Convex Clustering Redefined: Robust Learning[1] with the Median of Means Estimator

Sourav De *, Koustav Chowdhury*, Bibhabasu Mandal*, Sagar Ghosh†, Swagatam Das‡, Debolina Paul§, and Saptarshi Chakraborty¶,
*Indian Statistical Institute, Kolkata
†Department of Statistics and Data Science, University of Texas at Austin
‡Electronics and Communication Sciences, Indian Statistical Institute, Kolkata
§ Department of Statistics, University of Oxford
¶ Department of Statistics, University of Michigan
desourav02@gmail.com, koustavchowdhury2003@gmail.com, bibhabasumandal04@gmail.com, sagarghosh1729@utexas.edu,swagatam.das@isical.ac.in, ddebolina.paul@stats.ox.ac.uk, saptarsc@umich.edu

Abstract

Clustering approaches that utilize convex loss functions have recently attracted growing interest in the formation of compact data clusters. Although classical methods like $k-$means and its wide family of variants are still widely used, all of them require the number of clusters (k) to be supplied as input and many are notably sensitive to initialization. Convex clustering provides a more stable alternative by formulating the clustering task as a convex optimization problem, ensuring a unique global solution. However, it faces challenges in handling high-dimensional data, especially in the presence of noise and outliers. Additionally, strong fusion regularization, controlled by the tuning parameter, can hinder effective cluster formation within a convex clustering framework. To overcome these challenges, we introduce a robust approach that integrates convex clustering with the Median of Means (MoM) estimator, thus developing an outlier-resistant and efficient clustering framework that does not necessitate a prior knowledge of the number of clusters. By leveraging the robustness of MoM alongside the stability of convex clustering, our method enhances both performance and efficiency, especially on large-scale datasets. Theoretical analysis demonstrates weak consistency under specific conditions, while experiments on synthetic and real-world datasets validate the method's superior performance compared to existing approaches.

Github Repository: https://tinyurl.com/2v3dx75x
Benchmark Dataset (CC BY-NC-ND 4.0): https://tinyurl.com/2zatwkf3
ASU Datasets (GPLv2): https://tinyurl.com/49n36ume
Micro-array Datasets: https://tinyurl.com/2f2pjz7j
Brain Dataset: https://tinyurl.com/4ntav7b9
Wisconsin Dataset: https://tinyurl.com/58wxjha5

I. Introduction

Clustering is a fundamental task in unsupervised learning, aiming to organize unlabeled data into coherent groups for better interpretation and downstream applications. It plays a critical role in diverse areas such as customer segmentation [1], image analysis [2], and anomaly detection [3]. Traditional algorithms, such as k-means, approach clustering as a non-convex optimization problem [4], typically solved using greedy heuristics. Although computationally efficient and widely used, these methods suffer from several well-known limitations [5]: they require pre-specifying the number of clusters [6], [7], are sensitive to initialization [8], [9], and degrade in performance in high-dimensional spaces or when the data contains noise and outliers [10]–[12].

To overcome these challenges, convex relaxations of non-convex clustering problems have gained significant attention [13]. A prominent example is *convex clustering* (or sum-of-norms clustering), which enjoys strong theoretical guarantees such as global optimality and convergence, while remaining broadly applicable in practice [14]–[16].

[1] Sourav De, Koustav Chowdhury and Bibhabasu Mandal contributed equally to this work.

Given a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where each row represents a data point in d-dimensional Euclidean space, convex clustering solves the following objective:

$$\min_{\mathbf{u}} \frac{1}{2} \left[\|\mathbf{x}_{i.} - \mathbf{u}_{i.}\|_2^2 + \gamma \sum_{i,j} \theta_{ij} \|\mathbf{u}_{i.} - \mathbf{u}_{j.}\|_p^2 \right], \tag{I.1}$$

where $\mathbf{u}_{i.}$ denotes the i-th row of $\mathbf{u}$, θ_{ij} are edge weights, and $\|\cdot\|_p$ is the ℓ^p norm. The first term encourages each point to remain close to its centroid, while the second term (controlled by tuning parameter $\gamma > 0$) promotes fusion across centroids, effectively determining the number of clusters [17]. Although convex clustering is effective even in large-sample settings [18], strong regularization can lead to undesirable merging of outliers with genuine clusters, especially in high-dimensional data [19].

This paper focuses on addressing the **robustness challenges** of clustering in the presence of noise and outliers. Robust methods can mitigate these effects by either discarding outlier features [20] or directly controlling the influence of anomalous data points. One powerful approach is the *Median-of-Means (MoM)* estimator, which provides strong robustness and concentration guarantees under mild assumptions [21]–[25]. Related work by [26] further unifies robust center-based clustering under general dissimilarity measures.

Consider n data points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ to be grouped into k clusters. Each cluster is represented by a centroid $\boldsymbol{\theta}_j \in \mathbb{R}^d$, and the set of centroids forms a matrix $\boldsymbol{\Theta} \in \mathbb{R}^{k \times d}$. Using a Bregman divergence $d_\phi(\cdot, \cdot)$ as the dissimilarity measure, where $\phi : \mathbb{R}^d \to \mathbb{R}$ is a differentiable convex function, clustering can be formulated as

$$f_{\boldsymbol{\Theta}}(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \frac{1}{n} \sum_{i=1}^{n} \Psi_\alpha(d_\phi(\mathbf{x}_i, \boldsymbol{\theta}_1), \ldots, d_\phi(\mathbf{x}_i, \boldsymbol{\theta}_k)), \tag{I.2}$$

where $\Psi_\alpha : \mathbb{R}_{\geq 0}^k \to \mathbb{R}_{\geq 0}$ is a non-decreasing function, $\Psi_\alpha(0) = 0$, and α is a hyperparameter. Different choices of ϕ and Ψ_α recover well-known clustering algorithms such as k-means, power k-means, and k-harmonic-means.

Instead of directly minimizing (I.2), MoM partitions the data into L disjoint subsets $B_1, \ldots, B_L$, each containing b samples, and optimizes a robust median-based objective:

$$\mathrm{MoM}_L^n(\boldsymbol{\Theta}) = \mathrm{Median}\left(\frac{1}{b} \sum_{i \in B_1} f_{\boldsymbol{\Theta}}(\mathbf{x}_i), \ldots, \frac{1}{b} \sum_{i \in B_L} f_{\boldsymbol{\Theta}}(\mathbf{x}_i) \right). \tag{I.3}$$

Because outliers typically contaminate only a fraction of the partitions, the median effectively suppresses their influence, ensuring robustness even in adversarial settings. Formal breakdown-point analyses of MoM estimators support this intuition [27], [28].

To demonstrate our method, Figure 1 illustrates results on a benchmark dataset with $k = 5$, $N = 1000$, $\gamma = 5000$, 20% noise, and varying μ. Outliers near the boundary are successfully isolated. When any pair μ_i, μ_j exceeds the separation threshold μ, the connecting edge is dropped, preventing spurious merging. While most outliers are identified, some deeply embedded points remain undetected.

(a) Dataset (b) $\mu = 0.4$ (c) $\mu = 0.2$ (d) $\mu = 0.12$

Fig. 1: Figure 1a shows the original dataset in yellow, with 20% added noise represented by blue dots. As μ decreases, our method progressively identifies more noise points as outliers, which are marked by purple dots in Figures 1b, 1c, and 1d respectively.

II. Contributions

In this paper, we present a novel clustering framework that extends convex clustering with enhanced robustness using the Median-of-Means (MoM) estimator. Our main contributions are summarized below:

- **Robust Convex Clustering Framework:** We propose a new clustering method that integrates the MoM estimator into the convex clustering paradigm, effectively mitigating the adverse impact of outliers and noisy data. We also develop a dedicated algorithm for the proposed framework, ensuring practical applicability and computational efficiency.
- **Theoretical Guarantees:** We establish uniform deviation bounds and concentration inequalities under standard regularity assumptions, providing strong theoretical reliability for our method.
- **Empirical Validation:** Extensive simulation studies demonstrate that our method consistently outperforms conventional clustering approaches, particularly in terms of robustness and efficiency under data contamination.

III. Related Works

Convex Clustering and Semi-definite Programming: Since the introduction of Convex Clustering by [29], various extensions and perspectives have been explored [30], [31], [32]. Pelckmans and De Moor introduced a shrinkage term to induce sparsity between centroids, enabling hierarchical clustering by tuning the trade-off parameter. [15] propose a convex relaxation-based clustering algorithm that efficiently traces a regularization path, achieves state-of-the-art performance on non-convex clusters, and simultaneously infers a hierarchical tree structure from the data. In [33], two optimization approaches — ADMM and a variant of AMA were introduced to solve convex clustering problems for practical applications. Additionally, convex relaxations of the $k-$Means problem via Semi-Definite Programming (SDP) have been developed [34]–[36], replacing the $k-$means objective with a trace-based formulation. [36] further showed that this SDP relaxation achieves perfect recovery with high probability under the stochastic unit-ball model in $\mathbb{R}^d$, given mild regularity conditions.

Robustness and Feature Selection: Robustness to outliers is essential for ensuring learning algorithms remain stable under adversarial or noisy conditions. To address this, [37] proposed a Robust Multi-Task Feature Learning model that not only identifies shared features across tasks but also detects outlier tasks. Similarly, [33] introduced a robust multi-task learning framework combining a low-rank structure for related tasks and a sparse group structure to isolate outlier tasks.

Median of Means based Clustering: The Median of Means (MoM) estimator provides a robust and efficient framework for mean estimation with strong theoretical guarantees. [38] introduced a bootstrap-based MoM method, forming blocks with replacement, which improves the breakdown point over standard MoM when enough blocks are used. In the context of interpretable clustering, [39] proposed a method using small decision trees to partition data, enabling clear cluster characterization. They further analyzed whether such tree-induced clusterings can match the cost of optimal unconstrained clustering and how to compute them efficiently.

IV. Proposed Method

In this section, we outline our proposed clustering technique based on the median of means estimate in sufficient detail. We also include an Adam-based gradient descent method to optimize our non-convex objective function effectively.

Let $\mathbf{X}_{n\times d} \in \mathbb{R}^{n\times d}$ be the data matrix, where each row $\{\mathbf{x}_i\}_{i=1}^n$ is a data point, and $\mathbf{x}_i \in \mathbb{R}^d$ for each $i \in \{1, 2, ..., n\}$. Let $\mathbf{u}_i$ be the agent corresponding to point $\mathbf{x}_i$ $\forall i \in \{1, 2, \cdots, n\}$, and we define $\mathbf{U}_{n\times d} \in \mathbb{R}^{n\times d}$ as the agent matrix.

One of the most challenging problems in convex clustering is assigning weights to every pair of neighbours based on certain similarity measures. This enforces a restriction on its performance in high dimensions, which depends heavily on the choice of the pairwise similarities, although most people follow a $k-$nearest neighbour-based approach [40], coupled with a Gaussian similarity measure:

$$w_{ij} = \mathbb{1}_{ij,k} e^{-\phi||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2}, \tag{IV.1}$$

where $\mathbb{1}_{ij,k} = 1$ if x_i is one of the k-nearest neighbours of x_j with respect to the $\|\cdot\|_2$ and 0 otherwise. Here, ϕ represents the bandwidth of the Gaussian kernel, and smaller values of ϕ indicate greater similarities between the two nodes. Arbitrary choices of ϕ can lead to poor cluster generation, formation of arbitrary clusters, or even collapse of all cluster centroids to a global centroid [15], hindering the overall effectiveness of the $k-$nearest neighbour-based heuristics.

Next, we introduce a Random Binning strategy to partition the dataset before minimizing a non-convex objective function. This class of Random Binning (RB) techniques was originally proposed in [41] and subsequently revisited in [42], where it was shown to yield faster convergence than other Random Features methods when scaling large-scale kernel machines. Although these previous approaches typically employ a parametrized feature map [43], incorporating both bin widths and offsets, our method adopts a simplified variant of this strategy - designed specifically to randomly partition the dataset into $\mathcal{O}(n)$ number of bins, each containing a fixed number of samples drawn from the observables. Formally, we partition the index set $1, 2, \cdots, n$ into $l = \mathcal{O}(n)$ subsets, denoted by $B = \{B_i\}_{i=1}^{l}$, where each B_i contains exactly $b(= \lfloor \frac{n}{l} \rfloor)$ elements. If n is not divisible by l, a small number of elements are discarded to maintain uniform bin sizes across all partitions.

Henceforth, we define the "contribution" of point $\mathbf{x}_r$ in a "convex" type cost function as

$$f_U(\mathbf{x}_r) = \frac{1}{2}||\mathbf{x}_r - \mathbf{u}_r||_2^2 + \frac{\gamma}{2}\sum_{i,j} w_{ij}||\mathbf{u}_i - \mathbf{u}_j||_2^2. \tag{IV.2}$$

By our aforementioned MoM framework, instead of directly minimizing $\frac{1}{n}\sum_{r=1}^{n} f_{\mathbf{U}}(\mathbf{x}_r)$, we aim to minimize an objective function of the form

$$C(\mathbf{U}) = Median\left(\left\{\frac{1}{b}\sum_{r\in B_j} f_{\mathbf{U}}(\mathbf{x}_r)\right\}_{j=1}^{l}\right). \tag{IV.3}$$

Noting that the second term in (IV.2) is independent of r, we define $l_t \in \{1, 2, \cdots, l\}$ such that

$$\begin{aligned} MoM_B(\mathbf{U}) &:= Median\left(\left\{\frac{1}{2b}\sum_{i\in B_j} ||\mathbf{x}_i - \mathbf{u}_i||_2^2\right\}_{j=1}^{l}\right) \\ &= \frac{1}{2b}\sum_{i\in B_{l_t}} ||\mathbf{x}_i - \mathbf{u}_i||_2^2. \end{aligned} \tag{IV.4}$$

Next, we rewrite the cost function as

$$C(\mathbf{U}) = MoM_B(\mathbf{U}) + \frac{\gamma}{2}\sum_{i,j} w_{ij}||\mathbf{u}_i - \mathbf{u}_j||_2^2. \tag{IV.5}$$

Before initiating the optimization procedure of the cost function in IV.5, we will involve another robustness criterion to make our objective function more stable from outliers: we use $\sum_{i,j} w_{ij}\min\left(\mu, ||\mathbf{u}_i - \mathbf{u}_j||_2^2\right)$ instead of $\sum_{i,j} w_{ij}||\mathbf{u}_i - \mathbf{u}_j||_2^2$. By clipping the maximum pairwise distances by another hyperparameter μ, we can significantly remove the effect of such outliers or other distant clusters.

Now, we are in a position to write down the final cost function, which is

$$C(\mathbf{U}) = MoM_B(\mathbf{U}) + \frac{\gamma}{2}\sum_{i,j} w_{ij}\min\left\{\mu, ||\mathbf{u}_i - \mathbf{u}_j||_2^2\right\}. \tag{IV.6}$$

Due to the non-convex nature of this objective function, we use the ADAM gradient descent algorithm [44] to minimize it. The gradient of $C(\mathbf{U})$ with respect to $\mathbf{u}_i$ is

$$g_i := \frac{\partial C(\mathbf{U})}{\partial \mathbf{u}_i} = \frac{1}{b}(\mathbf{u}_i - \mathbf{x}_i)\mathbb{1}(i \in B_{l_t}) + \gamma\sum_j w_{ij}(\mathbf{u}_i - \mathbf{u}_j)\mathbb{1}(||\mathbf{u}_i - \mathbf{u}_j||_2^2 < \mu). \tag{IV.7}$$

After N iterations, we construct a graph with $\{\mathbf{u}_i\}_{i=1}^{n}$ as vertices and where $\mathbf{u}_i$ and $\mathbf{u}_j$ are adjacent if $||\mathbf{u}_i - \mathbf{u}_j|| < \eta_1$. We assign each connected component of this graph as a cluster and combine all clusters with less than half the average cluster size into a single cluster, marking this combined cluster as noise.

V. Theoretical Properties

This section establishes the theoretical properties of the (global) optimal solutions of the proposed objective function. We also analyse computational complexity and discuss the convergence properties of our method.

Algorithm 1 COMET : Convex Clustering with Median of Mean Estimator and Adam Optimization

Input: Data $\{\mathbf{x}_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$
Hyperparameters: $N, k, \phi, \gamma, \mu, \eta_1$
Output: Cluster assignment $\{Z_i\}_{i=1}^n$ where $Z_i \in \mathbb{N}$

Construct a $k-$NN graph on $\{\mathbf{x}_i\}_{i=1}^n$ and assign $w_{ij} = e^{-\phi||\mathbf{x}_i - \mathbf{x}_j||_2^2}$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are adjacent, $w_{ij} = 0$ otherwise
Initialize $\mathbf{m}_i^{(0)} = 0$, $\mathbf{v}_i^{(0)} = 0$ and $\mathbf{u}_i^{(0)} = \mathbf{x}_i$
for $t = 0$ to $N-1$ **do**
  Construct a partition, $B = \{B_i\}_{i=1}^l$, of $\{1, 2, \cdots, n\}$ into l bins each of size b
  Find $B_{l_t} \in B$ such that $MoM_B(\mathbf{U}^{(t)}) = \dfrac{1}{2b} \sum_{i \in B_{l_t}} ||\mathbf{x}_i - \mathbf{u}_i^{(t)}||_2^2$
  $\mathbf{g}_i^{(t)} = \dfrac{1}{b}(\mathbf{u}_i^{(t)} - \mathbf{x}_i)\mathbb{1}(i \in B_{l_t}) + \gamma \sum_j w_{ij}(\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)})\mathbb{1}(||\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}||_2^2 < \mu)$
  $\mathbf{m}_i^{(t)} = \boldsymbol{\beta}_1 \mathbf{m}_i^{(t-1)} + (1 - \boldsymbol{\beta}_1)\mathbf{g}_i^{(t)}$
  $\mathbf{v}_i^{(t)} = \boldsymbol{\beta}_2 \mathbf{v}_i^{(t-1)} + (1 - \boldsymbol{\beta}_2)\left(\mathbf{g}_i^{(t)} \odot \mathbf{g}_i^{(t)}\right)$
  Calculate $\mathbf{u}_i^{(t+1)}$ from $\mathbf{u}_i^{(t)}$ with the help of $\hat{\mathbf{m}}_i^{(t)}$ and $\hat{\mathbf{v}}_i^{(t)}$ using the ADAM update rule. Refer to the supplementary material (A-A) for the exact update rule.
end for
Construct a graph on $\left\{\mathbf{u}_i^{(N)}\right\}_{i=1}^n$ where $\mathbf{u}_i$ and $\mathbf{u}_j$ are adjacent if $||\mathbf{u}_i - \mathbf{u}_j|| < \eta_1$
Assign each connected component of this graph as a cluster
Combine all clusters with less than half the average cluster size into a single cluster and mark this combined cluster as noise

A. Finite Sample Error Bounds and Weak Consistency

We begin our statistical analysis of COMET by providing finite sample error bounds on the prediction error, using the widely used Hanson-Wright inequalities, especially the recent uniform versions by [45]. These bounds provide sufficient conditions for the consistency of the centroid and weight estimates.

Recall the objective function (IV.5),

$$\min_U \left\{ \frac{1}{2b} \sum_{i \in B_{l_t}} ||\mathbf{x}_i - \mathbf{u}_i||_2^2 + \frac{\gamma}{2} \sum_{i,j} w_{ij} ||\mathbf{u}_i - \mathbf{u}_j||_2^2 \right\},$$

where l_t is such that $\frac{1}{2b}\sum_{i \in B_{l_t}} ||\mathbf{x}_i - \mathbf{u}_i||_2^2 = Median\left(\left\{\frac{1}{2b}\sum_{i \in B_j} ||\mathbf{x}_i - \mathbf{u}_i||_2^2\right\}_{j=1}^l\right)$, $l_t \in \{1, \ldots, l\}$.
Let $\boldsymbol{x} = vec(X)$ and $\boldsymbol{u} = vec(U)$, where $vec(\cdot)$ means to vectorize a matrix by appending its columns together. So, $\boldsymbol{x}, \boldsymbol{u} \in \mathbb{R}^{nd}$ and $\boldsymbol{x}_{d(i-1)+j} = X_{ij}$, $\boldsymbol{u}_{d(i-1)+j} = U_{ij}$.
Consider $\boldsymbol{I}_{B_{l_t}}$ to be an $nd \times nd$ diagonal matrix with i-th diagonal element = 1 if $bd \leq i < (b+1)d$ where $b \in B_{l_t}$ and all other elements 0. So, we can write

$$\sum_{i \in B_{l_t}} ||\mathbf{x_i} - \mathbf{u_i}||_2^2 = (\boldsymbol{x} - \boldsymbol{u})^\top I_{B_{l_t}} (\boldsymbol{x} - \boldsymbol{u}).$$

Also note that w_{ij}'s remain fixed in each iteration of the algorithm. Since w_{ij}'s are either 0 or < 1, we work with an upper bound of the cost function, where each w_{ij} is replaced by $w'_{ij} = \mathbb{1}(w_{ij} > 0)$. Let $D^{n(n-1)d \times nd}$ be such that $\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u} = \mathbf{u}_i - \mathbf{u}_j$, where $\mathcal{C}(i,j)$ is an index set: then the objective function can be written as

$$\min \left\{ \frac{1}{2b}(\boldsymbol{x} - \boldsymbol{u})^\top I_{B_{l_t}} (\boldsymbol{x} - \boldsymbol{u}) + \frac{\gamma}{2} \sum_{(i,j) \in \mathcal{E}} \|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\|_2^2 \right\}, \tag{V.1}$$

where $\mathcal{E} \subseteq \{(i,j) : i, j \in \{1, 2, \ldots, n\}\}$ is an index set. We will assume the model $\boldsymbol{x} = \boldsymbol{u} + \boldsymbol{\epsilon}$, where $\epsilon \in \mathbb{R}^{nd}$ is a vector of independent noise variables and $\mathbb{E}(\epsilon) = 0$. This model is fairly standard for analysing the large-sample

behaviour of convex clustering methods [46]; [47]. For all practical purposes, one may assume that the error terms are almost surely bounded, that is, for some $M > 0$, $|\epsilon_i| \leq M$ for all $i = 1, \ldots, nd$. For notational simplicity, we write $\|\boldsymbol{y}\|_{\boldsymbol{A}}^2 = \boldsymbol{y}^\top \boldsymbol{A}\boldsymbol{y}$, for any positive semidefinite matrix $\boldsymbol{A}$. The goal of this analysis is to find probabilistic bounds on $\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|_{\hat{\boldsymbol{I}}_{B_{l_t}}}^2$, where $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{I}}_{B_{l_t}}$ are obtained by minimising the objective function in (V.1).

Theorem 1. *Suppose the model behaves as $\boldsymbol{x} = \boldsymbol{u} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \in \mathbb{R}^{nd}$ is a vector of independent bounded random variables, with mean 0, covariance matrix $\sigma^2 \boldsymbol{I}_{nd\times nd}$ and $|\epsilon_i| \leq M$, for all $i = 1, \ldots, nd$. Further assume that $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{I}}_{B_{l_t}}$ are obtained from minimizing* (V.1)*, then if $\gamma' \geq \frac{M}{ndb\sqrt{n}}$ the following holds with probability at least 1 - δ,*

$$\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|_{\hat{\boldsymbol{I}}_{B_{l_t}}}^2 \leq M^2\left(\frac{\sqrt{db} + d\sqrt{n}}{n\sqrt{db}}\right) + M^2\left(\frac{c}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)} + \frac{c\log\left(\frac{1}{\delta}\right)}{ndb}\right) + \gamma'\frac{|\mathcal{E}|}{4} + \gamma'\left[\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2 + \sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2^2\right]. \tag{V.2}$$

The proof of this theorem is deferred to the supplemntary material (A-B). From this theorem, we also arrive at the following two corollaries: Corollary 1.1 addresses the convergence of the centroid estimates under a minimum constraint on the hyperparameter, for the number of features being small enough with respect to the number of sample points, while Corollary 1.2 elaborates on the rate of convergence of those estimates under the constraint on the hyperparameter only.

Corollary 1.1. *Suppose $\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2 \leq C$, for all $1 \leq i, j \leq n$, for some constant C, $|\mathcal{E}| \leq kn$ and $\gamma' \geq \frac{M}{ndb\sqrt{n}}$. If $d = o(n)$, then $\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|_{\hat{\boldsymbol{I}}_{B_{l_t}}}^2 \xrightarrow{p} 0$ as $n, d \to \infty$.*

Corollary 1.2. *Suppose $\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2 \leq C$, for all $1 \leq i, j \leq n$, for some constant C, $|\mathcal{E}| \leq kn$ and $\gamma' \geq \frac{M}{ndb\sqrt{n}}$. Then $\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|_{\hat{\boldsymbol{I}}_{B_{l_t}}}^2 = O\left(\frac{1}{\sqrt{n}}\right)$.*

We lay out the complete proofs of Corollary 1.1 and Corollary 1.2 in the supplementary material (A-C) and (A-D), respectively.

B. Computational Complexity and Other Competing Methods

We compare the efficiency of our algorithm with other popular algorithms such as Convex Clustering ([48]), Robust Continuous Clustering ([49]), and Robust Convex Clustering ([20]). For a detailed explanation, refer to supplementary material (A-E).

Algorithm	**Complexity**
COMET	$\mathcal{O}(Nnkd)$
Convex-Clustering	$\mathcal{O}(N(n^2d + d\epsilon))$
Robust Continuous Clustering	$\mathcal{O}(N(n^2d + nkd)$
Robust Convex Clustering	$\mathcal{O}(Nnkd)$

TABLE I: Comparison of Runtime Complexity with other SOTA methods

From the above table, it is clear that COMET is better or at least on par in terms of computational cost with recent and most widely used robust clustering algorithms.

VI. Experiments and Results

In this section, we demonstrate the superiority of our proposed algorithm, COMET, over different variants of existing clustering algorithms, including both real and simulated datasets. The description of the datasets is given in the supplementary material (A-F). For simulated datasets, the generation procedure is described later.

A. Algorithms under consideration

We consider the following well-known clustering algorithms to assess the effectiveness of COMET: $k-$means (KM) [50], Convex Clustering (CC) [48], MoM $k-$means (MKM) [26], Robust Convex Clustering (RConv) [20], Robust Continuous Clustering (RCC) [49] and Robust Bregman $k-$means (RBKM) [51].

B. Performance Measures

For evaluating our proposed COMET algorithm against competing methods, we adopt the following metrics and resources:

- **Evaluation Metrics:** Since ground-truth cluster labels are available for all real and simulated datasets, we evaluate clustering performance using
  - **Adjusted Rand Index (ARI)**
  - **Adjusted Mutual Information (AMI)**

  Both metrics provide robust comparisons across algorithms.
- **Estimated Number of Clusters:** We also report the average number of clusters estimated by each algorithm to further assess performance.

C. Experimental Set-up

We apply all the selected algorithms on the datasets listed in the supplementary material (A-F). Our main goal is to make a proper comparison of robustness of these algorithms to the presence of noise and outliers in the data. We artificially add different levels of noise and outliers to the datasets under study and record the performances of the algorithms.

To add noise of level $p\%$ to a dataset, we first consider the smallest axis-parallel hypercube containing the whole original dataset. Then, we simulate $\lfloor \frac{np}{100} \rfloor$ points uniformly from the hypercube and add them to the original dataset, labeled as "noise". All the algorithms are run on this modified dataset, and the ARI/AMI is calculated based on the obtained cluster labels of the original data points only. We vary p to observe the change in performance of the algorithms with the introduction of noise. $k-$means, MoM $k-$means and Robust Bregman $k-$means require the exact number of clusters to be given as input, but that gives these algorithms an unfair advantage considering Convex Clustering, Robust Convex Clustering, Robust Continuous Clustering as well as COMET determine the number clusters automatically. Hence, to ensure a fair comparison, we used *Gapstat* ([6]) in those three algorithms to get an estimate of the number of clusters from the data itself and used that value for the clustering. We run all algorithms according to the recommended specification of hyperparameters or tune them to achieve maximum ARI. $k-$means, MoM $k-$means, and RBKM are run till there is no further update in the cluster assignment matrix. Convex Clustering and Robust Convex Clustering were run on each dataset after tuning its hyperparameters for 150 epochs. RCC is run according to the hyperparameter recommendations and termination condition specified in [49]. $k-$means, MoM $k-$means and Robust Bregman $k-$means are dependent on the choice of initial centroids, each of them were run 25 times for every noise level and the mean performance is reported with their standard deviation. The random noise was added to the data using numpy.random.default_rng(0) in numpy library of python 3 (ipykernel).

We perform the experiments for both generated and real-life datasets. In the next section we will focus on the results for real-life datasets. **For the detailed study on generated datasets refer to the supplementary material (A-G)**

D. Real-Life Datasets

Here we show the clustering results of our algorithm COMET and other selected algorithms on some of the real-life datasets with 10% noise. For the results on other datasets refer to the supplementary material (A-H). Here, k^* refers to an estimated number of clusters. The actual number of clusters is indicated as k. Here the standard deviation is that of the performance measure, not of the mean statistic.

1) Discussion: Table II shows that COMET outperforms other algorithms, achieving nearly accurate cluster numbers with low standard deviation across most datasets. However, in datasets like ORLRaws10P (Table II), Brain, and Wisconsin (present in the supplementary material (A-H)), the detected cluster size slightly deviates from the actual value, likely due to limitations in the $k-$NN graph structure. This issue is more pronounced in other algorithms. Despite this, COMET still provides better clustering patterns, with higher ARI and AMI than the others.

E. Significance of Our Results: Wilcoxon-Rank Sum Test

For various datasets, we want to test if the ARI and AMI produced by our algorithm are "significantly higher" than other selected clustering algorithms. We use Wilcoxon-Rank Sum test for this purpose. Refer to the supplementary material (A-K) for detailed discussion related to this.

Dataset	Index	KM	MKM	CC	RCC	RConv	RBKM	COMET
Newthyroid (k = 3)	k^*	3.08±1.28	2.94±1.38	14.14±1.23	212.13±3.36	3.79±0.58	2.00±0.00	4.14±0.36
	ARI	0.34±0.21†	0.40±0.26†	0.69±0.04†	0.00±0.00†	0.81±0.21†	0.11±0.03†	**0.97±0.01**
	AMI	0.34±0.19†	0.39±0.25†	0.52±0.03†	0.003±0.004†	0.77±0.16†	0.08±0.03†	**0.90±0.02**
Wisconsin (k = 2)	k^*	2.25±0.63	2.00±0.82	15±1.86	477±9.06	2.00±1.04	2.00±0.00	3.00±0.00
	ARI	0.52±0.35†	0.47±0.39†	0.81±0.01†	0.01±0.00†	0.85±0.03$^\sim$	0.15±0.06†	**0.87±0.01**
	AMI	0.48±0.31†	0.41±0.34†	0.67±0.01†	0.07±0.003†	0.75±0.03†	0.19±0.05†	**0.76±0.01**
Wine (k = 3)	k^*	3.14±1.18	3.17±1.35	25.29±2.16	178±0.00	2.43±0.51	2.00±0.00	4.64±0.84
	ARI	0.66±0.31$^\sim$	0.59±0.29†	0.59±0.15†	0.0±0.0†	0.22±0.28†	0.01±0.02†	**0.79±0.15**
	AMI	0.67±0.29$^\sim$	0.61±0.28†	0.59±0.10†	0.00±0.00†	0.32±0.31†	0.04±0.03†	**0.80±0.09**
Dermatology (k = 6)	k^*	5.16±1.93	4.77±2.09	4.00±0.00	358±0.00	5.00±0.00	2.00±0.00	5.85±0.53
	ARI	0.61±0.17†	0.56±0.17†	0.21±0.00†	0.00±0.00†	0.66±0.01†	0.004±0.02†	**0.81±0.06**
	AMI	0.78±0.10†	0.73±0.13†	0.44±0.00†	0.00±0.00†	0.79±0.01†	0.04±0.04†	**0.86±0.04**
Lung-Discrete (k = 7)	k^*	5.91±1.57	5.23±1.39	2.36±0.63	14.9±16.8	9.79±0.43	2±0.00	9.21±0.80
	ARI	0.44±0.09†	0.50±0.10†	0.07±0.03†	0.41±0.12†	0.39±0.05†	0.01±0.01†	**0.71±0.02**
	AMI	0.53±0.07†	0.58±0.08†	0.20±0.07†	0.51±0.15†	0.51±0.04†	0.07±0.04†	**0.69±0.01**
ORLRaws10p (k = 10)	k^*	4.85±1.83	4.89±1.72	42±0.00	100±0.00	16±0.00	2±0.00	14±0.00
	ARI	0.33±0.11†	0.33±0.10†	0.53±0.00†	0.00±0.00†	0.54±0.002†	0.02±0.01†	**0.73±0.00**
	AMI	0.58±0.11†	0.61±0.09†	0.61±0.00†	0.00±0.00†	0.69±0.001†	0.11±0.03†	**0.81±0.00**

TABLE II: Results for Real-life Datasets

† : significantly different from the best performing algorithm, $^\sim$: statistically similar to the best performing algorithm .

F. Case Study on Brain dataset

We evaluate our algorithm's performance on the Brain dataset, a Microarray dataset with 42 instances (brain tumor patients) and 5597 features. The dataset includes 5 categories: 10 medulloblastomas, 10 malignant gliomas, 10 AT/RT, 4 normal cerebellums, and 8 supratentorial PNETs, as described in [52].

Index	Noise(%)	KM	MKM	CC	RCC	RConv	RBKM	COMET
ARI	0	0.28±0.10†	0.23±0.11†	0.64±0.00†	0.00±0.00†	0.56±0.00†	0.01±0.01†	**0.65±0.00**
	5	0.31±0.13†	0.31±0.13†	0.64±0.00†	0.00±0.00†	0.56±0.01†	0.01±0.01†	**0.66±0.00**
	10	0.26±0.10†	0.26±0.10†	0.64±0.02$^\sim$	0.00±0.00†	0.56±0.06†	0.016±0.02†	**0.66±0.03**
	15	0.22±0.09†	0.10±0.08†	0.63±0.02$^\sim$	0.00±0.00†	0.55±0.06†	0.02±0.02†	**0.66±0.03**
	20	0.19±0.11†	0.08±0.07†	0.63±0.04†	0.00±0.00†	0.63±0.03†	0.02±0.02†	**0.65±0.02**
AMI	0	0.35±0.10†	0.32±0.10†	0.62±0.00†	0.00±0.00†	0.62±0.00†	0.017±0.01†	**0.67±0.00**
	5	0.38±0.13†	0.28±0.14†	0.62±0.00†	0.00±0.00†	0.62±0.01†	0.02±0.02†	**0.72±0.00**
	10	0.33±0.10†	0.27±0.11†	0.62±0.03†	0.00±0.00†	0.62±0.05†	0.03±0.04†	**0.72±0.03**
	15	0.29±0.11†	0.18±0.12†	0.62±0.01†	0.00±0.00†	0.62±0.04†	0.03±0.04†	**0.72±0.02**
	20	0.27±0.13†	0.15±0.12†	0.62±0.01†	0.00±0.00†	0.62±0.05†	0.03±0.05†	**0.72±0.03**
k^*	0	5±1.92	5±1.50	18±0.00	42±0.00	5±0.00	2±0.00	**5±0.00**
	5	5±1.85	5±1.46	18±0.00	42±0.00	5±0.00	2±0.00	**4±0.00**
	10	5±1.92	5±1.49	18±0.00	42±0.00	5±0.36	2±0.50	**4±0.00**
	15	4±1.90	4±1.41	18±0.00	42±0.00	5.1±0.22	2±0.50	**4±0.00**
	20	3±1.45	3±1.42	18±0.00	42±0.00	18±0.52	2±0.5	**4±0.00**

TABLE III: Performance of Different Algorithms on **Brain** on Different Noise Levels

† : significantly different from the best performing algorithm, $^\sim$: statistically similar to the best performing algorithm .

We compare the algorithms under varying noise levels (0%, 5%, 10%, 15%, 20%) using the procedure outlined earlier, without applying any feature reduction methods like PCA. The results, shown in Table III and Figure 2, show that COMET consistently outperforms all other algorithms, maintaining an ARI above 0.6 across all noise levels. Here, the standard deviation is that of the performance measure, not of the mean statistic. Convex clustering and Robust Convex clustering perform well but are outpaced by COMET. $k-$means and MoM $k-$means show poor results, worsening with increased noise. RCC and RBKM perform very poorly, as reflected in the results. The *t*-SNE plots for the clustering results for various algorithms on this dataset are provided in the supplementary material (A-J).

Also, refer to the supplementary material (A-I) for a case study on the Wisconsin dataset.

VII. Ablation Study

A sensitivity analysis of the performance of our algorithm for the hyperparameter γ is illustrated in the supplementary material (A-L1). For a detailed illustration of the tuning process of our hyperparameters on the **Wisconsin**

Fig. 2: Line plot for performance of different algorithms on **Brain** dataset

Breast Cancer dataset, refer to the supplementary material (A-L2). Another ablation study on the **NewThyroid** Dataset is discussed in the supplementary material (A-L3).

VIII. Limitations

While we tested our algorithm on specific distributions of noise for clustering synthetic data, a systematic method to choose an appropriate cost function needs to be developed. However, it is still obscure to us how the cost function can be modified in case the noise follows some definite pattern and is not randomly distributed. Also, in our proof of theoretical consistency 1, we assumed $d = o(n)$. However, modifications to the clustering procedure need to be done for higher-dimensional datasets. Overall, given the flexibility of our clustering framework, other possibilities can be explored by incorporating different methods for different steps. One may further explore to relax assumptions on the errors 1 for consistency in a more general settings.

IX. Conclusion

In this study, we introduce a robust and interpretable clustering framework designed for multivariate datasets affected by noise. Our method reformulates the underlying cost function to reduce the adverse effects of random noise that often undermine conventional convex clustering techniques. We provide rigorous theoretical guarantees by establishing both the consistency and the convergence rate of the proposed estimators. Furthermore, extensive empirical evaluations—including experiments on diverse real-world datasets, detailed case studies, and ablation studies—demonstrate the practical effectiveness and reliability of our approach.

References

[1] T. Kansal, S. Bahuguna, V. Singh, and T. Choudhury, "Customer segmentation using k-means clustering," in *2018 international conference on computational techniques, electronics and mechanical systems (CTEMS)*. IEEE, 2018, pp. 135–139.

[2] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in *Gi/itg workshop mmbnet*, vol. 7, 2007.

[3] G. B. Coleman and H. C. Andrews, "Image segmentation by clustering," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 773–785, 1979.

[4] Y. Lu and H. H. Zhou, "Statistical and computational guarantees of lloyd's algorithm and its variants," *arXiv preprint arXiv:1612.02099*, 2016.

[5] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[6] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the Number of Clusters in a Data Set Via the Gap Statistic," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 63, no. 2, pp. 411–423, Jul. 2001. [Online]. Available: https://academic.oup.com/jrsssb/article/63/2/411/7083348

[7] G. Hamerly and C. Elkan, "Learning the k in k-means," *Advances in Neural Information Processing Systems*, vol. 17, 03 2004.

[8] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy, "The effectiveness of lloyd-type methods for the k-means problem," *J. ACM*, vol. 59, no. 6, Jan. 2013. [Online]. Available: https://doi.org/10.1145/2395116.2395117

[9] J. Xu and K. Lange, "Power k-Means Clustering," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, Jun. 2019, pp. 6921–6931. [Online]. Available: https://proceedings.mlr.press/v97/xu19a.html

[10] D. M. Witten and R. Tibshirani, "A Framework for Feature Selection in Clustering," *Journal of the American Statistical Association*, vol. 105, no. 490, pp. 713–726, Jun. 2010. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1198/jasa.2010.tm09415

[11] R. C. De Amorim, "A Survey on Feature Weighting Based K-Means Algorithms," *Journal of Classification*, vol. 33, no. 2, pp. 210–242, Jul. 2016. [Online]. Available: http://link.springer.com/10.1007/s00357-016-9208-4

[12] S. Chakraborty and S. Das, "Detecting Meaningful Clusters From High-Dimensional Data: A Strongly Consistent Sparse Center-Based Clustering Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 2894–2908, 2022.

[13] J. Tropp, "Just relax: convex programming methods for identifying sparse signals in noise," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[14] K. Pelckmans, J. D. Brabanter, B. D. Moor, and J. A. K. Suykens, "Convex Clustering Shrinkage," in *Convex Clustering Shrinkage*, 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:125967021

[15] T. Hocking, J.-P. Vert, F. Bach, and A. Joulin, "Clusterpath An Algorithm for Clustering using Convex Fusion Penalties," in *Clustering Algorithm for Convex Fusion Penalties*, Jun. 2011, pp. 745–752.

[16] F. Lindsten, H. Ohlsson, and L. Ljung, "Clustering Using Sum-Of-Norms Regularization; with Application to Particle Filter Output Computation," in *IEEE Workshop on Statistical Signal Processing Proceedings*, Jun. 2011.

[17] E. C. Chi and S. Steinerberger, "Recovering trees with convex clustering," *SIAM Journal on Mathematics of Data Science*, vol. 1, no. 3, pp. 383–407, 2019.

[18] P. Radchenko and G. Mukherjee, "Convex clustering via l 1 fusion penalization," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 79, no. 5, pp. 1527–1546, 2017.

[19] Q. Feng, C. P. Chen, and L. Liu, "A review of convex clustering from multiple perspectives: models, optimizations, statistical properties, applications, and connections," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[20] Q. Wang, P. Gong, S. Chang, T. S. Huang, and J. Zhou, "Robust Convex Clustering Analysis," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. Barcelona, Spain: IEEE, Dec. 2016, pp. 1263–1268. [Online]. Available: http://ieeexplore.ieee.org/document/7837983/

[21] G. Lugosi and S. Mendelson, "Regularization, sparse recovery, and median-of-means tournaments," *Bernoulli*, vol. 25, 01 2017.

[22] M. Lerasle, "Lecture notes: Selected topics on robust statistical learning theory," *arXiv preprint arXiv:1908.10761*, 2019.

[23] G. Lecué and M. Lerasle, "Robust machine learning by median-of-means: Theory and practice," *The Annals of Statistics*, vol. 48, no. 2, pp. 906 – 931, 2020. [Online]. Available: https://doi.org/10.1214/19-AOS1828

[24] P. Laforgue, S. Clemencon, and P. Bertail, "On medians of (Randomized) pairwise means," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 1272–1281. [Online]. Available: https://proceedings.mlr.press/v97/clemencon19a.html

[25] P. Bartlett, S. Boucheron, and G. Lugosi, "Model selection and error estimation," *Machine Learning*, vol. 48, pp. 85–113, 01 2002.

[26] D. Paul, S. Chakraborty, S. Das, and J. Xu, "Uniform Concentration Bounds toward a Unified Framework for Robust Clustering," Oct. 2021, arXiv:2110.14148 [cs, math, stat]. [Online]. Available: http://arxiv.org/abs/2110.14148

[27] D. Rodriguez and M. Valdora, "The breakdown point of the median of means tournament," *Statistics & Probability Letters*, vol. 153, pp. 108–112, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167715219301415

[28] L. Guillaume and L. Matthieu, "Learning from mom's principles: Le cam's approach," 2017. [Online]. Available: https://arxiv.org/abs/1701.01961

[29] K. Pelckmans, J. De Brabanter, J. A. Suykens, and B. De Moor, "Convex clustering shrinkage," in *PASCAL workshop on statistics and optimization of clustering workshop*, vol. 1524, 2005.

[30] F. Lindsten, H. Ohlsson, and L. Ljung, "Clustering using sum-of-norms regularization: With application to particle filter output computation," in *2011 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2011, pp. 201–204.

[31] T. D. Hocking, A. Joulin, F. Bach, and J.-P. Vert, "Clusterpath an algorithm for clustering using convex fusion penalties," in *28th international conference on machine learning*, 2011, p. 1.

[32] C. Zhu, H. Xu, C. Leng, and S. Yan, "Convex optimization procedure for clustering: Theoretical revisit," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[33] J. Chen, J. Zhou, and J. Ye, "Integrating low-rank and group-sparse structures for robust multi-task learning," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 42–50.

[34] J. Peng and Y. Wei, "Approximating k-means-type clustering via semidefinite programming," *SIAM journal on optimization*, vol. 18, no. 1, pp. 186–205, 2007.

[35] P. Awasthi, A. S. Bandeira, M. Charikar, R. Krishnaswamy, S. Villar, and R. Ward, "Relax, no need to round: Integrality of clustering formulations," in *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, 2015, pp. 191–200.

[36] D. G. Mixon, S. Villar, and R. Ward, "Clustering subgaussian mixtures by semidefinite programming," *Information and Inference: A Journal of the IMA*, vol. 6, no. 4, pp. 389–415, 2017.

[37] P. Gong, J. Ye, and C. Zhang, "Robust multi-task feature learning," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 895–903.

[38] C. Brunet-Saumard, E. Genetay, and A. Saumard, "K-bmom: A robust lloyd-type clustering algorithm based on bootstrap median-of-means," *Computational Statistics & Data Analysis*, vol. 167, p. 107370, 2022.

[39] M. Moshkovitz, S. Dasgupta, C. Rashtchian, and N. Frost, "Explainable k-means and k-medians clustering," in *International conference on machine learning*. PMLR, 2020, pp. 7055–7065.

[40] E. C. Chi and K. Lange, "Splitting methods for convex clustering," *Journal of Computational and Graphical Statistics*, vol. 24, no. 4, pp. 994–1013, 2015.

[41] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Advances in neural information processing systems*, vol. 20, 2007.

[42] L. Wu, I. E. Yen, J. Chen, and R. Yan, "Revisiting random binning features: Fast convergence and strong parallelizability," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1265–1274.

[43] L. Wu, P.-Y. Chen, I. E.-H. Yen, F. Xu, Y. Xia, and C. Aggarwal, "Scalable spectral clustering using random binning features," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2506–2515.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[45] O. Bousquet, Y. Klochkov, and N. Zhivotovskiy, "Sharper bounds for uniformly stable algorithms," in *Proceedings of Thirty Third Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, J. Abernethy and S. Agarwal, Eds., vol. 125. PMLR, 09–12 Jul 2020, pp. 610–626. [Online]. Available: https://proceedings.mlr.press/v125/bousquet20b.html

[46] K. M. Tan and D. Witten, "Statistical Properties of Convex Clustering," 2015. [Online]. Available: https://arxiv.org/abs/1503.08340

[47] B. Wang, Y. Zhang, W. W. Sun, and Y. Fang, "Sparse convex clustering," *Journal of Computational and Graphical Statistics*, vol. 27, no. 2, p. 393–403, Apr. 2018. [Online]. Available: http://dx.doi.org/10.1080/10618600.2017.1377081

[48] E. C. Chi and K. Lange, "Splitting Methods for Convex Clustering," *Journal of Computational and Graphical Statistics*, vol. 24, no. 4, pp. 994–1013, Oct. 2015. [Online]. Available: http://www.tandfonline.com/doi/full/10.1080/10618600.2014.948181

[49] S. A. Shah and V. Koltun, "Robust continuous clustering," *Proceedings of the National Academy of Sciences*, vol. 114, no. 37, pp. 9814–9819, Sep. 2017. [Online]. Available: https://pnas.org/doi/full/10.1073/pnas.1700770114

[50] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Applied Statistics*, vol. 28, no. 1, p. 100, 1979. [Online]. Available: https://www.jstor.org/stable/10.2307/2346830?origin=crossref

[51] C. Brécheteau, A. Fischer, and C. Levrard, "Robust Bregman clustering," *The Annals of Statistics*, vol. 49, no. 3, Jun. 2021. [Online]. Available: https://projecteuclid.org/journals/annals-of-statistics/volume-49/issue-3/Robust-Bregman-clustering/10.1214/20-AOS2018.full

[52] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub, "Prediction of central nervous system embryonal tumour outcome based on gene expression," *Nature*, vol. 415, no. 6870, pp. 436–442, Jan. 2002. [Online]. Available: https://www.nature.com/articles/415436a

APPENDIX A
SUPPLEMENTARY MATERIAL

A. ADAM update rule for optimization

For a chosen $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ the update rule to be followed is as follows:

$$\hat{\mathbf{m}}_i^{(t)} = \frac{\mathbf{m}_i^{(t)}}{1-\boldsymbol{\beta}_1^t} \ ,$$

$$\hat{\mathbf{v}}_i^{(t)} = \frac{\mathbf{v}_i^{(t)}}{1-\boldsymbol{\beta}_2^t} \ ,$$

$$\mathbf{u}_i^{(t+1)} = \mathbf{u}_i^{(t)} - \frac{\alpha \hat{\mathbf{m}}_i^{(t)}}{\sqrt{\hat{\mathbf{v}}_i^{(t)} + \epsilon}}.$$

B. Proof of Theorem 1

Theorem 1. *Suppose $\boldsymbol{x} = \boldsymbol{u} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \in \mathbb{R}^{nd}$ is a vector of independent bounded random variables, with mean 0, covariance matrix $\sigma^2 \boldsymbol{I}$ and $|\epsilon_i| \leq M$, for all $i = 1, \ldots, nd$. Suppose that $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{I}}_{B_{l_t}}$ are obtained from minimizing equation (10), then if $\gamma' \geq \frac{M}{ndb\sqrt{n}}$ the following holds with probability at least 1 - δ:*

$$\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} \leq \quad M^2 \left(\frac{\sqrt{db/n} + d}{\sqrt{ndb}} + \quad c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)} + c\frac{\log\left(\frac{1}{\delta}\right)}{ndb} \right)$$
$$+ \quad \gamma' \frac{|\mathcal{E}|}{4} + \gamma' \left[\sum_{(i,j)\in\mathcal{E}} \left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2 + \sum_{(i,j)\in\mathcal{E}} \left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2^2 \right]$$

Proof. Let $\boldsymbol{D} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}_{\boldsymbol{\beta}}^\top$ be the singular value decomposition (SVD) of $\boldsymbol{D}$, where $\boldsymbol{V}_{\boldsymbol{\beta}} \in \mathbb{R}^{nd\times(n-1)d}$. We construct $\boldsymbol{V}_{\boldsymbol{\alpha}} \in \mathbb{R}^{nd\times d}$ such that $\boldsymbol{V} = [\boldsymbol{V}_{\boldsymbol{\alpha}}, \boldsymbol{V}_{\boldsymbol{\beta}}]$ is an $nd \times nd$ orthonormal matrix.

Next, define $\boldsymbol{\beta} = \boldsymbol{V}_{\boldsymbol{\beta}}^\top \boldsymbol{u}$, $\boldsymbol{\alpha} = \boldsymbol{V}_{\boldsymbol{\alpha}}^\top \boldsymbol{u}$ and $\gamma' = \frac{\gamma}{2nd}$. Also $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Lambda}$ and $\boldsymbol{Z}^-$ is the left inverse of $\boldsymbol{Z}$. Thus, the optimization problem becomes:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{I}_{B_{l_t}}} \frac{1}{2ndb}\left(\boldsymbol{x} - \boldsymbol{V}_{\boldsymbol{\alpha}}\boldsymbol{\alpha} - \boldsymbol{V}_{\boldsymbol{\beta}}\boldsymbol{\beta}\right)^\top \boldsymbol{I}_{B_{l_t}}\left(\boldsymbol{x} - \boldsymbol{V}_{\boldsymbol{\alpha}}\boldsymbol{\alpha} - \boldsymbol{V}_{\boldsymbol{\beta}}\boldsymbol{\beta}\right) + \gamma' \sum_{(i,j)\in\mathcal{E}} \left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\boldsymbol{\beta}\right\|_2^2 \tag{A.1}$$

Now, let $\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\beta}}$, and $\hat{\boldsymbol{I}}_{B_{l_t}}$ be the minimiser of the above cost function. Then, by definition,

$$\begin{aligned}
&\frac{1}{2ndb}\left\|\boldsymbol{x} - \boldsymbol{V}_{\boldsymbol{\alpha}}\hat{\boldsymbol{\alpha}} - \boldsymbol{V}_{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}\right\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} + \gamma' \sum_{(i,j)\in\mathcal{E}} \left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\hat{\boldsymbol{\beta}}\right\|_2^2 \\
&\leq \frac{1}{2ndb}\left\|\boldsymbol{x} - \boldsymbol{V}_{\boldsymbol{\alpha}}\boldsymbol{\alpha} - \boldsymbol{V}_{\boldsymbol{\beta}}\boldsymbol{\beta}\right\|^2_{\boldsymbol{I}_{B_{l_t}}} + \gamma' \sum_{(i,j)\in\mathcal{E}} \left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\boldsymbol{\beta}\right\|_2^2 \\
&\leq \frac{1}{2ndb}\left\|\boldsymbol{x} - \boldsymbol{V}_{\boldsymbol{\alpha}}\boldsymbol{\alpha} - \boldsymbol{V}_{\boldsymbol{\beta}}\boldsymbol{\beta}\right\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} + \frac{1}{2ndb}\boldsymbol{\epsilon}^\top(\boldsymbol{I}_{B_{l_t}} - \hat{\boldsymbol{I}}_{B_{l_t}})\boldsymbol{\epsilon} + \gamma' \sum_{(i,j)\in\mathcal{E}} \left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\boldsymbol{\beta}\right\|_2^2 \\
&\leq \frac{1}{2ndb}\left\|\boldsymbol{x} - \boldsymbol{V}_{\boldsymbol{\alpha}}\boldsymbol{\alpha} - \boldsymbol{V}_{\boldsymbol{\beta}}\boldsymbol{\beta}\right\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} + \frac{1}{n}M^2 + \gamma' \sum_{(i,j)\in\mathcal{E}} \left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\boldsymbol{\beta}\right\|_2^2
\end{aligned}$$

On further simplification, we get the following,

$$\frac{1}{2ndb}\left\|\boldsymbol{V}_{\boldsymbol{\alpha}}\left(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\right) + \boldsymbol{V}_{\boldsymbol{\beta}}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} + \gamma' \sum_{(i,j)\in\mathcal{E}} \left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\hat{\boldsymbol{\beta}}\right\|_2^2 \leq \frac{1}{ndb}G\left(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{I}}_{B_{l_t}}\right) + \frac{1}{n}M^2 + \gamma' \sum_{(i,j)\in\mathcal{E}} \left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\boldsymbol{\beta}\right\|_2^2$$

where $G(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{I}}_{B_{l_t}}) = \boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} (\boldsymbol{V_\alpha}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}) + \boldsymbol{V_\beta}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}))$. Since $\hat{\boldsymbol{\alpha}}$ is the minimiser, we can choose $\hat{\boldsymbol{\alpha}}$ such that $\boldsymbol{x} - \boldsymbol{V_\alpha}\hat{\boldsymbol{\alpha}} - \boldsymbol{V_\beta}\hat{\boldsymbol{\beta}} = 0$. Therefore, $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon}$. Now, we can bound,

$$
\begin{aligned}
&\frac{1}{ndb}\left|G(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{I}}_{B_{l_t}})\right| \\
&= \frac{1}{ndb}\left|\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \left(\boldsymbol{V_\alpha}\left(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\right) + \boldsymbol{V_\beta}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right)\right| \\
&= \frac{1}{ndb}\left|\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \left(\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon} + \boldsymbol{V_\beta}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right)\right| \\
&\leq \frac{1}{ndb}\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon} + \frac{1}{ndb}\left|\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\beta}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right| \\
&= \frac{1}{ndb}\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon} + \frac{1}{ndb}\left|\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\beta}\boldsymbol{Z}^- \boldsymbol{Z}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right| \\
&= \frac{1}{ndb}\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon} + \frac{1}{ndb}\left|\sum_{(i,j)\in\mathcal{E}} \boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\beta}\boldsymbol{Z}^-_{\mathcal{C}(i,j)}\boldsymbol{Z}_{\mathcal{C}(i,j)}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right| \\
&\leq \frac{1}{ndb}\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon} + \frac{1}{ndb}\sum_{(i,j)\in\mathcal{E}}\left|\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\beta}\boldsymbol{Z}^-_{\mathcal{C}(i,j)}\boldsymbol{Z}_{\mathcal{C}(i,j)}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right| \\
&\leq \frac{1}{ndb}\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon} + \frac{1}{ndb}\sum_{(i,j)\in\mathcal{E}}\left(\left\|(\boldsymbol{Z}^-_{\mathcal{C}(i,j)})^\top \boldsymbol{V_\beta}^\top \hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{\epsilon}\right\|_2 \cdot \left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right\|_2\right) \\
&\leq \frac{1}{ndb}\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon} + \frac{1}{ndb}\max_{(i,j)\in\mathcal{E}}\left\|(\boldsymbol{Z}^-_{\mathcal{C}(i,j)})^\top \boldsymbol{V_\beta}^\top \hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{\epsilon}\right\|_2 \cdot \sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right\|_2
\end{aligned}
$$

Next, we derive high-probability bounds for the terms $\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon}$ and $\max_{(i,j)\in\mathcal{E}}\left\|(\boldsymbol{Z}^-_{\mathcal{C}(i,j)})^\top \boldsymbol{V_\beta}^\top \hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{\epsilon}\right\|_2$. Now, using the Hanson Wright Inequality,

$$
\begin{aligned}
&\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}} \boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon} \\
&\leq \mathbb{E}\left[\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}} \boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon}\right] + c\left(M\sqrt{r}\ \mathbb{E}\left[\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}}\left\|\hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon}\right\|_{sp}\right]\right) + c\left(rM^2 \sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}}\left\|I_{B_{l_t}}\right\|\right) \\
&\leq \mathbb{E}\left[\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}} \operatorname{tr}\left(\boldsymbol{\epsilon}^\top \hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon}\right)\right] + c\left(M\sqrt{r}\ \mathbb{E}\left[\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}}\left\|\hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\right\|_{sp}\|\boldsymbol{\epsilon}\|_2\right]\right) + c\left(rM^2 \sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}} 1\right) \\
&= \mathbb{E}\left[\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}} \operatorname{tr}\left(\hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right)\right] + c\left(M\sqrt{r} \sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}}\left\|\hat{\boldsymbol{I}}_{B_{l_t}} \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\right\|_{sp}\ \mathbb{E}\left[\|\boldsymbol{\epsilon}\|_2\right] + rM^2\right) \\
&\leq \mathbb{E}\left[\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}} \sqrt{\operatorname{tr}\left(\hat{\boldsymbol{I}}^2_{B_{l_t}}\right)} \cdot \sqrt{\operatorname{tr}\left(\left(\boldsymbol{V}_\alpha \boldsymbol{V}_\alpha^\top \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right)^\top \left(\boldsymbol{V}_\alpha \boldsymbol{V}_\alpha^\top \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right)\right)}\right] + c\left(M\sqrt{r} \sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}}\left\|\hat{\boldsymbol{I}}_{B_{l_t}}\right\|_{sp}\left\|\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\right\|_{sp}\ \mathbb{E}\left[M\sqrt{nd}\right]\right) \\
&\quad + crM^2 \\
&= \mathbb{E}\left[\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}} \sqrt{db}\sqrt{\operatorname{tr}\left(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right)}\right] + c\left(M^2\sqrt{ndr} + rM^2\right) \\
&= \sqrt{db}\ \mathbb{E}\left[\sqrt{\operatorname{tr}\left(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right)}\right] + c\left(M^2\sqrt{ndr} + rM^2\right) \\
&= \sqrt{db}\ \mathbb{E}\left[\|\boldsymbol{\epsilon}\|_2\sqrt{\operatorname{tr}\left(\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right)}\right] + c\left(M^2\sqrt{ndr} + rM^2\right)
\end{aligned}
$$

$$
\begin{aligned}
&\leq Md\sqrt{nb}\ \mathbb{E}\left[\sqrt{\text{tr}\left(\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right)}\right] + c\left(M^2\sqrt{ndr} + rM^2\right)\\
&\leq Md\sqrt{nb}\ \sqrt{\mathbb{E}\left[\text{tr}\left(\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right)\right]} + c\left(M^2\sqrt{ndr} + rM^2\right)\\
&= Md\sqrt{nb}\ \sqrt{\text{tr}\left(\mathbb{E}\left[\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right]\right)} + c\left(M^2\sqrt{ndr} + rM^2\right)\\
&= Md\sqrt{nb}\ \sqrt{\text{tr}\left(\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\mathbb{E}\left[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right]\right)} + c\left(M^2\sqrt{ndr} + rM^2\right)\\
&= Md\sqrt{nb}\ \sqrt{\text{tr}\left(\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\left(\sigma^2 I\right)\right)} + c\left(M^2\sqrt{ndr} + rM^2\right)\\
&= Md\sigma\sqrt{nb}\ \sqrt{\text{tr}\left(\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\right)} + c\left(M^2\sqrt{ndr} + rM^2\right)\\
&= M^2d\sqrt{nb}\ \sqrt{\text{tr}\left(\boldsymbol{V_\alpha}^\top\boldsymbol{V_\alpha}\right)} + c\left(M^2\sqrt{ndr} + rM^2\right)\\
&= M^2d\sqrt{ndb} + c\left(M^2\sqrt{ndr} + rM^2\right)\\
&= M^2\left(d\sqrt{ndb} + c\sqrt{ndr} + cr\right)
\end{aligned}
$$

Thus, from the above analysis, we get

$$
\mathbb{P}\left(\frac{1}{ndb}\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}}\boldsymbol{\epsilon}^\top\hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\boldsymbol{\epsilon} \geq M^2\left(\frac{d}{\sqrt{ndb}} + c\frac{1}{b\sqrt{nd}}\sqrt{r} + \frac{cr}{ndb}\right)\right) \leq e^{-r}
$$

Taking $r = \log\left(\frac{1}{\delta}\right)$, we get,

$$
\mathbb{P}\left(\frac{1}{ndb}\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}}\boldsymbol{\epsilon}^\top\hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\boldsymbol{\epsilon} \geq M^2\left(\frac{d}{\sqrt{ndb}} + c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)} + c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right)\right) \leq \delta
$$

Thus, with probability atleast $1-\delta$,

$$
\frac{1}{ndb}\boldsymbol{\epsilon}^\top\hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\boldsymbol{\epsilon} \leq \frac{1}{ndb}\sup_{\boldsymbol{I}_{B_{l_t}}\in\mathcal{I}}\boldsymbol{\epsilon}^\top\hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{V_\alpha}\boldsymbol{V_\alpha}^\top\boldsymbol{\epsilon} \leq M^2\left(\frac{d}{\sqrt{ndb}} + c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)} + c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right)
$$

Let $y_j = \boldsymbol{e}_j^\top(\boldsymbol{Z}^-_{\mathcal{C}(i,j)})^\top\boldsymbol{V_\beta}^\top\hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{\epsilon}$. Now, y_j is a univariate, bounded random variable with $|y_j| \leq \frac{M}{\sqrt{n}}$. Thus,

$$
\begin{aligned}
&\max_{(i,j)\in\mathcal{E}}\left\|(\boldsymbol{Z}^-_{\mathcal{C}(i,j)})^\top\boldsymbol{V_\beta}^\top\hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{\epsilon}\right\|_\infty = \max_j|y_j| \leq \frac{M}{\sqrt{n}}\\
\Rightarrow &\frac{1}{ndb}\max_{(i,j)\in\mathcal{E}}\left\|(\boldsymbol{Z}^-_{\mathcal{C}(i,j)})^\top\boldsymbol{V_\beta}^\top\hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{\epsilon}\right\|_2 \leq \frac{1}{ndb}\max_{(i,j)\in\mathcal{E}}\left\|(\boldsymbol{Z}^-_{\mathcal{C}(i,j)})^\top\boldsymbol{V_\beta}^\top\hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{\epsilon}\right\|_\infty \leq \frac{M}{ndb\sqrt{n}}
\end{aligned}
$$

Note $\gamma' \geq \frac{M}{ndb\sqrt{n}}$ implies that,

$$
\gamma' \geq \frac{1}{ndb}\max_{(i,j)\in\mathcal{E}}\left\|(\boldsymbol{Z}^-_{\mathcal{C}(i,j)})^\top\boldsymbol{V_\beta}^\top\hat{\boldsymbol{I}}_{B_{l_t}}\boldsymbol{\epsilon}\right\|_2.
$$

So we get,

$$\frac{1}{ndb}\left|G(\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{\beta}},\hat{\boldsymbol{I}}_{B_{l_t}})\right| \leq M^2\left(\frac{d}{\sqrt{ndb}}+c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)}+c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right)+\gamma'\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\left(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right)\right\|_2$$

holds with probability atleast $1-\delta$. Now combining all the results we get, with probability atleast $1-\delta$

$$\frac{1}{2ndb}\left\|\boldsymbol{V}_{\boldsymbol{\alpha}}\left(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}\right)+\boldsymbol{V}_{\boldsymbol{\beta}}\left(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right)\right\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}}+\gamma'\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\hat{\boldsymbol{\beta}}\right\|_2^2$$
$$\leq M^2\left(\frac{d}{\sqrt{ndb}}+c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)}+c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right)+\frac{1}{n}M^2+\gamma'\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\left(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right)\right\|_2+\gamma'\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\boldsymbol{\beta}\right\|_2^2$$

Upon rearranging the terms, we obtain that with probability atleast $1-\delta$,

$$\frac{1}{2ndb}\left\|\boldsymbol{V}_{\boldsymbol{\alpha}}\left(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}\right)+\boldsymbol{V}_{\boldsymbol{\beta}}\left(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right)\right\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}}$$
$$\leq M^2\left(\frac{1}{n}+\frac{d}{\sqrt{ndb}}+c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)}+c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right)$$
$$+\gamma'\left[\sum_{(i,j)\in\mathcal{E}}\left(\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\hat{\boldsymbol{\beta}}\right\|_2-\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\hat{\boldsymbol{\beta}}\right\|_2^2\right)+\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\boldsymbol{\beta}\right\|_2+\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\boldsymbol{\beta}\right\|_2^2\right]$$
$$\leq M^2\left(\frac{1}{n}+\frac{d}{\sqrt{ndb}}+c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)}+c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right)+\gamma'\left[\frac{|\mathcal{E}|}{4}+\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\boldsymbol{\beta}\right\|_2+\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{Z}_{\mathcal{C}(i,j)}\boldsymbol{\beta}\right\|_2^2\right]$$
$$\leq M^2\left(\frac{1}{n}+\frac{d}{\sqrt{ndb}}+c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)}+c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right)+\gamma'\left[\frac{|\mathcal{E}|}{4}+\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2+\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2^2\right]$$

□

C. Proof of Corollary 1

Corollary 1.1. *Suppose $\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2 \leq C$, for all $1\leq i,j\leq n$, for some constant C, $|\mathcal{E}|\leq kn$ and $\gamma'\geq\frac{M}{ndb\sqrt{n}}$. If $d=o(n)$, then $\frac{1}{2ndb}\|\hat{\boldsymbol{u}}-\boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}}\xrightarrow{p}0$ as $n,d\to\infty$.*

Proof. For any fixed δ, we know from Theorem 1 that with probability atleast $1-\delta$.

$$\frac{1}{2ndb}\|\hat{\boldsymbol{u}}-\boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}}\leq M^2\left(\frac{1}{n}+\frac{d}{\sqrt{ndb}}+c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)}+c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right)+\gamma'\frac{|\mathcal{E}|}{4}$$
$$+\gamma'\left[\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2+\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2^2\right]$$
$$\leq M^2\left(\frac{1}{n}+\frac{d}{\sqrt{ndb}}+c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)}+c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right)+\gamma'\frac{|\mathcal{E}|}{4}+\left(C+C^2\right)\gamma'|\mathcal{E}|$$
$$\leq M^2\left(\frac{1}{n}+\frac{d}{\sqrt{ndb}}+c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)}+c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right)+\gamma'\frac{kn}{4}+\left(C+C^2\right)\gamma' kn$$
$$\to 0 \quad \text{as} \quad n,d\to\infty.$$

Thus, for any fixed $\epsilon > 0$, $P\left(\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} > \epsilon\right) \leq \delta$ as $n, p \to \infty$. Hence, $\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} \xrightarrow{p} 0$.

□

D. Proof of Corollary 2

Corollary 1.2. *Suppose* $\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2 \leq C$, *for all* $1 \leq i, j \leq n$, *for some constant* C, $|\mathcal{E}| \leq kn$ *and* $\gamma' \geq \frac{M}{ndb\sqrt{n}}$. *Then* $\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} = O\left(\frac{1}{\sqrt{n}}\right)$.

Proof. For any fixed δ, we know from Theorem 1 that with probability atleast $1 - \delta$.

$$
\begin{aligned}
&\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} \\
&\leq M^2\left(\frac{1}{n} + \frac{d}{\sqrt{ndb}} + c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)} + c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right) + \gamma'\frac{|\mathcal{E}|}{4} + \gamma'\left[\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2 + \sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2^2\right] \\
&\leq M^2\left(\frac{1}{n} + \frac{d}{\sqrt{ndb}} + c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)} + c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right) + \gamma'\frac{|\mathcal{E}|}{4} + \left(C + C^2\right)\gamma'|\mathcal{E}| \\
&\leq M^2\left(\frac{1}{n} + \frac{d}{\sqrt{ndb}} + c\frac{1}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)} + c\frac{\log\left(\frac{1}{\delta}\right)}{ndb}\right) + \gamma'\frac{kn}{4} + \left(C + C^2\right)\gamma' kn \\
&= O\left(\frac{1}{\sqrt{n}}\right)
\end{aligned}
$$

Thus, $\sqrt{n}\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} = O(1)$. This implies that there exists a constant ϵ such that, $P\left(\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} \leq \epsilon\right) \geq 1 - \delta$, for all $n \in \mathbb{N}$. Hence, $\sqrt{n}\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}}$ is tight.

□

E. Derivation of Time Complexity

The main loop in Algorithm 1 performs N iterations. It is enough to calculate the complexity of each step inside this loop. Constructing a random partition of l buckets of b points takes $O(lb) = O(n)$ steps. Calculating $MoM_B(\mathbf{U})$ for each bucket takes $O(bd)$ steps. Therefore, finding the median bucket B_{l_t}, takes $O(lbd) = O(nd)$ steps. Calculating each g_i requires checking whether the index i is in B_{l_t}, which takes at most $O(b)$ checks, evaluating $(\mathbf{u}_i^{(t)} - \mathbf{x}_i)$, which takes $O(d)$ constant time evaluations, and evaluating $\sum_j w_{ij}(\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)})\mathbb{1}(||\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}||_2^2 < \mu)$, which takes $O(kd)$ constant-time evaluations. Therefore, calculating all g_i's require at most $O(n(b+d+kd)) = O(nkd)$ steps. Calculating each m_i, v_i and u_i takes $O(d)$ constant-time evaluations. Therefore, calculating all m_i's, v_i's, and u_i's require $O(nd)$ constant-time evaluations. Hence, the per-iteration complexity of the main loop is $O(nkd)$. Therefore, the complexity of Algorithm 1 is $O(Nnkd)$.

F. Description of datasets

Table IV contains the name and description of all 18 datasets that were used in the experiments.

G. Synthetic Data

In this section, we get empirical results on 3 simulated datasets. We contaminate every dataset by adding noise points following a specific uniform distribution. The contamination levels are: 0%, 5%, 10%, 15% and 20%.

- **Blobs**: 3 blobs of data points containing 500 points each simulated from bi-variate Gaussian distribution with different means and covariance matrix as the $\mathbf{I}_{2\times 2}$. Noise is simulated uniformly from the smallest enclosing axis-parallel rectangle.
- **Circles**: This dataset contains a large circle of radius 1 (consisting of 350 points) containing a smaller circle of radius 0.25 (consisting of 350 points) in two-dimensional space, forming a non-linearly separable pattern. Each

	Datasets	Type	No. of Data-points	Input Dimension	No. of Clusters
1.	Iris	Real	150	4	3
2.	Newthyroid	Real	215	5	3
3.	Ecoli	Real	336	7	8
4.	Wisconsin	Real	683	9	2
5.	Wine	Real	178	13	3
6.	Zoo	Real	101	16	7
7.	Dermatology	Real	358	34	6
8.	Brain	Real	42	5597	5
9.	Lung	Real	203	3312	5
10.	Lymphoma (bio)	Real	96	4026	9
11.	Coil 20	Real	1440	1024	20
12.	Wdbc	Real	569	30	2
13.	Lung-discrete	Real	73	325	7
14.	ORLRaws10p	Real	100	10304	10
15.	Lymphoma (microarray)	Real	62	4026	3
16.	Blobs	Simulated	1500	2	3
17.	Circles	Simulated	700	2	2
18.	Moons	Simulated	1500	2	2

TABLE IV: List of Datasets

point in the dataset is assigned to one of two classes, representing the respective circle to which it belongs. We artificially contaminate the dataset by introducing random points in the middle-annulus to understand which algorithm best separates the two clusters.

- **Moons**: The two moons dataset is a synthetic dataset consisting of two interleaving crescent-shaped clusters resembling two half-moons. The data contains 1500 two-dimensional points, and the two clusters are non-linearly separable. Noise is simulated uniformly from the smallest enclosing axis-parallel rectangle.

Fig. 3: Performance of Algorithms on Blobs Dataset (ARI and AMI Values)

Figures 3, 4 and 5 represent a comparison of COMET with other SOTA algorithms for different noise levels on different datasets. Figure 14 gives a visual representation of the outputs of different algorithms at 10% noise. It is clearly visible that COMET captures the clustering pattern well and also properly classifies noise.

Discussion: As shown in Figure 3, 4, 5 and 14, clustering results vary significantly across algorithms. Our proposed algorithm, **COMET**, excels in identifying true clusters, maintaining high ARI values even as noise increases, demonstrating robustness. In contrast, $k-$means and MoM $k-$means struggle to detect the underlying structure which is a limitation of the $k-$means framework. Convex Clustering performs similarly to COMET in identifying cluster structure and count, but struggles with noise, whereas COMET effectively isolates noise, showcasing its superior performance in noisy environments.

Fig. 4: Performance of Algorithms on Circles Datasets (ARI and AMI Values)

Fig. 5: Performance of Algorithms on Moons Dataset (ARI and AMI Values)

H. Real-life Data

Here we have provided the performance of the selected algorithms on the real-life datasets mentioned earlier in Table IV, which are not included in Table 2 in the main paper. Here, k^* refers to an estimated number of clusters. The actual number of clusters is indicated as k. Here the standard deviation is that of the performance measure, not of the mean statistic.

Observe in Table V, that even though in some datasets like Zoo, Coil20, Lymphoma (Microarray), etc., COMET is not the best performing algorithm, still it gives performance close to the respective best performing algorithms.

I. Case study on Wisconsin Dataset

We evaluate our algorithm's performance on the Wisconsin dataset, available, and compare it with other algorithms. The dataset contains 699 cases from a study on breast cancer patients, with 9 categorical features and two classes: Benign and Malignant. After refining the dataset to exclude missing values, we work with 683 instances.

The case study focuses on how our algorithm performs in the presence of noise, which is common in real-life clustering tasks. We introduce varying levels of uniform noise and test the algorithms at 0%, 5%, 10%, 15%, and 20% noise. For each noise level, we add $\lfloor np \rfloor$ new "noise" datapoints uniformly within the minimum hypercube containing the original data. The results, shown in Table VI and Figure 6, reveal that COMET outperforms all algorithms, achieving the highest ARI and AMI across noise levels. Here the standard deviation is that of the performance measure, not of the mean statistic. Convex clustering and Robust Convex clustering also perform well,

Dataset	Index	KM	MKM	CC	RCC	RConv	RBKM	COMET
Iris (k = 3)	k^*	3.21±1.01	2.91±0.79	2.57±0.51	147.90±0.70	3.71±0.47	2.90±0.80	3.21±0.43
	ARI	0.56±0.06$^\sim$	**0.59±0.06**	0.55±0.01†	0.001±0.00†	0.46±0.06†	**0.59±0.06**	0.55±0.038$^\sim$
	AMI	0.66±0.05$^\sim$	0.67±0.04$^\sim$	**0.71±0.01**	0.006±0.002†	0.61±0.03†	0.67±0.04$^\sim$	0.65±0.03†
Ecoli (k = 8)	k^*	3.20±2.24	1.92±1.82	34.21±1.25	334.80±2.56	16.86±2.66	2.00±0.00	10.6±1.95
	ARI	0.27±0.23†	0.10±0.17†	0.47±0.06$^\sim$	0.00±0.00†	**0.51±0.04**	0.04±0.01†	0.46±0.04$\sim$
	AMI	0.25±0.21†	0.09±0.16†	**0.46±0.02**	0.002±0.004†	0.46±0.01$^\sim$	0.09±0.01†	0.43±0.02$^\sim$
Zoo (k = 7)	k^*	4.43±2.39	4.37±1.87	16.78±2.26	66.93±3.95	6.00±0.88	2.00±0.00	6.57±0.85
	ARI	0.53±0.22†	0.64±0.23†	0.71±0.10†	0.11±0.02†	**0.89±0.09**	0.03±0.04†	0.85±0.02$^\sim$
	AMI	0.61±0.21†	0.69±0.21†	0.75±0.05†	0.27±0.03†	0.84±0.06$^\sim$	0.05±0.02†	**0.85±0.02**
Brain (k = 5)	k^*	5±1.92	5±1.49	18±0.00	42±0.00	5±0.36	2±0.50	4±0.00
	ARI	0.26±0.10†	0.26±0.10†	0.64±0.02$^\sim$	0.00±0.00†	0.56±0.06†	0.016±0.02†	**0.66±0.03**
	AMI	0.33±0.10†	0.27±0.11†	0.62±0.03†	0.00±0.00†	0.62±0.05†	0.03±0.04†	**0.72±0.03**
Lung (k = 5)	k^*	4.21±1.63	4.41±1.64	23.86±0.36	7.40±2.16	3.00±0.00	1.04±0.20	5.00±0.00
	ARI	0.39±0.20†	**0.53±0.19**	0.35±0.004†	0.31±0.13†	0.35±0.003†	0.00±0.00†	0.36±0.003†
	AMI	0.49±0.20†	**0.55±0.17**	0.35±0.001†	0.22±0.09†	0.37±0.001†	0.00±0.00†	0.54±0.01$^\sim$
Lymphoma(bio) (k = 9)	k^*	4.28±1.61	6.32±1.41	96±0.00	96±0.00	6.57±0.51	2±0.00	6.93±0.82
	ARI	0.37±0.12†	**0.49±0.09**	0.00±0.00†	0.00±0.00†	0.26±0.09†	0.06±0.06†	0.41±0.02†
	AMI	0.47±0.08†	**0.56±0.07**	0.00±0.00†	0.00±0.00†	0.45±0.08†	0.09±0.06†	0.51±0.01$^\sim$
Coil20 (k = 20)	k^*	8.66±3.72	5.62±1.69	90.07±5.01	1440±0.00	20±0.00	1.31±0.47	19±0.00
	ARI	0.30±0.11†	0.21±0.05†	0.69±0.001†	0.00±0.00†	**0.82±0.00**	0.00±0.00†	0.80±0.00†
	AMI	0.58±0.09†	0.50±0.07†	0.86±0.00†	0.00±0.00†	**0.93±0.00**	0.00±0.00†	0.92±0.00†
Wdbc (k = 2)	k^*	1±0.00	1±0.00	136.43±5.65	569±0.00	6.78±0.58	2±0.00	2.21±0.42
	ARI	0.00±0.00†	0.00±0.00†	**0.38±0.03**	0.00±0.00†	0.09±0.01†	0.001±0.002†	0.17±0.31†
	AMI	0.00±0.00†	0.00±0.00†	**0.24±0.01**	0.00±0.00†	0.08±0.01†	0.01±0.003†	0.14±0.26†
Lymphoma (Microarray) (k = 3)	k^*	2.44±1.63	2.23±1.48	3±0.00	3±0.85	2.43±0.51	1.06±0.25	3±0.00
	ARI	0.22±0.27†	0.22±0.28†	0.79±0.00†	**0.86±0.03**	0.33±0.42†	0.00±0.011†	0.79±0.00†
	AMI	0.26±0.29†	0.27±0.31†	0.71±0.00†	**0.81±0.03**	0.29±0.37†	0.00±0.001†	0.71±0.00†

TABLE V: Results for Real Life Datasets

† : significantly different from the best performing algorithm , $^\sim$: statistically same as the best performing algorithm .

Index	Noise	KM	MKM	CC	RCC	RConv	RBKM	COMET
ARI	0	0.52±0.36†	0.55±0.39†	0.79±0.00†	0.01±0.00†	0.85±0.00†	0.002±0.001†	**0.88±0.00**
	5	0.58±0.36†	0.65±0.33†	0.82±0.01†	0.01±0.01†	0.84±0.00†	0.08±0.03†	**0.87±0.00**
	10	0.52±0.35†	0.47±0.39†	0.81±0.01†	0.01±0.00†	0.85±0.03†	0.15±0.06†	**0.87±0.01**
	15	0.66±0.29†	0.54±0.38†	0.80±0.02†	0.01±0.01†	0.84±0.02†	0.24±0.05†	**0.86±0.01**
	20	0.40±0.39†	0.53±0.38†	0.82±0.03†	0.01±0.01†	0.86±0.03†	0.27±0.11†	**0.88±0.02**
AMI	0	0.47±0.32†	0.47±0.34†	0.66±0.00†	0.08±0.00†	0.73±0.00†	0.002±0.001†	**0.80±0.00**
	5	0.52±0.32†	0.57±0.29†	0.69±0.01†	0.06±0.001†	0.73±0.01†	0.12±0.02†	**0.76±0.02**
	10	0.48±0.31†	0.41±0.34†	0.67±0.01†	0.07±0.003†	0.75±0.03†	0.19±0.05†	**0.76±0.01**
	15	0.60±0.25†	0.47±0.33†	0.67±0.02†	0.08±0.005†	0.72±0.02$^\sim$	0.27±0.03†	**0.75±0.03**
	20	0.37±0.35†	0.46±0.33†	0.68±0.02†	0.07±0.01†	0.76±0.03$^\sim$	0.31±0.06†	**0.77±0.02**
k^*	0	2.09±0.85	1.92±0.83	15±0.00	462±0.00	3±0.00	2±0.00	**3±0.00**
	5	2.13±0.45	2.2±0.81	15±1.86	508±7.12	3±0.56	2±0.00	**2±0.00**
	10	2.25±0.63	2.00±0.82	15±1.86	477±9.06	2±1.04	2±0.00	**3±0.00**
	15	2.47±0.62	2.04±0.92	15±1.42	457±±8.57	3±0.97	2±0.00	**3±0.00**
	20	1.93±0.69	2.12±0.83	15±1.42	481±9.21	3±0.97	2±0.00	**2±0.00**

TABLE VI: Performance of Different Algorithms on **Wisconsin** on different Noise levels

† : significantly different from the best performing algorithm, $^\sim$: statistically same as the best performing algorithm .

but slightly worse than COMET. $k-$means and MoM $k-$means are similar but much less effective, while RBKM and RCC perform poorly, with RBKM improving as noise increases. The *t*-SNE plots showing the clustering results of various algorithms on this datsets are provided in the appendix (see A-J).

J. t-SNE Plots for Wisconsin and Brain dataset

t-SNE plots for the clustering results on Wisconsin and Brain dataset using various clustering algorithms are given in Figure 7, 8. The *t*-SNE plots help in understanding the clustering pattern picked by various algorithms better for high dimensional datasets.

Fig. 6: Line plot for performance of different algorithms on **Wisconsin**

(a) Dataset (b) KM (c) MKM (d) CC

(e) RCC (f) RConv (g) RBKM (h) COMET

Fig. 7: t-SNE plot of the Wisconsin dataset after clustering under various algorithms at 10% noise.

K. Wilcoxon-Rank Sum Test

To assess whether the ARI and AMI scores produced by our algorithm are *significantly higher* than those of selected baseline clustering methods, we employ the Wilcoxon Rank-Sum test. The ARI and AMI scores are computed for various algorithms on different 10% contaminated datasets, and the corresponding p-values of the Wilcoxon Rank-Sum test are estimated using *Monte Carlo simulation*. The results are presented in Table VII and VIII. For any value in the table with $p \leq 0.05$, we consider the difference to be *statistically significant*, indicating that our algorithm produces higher ARI and AMI scores under the tested conditions.

Based on the values in the Table VII and VIII, we observe that the performance measures for our algorithm are significantly higher than those for other algorithms. Our closest competitor is RConv, which performs better than us on 4 datasets out of the 15 we have tested on.

L. Ablation Study

1) Gamma: We will do a sensitivity analysis of the hyperparameter γ on the performance of our algorithm For the convex clustering cost function, once a graph is created using $k-$NN for some k, the u_i's converge to the mean of the connected components if the value of γ is high. During our test runs, we also took a large value of γ, say

(a) Dataset (b) KM (c) MKM (d) CC

(e) RCC (f) RConv (g) RBKM (h) COMET

Fig. 8: *t*-SNE plot of the Brain dataset after clustering under various algorithms.

Dataset	**CC**	**RCC**	**RConv**	**KM**	**MKM**	**RBKM**
Iris	0.9409	2.32E-06	0.000385	0.97549	0.99922	0.998
Newthyroid	3.35E-06	1.96E-06	1.13E-06	1.61E-09	1.42E-09	1.48E-09
Ecoli	0.7399	1.37E-06	0.9967	2.52E-09	2.52E-09	1.31E-09
Wisconsin	3.35E-06	3.72E-06	1.78E-06	2.79E-05	1.41E-07	1.27E-09
Wine	0.0001	4.49E-07	1.64E-05	0.18801	0.001763	1.63E-09
Zoo	3.88E-05	2.18E-06	0.9984	5.96E-09	8.90E-07	1.49E-09
Dermatology	3.35E-06	3.62E-07	2.6E-05	1.05E-05	2.07E-07	1.61E-09
Brain	0.9993	2.41E-07	4.96E-05	1.98E09	1.72E-09	1.55E-10
Lung	3.35E-06	0.0159	5.28E-07	0.913832	0.9999	5.29E-17
Lymphoma	3.35E-06	4.44E-07	1.31E-06	0.03995	0.9997	1.56E-09
Coil-20	3.55E-06	1.17E-07	0.999	1.74E-09	1.73E-09	5.35E-15
WDBC	0.9949	4.46E-07	0.9895	9.09E-21	9.09E-21	0.00126
Lung-discrete	3.35E-06	3.50E-06	9.7E-07	2.04E-09	6.03E-09	1.27E-09
ORLRaw10P	3.35E-06	1.17E-07	3.55E-07	1.65E-09	1.63E-09	1.21E-09
Lymphoma (micro)	0.5	0.9999	0.000241	1.67E-08	9.58E-08	5.7E-16

TABLE VII: COMET vs Other Algorithms (ARI values only)

50000. We observe that, for our algorithm, the value of γ doesn't influence the final result, provided it is large (say, $\geq$ 1000). Refer to the Figure 9 for the plot.

2) Ablation study on Wisconsin Dataset: We will study the fluctuations in our performance measures with varying hyperparameters. We will turn to the **Wisconsin Breast Cancer** dataset for our ablation studies. We have shown in section A-L1 that the hyperparameter γ does not have much influence on the final clustering of the dataset. Two hyperparameters, k, which is a hyperparameter for the $k-$NN graph structure, and μ, require tuning to achieve optimal performance of our algorithm. For our experiments, we are varying k in $\{24, 27, 30, 33, 36\}$, μ in $[4, 17]$ and p(noise level) in $\{0\%, 5\%, 10\%, 15\%, 20\%\}$. For each pair (p,k), we are varing μ and reporting the mean of ARI and AMI. For each noise level, we observe that both the AMI and ARI values increase stochastically with μ and converge to a value. The fluctuations in ARI increase gradually with noise level for every value of k. Within each noise level however, ARI is most stable for the mid-range of k, which is 27 and 30, indicating that values of k from 27 to 30 have higher stabilities. Same observation can be made for AMI as well. All the fluctuations can be attributed to the randomness in adding noise and the optimization procedure of our objective function. The figures 10 and 11 correspond to the variation in ARI and AMI respectively for Wisconsin dataset.

Dataset	CC	RCC	RConv	KM	MKM	RBKM
Iris	0.9999	2.32E-06	0.00029	0.58855	0.9992	0.9669
Newthyroid	3.35E-06	1.96E-06	7.47E-05	1.61E-09	1.42E-09	1.5E-09
Ecoli	0.9999	1.37E-06	0.9997	2.52E-09	2.52E-09	1.31E-09
Wisconsin	3.35E-06	3.72E-06	1.78E-06	0.004352	9.95E-07	1.27E-09
Wine	6.85E-06	4.49E-07	1.64E-05	0.17594	0.000765	1.63E-09
Zoo	5.15E-06	2.18E-06	0.4583	3.24E-09	1.44E-07	1.49E-09
Dermatology	3.35E-06	3.62E-07	2.6E-05	0.000254	1.65E-06	1.61E-09
Brain	3.35E-06	2.41E-07	4.96E-05	1.73E-09	1.72E-09	1.55E-10
Lung	3.35E-06	1.87E-06	5.28E-07	0.83273	0.9847	5.3E-17
Lymphoma	3.35E-06	4.44E-07	0.0921	0.0508	0.9996	1.6E-09
Coil-20	3.54E-06	1.17E-07	0.999	1.74E-09	1.73E-09	5.6E-15
WDBC	0.9949	4.46E-07	0.994	1.79E-11	9.09E-21	0.8603
Lung-discrete	3.35E-06	3.50E-06	9.7E-07	2.34E-09	2.51E-07	1.27E-09
ORLRaw10P	3.35E-06	1.17E-07	3.55E-07	1.65E-09	1.63E-09	1.21E-09
Lymphoma (micro)	0.5	0.9999	0.000241	1.06E-07	1.28E-06	5.7E-16

TABLE VIII: COMET vs Other Algorithms (AMI values only)

Fig. 9: ARI for different values of γ after adding 10% noise

3) Ablation study on Newthyroid Dataset: We will turn to the Newthyroid dataset for our ablation studies. We have shown in section A-L1 that the hyperparameter γ does not have much influence on the final clustering of the dataset. Two hyperparameters, k, which is a hyperparameter for the $k-$NN graph structure, and μ, require tuning to achieve optimal performance of our algorithm. For our experiments, we are varying k in $\{31, 38, 45, 52, 59\}$, μ in $[12, 110]$ and p(noise level) in $\{0\%, 5\%, 10\%, 15\%, 20\%\}$. For each pair ($p$,$k$), we are varying μ and reporting the mean of ARI and AMI. The ARI is almost always between 0.93 and 0.98 and very rarely drops to 0.92 at higher noise levels. The AMI values are also between 0.82 and 0.92. The Here we did not notice much fluctuations for both ARI and AMI values even for 20% noise level. All the fluctuations can be attributed to the randomness of adding noise and the optimization procedure of our objective function. The fluctuations increase with increasing p, however the increase is only slight. The fluctuations decrease while choosing higher values of k (such as 52 in our case). The effect of changing μ is also not apparent from the plots indicating that there are well-separated clusters in this dataset. However, for much smaller values of μ (say below 10), the ARI and AMI values are expected to drop. The figures 12 and 13 correspond to the variation in ARI and AMI respectively for Brain dataset.

M. Table of Plots for Simulated Datasets

(a) 0 % noise (b) 5 % noise (c) 10 % noise

(d) 15 % noise (e) 20 % noise

Fig. 10: Ablation Studies of the ARI Values obtained from the Wisconsin Breast Cancer Dataset. Each subfigure corresponds to a different level of noise introduced into the dataset.

(a) 0 % noise (b) 5 % noise (c) 10 % noise

(d) 15 % noise (e) 20 % noise

Fig. 11: Ablation Studies of the AMI Values obtained from the Wisconsin Breast Cancer Dataset. Each subfigure corresponds to a different level of noise introduced into the dataset.

(a) 0 % noise (b) 5 % noise (c) 10 % noise

(d) 15 % noise (e) 20 % noise

Fig. 12: Ablation Studies of the ARI Values obtained from the NewThyroid Dataset. Each subfigure corresponds to a different level of noise introduced into the dataset.

(a) 0 % noise (b) 5 % noise (c) 10 % noise

(d) 15 % noise (e) 20 % noise

Fig. 13: Ablation Studies of the AMI Values obtained from the NewThyroid Dataset. Each subfigure corresponds to a different level of noise introduced into the dataset.

Fig. 14: Clustering results for 7 algorithms across 3 datasets. Each row corresponds to an algorithm, and each column corresponds to a dataset.