functional MRI [48] benefit from manifold learning approaches [48]. It would be interesting to explore other approaches such as a heteroscedastic variational autoencoder [49] to expand the possible applications of heteroscedastic data learning.

APPENDIX

A. PCA Bound Experiment

This section focuses on providing empirical verification of our subspace bound in (7). Before doing so, we mention that the random matrix theory bound in (3) depends on a universal constant c_1 , independent of D and N , that is not calculated in the source paper [18]. Let $A = [a]$ be a 1×1 matrix with element $a \sim \mathcal{N}(0, 1)$. For the LHS in (3), the spectral norm in this instance is $\|A\|_2 = |a|$. This implies that $\mathbb{E}[\|A\|_2]$ is equivalent to calculating the mean of a folded normal distribution. Since a is a standard normal random variable, $\mathbb{E}[|a|] = \sqrt{2/\pi}$. One can verify from (4) (5) (6) that the RHS in (3) simplifies to $2 + \sqrt[4]{3}$. Solving for c_1 , the inequality becomes $c_1 \geq \sqrt{2/\pi}/(2 + \sqrt[4]{3}) \approx 0.24$. In our subspace bound (7), both sides are squared and a factor of 2 exists in (2), therefore the

constant in (7) is $c = 4c_1^2 \approx 0.22$. Knowing this constant, it is now possible to experimentally verify (7). The experimental setup consists of generating random rank-3 subspaces within a 100 dimensional ambient space. The data samples consist of two groups, one with $n_1 = 30$ samples, $\nu_1 = 0.1$ and the other with $n_2 = 970$ samples and a varying $\nu_2 \in \{0.1, \dots, 75\}$. During the course of 50 trials, we computed the mean spectral norm projection error, i.e., $\|\hat{U}\hat{U}' - UU'\|_2^2$, along with the minimum and maximum error values for that ν_2 instance. Fig. 8 illustrates that PCA scales similarly to the bound in (7), yet our method, LR-ALPCA, empirically did not degrade at the same rate, indicating robustness to heteroscedasticity.

ACKNOWLEDGMENT

The authors thank Jason Hu for helpful discussions related to algorithm convergence for Thm. 1. The authors thank David Hong for suggesting heteroscedastic data applications such as astronomy spectra and single-cell RNA sequencing. The authors have declared no financial conflicts of interest in this work.

REFERENCES

- [1] Y. Chen, Y. Tang, C. Wang, X. Liu, L. Zhao, and Z. Wang, "ADHD classification by dual subspace learning using resting-state functional connectivity," *Artif. Intell. Med.*, vol. 103, 2020, Art. no. 101786.
- [2] Q. Jia, J. Cai, X. Jiang, and S. Li, "A subspace ensemble regression model based slow feature for soft sensing application," *Chin. J. Chem. Eng.*, vol. 28, no. 12, pp. 3061–3069, 2020.
- [3] W. He, N. Yokoya, and X. Yuan, "Fast hyperspectral image recovery of dual-camera compressive hyperspectral imaging via non-iterative subspace-based fusion," *IEEE Trans. Image Process.*, vol. 30, pp. 7170–7183, 2021.
- [4] Y. Fu, W. Wang, and C. Wang, "Image change detection method based on RPCA and low-rank decomposition," in *Proc. 35th Chin. Control Conf. (CCC)*, Piscataway, NJ, USA: IEEE Press, 2016, pp. 9412–9417.
- [5] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma, "Stable principal component pursuit," in *Proc. IEEE Int. Symp. Inf. Theory*, Piscataway, NJ, USA: IEEE Press, 2010, pp. 1518–1522.
- [6] R. Otazo, E. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *Magnetic Reson. Med.*, vol. 73, no. 3, pp. 1125–1136, 2015.
- [7] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, Jul. 2018.
- [8] U. E. P. Agency, "Air quality system data mart." [Online]. Available: <https://www.epa.gov/outdoor-air-quality-data>
- [9] P. Tsalantza and D. W. Hogg, "A data-driven model for spectra: Finding double redshifts in the sloan digital sky survey," *Astrophys. J.*, vol. 753, no. 2, 2012, Art. no. 122.
- [10] Y. Cao, A. Zhang, and H. Li, "Multisample estimation of bacterial composition matrices in metagenomics data," *Biometrika*, vol. 107, no. 1, pp. 75–92, Dec. 2019, doi: 10.1093/biomet/asz062.
- [11] D. Hong, L. Balzano, and J. A. Fessler, "Asymptotic performance of PCA for high-dimensional heteroscedastic data," *J. Multivar. Anal.*, vol. 167, no. C, pp. 435–452, Sep. 2018, doi: 10.1016/j.jmva.2018.06.002.
- [12] M. E. Tipping and C. Bishop, "Probabilistic principal component analysis," *J. Roy. Statist. Soc., Ser. B*, vol. 21, no. 3, pp. 611–622, Jan. 1999, Available from: <http://www.ncrg.aston.ac.uk/Papers/index.html>. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/probabilistic-principal-component-analysis/>
- [13] D. Hong, K. Gilman, L. Balzano, and J. A. Fessler, "HePPCAT: Probabilistic PCA for data with heteroscedastic noise," *IEEE Trans. Signal Process.*, vol. 69, pp. 4819–4834, 2021.
- [14] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.
- [15] J. S. Cavazos, J. A. Fessler, and L. Balzano, "ALPCA: Sample-wise heteroscedastic PCA with tail singular value regularization," in *Proc. Int.*

- Conf. Sampling Theory Appl. (SampTA)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 1–6.
- [16] Y. Chi, Y. M. Lu, and Y. Chen, “Nonconvex optimization meets low-rank matrix factorization: An overview,” *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5239–5269, Oct. 2019.
- [17] R. Vershynin, *High-Dimensional Probability: An Introduction With Applications in Data Science*. Cambridge, U.K.: Cambridge Univ. Press, 2018, vol. 47.
- [18] R. Latała, “Some estimates of norms of random matrices,” *Proc. Amer. Math. Soc.*, vol. 133, no. 5, pp. 1273–1282, 2005.
- [19] A. R. Zhang, T. T. Cai, and Y. Wu, “Heteroskedastic PCA: Algorithm, optimality, and applications,” *Ann. Statist.*, vol. 50, no. 1, pp. 53–80, 2022.
- [20] A. Collas, F. Bouchard, A. Breloy, G. Ginolhac, C. Ren, and J.-P. Ovarlez, “Probabilistic pca from heteroscedastic signals: geometric framework and application to clustering,” *IEEE Trans. Signal Process.*, vol. 69, pp. 6546–6560, 2021.
- [21] L. Delchambre, “Weighted principal component analysis: a weighted covariance eigendecomposition approach,” *Monthly Notices Roy. Astronomical Soc.*, vol. 446, no. 4, pp. 3545–3555, 2015.
- [22] T.-H. Oh, Y.-W. Tai, J.-C. Bazin, H. Kim, and I. S. Kweon, “Partial sum minimization of singular values in robust PCA: Algorithm and applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 744–758, Apr. 2016.
- [23] N. Johnston and O. Blog, “Ky Fan norms, schatten norms, and everything in between,” *Nathaniel Johnston. Np*, vol. 21, p. 1, Aug. 2009.
- [24] E. Dobriban, “Permutation methods for factor analysis and PCA,” *Ann. Statist.*, vol. 48, no. 5, pp. 2824–2847, 2020, doi: 10.1214/19-AOS1907.
- [25] D. Hong, Y. Sheng, and E. Dobriban, “Selecting the number of components in PCA via random signflips,” 2020, *arXiv:2012.02985*.
- [26] S. Boyd et al., “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends® Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [27] Z. Lin, M. Chen, L. Wu, and Y. Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” Coordinated Science Laboratory Report no. UILU-ENG-09-2215, DC-247, 2009.
- [28] K. Guo, D. Han, and T.-T. Wu, “Convergence of alternating direction method for minimizing sum of two nonconvex functions with linear constraints,” *Int. J. Comput. Math.*, vol. 94, no. 8, pp. 1653–1669, 2017.
- [29] B. Mishra and W. Franz, “Algorithmic algebra,” *SIAM Rev.*, vol. 37, no. 3, p. 479, 1995.
- [30] H. Attouch, J. Bolte, and B. F. Svaiter, “Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized Gauss–Seidel methods,” *Math. Program.*, vol. 137, nos. 1–2, pp. 91–129, 2013.
- [31] L. van den Dries and C. Miller, “Geometric categories and o-minimal structures,” *Duke Math. J.*, vol. 84, no. 2, pp. 497–540, 1996, doi: 10.1215/S0012-7094-96-08416-1.
- [32] Y. Chi, Y. M. Lu, and Y. Chen, “Nonconvex optimization meets low-rank matrix factorization: An overview,” *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5239–5269, Oct. 2019.
- [33] C. L. Byrne, “Alternating minimization as sequential unconstrained minimization: A survey,” *J. Optim. Theory Appl.*, vol. 156, pp. 554–566, Aug. 2012.
- [34] S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi, and B. Recht, “Low-rank solutions of linear matrix equations via procrustes flow,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2016, pp. 964–973.
- [35] C. Eckart and G. Young, “The approximation of one matrix by another of lower rank,” *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.
- [36] O. L. Mangasarian, “Pseudo-convex functions,” *J. Soc. Ind. Appl. Math. Ser. A: Control*, vol. 3, no. 2, pp. 281–290, 1965.
- [37] L. Grippo and M. Sciandrone, “On the convergence of the block nonlinear Gauss–Seidel method under convex constraints,” *Oper. Res. Lett.*, vol. 26, no. 3, pp. 127–136, 2000.
- [38] A. Barg and D. Y. Nogin, “Bounds on packings of spheres in the Grassmann manifold,” *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2450–2454, Sep. 2002.
- [39] R. Ahumada et al., “The 16th data release of the sloan digital sky surveys: First release from the APOGEE-2 southern survey and full release of eBOSS spectra,” *Astrophys. J. Suppl. Ser.*, vol. 249, no. 1, 2020, p. 3.
- [40] B. W. Lyke et al., “The sloan digital sky survey quasar catalog: Sixteenth data release,” *Astrophys. J. Suppl. Ser.*, vol. 250, no. 1, 2020.
- [41] D. Hong, F. Yang, J. A. Fessler, and L. Balzano, “Optimally weighted PCA for high-dimensional heteroscedastic data,” *SIAM J. Math. Data Sci.*, vol. 5, no. 1, pp. 222–250, 2023.
- [42] E. Z. Macosko et al., “Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets,” *Cell*, vol. 161, no. 5, pp. 1202–1214, 2015.
- [43] T. S. Andrews and M. Hemberg, “Identifying cell populations with scRNAseq,” *Mol. Aspects Med.*, vol. 59, pp. 114–122, Feb. 2018.
- [44] P. V. Kharchenko, “The triumphs and limitations of computational methods for scRNA-seq,” *Nature Methods*, vol. 18, no. 7, pp. 723–732, 2021.
- [45] Y. Cao, A. Zhang, and H. Li, “Multisample estimation of bacterial composition matrices in metagenomics data,” *Biometrika*, vol. 107, no. 1, pp. 75–92, 2020.
- [46] J. Salmon, Z. Harmany, C.-A. Deledalle, and R. Willett, “Poisson noise reduction with non-local PCA,” *J. Math. Imag. Vis.*, vol. 48, pp. 279–294, Apr. 2014.
- [47] K. Gilman, D. Hong, J. A. Fessler, and L. Balzano, “Streaming probabilistic PCA for missing data with heteroscedastic noise,” 2023, *arXiv:2310.06277*.
- [48] J.-H. Kim, Y. Zhang, K. Han, Z. Wen, M. Choi, and Z. Liu, “Representation learning of resting state fmri with variational autoencoder,” *NeuroImage*, vol. 241, 2021, Art. no. 118423.
- [49] I. Higgins et al., “beta-vae: Learning basic visual concepts with a constrained variational framework,” *ICLR (Poster)*, vol. 3, Feb. 2017, pp. 1–22.



Javier Salazar Cavazos (Graduate Student Member, IEEE) received dual B.S. degrees in electrical engineering and mathematics from the University of Texas, Arlington, TX, USA, in 2020, and the M.S. degree in electrical and computer engineering (ECE) from the University of Michigan, Ann Arbor, MI, USA, in 2023. He is currently working toward the Ph.D. degree in electrical and computer engineering (ECE) working with Professor Laura Balzano and Jeffrey Fessler both with the University of Michigan, Ann Arbor, MI, USA. His main research interests include signal and image processing, machine learning, deep learning, and optimization. Specifically, current work focuses on topics such as low-rank modeling, clustering, and Alzheimer’s disease diagnosis and prediction from MRI data.



Jeffrey A. Fessler (Fellow, IEEE) received the B.S.E.E. degree from Purdue University, West Lafayette, IN, USA, in 1985, the M.S.E.E. degree from Stanford University, Stanford, CA, USA, in 1986, and the M.S. degree in statistics and the Ph.D. degree in electrical engineering from Stanford University, in 1989 and 1990, respectively. From 1985 to 1988, he was a National Science Foundation Graduate Fellow with Stanford. Currently, he is the William L. Root Distinguished University Professor of EECS and an ECE Interim Chair with the University of Michigan, Ann Arbor, MI, USA. His research focuses on various aspects of imaging problems, and he has supervised doctoral research in PET, SPECT, X-ray CT, MRI, and optical imaging problems. In 2006, he became a fellow of the IEEE for contributions to the theory and practice of image reconstruction.



Laura Balzano (Senior Member, IEEE) received the Ph.D. degree in ECE from the University of Wisconsin, Madison, WI, USA. Currently, she is an Associate Professor of electrical engineering and computer science with the University of Michigan, Ann Arbor, MI, USA. Her main research focuses on modeling and optimization with big, messy data – highly incomplete or corrupted data, uncalibrated data, and heterogeneous data – and its applications in a wide range of scientific problems. She is an Associate Editor for IEEE OPEN JOURNAL OF SIGNAL PROCESSING and *SIAM Journal of the Mathematics of Data Science*. She was a recipient of the NSF Career Award, the ARO Young Investigator Award, and the AFOSR Young Investigator Award. She was also a recipient of the Sarah Goddard Power Award at the University of Michigan.