# ALPCAHUS: Subspace Clustering for Heteroscedastic Data

Javier Salazar Cavazos, *Graduate Student Member, IEEE*,
Jeffrey A. Fessler, *Fellow, IEEE*, and Laura Balzano, *Senior Member, IEEE*
EECS Department, University of Michigan, Ann Arbor, Michigan, United States
Email: {javiersc, fessler, girasole}@umich.edu

***Abstract*—Principal component analysis (PCA) is a key tool in the field of data dimensionality reduction. Various methods have been proposed to extend PCA to the union of subspace (UoS) setting for clustering data that comes from multiple subspaces like $K$-Subspaces (KSS). 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 heteroscedastic-focused subspace clustering method, named ALPCAHUS, that can estimate the sample-wise noise variances and use this information to improve the estimate of the subspace bases associated with the low-rank structure of the data. This clustering algorithm builds on K-Subspaces (KSS) principles by extending the recently proposed heteroscedastic PCA method, named LR-ALPCAH, for clusters with heteroscedastic noise in the UoS setting. Simulations and real-data experiments show the effectiveness of accounting for data heteroscedasticity compared to existing clustering algorithms. Code available at https://github.com/javiersc1/ALPCAHUS.**

***Index Terms*—Heteroscedastic data, heterogeneous data quality, subspace bases estimation, subspace clustering, union of subspace model, unsupervised learning.**

## I. Introduction

Many modern data science problems require learning an approximate signal subspace basis for some collection of data. This is important for downstream tasks involving subspace basis coefficients such as classification [1], regression [2], and compression [3]. Besides subspace learning, one may be interested in clustering data points that originate from multiple subspaces. Formally, subspace clustering, or union of subspace (UoS) modeling, is an unsupervised machine learning problem where the goal is to cluster unlabeled data and find the subspaces associated with each data cluster. When the cluster assignments are known, it is easy to find the subspaces, and vice versa. This problem becomes nontrivial when both components must be estimated [4]. This clustering problem has many applications, such as image segmentation [5], motion segmentation [6], image compression [7], and system identification [8].

Some applications involve heterogeneous data samples that vary in quality due in part to noise characteristics associated with each sample. Some examples of heteroscedastic datasets include environmental air quality data [9], astronomical spectral data [10], and biological sequencing data [11]. In heteroscedastic settings, the noisier data samples can significantly corrupt the basis estimates [12]. In turn, this corruption can worsen clustering performance. Popular clustering methods such as Sparse Subspace Clustering (SSC) [13], $K$-Subspaces (KSS) [14], and Subspace Clustering via Thresholding (TSC) [15] implicitly assume that data quality is consistent. For example, in SSC, the method relies on the self-expressiveness property of data that requires using other samples to estimate similarity. From our experiments, we found that this implicit data quality assumption can degrade clustering quality for heteroscedastic data.

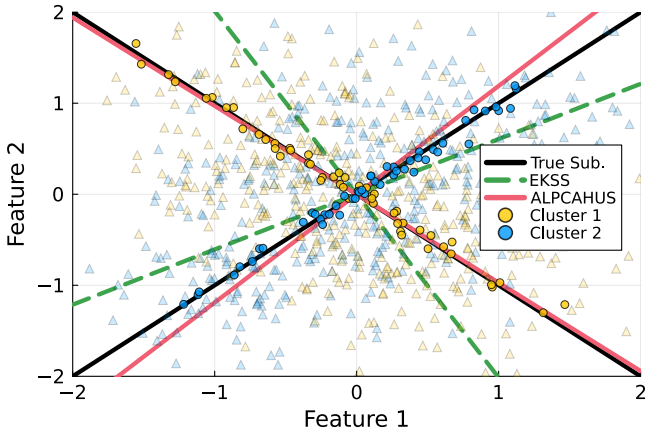



Fig. 1: Two 1D subspaces, colored blue and yellow, with data consisting of two noise groups shown with circle and triangle marker types.

Because of these limitations, we developed a subspace clustering algorithm, inspired from KSS principles, that explicitly models noise variance terms, without assuming that data quality is known. The method adaptively clusters data while learning noise characteristics. See Fig. 1 for a visualization where Ensemble KSS (EKSS) [16] returns poor subspace bases estimates whereas our method found more accurate subspace bases and improved clustering quality.

We extend our previous work [17] by generalizing the LR-ALPCAH formulation to the UoS setting for clustering heteroscedastic data. The proposed approach achieved $\sim$3 times lower clustering error than existing methods, and it achieved relatively low clustering error even when very few high quality samples are available. The paper is divided into a few key sections. Section II introduces the heteroscedastic problem formulation for subspace clustering. Section III discusses related work in subspace clustering and reviews the heteroscedastic

subspace algorithm LR-ALPCAH. Section IV introduces the proposed subspace clustering method named ALPCAHUS. Section V covers synthetic and real data experiments that illustrate the effectiveness of modeling heteroscedasticity in a clustering context. Sec. VI discusses some limitations of our method and possible extensions.

## II. Problem Formulation

Let $K$ denote the number of subspaces that is either known beforehand or estimated using other methods. Before describing the general union of subspace model, the single subspace model ($K = 1$) is introduced.

### A. Single-Subspace Model ($K = 1$)

Let $\boldsymbol{y}_i \in \mathbb{R}^D$ denote the data samples for index $i \in \{1, \dots, N\}$, where $D$ denotes the ambient dimension. Let $\boldsymbol{x}_i$ represent the low-dimensional data sample generated by $\boldsymbol{x}_i = \boldsymbol{U}\boldsymbol{z}_i$ where $\boldsymbol{U} \in \mathbb{R}^{D\times d}$ is an unknown basis for a subspace of dimension $d$ and $\boldsymbol{z}_i \in \mathbb{R}^d$ are the corresponding basis coordinates. Then the heteroscedastic model we consider is

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

assuming Gaussian noise with variance $\nu_i$, where $\boldsymbol{I}$ denotes the $D \times D$ identity matrix. We consider two cases: one where each data sample may have its own noise variance, and one where the case where there are $G$ groups of data having shared noise variance terms $\{\nu_1, \dots, \nu_G\}$. Sec. III-B discusses an optimization problem based on this model that estimates the heterogeneous noise variances $\{\nu_i\}$ and the subspace basis $\boldsymbol{U}$.

### B. Union of Subspaces Model ($K \geq 1$)

Let $\boldsymbol{Y} = \begin{bmatrix}\boldsymbol{y}_1 & \dots & \boldsymbol{y}_N\end{bmatrix} \in \mathbb{R}^{D\times N}$ denote a matrix whose columns consist of all $N$ data points $\boldsymbol{y}_i \in \mathbb{R}^D$. We generalize (1) to model the data with a union of subspaces model as follows

$$\begin{aligned} \boldsymbol{y}_i &= \boldsymbol{x}_i + \boldsymbol{\epsilon}_i \\ \boldsymbol{x}_i &= \boldsymbol{U}_{k_i}\boldsymbol{z}_i \text{ for some } k_i \in \{1, \dots, K\}, \end{aligned} \tag{2}$$

where $\boldsymbol{U}_k \in \mathbb{R}^{D\times d_k}$ is a subspace basis that has subspace dimension $d_k$. Here $\boldsymbol{z}_i \in \mathbb{R}^{d_k}$ denotes the basis coefficients associated with $\boldsymbol{x}_i$, and $\boldsymbol{\epsilon}_i \in \mathbb{R}^D$ denotes noise for that point drawn from $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \nu_i \boldsymbol{I})$.

If the subspace bases were known, then one would like to find the associated subspace label $c_i \in \{1, \dots, K\}$ for each data sample by solving the following optimization problem

$$c_i = \arg\min_k \|\boldsymbol{y}_i - \boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{y}_i\|_2^2, \quad \forall \boldsymbol{y}_i \in \boldsymbol{Y}. \tag{3}$$

This label describes the subspace association of a data point $\boldsymbol{y}_i$ that has the lowest residual between the original sample and its reconstructed value given the subspace model. In general, the goal is to estimate all of the subspace bases $\mathcal{U} = (\boldsymbol{U}_1, \dots, \boldsymbol{U}_K)$ and cluster assignments $\mathcal{C} = (c_1, \dots, c_N)$ that describe the subspace labels of all points.

## III. Related Works

### A. Subspace Clustering

Many subspace clustering algorithms fall into a general umbrella of categories such as algebraic methods [18], iterative methods [19], statistical methods [20], and spectral clustering methods [21] [22]. In recent years, both spectral clustering-based methods and iterative methods have become popular.

*1) Spectral Clustering:* Many methods build on spectral clustering. This method is often used for clustering nodes in graphs. Spectral clustering aims to find the minimum cost "cuts" in the graph to partition nodes into clusters. Given some collection of data points, one way to construct a graph is to assume that "nearby" data samples are highly connected in the graph. Thus, it is possible to construct a graph from data samples, with various levels of connectedness, by using some metric to assign affinity/similarity for each pair of points. Let $\mathcal{G}(\mathcal{V}, \boldsymbol{W})$ correspond to a graph that consists of vertices $\mathcal{V} = \{v_1, \dots, v_N\}$ and edge weights $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ such that $w_{ij}$ corresponds to some nonnegative weight, or similarity, between $v_i$ and $v_j$. The graph $\mathcal{G}$ is assumed to be undirected, i.e., $\boldsymbol{W} = \boldsymbol{W}^T$. The degree of a node is defined as $d_i = \sum_{j=1}^N w_{ij}$ and can be collected to form a degree matrix such that $\boldsymbol{D} = \text{diag}(d_1, \dots, d_N)$. Let $\mathcal{A}_1, \dots, \mathcal{A}_K$ form a $K$-partition on $\mathcal{G}(\mathcal{V}, \boldsymbol{W})$ and $\text{cut}(\mathcal{A}, \mathcal{B}) = \sum_{i\in\mathcal{A}, j\in\mathcal{B}} w_{ij}$. Then, spectral clustering aims to solve the following optimization problem:

$$\arg\min_{\mathcal{A}_1,\dots,\mathcal{A}_K} \frac{1}{2}\sum_{i=1}^K \frac{\text{cut}(\mathcal{A}_i, \bar{\mathcal{A}}_i)}{|\mathcal{A}_i|} \tag{4}$$

where $\bar{\mathcal{A}}_i$ is all vertices not belonging in $\mathcal{A}_i$ and $|\mathcal{A}_i|$ is the number of vertices belonging to the $i$th partition. Instead of working with $\boldsymbol{W}$ directly, one computes a normalized graph Laplacian such as $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{W}$ to make the influence of heavy degree nodes more similar to low degree nodes. Collecting the indicator values

$$h_{ij} \triangleq \left\{ \begin{array}{ll} 1/\sqrt{|\mathcal{A}_j|} & \text{if } v_i \in \mathcal{A}_j \\ 0 & \text{otherwise} \end{array} \right\}$$

into a matrix $\boldsymbol{H} \in \mathbb{R}^{N\times K}$, one can rewrite (4) as

$$\frac{1}{2}\sum_{i=1}^K \frac{\text{cut}(\mathcal{A}_i, \bar{\mathcal{A}}_i)}{|\mathcal{A}_i|} = \sum_{i=1}^K (\boldsymbol{H}^T\boldsymbol{L}\boldsymbol{H})_{ii} = \text{Tr}(\boldsymbol{H}^T\boldsymbol{L}\boldsymbol{H}), \tag{5}$$

where $\text{Tr}(\boldsymbol{H}) = \sum_i \boldsymbol{H}_{ii}$ denotes the trace of a matrix.

The indicator vectors are discrete, which makes the problem computationally challenging. One way to relax the problem is to allow arbitrary real values for $\boldsymbol{H}$ and instead solve

$$\hat{\boldsymbol{H}} = \arg\min_{\boldsymbol{H}\in\mathbb{R}^{N\times K}} \text{Tr}(\boldsymbol{H}^T\boldsymbol{L}\boldsymbol{H}) \text{ s.t. } \boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{I} \tag{6}$$

where the notation s.t. means subject to some conditions. Trace minimization problems with semi-unitary constraints are well-studied in the literature, and it is easy to see that $\hat{\boldsymbol{H}}$ consists of the first $K$ eigenvectors of $\boldsymbol{L}$ assuming that $\lambda_1(\boldsymbol{L}) \leq \dots \leq \lambda_N(\boldsymbol{L})$ where $\lambda_i(\boldsymbol{L})$ denotes the $i$th eigenvalue of $\boldsymbol{L}$. Once $\hat{\boldsymbol{H}}$ is computed, the $K$-means algorithm is applied to $\hat{\boldsymbol{H}}^T$, treating each row of $\hat{\boldsymbol{H}}$ as the spectral embedding of the associated $v_i$ vertex. The next section, Sec. III-A2, now describes some

subspace clustering methods that use this technique by first forming a weight matrix $\boldsymbol{W}$ and then applying the spectral clustering method.

*2) Self-Expressive Methods:* Self-expressive methods exploit the "self-expressiveness property" [23] of data that hypothesizes that a single data point can be expressed as a linear combination of other data points in its cluster, which is trivially true if the data exactly follows a subspace model. The goal is to learn those linear coefficients and which is often achieved by adopting different regularizers in the formulation. These self-expressive algorithms, e.g., SSC [13] and LRSC [24], learn the coefficient matrix $\boldsymbol{P} \in \mathbb{R}^{N\times N}$ by solving special cases of the following general optimization problem

$$\arg\min_{\boldsymbol{P}} f(\boldsymbol{Y}-\boldsymbol{Y}\boldsymbol{P}) + \lambda\mathcal{R}(\boldsymbol{P}) \;\text{ s.t. }\; \boldsymbol{P}\in\mathcal{S}_{\boldsymbol{P}}. \tag{7}$$

The function $f(\boldsymbol{Y}-\boldsymbol{Y}\boldsymbol{P})$ is a data fidelity term, $\mathcal{R}(\boldsymbol{P})$ is a regularizer, and $\mathcal{S}_{\boldsymbol{P}}$ is some constrained set to encourage $\boldsymbol{P}$ to satisfy certain conditions. In the case of SSC the optimization problem is

$$\arg\min_{\boldsymbol{P}} \|\boldsymbol{Y}-\boldsymbol{Y}\boldsymbol{P}\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{P}\|_{1,1} \;\text{ s.t. }\; \boldsymbol{P}_{ii}=0 \;\;\forall i \tag{8}$$

where $\|\cdot\|_{1,1} = \|\mathrm{vec}(\cdot)\|_1$ is the vectorized 1-norm and $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm of a matrix. Observe that (8) approximates each data sample as a sparse linear combination of other data points to form $\boldsymbol{P}$. Let $\mathrm{abs}(\boldsymbol{P})$ represent the element-wise absolute value of matrix $\boldsymbol{P}$. Then, spectral clustering is performed on $\boldsymbol{W} = \frac{1}{2}(\mathrm{abs}(\boldsymbol{P}^T) + \mathrm{abs}(\boldsymbol{P}))$ by applying the $K$-means method to the spectral embedding of the affinity matrix $\boldsymbol{W}$. Ideally, (8) would select the high quality data to represent worse samples in $\boldsymbol{P}$ and this information would be retained in $\boldsymbol{W}$. However, this data quality awareness condition is not guaranteed in self-expressive methods to our knowledge. In our experiments, Fig. 2f and Fig. 6 show that there must be an issue constructing an ideal $\boldsymbol{P}$ due to worse clustering performance in heteroscedastic conditions.

*3) Subspace Clustering via Thresholding (TSC):* In the example to follow, assume that there exists two unique 1-dimensional subspaces. Given two data samples $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$, intuitively their dot product $\langle \boldsymbol{y}_i, \boldsymbol{y}_j\rangle$ will be high if $c_i = c_j$ meaning they belong to the same subspace. Likewise, if $c_i \neq c_j$, then $\langle \boldsymbol{y}_i, \boldsymbol{y}_j\rangle = 0$ assuming orthogonal subspaces. In non-orthogonal scenarios, $\langle \boldsymbol{y}_i, \boldsymbol{y}_j\rangle$ will be relatively small under $c_i \neq c_j$ compared to $c_i = c_j$. Using this idea, in more general D-dimensional subspace settings, TSC constructs a matrix $\boldsymbol{Z} \in \mathbb{R}^{N\times N}$ such that

$$\boldsymbol{Z}_{ij} = \exp(-2\arccos(|\langle \boldsymbol{y}_i, \boldsymbol{y}_j\rangle|)) \;\text{ s.t. }\; \boldsymbol{Z}_{ii} = 0\;\forall i. \tag{9}$$

This matrix is then thresholded to retain only the top $q$ values for each row, i.e.,

$$\boldsymbol{W} = \arg\min_{\boldsymbol{W}} \|\boldsymbol{W}-\boldsymbol{Z}\|_{\mathrm{F}}^2 \;\text{ s.t. }\; \|\boldsymbol{W}_{i,:}\|_0 = q \;\;\forall i \tag{10}$$

where $\|\boldsymbol{W}_{i,:}\|_0 = q$ is the $l_0$ pseudo-norm. In other words, one constructs $\boldsymbol{W}_{i,:}$ using the q nearest neighbors that correspond to the highest magnitude values. Then, spectral clustering is applied to $\boldsymbol{W}$ to find the subspace clustering associations.

*4) Ensemble $K$-Subspaces (EKSS):* Iterative methods are based on the idea of alternating between cluster assignments and subspace basis approximation, with KSS being highly prominent [25]. The KSS algorithm seeks to solve the following optimization problem:

$$\arg\min_{\mathcal{C},\mathcal{U}} \sum_{k=1}^{K} \sum_{c_i=k,\forall i} \|\boldsymbol{y}_i - \boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{y}_i\|_2^2. \tag{11}$$

The goal is to minimize the sum of residual norms by alternating between performing PCA on each cluster to update $\mathcal{U}$ and using the subspace bases to calculate new cluster assignments $\mathcal{C}$ in a similar fashion to the $K$-means algorithm.

The quality of the solution depends highly on the initialization. Recent work provides convergence guarantees and spectral initialization schemes that provably perform better than random initialization [26]. However, this problem (11), as shown in [27], is NP-hard. Further, it is prone to local minima [28]. To overcome this, consensus clustering [29] is a tool that leverages information from many trials and combines results together. This approach, known as Ensemble KSS (EKSS) [16] in this context, creates an affinity matrix whose $(i,j)$th entry represents the number of times the two points were clustered together in a trial. Then, spectral clustering is performed on the affinity matrix to get the final clustering from the many base clusterings. The use of PCA makes it challenging to learn clusters and subspace bases in the heteroscedastic regime since PCA implicitly assumes the same noise variance across all samples [30]. The proposed ALPCAHUS approach in Sec. IV builds on KSS, and its ensemble version, by generalizing (11) to the heteroscedastic regime.

### B. Single Subspace Heteroscedastic PCA Method

Before extending (11) to the heteroscedastic setting, a review of how LR-ALPCAH [17] solves for a single subspace ($K=1$) in the heteroscedastic setting using the model described in (1) is necessary. For the measurement model $\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{x}_i, \nu_i\boldsymbol{I})$ in (1), the probability density function for a single data sample $\boldsymbol{y}_i$ is

$$\frac{1}{\sqrt{(2\pi)^D|\nu_i\boldsymbol{I}|}} \exp\left[-\frac{1}{2}(\boldsymbol{y}_i-\boldsymbol{x}_i)'(\nu_i\boldsymbol{I})^{-1}(\boldsymbol{y}_i-\boldsymbol{x}_i)\right]. \tag{12}$$

After further manipulation, the negative log-likelihood is expressed in matrix notation as

$$\frac{1}{2}\|(\boldsymbol{Y}-\boldsymbol{X})\boldsymbol{\Pi}^{-1/2}\|_{\mathrm{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}| \tag{13}$$

where $\boldsymbol{\Pi} = \mathrm{diag}(\nu_1,\ldots,\nu_N)$. To reduce computation and enforce a low-rank solution, LR-ALPCAH [17] took inspiration from the matrix factorization literature [31] and factorized $\boldsymbol{X} \in \mathbb{R}^{D\times N} \approx \boldsymbol{L}\boldsymbol{R}'$ where $\boldsymbol{L} \in \mathbb{R}^{D\times\hat{d}}$ and $\boldsymbol{R} \in \mathbb{R}^{N\times\hat{d}}$ for some rank estimate $\hat{d}$. Using this idea, LR-ALPCAH estimates $\boldsymbol{X}$ by solving for $\boldsymbol{L}$ and $\boldsymbol{R}$ in the following optimization problem

$$\arg\min_{\boldsymbol{L},\boldsymbol{R},\boldsymbol{\Pi}} \frac{1}{2}\|(\boldsymbol{Y}-\boldsymbol{L}\boldsymbol{R}')\boldsymbol{\Pi}^{-1/2}\|_{\mathrm{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}|. \tag{14}$$

The next section, Sec. IV, combines ideas from (11) and (14) to tackle the general union of subspaces setting ($K>1$).

### C. Other heteroscedastic models

This paper focuses on heteroscedastic noise across the data samples. There are other subspace learning methods in the literature that explore heteroscedasticity in different ways. For example, HeteroPCA considers heteroscedasticity across the feature space [32]. 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 assumes that the signal itself is heteroscedastic [33]. That work considers clustering applications where the power fluctuating (i.e., heteroscedastic) signals are embedded in white Gaussian noise. We exclude comparisons with those methods since their models are very different. These different models each have their own applications.

## IV. Proposed Subspace Clustering Method

For notational simplicity, let $\boldsymbol{Y}_k \triangleq \boldsymbol{Y}_{\mathcal{C}_k} \in \mathbb{R}^{D\times N_k}$ denote the submatrix of $\boldsymbol{Y}$ having columns corresponding to data samples that are estimated to belong in the $k$th subspaces, i.e., $\boldsymbol{Y}_k \triangleq \boldsymbol{Y}_{\mathcal{C}_k} \triangleq \text{matrix}(\{\boldsymbol{y}_i : c_i = k\})$. This notation applies similarly to other matrices such as $\boldsymbol{\Pi}_k \triangleq \boldsymbol{\Pi}_{\mathcal{C}_k} \triangleq \text{diag}(\{\nu_i : c_i = k\})$. For the union of subspace measurement model (2), we generalize LR-ALPCAH (14) with the following optimization problem

$$\underset{\mathcal{L},\mathcal{R},\boldsymbol{\Pi},\mathcal{C}}{\arg\min} \sum_{k=1}^{K} \frac{1}{2}\left\|(\boldsymbol{Y}_k - \boldsymbol{L}_k\boldsymbol{R}_k^T)\boldsymbol{\Pi}_k^{-1/2}\right\|_{\text{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}_k| \quad (15)$$

where $\mathcal{C},\boldsymbol{\Pi},\mathcal{L},\mathcal{R}$ denote the sets of estimated clusters, noise variances, and factorized matrices respectively for each cluster $k = 1,\dots,K$. Specifically $\mathcal{L} \triangleq \{\boldsymbol{L}_1,\dots,\boldsymbol{L}_K\}$ and $\mathcal{R} \triangleq \{\boldsymbol{R}_1,\dots,\boldsymbol{R}_K\}$. Our algorithm for solving (15) is called ALPCAHUS (**ALPCAH** for **U**nion of **S**ubspaces). Similar to KSS, ALPCAHUS alternates between updating subspace bases given cluster estimates and updating cluster estimates by projection given subspace bases estimates. We solve (15) with the alternating minimization technique. The method alternates between basis estimation, $\forall k \in \{1,\dots,K\}$ solve

$$\underset{\boldsymbol{L}_k,\boldsymbol{R}_k,\boldsymbol{\Pi}_k}{\arg\min} \frac{1}{2}\left\|(\boldsymbol{Y}_k - \boldsymbol{L}_k\boldsymbol{R}_k^T)\boldsymbol{\Pi}_k^{-1/2}\right\|_{\text{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}_k| \;\text{ s.t. }\; \boldsymbol{\Pi}_k \succeq \alpha\boldsymbol{I} \quad (16)$$

and data sample reassignment, $\forall i \in \{1,\dots,N\}$ solve

$$\begin{aligned} &c_i^{(t_2+1)} = \underset{k}{\arg\min}\, J_i(k) = \|\boldsymbol{y}_i - \boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{y}_i\|_2^2 \;\text{ s.t.}\\ &c_i^{(t_2+1)} \leftarrow c_i^{(t_2)} \text{ if } \exists \tilde{k} \in \mathcal{S}_{J_i} \neq c_i^{(t_2)} \text{ such that}\\ &J_i(c_i^{(t_2)}) = J_i(\tilde{k}) \text{ occurs.} \end{aligned} \quad (17)$$

The solution to (16) is described in [17] with convergence theory. For completeness and notational consistency, the updates are included in (18) (19) (21). For the $(t_1+1)$ iteration, given current $t_1$ iterates, the updates are given as

$$\begin{aligned} \boldsymbol{L}_k^{(t_1+1)} &= \underset{\boldsymbol{L}_k}{\arg\min}\, f(\boldsymbol{L}_k, \boldsymbol{R}_k^{(t_1)}, \boldsymbol{\Pi}_k^{(t_1)})\\ &= \boldsymbol{Y}_k\boldsymbol{\Pi}_k^{(t_1)}\boldsymbol{R}_k^{(t_1)}({\boldsymbol{R}_k^{(t_1)}}^T\boldsymbol{\Pi}_k^{(t_1)}\boldsymbol{R}_k^{(t_1)})^{-1} \quad (18)\\ \boldsymbol{R}_k^{(t_1+1)} &= \underset{\boldsymbol{R}_k}{\arg\min}\, f(\boldsymbol{L}_k^{(t_1)}, \boldsymbol{R}_k, \boldsymbol{\Pi}_k^{(t_1)})\\ &= \boldsymbol{Y}_k^T\boldsymbol{L}_k^{(t_1)}({\boldsymbol{L}_k^{(t_1)}}^T\boldsymbol{L}_k^{(t_1)})^{-1}. \quad (19) \end{aligned}$$

Let the following function be defined

$$p_k(j) \triangleq \max(\alpha, |\mathcal{C}_k|^{-1}\|(\boldsymbol{Y}_k - \boldsymbol{L}_k^{(t_1)}{\boldsymbol{R}_k^{(t_1)}}^T)\boldsymbol{e}_j\|_2^2) \quad (20)$$

where $\boldsymbol{e}_j$ denotes the $j$th standard canonical basis vector that is used to select the $j$th column of some matrix and $\alpha > 0$ is some noise variance threshold parameter. Then, the noise variance matrix update is given as

$$\begin{aligned} \boldsymbol{\Pi}_k^{(t_1+1)} &= \underset{\boldsymbol{\Pi}_k}{\arg\min}\, f(\boldsymbol{L}_k^{(t_1)}, \boldsymbol{R}_k^{(t_1)}, \boldsymbol{\Pi}_k) \implies\\ &= \text{diag}(p_k(1),\dots,p_k(|\mathcal{C}_k|)). \end{aligned} \quad (21)$$

Because $\boldsymbol{\Pi}_k$ is a diagonal matrix, the $\boldsymbol{\Pi}_k \succeq \alpha\boldsymbol{I}$ majorization condition in (16) implies that $\forall i, \lambda_i(\boldsymbol{\Pi}_k) \geq \alpha$ and leads to a projection to the positive orthant, i.e., the $\max(\alpha,\cdot)$ condition in (20). This condition is necessary to ensure convergence as proven in [17, Thm. 2] and used in this work to further prove ALPCAHUS convergence in Thm. 1 using some $\alpha \in \mathbb{R}_+$.

The solution to (17) involves the following procedure. Given a data sample $\boldsymbol{y}_i$, find the lowest subspace projection residual out of all subspaces and assign it to $c_i^{(t_2+1)}$ at the $(t_2+1)$ iteration. In the event of a tie, meaning there is more than one subspace that has equal residual, copy the past label $c_i^{(t_2)}$ to $c_i^{(t_2+1)}$. In (17), $\mathcal{S}_{J_i}$ refers to the set of possible $k$ that achieves the lowest cost $J_i(k)$, i.e., $J_i(k_1) = J_i(k_2)$ for $k_1 \neq k_2$. By doing this, cycling between cluster labels is prevented for all data samples. This cluster reassignment criteria will play a big role in ensuring convergence as proven in Thm. 1.

To further improve clustering performance, consensus clustering can be leveraged over many trials. Initially, we tried using this approach with only one trial as seen in Fig. 2b and ALPCAHUS ($B = 1$) result in Fig. 6a. However, since $K$-subspaces in general is sensitive to initialization, during experimentation there was great success in using a consensus approach with more than one trial as shown in Fig. 2d and ALPCAHUS ($B = 16$) result in Fig. 6a.

### A. Ensemble Extension for ALPCAHUS

This section presents the ensemble algorithm with base clustering parameter $B$ to combine multiple trials for finding $\mathcal{C}$ in (15). Any $B > 1$ leverages consensus clustering by forming an affinity matrix $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ where

$$\begin{aligned} \boldsymbol{W}_{i,j} = &\frac{1}{B}|\{\forall b \in \{1,\dots,B\} \text{ such that}\\ &\boldsymbol{y}_i, \boldsymbol{y}_j \text{ are co-clustered in } \mathcal{C}^{(b)}\}| \end{aligned} \quad (22)$$

**Algorithm ALPCAHUS** (unknown noise variances, unknown noise grouping)

**Input:** $\boldsymbol{Y} \in \mathbb{R}^{D\times N}$: data, $K \in \mathbb{Z}^+$: number of subspaces, $\{\hat{d}_k \in \mathbb{Z}^+ \ \ \forall k \in \{1,\dots,K\}\}$: candidate dimension for all clusters, $q \in \mathbb{Z}^+$: threshold parameter, $B \in \mathbb{Z}^+$: base clusterings, $T_1 \in \mathbb{Z}^+$: LR-ALPCAH iterations, $T_2 \in \mathbb{Z}^+$: maximum alternating updates (cluster reassignments)
**Output:** $\mathcal{C} = \{c_1,\dots,c_K\}$: clusters of $\boldsymbol{Y}$
**for** $b = 1,\dots,B$ (in parallel) **do**
  $\mathcal{C}_k \sim \{1,\dots,N\}$ s.t. $|\mathcal{C}_k| \approx \frac{N}{K}$ for $k = 1,\dots,K$ — Initialize clusters randomly without replacement
  **for** $k = 1,\dots,K$ (in parallel) **do**
    Apply LR-ALPCAH from [17] with rank $\hat{d}_k$ matrices for $\boldsymbol{L}_k$ and $\boldsymbol{R}_k$
    $\boldsymbol{L}_k, \boldsymbol{R}_k, \boldsymbol{\Pi}_k \leftarrow$ Initialize $\boldsymbol{\Pi}_k = \boldsymbol{I}$ and compute (18) (19) (21) using $\boldsymbol{Y}_k$ for $T_1$ iterations
  **end for**
  **for** $t_2 = 1,\dots,T_2$ (in sequence) **do**
    **for** $k = 1,\dots,K$ (in parallel) **do**
      $\boldsymbol{L}_k, \boldsymbol{R}_k, \boldsymbol{\Pi}_k \leftarrow$ Compute (18) (19) (21) using $\boldsymbol{Y}_k$ for $T_1$ iterations — Apply LR-ALPCAH from [17] with rank $\hat{d}_k$
    **end for**
    $\mathcal{C}_k \leftarrow \{\forall \boldsymbol{y}_i \in \boldsymbol{Y}$ : Solve (17) with $\boldsymbol{U}_k$ extracted from SVD$(\boldsymbol{L}_k), \forall k\}$ — Update current cluster estimates by projection
    Compute cost function value in (31) and stop if $\text{cost}^{(t_1+1,t_2+1)} = \text{cost}^{(t_1,t_2)}$ — Stopping criteria
  **end for**
  $\mathcal{C}^{(b)} \leftarrow \mathcal{C}_k = \{c_1,\dots,c_N\}$ — Collect results from all trials
**end for**
$\boldsymbol{W}_{i,j} \leftarrow$ Form $\boldsymbol{W}$ using (22) — Form affinity matrix of similar clusterings
**for** $i = 1,\dots,N$ (in parallel) **do**
  $\boldsymbol{Z}^{\text{row}} \leftarrow$ Threshold rows by solving (23) — Retain top $q$ elements for each row
  $\boldsymbol{Z}^{\text{col}} \leftarrow$ Threshold columns by solving (24) — Retain top $q$ elements for each column
**end for**
$\boldsymbol{W} \leftarrow \frac{1}{2}\left(\boldsymbol{Z}^{\text{row}} + \boldsymbol{Z}^{\text{col}}\right)$ — Average affinity matrix
$\mathcal{C} \leftarrow$ Perform spectral clustering via (6) and apply $K$-means on the rows of $\hat{H}$ — Final clustering

and $\mathcal{C}^{(b)} = \{c_1^{(b)},\dots,c_N^{(b)}\}$ refers to the cluster labels for the $b$th trial ranging from 1 to $B$. Then, the rows and columns of $\boldsymbol{W}$ are thresholded to retain the top $q$ values by solving

$$\boldsymbol{Z}^{\text{row}} = \arg\min_{\boldsymbol{Z}} \|\boldsymbol{W} - \boldsymbol{Z}\|_{\text{F}}^2 \ \text{ s.t. } \ \|\boldsymbol{W}_{i,:}\|_0 = q \ \ \forall i \tag{23}$$

$$\boldsymbol{Z}^{\text{col}} = \arg\min_{\boldsymbol{Z}} \|\boldsymbol{W} - \boldsymbol{Z}\|_{\text{F}}^2 \ \text{ s.t. } \ \|\boldsymbol{W}_{:,i}\|_0 = q \ \ \forall i. \tag{24}$$

Finally, spectral clustering is applied to $\frac{1}{2}(\boldsymbol{Z}^{\text{col}} + \boldsymbol{Z}^{\text{row}})$ to get the clusters from the results of the ensemble. One could select the base clusterings parameter $B = 1$ to reduce to one trial of the optimization problem (15). Alg. ALPCAHUS summarizes the procedure for subpace clustering with optional ensemble learning. For Julia code implementations, refer to https://github.com/javiersc1/ALPCAHUS.

### *B. Rank Estimation*

In subspace clustering, some algorithms like SSC do not require subspace dimension to be known or estimated, whereas others like KSS require this. For the group of algorithms that require this parameter to be estimated, there is great interest in adaptive methods that can learn the subspace dimension. In recent work, [26] proposed using an eigengap heuristic on the sample covariance matrix of each cluster, i.e., $S_k = \frac{1}{|\mathcal{C}_k|}\boldsymbol{Y}_k\boldsymbol{Y}_k^T$, to estimate dimension. This means calculating the following

$$\hat{d}_k = \arg\max_i |\lambda_i(S_k) - \lambda_{i+1}(S_k)| \tag{25}$$

where $\lambda_i(S_k)$ denotes the $i$th eigenvalue of $S_k$ assuming $\lambda_1 \geq \dots \geq \lambda_D$. For heteroscedastic data, the eigengap heuristic can break down, making it challenging to determine rank, as shown in Sec. V, Fig. 5. In recent work, [34] developed a parallel analysis algorithm to estimate the rank of a matrix that is consistently shown to work well in the heteroscedastic regime if the data comes from **one** subspace only. It works by creating an i.i.d. Bernoulli $(p = 0.5)$ matrix denoted as $\boldsymbol{M}$ and analyzing the singular values of $\boldsymbol{M} \odot \boldsymbol{Y}$ for a matrix of interest $\boldsymbol{Y}$. One can distinguish what singular values are associated with the signal and noise component of the data through this process. We generalize this line of work and apply it to the union of subspace setting by starting off over-parameterized and adaptively shrinking subspace bases as follows:

$$\begin{aligned} &\forall k \in \{1,\dots,K\} \ \text{ do} \\ &\tilde{\sigma}^{(r)} = \text{SingularValues}(\boldsymbol{M} \odot \boldsymbol{Y}_k) \ \ \forall r \in \{1,\dots,R\} \qquad (26) \\ &\hat{d}_k = \text{smallest } d \text{ that satisfies} \\ &\quad \sigma_{d+1}(\boldsymbol{Y}_k) \leq \alpha\text{-percentile of } \{\tilde{\sigma}_{d+1}^{(1)},\dots,\tilde{\sigma}_{d+1}^{(R)}\}. \qquad (27) \end{aligned}$$

This is done for each estimated cluster $\boldsymbol{Y}_k$ over $R$ random trials where $1 \leq R \ll T_2 =$ ALPCAHUS maximum total iterations. Here $\sigma_{d+1}(\boldsymbol{Y}_k)$ denotes the $d+1$th singular value of $\boldsymbol{Y}_k$. This process is repeated for all cluster subsets $\{\boldsymbol{Y}_1,\dots,\boldsymbol{Y}_K\}$ after the cluster reassignment update in (17). To reduce computation, this dimension estimation is performed sparingly every 10% of ALPCAHUS iterations.

### C. Cluster Initialization

For the non-ensemble version of ALPCAHUS ($B = 1$), it previously remained to be seen whether there exists an initialization scheme that performs better in expectation than random cluster assignment in the heteroscedastic regime. In recent work, [26] proposes a thresholding inner-product based spectral initialization method (TIPS), designed for homoscedastic data to be used with KSS, where an affinity matrix is generated by

$$\boldsymbol{W}_{ij} = 1 \text{ if } |\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle| \geq \tau \text{ and } i \neq j \tag{28}$$

given a thresholding parameter $\tau > 0$ and zero otherwise. Then, the cluster assignments $\mathcal{C} = \{c_1, \ldots, c_N\}$ are calculated by applying the spectral clustering method on $\boldsymbol{W}$. Recall that $\boldsymbol{y}_i = \boldsymbol{x}_i + \boldsymbol{\epsilon}_i$. Upon closer analysis, this metric in expectation, ignoring absolute value, gives

$$\begin{aligned}\mathbb{E}[\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle] &= \mathbb{E}[\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle] + \mathbb{E}[\langle \boldsymbol{x}_i, \boldsymbol{\epsilon}_j \rangle] \\ &+ \mathbb{E}[\langle \boldsymbol{\epsilon}_i, \boldsymbol{x}_j \rangle] + \mathbb{E}[\langle \boldsymbol{\epsilon}_i, \boldsymbol{\epsilon}_j \rangle] = \mathbb{E}[\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle]. \end{aligned} \tag{29}$$

Therefore, the metric is independent of the noise variances, meaning the affinity matrix constructed does not have highly unbalanced, asymmetric edge weights from noisy samples. Thus this metric is more robust to heteroscedastic noise than others such as Euclidean norm where

$$\mathbb{E}[\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2] = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 + D(\boldsymbol{\nu}_i + \boldsymbol{\nu}_j). \tag{30}$$

Clearly, this is inflated by the noise terms $\boldsymbol{\nu}_i$ and $\boldsymbol{\nu}_j$. To note, TIPS initialization is only useful when $B = 1$ for one base clustering since it is a deterministic initialization; it provably performs better than random initialization as illustrated in Fig. 4. Otherwise, random initialization is used when $B > 1$ to leverage consensus information in the ensemble process.

### D. ALPCAHUS Convergence

ALPCAHUS with a single cluster ($K = 1$) has a sequence of cost function values that provably converges to local minima, since the cost function and algorithm simplifies to LR-ALPCAH (14) that has convergence guarantees shown in [17, Thm. 2]. In the multi-cluster setting ($K > 1$), Thm. 1 shows that one can guarantee convergence of the sequence of cost function values in (15) by ensuring two things: that the noise variances are kept above some small threshold value, and that the cluster assignment criteria makes assignment changes only if the cost function strictly decreases. The argument in Thm. 1 below is a natural extension of that in the original k-subspaces paper [25, Thm. 7]. The Appendix provides an experimental result, Fig. 9, that corroborates this theorem.

**Theorem 1.** *Consider the non-ensemble version of the ALPCAHUS algorithm. Assume a stopping criteria where the loop over $t_2$ in the algorithm continues until there is an iterate where the cost function does not decrease. Assume a noise variance threshold parameter $\alpha \in \mathbb{R} > 0$ that lower bounds all $\nu_i$ and a cluster assignment criteria that accepts changes only if there is a cluster reassignments of points that strictly decrease the cost function, as expressed in (17). Then, the ALPCAHUS algorithm generates a sequence of cost function values that converges to a fixed point.*

*Proof.* To prove that the sequence of cost function values converges, one needs to assess whether each step decreases the cost function, and whether the algorithm terminates at a fixed point where the variables are no longer updated. Each step consists of two sub-steps: first, the subspace estimation sub-step where LR-ALPCAH is used, and second, the clustering sub-step where projection onto the estimated subspaces is done to cluster with respect to the smallest residual. For the subspace estimation sub-step, with cluster assignments fixed, it is known that LR-ALPCAH has a sequence of cost function values that will converge according to guarantees given in [17, Thm. 2]. For the clustering sub-step, given our stopping criteria, once the clusters do not change, the algorithm has a sequence of cost function values that reaches a fixed point. That leaves only the case where the clusters do change, which will be the focus in the remainder of the proof. Let $\boldsymbol{Y}_k^{(t)}, \boldsymbol{L}_k^{(t)}, \boldsymbol{R}_k^{(t)}, \boldsymbol{\Pi}_k^{(t)}$ be the optimization variables that produce the cost

$$\begin{aligned}\text{cost}^{(t_1,t_2)} &= \sum_k \frac{1}{2} \|(\boldsymbol{Y}_k^{(t_2)} - \boldsymbol{L}_k^{(t_1)}[\boldsymbol{R}_k^{(t_1)}]^T)[\boldsymbol{\Pi}_k^{(t_1)}]^{-\frac{1}{2}}\|_F^2 \\ &+ \frac{D}{2} \log |\boldsymbol{\Pi}_k^{(t_1)}| \end{aligned} \tag{31}$$

where $t_1$ is associated with the variables in the subspace estimation sub-step and $t_2$ is associated with the variables in the clustering sub-step. Recall that the clustering is produced by solving

$$c_i = \arg\min_k \|\boldsymbol{y}_i - \boldsymbol{U}_k^{(t_1)}[\boldsymbol{U}_k^{(t_1)}]^T \boldsymbol{y}_i\|_2^2 \tag{32}$$

for all data points $y_i$, where $\boldsymbol{U}_k^{(t_1)}$ is found by performing an SVD on $\boldsymbol{L}_k^{(t_1)}$ or by $\boldsymbol{U}_k^{(t_1)} = \boldsymbol{L}_k^{(t_1)}[\boldsymbol{L}_k^{(t_1)}]^\dagger$. Under this setting, there is at least one $\boldsymbol{y}_i$ with label $c_i^{(t_2+1)} \neq c_i^{(t_2)}$. For those points, denote the previous cluster assignment as $\hat{k}$. Due to the cluster assignment criteria, it is know that

$$\begin{aligned}&\min_k \|\boldsymbol{y}_i - \boldsymbol{U}_k^{(t_1+1)}[\boldsymbol{U}_k^{(t_1+1)}]^T \boldsymbol{y}_i\|_2^2 \\ &\leq \|\boldsymbol{y}_i - \boldsymbol{U}_{\hat{k}}^{(t_1+1)}[\boldsymbol{U}_{\hat{k}}^{(t_1+1)}]^T \boldsymbol{y}_i\|_2^2 \qquad (33) \\ &\implies (\boldsymbol{y}_i \in \mathcal{C}_{\hat{k}}^{(t_2)} \implies \boldsymbol{y}_i \in \mathcal{C}_k^{(t_2+1)}). \qquad (34)\end{aligned}$$

It is easy to see that $\boldsymbol{X}_k = \boldsymbol{L}_k \boldsymbol{R}_k^T = \boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{Y}_k$ since the projection of $\boldsymbol{Y}_k$ onto $\mathcal{R}(\boldsymbol{U}_k)$ is $\boldsymbol{X}_k$. The cost can be expressed as

$$\begin{aligned}\text{cost}^{(t_1,t_2)} &= \sum_k \frac{1}{2} \|(\boldsymbol{Y}_k^{(t_2)} - \boldsymbol{X}_k^{(t_1)})[\boldsymbol{\Pi}_k^{(t_1)}]^{-1/2}\| \\ &+ \frac{D}{2} \log |\boldsymbol{\Pi}_k^{(t_1)}|. \end{aligned} \tag{35}$$

Then, the following observation is made regarding the cost

function written in (35).

$$\text{cost}^{(t_1,t_2)} = \sum_k \sum_{i\in\mathcal{C}_k^{(t_2)}} \frac{1}{\sqrt{\nu_i^{(t_1)}}} \|[\boldsymbol{Y}_k]_i - [\boldsymbol{X}_k]_i^{(t_1)}\|_2^2 + \frac{D}{2}\log|\boldsymbol{\Pi}_k^{(t_1)}| \tag{36}$$

$$\geq \sum_k \sum_{i\in\mathcal{C}_k^{(t_2)}} \frac{1}{\sqrt{\nu_i^{(t_1+1)}}} \|[\boldsymbol{Y}_k]_i - [\boldsymbol{X}_k]_i^{(t_1+1)}\|_2^2 + \frac{D}{2}\log|\boldsymbol{\Pi}_k^{(t_1+1)}| = \text{cost}^{(t_1+1,t_2)} \tag{37}$$

$$\geq \sum_k \sum_{i\in\mathcal{C}_k^{(t_2+1)}} \frac{1}{\sqrt{\nu_i^{(t_1+1)}}} \|[\boldsymbol{Y}_k]_i - [\boldsymbol{X}_k]_i^{(t_1+1)}\|_2^2 + \frac{D}{2}\log|\boldsymbol{\Pi}_k^{(t_1+1)}| = \text{cost}^{(t_1+1,t_2+1)} \tag{38}$$

Here, (37) is due to the subspace step not increasing the cost as shown in [17, Thm. 2], and (38) is due to the argument above in (33). What has been shown is that

$$\text{cost}^{(t_1,t_2)} \geq \text{cost}^{(t_1+1,t_2)} \geq \text{cost}^{(t_1+1,t_2+1)}. \tag{39}$$

Therefore, as the cluster assignments and subspace bases are updated, then the cost will either decrease or remain the same. Once the cost remains the same, then the algorithm has a sequence of cost function values that converges to a fixed point with locally optimal cluster labels. □

## V. Experiments

### A. Synthetic Experiments

*1) Experimental Setup:* A synthetic dataset is generated consisting of $K = 2$ clusters each of dimension $d = 3$ derived from random subspaces in $D = 100$ dimensional ambient space. Each cluster consisted of two data groups with group 1 containing $N_1 = 6$ samples, to explore the data constrained regime, with noise variance $\nu_1 = 0.1$ per cluster. For group 2, the $N_2$ samples and noise variance $\nu_2$ are varied. Cross validation is done for hyperparameters in any algorithms that require it with a separate training set. Both the estimated clusters from each clustering algorithm and the known clusters are necessary to calculate clustering error. Additionally, the problem of label permutations is dealt by using the Hungarian algorithm [35]. More concretely, clustering error is defined as

$$\text{clustering error} = \frac{100}{N}\left(1 - \max_{\pi}\sum_{i,j} \boldsymbol{Q}^{\text{out}}_{\pi(ij)}\boldsymbol{Q}^{\text{true}}_{ij}\right) \tag{40}$$

where $\pi$ is a permutation of cluster labels, and $\boldsymbol{Q}^{\text{out}}$ and $\boldsymbol{Q}^{\text{true}}$ are output labels and ground truth labelings of the data where the $(i,j)$th entry is one if point $j$ belongs to cluster $i$ and zero otherwise.

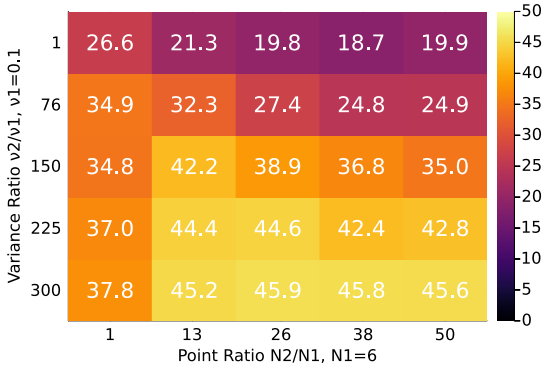
The average clustering error is computed out of 100 trials where each trial had different noise, basis coefficients, and subspace basis realizations for the clusters. Various subspace clustering algorithms are compared such as $K$-Subspaces (KSS) [14], Ensemble $K$-Subspaces (EKSS) [16], Subspace Clustering via Thresholding (TSC) [36], and Doubly Stochastic Sparse Subspace Clustering (ADSSC) [37]. Applying general non-subspace clustering methods such as $K$-means [38] resulted in poor clustering error so results of these methods are generally not shown.

*2) Clustering Error:* Table 2h shows the clustering quality of these algorithms for synthetic heteroscedastic data. Fig. 2 complements Table 2h by exploring the complete heteroscedastic landscape with a visual representation that is easier to understand. We define a method called "noisy oracle" where an oracle uses the true cluster assignments to apply PCA on the low noise data only per each cluster. Using these subspace basis estimates of each cluster, the oracle performs cluster assignments using (17). This oracle reference provides a kind of lower bound on realistic clustering performance since it can approximate subspace bases separately given the true labeling.

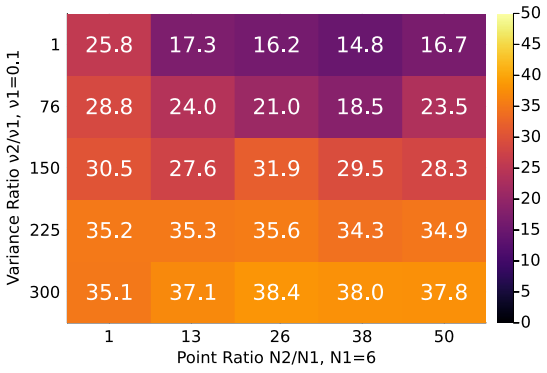
ALPCAHUS ($B = 1$) with TIPS initialization achieved lower clustering error than KSS with TIPS initialization. For the ensemble methods ($B > 1$), both EKSS and ALPCAHUS significantly improved. However, EKSS still had higher clustering error in more heteroscedastic regions. Meanwhile, ALPCAHUS remained very close to the noisy oracle, indicating it more accurately estimated the underlying true labeling. Compared to other methods like TSC and ADSSC, ALPCAHUS generally outperformed them with one exception for ADSSC ($N_2 = 1, \nu_2/\nu_1 = 300$). Here, ADSSC performed better than ALPCAHUS when the number of noisy points is roughly equal to the number of good points. Overall, the ensemble version of ALPCAHUS was generally more robust against heteroscedasticity compared to other subspace clustering algorithms. To summarize, the effects of data quality and data quantity on the heteroscedastic subspace clustering estimates is explored in different situations. To our knowledge, this is the first systematic analysis of heteroscedasticity effects on subspace clustering quality in the literature.

*3) Effects of Good Data:* Additionally, the effects of good data quantity on the subspace clustering problem is explored to understand at what point it becomes advantageous to use ALPCAHUS over KSS methods. The following parameters are fixed $\nu_1 = 0.1, N_2 = 500$ and the following vary $N_1, \nu_2$. The reporting metric is the percentage difference of EKSS clustering error (%) - ALPCAHUS clustering error (%), i.e., $\text{error}_{\text{EKSS}} - \text{error}_{\text{ALPCAHUS}}$, meaning higher values indicate our method is better. Figure 3 shows that ALPCAHUS performed better than EKSS even up to $N_1 = 160$ which represents about 33% of the total data. ALPCAHUS never performed worse than EKSS but the advantage gap narrowed as $N_1$ increased. Thus, there is a wide range of conditions for which ALPCAHUS is preferable to homoscedastic methods.

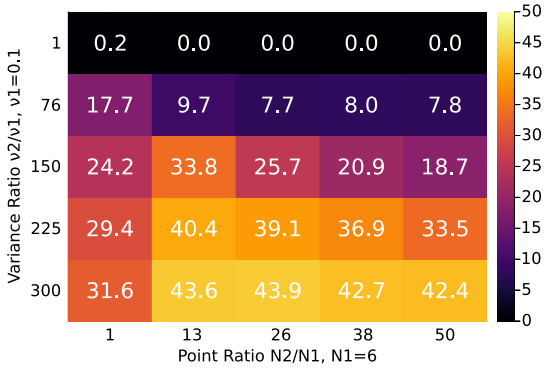
*4) Clustering Initialization:* Section IV proposed using the TIPS initialization scheme in this heteroscedastic context since the dot product metric used in constructing the affinity matrix in (29) is provably robust to heteroscedastic noise. The following parameters were fixed $N_1, N_2$ and the following varied $\nu_2$ while keeping $\nu_1 = 0.1$. The noisy oracle result is included to establish a realistic performance baseline. Figure 4 shows that TIPS initialization ALPCAHUS ($B = 1$) outperformed random initialization for the non-ensemble ALPCAHUS method ($B = 1$) across the entire heteroscedastic landscape. Thus, TIPS should always be used for the non-ensemble methods for clustering quality. The ensemble version of ALPCAHUS



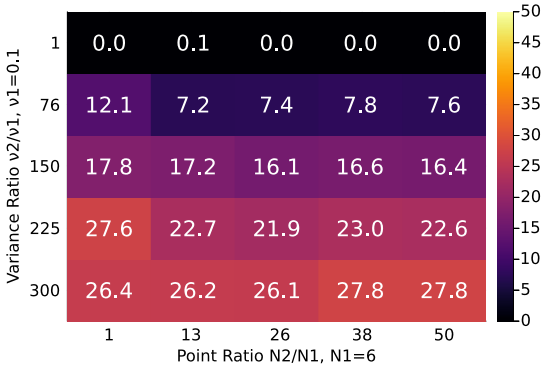

(a) KSS (TIPS) mean clustering error.



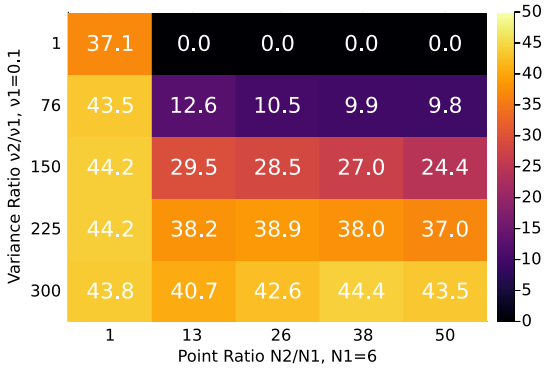

(b) ALPCAHUS ($B = 1$, TIPS) mean clustering error.



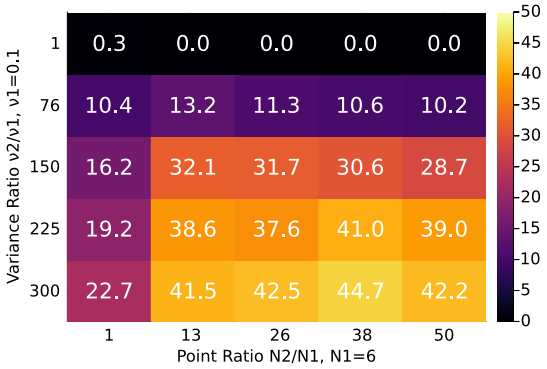

(c) EKSS (B=128) mean clustering error.



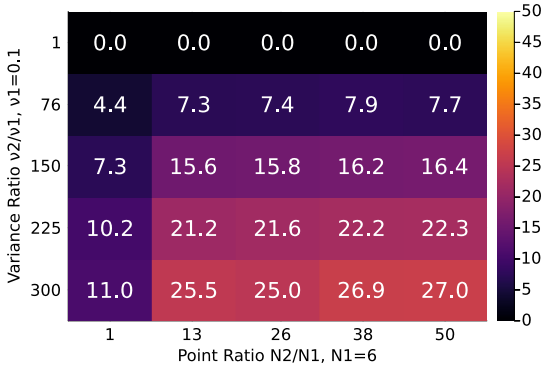

(d) ALPCAHUS (B=128) mean clustering error.




(e) TSC mean clustering error.




(f) ADSSC mean clustering error.




(g) Noisy oracle mean clustering error.

| $\nu_2/\nu_1$ | 1 | 1 | 300 | 300 | 150 | 225 | 76 |
|---|---|---|---|---|---|---|---|
| $N_2/N_1$ | 1 | 50 | 1 | 50 | 26 | 13 | 38 |
| Reference Level | | | | | | | |
| Noisy Oracle | 0.0 | 0.0 | 11.0 | 27.0 | 15.8 | 21.2 | 7.9 |
| Method Comparisons | | | | | | | |
| KSS | 26.6 | 19.9 | 37.8 | 45.6 | 38.9 | 44.4 | 24.8 |
| EKSS ($B = 128$) | 0.2 | **0.0** | 31.6 | 42.4 | 25.7 | 40.4 | 8.0 |
| ADSSC | 0.3 | **0.0** | **22.7** | 42.2 | 31.7 | 38.6 | 10.6 |
| TSC | 37.1 | **0.0** | 43.8 | 43.5 | 28.5 | 38.2 | 9.9 |
| ALPCAHUS ($B = 1$) | 25.8 | 16.7 | 35.1 | 37.8 | 31.9 | 35.3 | 18.5 |
| ALPCAHUS ($B = 128$) | **0.0** | **0.0** | 26.4 | **27.8** | **16.1** | **22.7** | **7.8** |

(h) Clustering error (%) results for heteroscedastic landscape. Visual results included in Fig. 2. Columns consist of the 4 corners and the 3 diagonal elements of the heatmaps in Fig. 2.

Fig. 2: Clustering error over the heteroscedastic landscape for various subspace clustering algorithms.

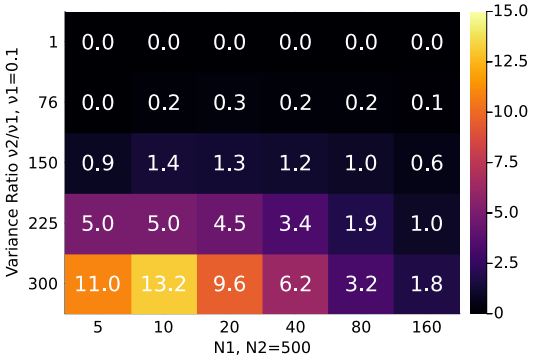



Fig. 3: Percentage difference (%) of ALPCAHUS clustering error subtracted from EKSS while good data amount varies.

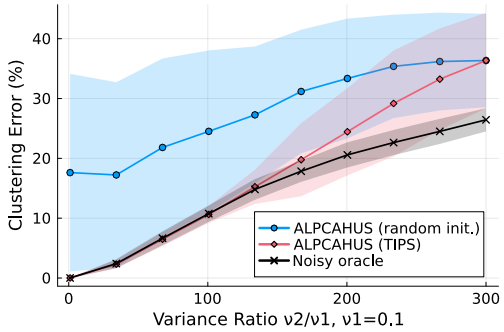



Fig. 4: Clustering error (%) for TIPS initialization scheme vs. random initialization for the ALPCAHUS method ($B = 1$).

($B > 1$) is not included in these results because the ensemble process inherently takes advantage of random initialization to achieve a different labeling each trial.

*5) Rank Estimation of Clusters:* In Sec. IV, it is proposed to use random sign flipping to adaptively find and shrink the subspace dimension of clusters when subspace dimension is unknown. For Fig. 5 synthetic subspaces are generated with true rank $d = 6$ to explore how the initial rank parameter affects the ability to estimate the true rank over 100 trials. This approach is compared against the eigengap heuristic by using ALPCAHUS with both adaptive rank schemes and report the estimated rank values from the final clustering. The eigengap approach consistently underestimated the rank regardless of the initial value, except when given the true rank value $d = 6$. Our sign flipping approach provided much better rank estimates with much smaller variances between trials.

### B. Real Data Experiments

*1) Quasar Flux Data:* Quasar spectra data from the Sloan Digital Sky Survey (SDSS) Data Release 16 [39] is investigated 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 for our experiments 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

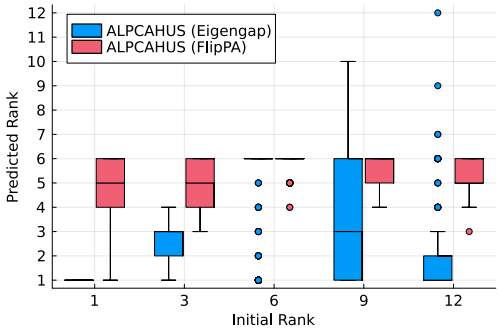
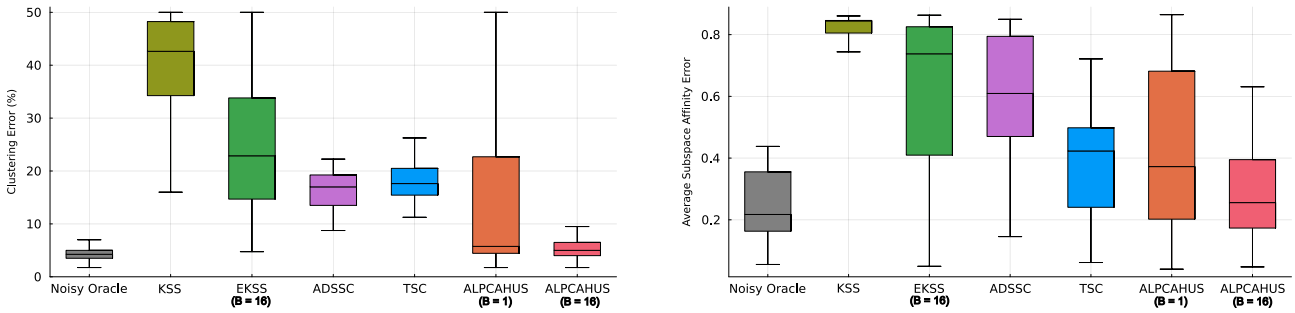



Fig. 5: Adaptive rank estimation using eigengap heuristic and proposed FlipPA approach (true rank $d = 6$).

[39]. In Fig. 7, a subset of the spectra data is shown for illustrative purposes to compare the visual differences in data quality. The preprocessing pipeline for the data included (filtering, interpolation, centering, and normalization) based on the supplementary material of [41]. Clusters are formed in these experiments by considering quasars with different properties, namely, two quasar groups ($K = 2$) with different redshift values ($z_1 = 1.0 - 1.1, z_2 = 2.0 - 2.1$). Additionally, for the second group, a query for broad absorption lines (BAL) type quasars is done. Since the downloaded data is queried separately, one knows which points belong to which cluster group, meaning one can compute clustering error for comparison purposes. A training set (5000 samples total) is set aside to learn any model parameters and the rest (5000 samples total) is used for a test dataset. Trials are formed by randomly selecting 400 quasar spectra samples per group and running clustering algorithms for 50 total trials. For rank estimation, the noisy oracle approach is used to find that there is no improvement to clustering quality for rank values greater than $\hat{d} = 3$, so this value is used for any applicable algorithms that require it.

Figure 6a shows clustering error for this quasar spectra multi-cluster data. The ensemble version of ALPCAHUS was very close in clustering quality to the noisy oracle, suggesting that it accurately learned the subspace bases while clustering the data groups. The non-ensemble version of ALPCAHUS ($B = 1$) had large variances in clustering error, indicating some challenges in getting close to the optimal cost function minima with this data. However, based on median values, it still managed to get a lower error than KSS, ADSSC, and TSC. Additionally, Fig. 6b shows the average subspace affinity error of these methods after applying LR-ALPCAH to the clusterings for all methods. ALPCAHUS better learned the subspace bases than the other methods when $B = 16$, which was chosen as the smallest value that had small run-to-run variances while not taking significantly longer to compute. Both of these results show the benefit of developing heteroscedastic specific algorithms in the subspace clustering context. Table I reports median time complexity and memory requirements along with mean clustering error and mean subspace error. In this data instance, the ensemble method gets close in time to the non-ensemble methods due to our multi-threaded implementation. Yet, because of the multi-threaded implementation, the memory




(a) Subspace clustering results for various methods.

(b) Average subspace affinity error results for various methods.

Fig. 6: Experimental results of quasar flux data for subspace clustering and learning. Methods involving a single run, i.e., KSS and ALPCAHUS ($B = 1$), use the TIPS initialization scheme.

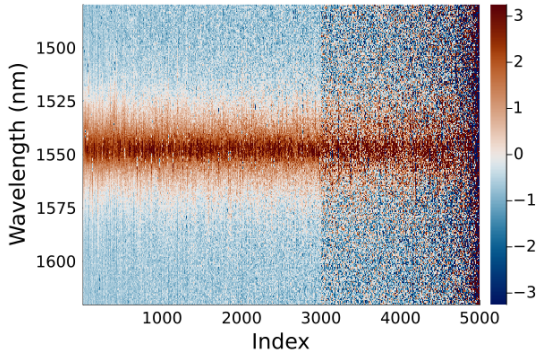



Fig. 7: Sample data matrix of quasar flux measurements across wavelengths for each column-wise sample.

| Methods | Time (ms) | Memory (MiB) | Mean Subspace Error | Mean Clustering Error (%) |
|---|---|---|---|---|
| KSS | 149.3 | 53.0 | 0.80 | 38.5 |
| EKSS (B=16) | 181.4 | 212.6 | 0.62 | 23.8 |
| ADSSC | 125.7 | 17.5 | 0.62 | 21.1 |
| TSC | **32.3** | **16.5** | 0.40 | 18.2 |
| ALPCAHUS (B=1) | 148.5 | 126.7 | 0.43 | 14.4 |
| ALPCAHUS (B=16) | 210.7 | 678.2 | **0.28** | **6.3** |

TABLE I: Subspace clustering results on quasar flux data. The KSS and ALPCAHUS ($B = 1$) methods use TIPS initialization.

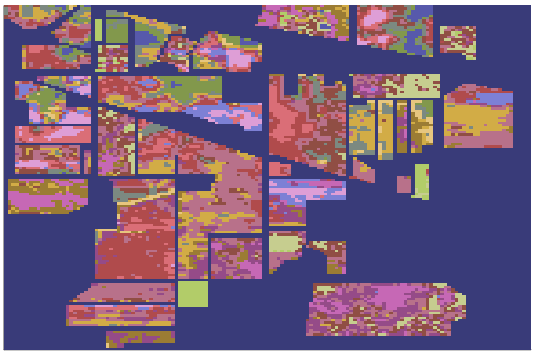
requirements are larger for the ensemble method as opposed to the non-ensemble method. Overall, relative to other clustering methods, ALPCAHUS is competitive time-wise at the cost of increased memory requirements.

*2) Indian Pines Data:* Hyperspectral image (HSI) segmentation data, called *Indian Pines* [42], is explored in this section. This image of size $145 \times 145$ contains $D = 200$ reflectance bands for each pixel that is around the $0.4 - 2.5\mu m$ wavelength. HSI data is known to be very noisy due to thermal effects, atmospheric effects, and camera electronics, leading to works that study per-pixel noise estimation in hyperspectral imaging [43]. In total, there are $K = 16$ classes composed of alfalfa, corn, grass, wheat, soybean, and others. There is one additional class (background) that is withheld from clustering as commonly done [33]. Therefore, $N = 10,249$ instead of $N = 145 \times 145$. In other works, researchers have found that $\hat{d} = 5$ is an appropriate rank parameter since it covers $95\%$ of the cumulative variance [33]. A subset of the data matrix $Y \in \mathbb{R}^{200 \times 10249}$ is split into $2,500$ samples to learn model parameters. From this, the learned subspaces are applied on $Y$ to cluster all data so that the heatmaps can be compared against the ground truth. The ground truth image is found in Fig. 8a. For perspective, randomly guessing would lead to 94% clustering error due to $K = 16$ clusters. For consistency, the Hungarian algorithm is again applied to the clustering results in Fig. 8 to make it easier to compare against the ground truth label image in Fig. 8a.

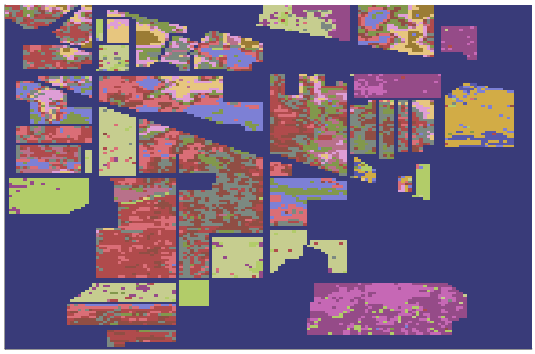
$K$-means is compared against in Fig. 8b to illustrate the difficulty of the problem. ALPCAHUS results are shown in Fig. 8d next to EKSS results in Fig. 8c. Clustering error is reported for these algorithms along with mean IOU to measure the similarity between the images. ALPCAHUS achieved the lowest clustering error and highest mIOU (53% / 31%) relative to the other approaches such as $K$-means (77% / 16%) and EKSS (64% / 27%). We note that our ALPCAHUS results are similar to [33] even though, in that work, they instead treat the noise as homoscedastic and instead model the signal as heteroscedastic. Further, the EKSS results on this dataset are similar to the homoscedastic PCA approach used in [33]. This indicates some utility in modeling heteroscedasticity in hyperspectral images. Yet, the reflectance bands themselves also appear to be heteroscedastic with some bands being noisier than others [44]. Thus, developing a method that is doubly heteroscedastic with respect to both the samples and features is an interesting direction of future work.

For reference, the SOTA result is about 10% misclassification rate in a *classification* setting (i.e., not clustering) [45]. In a clustering setting, recent works such as [46] have achieved a 40% clustering error which is 13% lower than our results. We do not claim SOTA results with HSI data, only that heteroscedastic union of subspace modeling improved results over homoscedastic union of subspace modeling in finding reflectance band groups with similar characteristics.

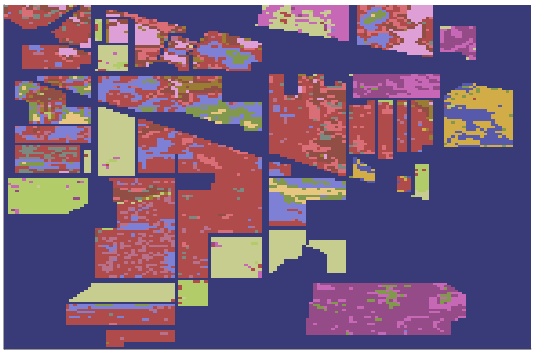

(a) Ground truth labels for *Indian Pines* dataset. NC means no class available (background).



(b) $K$-means results with clustering error = 77% and mIOU = 16%.



(c) EKSS ($B = 32, T = 3, q = K$) results with clustering error = 64% and mIOU = 27%.



(d) ALPCAHUS ($B = 32, T = 3, q = K$) results with clustering error = 53% and mIOU = 31%.

Fig. 8: Experimental results of *Indian Pines* HSI data in a subspace clustering context. Results reported are clustering error and mean IOU (intersection over union) for each algorithm.

## VI. Conclusion

This paper proposed ALPCAHUS, a subspace clustering algorithm that can find subspace clusters whose samples contain heteroscedastic noise. For future work, a union of manifolds generalization could possibly be useful. For example, Alzheimer's disease patients often have resting state functional MRI activations different than cognitively normal individuals [47], so one could consider these different classes to belong to different manifolds in the ambient space. Previous research has shown manifold learning to more correctly model temporal dynamics than subspace modeling in this domain [48]. Another direction of future work could be to consider heteroscedasticity across the feature space instead of the sample space. For example, one might be interested in biological sequencing of different species with similar genes where the gene marker counts naturally follow heteroscedastic distributions [49]. These generalizations are nontrivial so it is left for future work.

While our implementation of ALPCAHUS benefited from parallelization, it needed more computation time and memory than TSC. Because of this, there is an opportunity to further optimize our code base to remove certain matrix multiplication operations and instead use single vector dot products to reduce memory. Further, since our subspace basis step requires an SVD computation for an initial low-rank estimate, perhaps the Krylov-based Lanczos algorithm [50] can be used to further reduce time and memory for each trial leading to more significant improvements in memory usage and time. Overall, ALPCAHUS was more robust towards heteroscedastic noise than other clustering methods, making it easier to identify correct clusterings under this kind of heteroscedastic noise model.

## VII. Acknowledgments

This work was supported in part by NSF CAREER Grant CCF-1845076 and NSF Grant CCF-2331590. There are no financial conflicts of interest in this work. We thank David Hong for suggesting heteroscedastic data applications such as astronomy spectra used in this work.

## VIII. Appendix

### A. ALPCAHUS Convergence Experiment

This section focuses on providing empirical verification of the ALPCAHUS convergence theorem in Thm. 1. The quasar spectra flux data from SDSS (DR16Q catalog) is used in this experiment. The noise variance threshold parameter $\alpha$ is set to $10^{-6}$ to lower bound the noise variance estimation. Further, as shown in Thm. 1, it is necessary to use the cluster reassignment criteria in (17) that rejects repeated assignments of points to prevent cycling. For simplicity, only one base trial ($B = 1$) is run in this experiment for plotting purposes and to compare

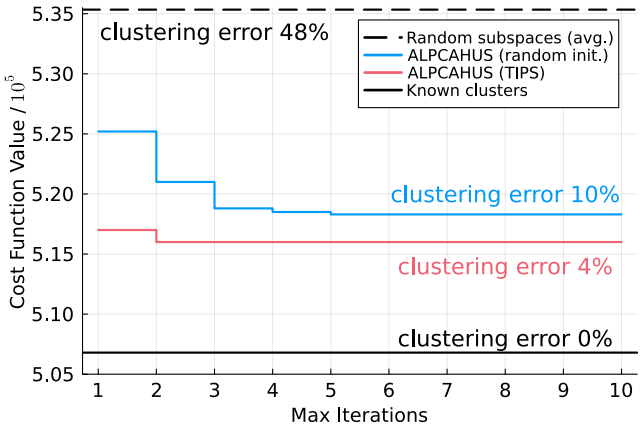



Fig. 9: ALPCAHUS ($B = 1$) cost function value convergence plot to corroborate the theorem in Thm. 1.

against the TIPS initialized variant of ALPCAHUS. This variant is only possible when $B = 1$ as discussed in the paper. In Fig. 9, the cost function value is plotted over iterations for both versions of the ALPCAHUS method.

Both upper and lower bound cost function reference values are provided in Fig. 9. The upper bound is generated by first using randomly initialized subspaces as the true subspaces and secondly, using (17) to get cluster labels. From this, the cost function in (15) is computed. Likewise, a lower bound is generated by first using the known cluster labels and secondly, estimating the subspaces and noise variances with (16). As observed in Fig. 9, ALPCAHUS converges relatively quickly with only a few iterations. Note that in this figure, the stopping criteria is not implemented. In other words, ALPCAHUS is run for a fixed number of iterations. However, as clearly demonstrated, convergence to a fixed point is achieved earlier for both versions of ALPCAHUS before reaching the maximum number of iterations. This indicates the usefulness of the stopping criteria to speed up the algorithm in various applications.

## 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," *Artificial intelligence in medicine*, vol. 103, p. 101786, 2020.

[2] Q. Jia, J. Cai, X. Jiang, and S. Li, "A subspace ensemble regression model based slow feature for soft sensing application," *Chinese Journal of Chemical Engineering*, 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 Transactions on Image Processing*, vol. 30, pp. 7170–7183, 2021.

[4] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[5] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[6] R. Vidal, R. Tron, and R. Hartley, "Multiframe motion segmentation with missing data using PowerFactorization and GPCA," *International Journal of Computer Vision*, vol. 79, pp. 85–105, 2008.

[7] W. Hong, J. Wright, K. Huang, and Y. Ma, "Multiscale hybrid linear models for lossy image representation," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3655–3671, 2006.

[8] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "An algebraic geometric approach to the identification of a class of linear hybrid systems," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 1. IEEE, 2003, pp. 167–172.

[9] [Online]. Available: https://www.epa.gov/outdoor-air-quality-data

[10] P. Tsalmantza and D. W. Hogg, "A data-driven model for spectra: Finding double redshifts in the sloan digital sky survey," *The Astrophysical Journal*, vol. 753, no. 2, p. 122, 2012.

[11] Y. Cao, A. Zhang, and H. Li, "Multisample estimation of bacterial composition matrices in metagenomics data," *Biometrika*, vol. 107, no. 1, pp. 75–92, 12 2019. [Online]. Available: https://doi.org/10.1093/biomet/asz062

[12] D. Hong, L. Balzano, and J. A. Fessler, "Asymptotic performance of PCA for high-dimensional heteroscedastic data," *J. Multivar. Anal.*, vol. 167, no. C, p. 435–452, sep 2018. [Online]. Available: https://doi.org/10.1016/j.jmva.2018.06.002

[13] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[14] P. K. Agarwal and N. H. Mustafa, "K-means projective clustering," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2004, pp. 155–165.

[15] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6320–6342, 2015.

[16] J. Lipor, D. Hong, Y. S. Tan, and L. Balzano, "Subspace clustering using ensembles of k-subspaces," *Information and Inference: A Journal of the IMA*, vol. 10, no. 1, pp. 73–107, 2021.

[17] J. S. Cavazos, J. A. Fessler, and L. Balzano, "ALPCAH: Subspace learning for sample-wise heteroscedastic data," *Transactions on Signal Processing (TSP)*, pp. 1–12, 2025.

[18] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, pp. 159–179, 1998.

[19] L. Lu and R. Vidal, "Combined central and subspace clustering for computer vision applications," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 593–600.

[20] S. R. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[21] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.

[22] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1796–1808, 2011.

[23] W. Qu, X. Xiu, H. Chen, and L. Kong, "A survey on high-dimensional subspace clustering," *Mathematics*, vol. 11, no. 2, p. 436, 2023.

[24] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014.

[25] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *Journal of Global optimization*, vol. 16, pp. 23–32, 2000.

[26] P. Wang, H. Liu, A. M.-C. So, and L. Balzano, "Convergence and recovery guarantees of the k-subspaces method for subspace clustering," in *International Conference on Machine Learning*. PMLR, 2022, pp. 22 884–22 918.

[27] A. Gitlin, B. Tao, L. Balzano, and J. Lipor, "Improving $k$-subspaces via coherence pursuit," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1575–1588, 2018.

[28] M. E. Timmerman, E. Ceulemans, K. De Roover, and K. Van Leeuwen, "Subspace k-means clustering," *Behavior research methods*, vol. 45, pp. 1011–1023, 2013.

[29] J. Ghosh and A. Acharya, "Cluster ensembles," *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, vol. 1, no. 4, pp. 305–315, 2011.

[30] D. Hong, K. Gilman, L. Balzano, and J. A. Fessler, "HePPCAT: Probabilistic PCA for data with heteroscedastic noise," *IEEE Transactions on Signal Processing*, vol. 69, pp. 4819–4834, 2021.

[31] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Transactions on Signal Processing*, vol. 67, no. 20, pp. 5239–5269, 2019.

[32] A. R. Zhang, T. T. Cai, and Y. Wu, "Heteroskedastic PCA: Algorithm, optimality, and applications," *The Annals of Statistics*, vol. 50, no. 1, pp. 53–80, 2022.

[33] 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 Transactions on Signal Processing*, vol. 69, pp. 6546–6560, 2021.

[34] D. Hong, Y. Sheng, and E. Dobriban, "Selecting the number of components in PCA via random signflips," 2023.

[35] M. Wright, "Speeding up the hungarian algorithm," *Computers & Operations Research*, vol. 17, no. 1, pp. 95–96, 1990.

[36] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6320–6342, 2015.

[37] D. Lim, R. Vidal, and B. D. Haeffele, "Doubly stochastic subspace clustering," *arXiv preprint arXiv:2011.14859*, 2020.

[38] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020.

[39] R. Ahumada, C. A. Prieto, A. Almeida, F. Anders, S. F. Anderson, B. H. Andrews, B. Anguiano, R. Arcodia, E. Armengaud, M. Aubert *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," *The Astrophysical Journal Supplement Series*, vol. 249, no. 1, p. 3, 2020.

[40] B. W. Lyke, A. N. Higley, J. McLane, D. P. Schurhammer, A. D. Myers, A. J. Ross, K. Dawson, S. Chabanier, P. Martini, H. D. M. Des Bourboux *et al.*, "The sloan digital sky survey quasar catalog: Sixteenth data release," *The Astrophysical Journal Supplement Series*, vol. 250, no. 1, p. 8, 2020.

[41] D. Hong, F. Yang, J. A. Fessler, and L. Balzano, "Optimally weighted PCA for high-dimensional heteroscedastic data," *SIAM Journal on Mathematics of Data Science*, vol. 5, no. 1, pp. 222–250, 2023.

[42] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3," Sep 2015. [Online]. Available: https://purr.purdue.edu/publications/1947/1

[43] A. Mahmood and M. Sears, "Per-pixel noise estimation in hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[44] S. Zhang, X. Kang, Y. Mo, and S. Li, "Noise analysis of hyperspectral images captured by different sensors," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 2707–2710.

[45] D. Uchaev and D. Uchaev, "Small sample hyperspectral image classification based on the random patches network and recursive filtering," *Sensors*, vol. 23, no. 5, p. 2499, 2023.

[46] S. Huang, H. Zhang, Q. Du, and A. Pi$\boldsymbol{v}$zurica, "Sketch-based subspace clustering of hyperspectral images," *Remote Sensing*, vol. 12, no. 5, p. 775, 2020.

[47] R. Sperling, "The potential of functional MRI as a biomarker in early alzheimer's disease," *Neurobiology of aging*, vol. 32, pp. S37–S43, 2011.

[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, p. 118423, 2021.

[49] R. K. Gupta and J. Kuznicki, "Biological and medical importance of cellular heterogeneity deciphered by single-cell rna sequencing," *Cells*, vol. 9, no. 8, p. 1751, 2020.

[50] R. M. Larsen, "Lanczos bidiagonalization with partial reorthogonalization," *DAIMI Report Series*, no. 537, 1998.


**Javier Salazar Cavazos** (Graduate Student Member, IEEE) received dual B.S. degrees in electrical engineering and mathematics from the University of Texas at Arlington, Arlington, TX, USA, 2020 and the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, 2023. He is currently working toward the Ph.D. degree in electrical and computer engineering 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. Since 1990, he has been with the University of Michigan. From 1991 to 1992, he was a Department of Energy Alexander Hollaender Postdoctoral Fellow with the Division of Nuclear Medicine. From 1993 to 1995, he was an Assistant Professor in Nuclear Medicine and the Bioengineering Program. He is currently a Professor with the Departments of Electrical Engineering and Computer Science, Radiology, and Biomedical Engineering. He is currently the William L. Root Professor of EECS 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. She is currently 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 the 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, the AFOSR Young Investigator Award. She was also a recipient of the Sarah Goddard Power Award at the University of Michigan.