On Approximability of ℓ_2^2 Min-Sum Clustering

Karthik C. S.
Rutgers University, Piscataway, NJ, USA

Euiwoong Lee
University of Michigan, Ann Arbor, MI, USA

Yuval Rabani
The Hebrew University of Jerusalem, Israel

Chris Schwiegelshohn
Aarhus University, Denmark

Samson Zhou
Texas A&M University, College Station, TX, USA

Abstract

The ℓ_2^2 min-sum k-clustering problem is to partition an input set into clusters $C_1, \ldots, C_k$ to minimize $\sum_{i=1}^k \sum_{p,q \in C_i} \|p-q\|_2^2$. Although ℓ_2^2 min-sum k-clustering is NP-hard, it is not known whether it is NP-hard to approximate ℓ_2^2 min-sum k-clustering beyond a certain factor.

In this paper, we give the first hardness-of-approximation result for the ℓ_2^2 min-sum k-clustering problem. We show that it is NP-hard to approximate the objective to a factor better than 1.056 and moreover, assuming a balanced variant of the Johnson Coverage Hypothesis, it is NP-hard to approximate the objective to a factor better than 1.327.

We then complement our hardness result by giving a fast PTAS for ℓ_2^2 min-sum k-clustering. Specifically, our algorithm runs in time $O(n^{1+o(1)} d \cdot 2^{(k/\varepsilon)^{O(1)}})$, which is the first nearly linear time algorithm for this problem. We also consider a learning-augmented setting, where the algorithm has access to an oracle that outputs a label $i \in [k]$ for input point, thereby implicitly partitioning the input dataset into k clusters that induce an approximately optimal solution, up to some amount of adversarial error $\alpha \in \left[0, \frac{1}{2}\right)$. We give a polynomial-time algorithm that outputs a $\frac{1+\gamma\alpha}{(1-\alpha)^2}$-approximation to ℓ_2^2 min-sum k-clustering, for a fixed constant $\gamma > 0$.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Facility location and clustering

Keywords and phrases Clustering, hardness of approximation, polynomial-time approximation schemes, learning-augmented algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2025.62

Related Version *Full Version*: `https://arxiv.org/abs/2412.03332`

Funding *Karthik C. S.*: Supported by NSF grants CCF-2313372 and CCF-2443697, a grant from the Simons Foundation, Grant Number 825876, Awardee Thu D. Nguyen, and partially funded by the Ministry of Education and Science of Bulgaria's support for INSAIT, Sofia University "St. Kliment Ohridski" as part of the Bulgarian National Roadmap for Research Infrastructure.
Euiwoong Lee: Supported in part by NSF grant CCF-2236669 and Google.
Yuval Rabani: Supported in part by ISF grants 3565-21 and 389-22, and by BSF grant 2023607.
Chris Schwiegelshohn: Supported in part by the Independent Research Fund Denmark (DFF) under a Sapere Aude Research Leader grant No 1051-00106B and Google.
Samson Zhou: Supported in part by NSF CCF-2335411. The work was conducted in part while the author was visiting the Simons Institute for the Theory of Computing as part of the Sublinear Algorithms program.

41st International Symposium on Computational Geometry (SoCG 2025).
Editors: Oswin Aichholzer and Haitao Wang; Article No. 62; pp. 62:1–62:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

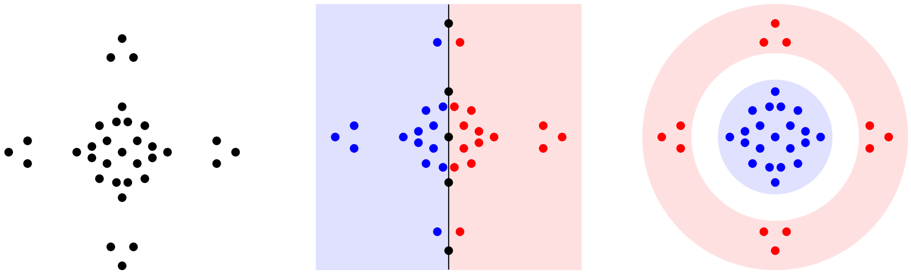
1 Introduction

Clustering is a fundamental technique that partitions an input dataset into distinct groups called clusters, which facilitate the identification and subsequent utilization of latent structural properties underlying the dataset. Consequently, various formulations of clustering are used across a wide range of applications, such as computational biology, computer vision, data mining, and machine learning [39, 70]. Ideally, the elements of each cluster are more similar to each other than to elements in other clusters. To formally capture this notion, a dissimilarity metric is often defined on the set of input elements, so that more closer objects in the metric correspond to more similar objects. Perhaps the most natural goal would be to minimize the intra-cluster dissimilarity in a partitioning of the input dataset. This objective is called the *min-sum k-clustering* problem and has received significant attention due to its intuitive clustering objective [33, 37, 58, 63, 11, 24, 22, 2, 12, 10, 18].

In this paper, we largely focus on the ℓ_2^2 min-sum k-clustering formulation. Formally, the input is a set X of n points in $\mathbb{R}^d$ and the goal is to partition $X = C_1 \dot\cup \cdots \dot\cup C_k$ into k clusters to minimize the quantity $\min_{C_1,\ldots,C_k} \sum_{i=1}^k \sum_{p,q\in C_i} \|p-q\|_2^2$, where $\|\cdot\|_2$ denotes the standard Euclidean ℓ_2 norm.

Whereas classical centroid-based clustering problems such as k-means and k-median leverage distances between data points and cluster centroids to identify convex shapes that partition the dataset, min-sum k-clustering is a density-based clustering that can handle complex structures in data that may not be linearly separable. In particular, min-sum k-clustering can be more effective than traditional centroid-based clustering in scenarios where clusters have irregular, non-convex shapes or overlapping clusters. A simple example of the ability of min-sum clustering to capture more natural structure is an input that consists of two concentric dense rings of points in the plane. Whereas min-sum clustering can partition the points into the separate rings, centroid-based clustering will instead create a separating hyperplane between these points, thereby "incorrectly" grouping together points of different rings. See Figure 1 for an example of the ability of min-sum clustering to capture natural structure in cases where centroid-based clustering fails.

Moreover, min-sum clustering satisfies Kleinberg's consistency axiom [47], which informally demands that the optimal clustering for a particular objective should be preserved when distances between points inside a cluster are shrunk and distances between points in different clusters are expanded. By contrast, many centroid-based clustering objectives, including k-means and k-median, do not satisfy Kleinberg's consistency axiom [57].



(a) Input dataset. **(b)** Centroid-based clustering. **(c)** Density-based clustering.

Figure 1 Clustering of input dataset in Figure 1a with $k = 2$. Figure 1b is an optimal centroid-based clustering, e.g., k-median or k-means, while the more natural clustering in Figure 1c is an optimal density-based clustering, e.g., ℓ_2 min-sum k-clustering.

On the other hand, theoretical understanding of density-based clustering objectives such as min-sum k-clustering is far less developed than that of their centroid-based counterparts. It can be shown that min-sum k-clustering with the ℓ_2^2 cost function is NP-hard, using arguments from [2]. The problem is NP-hard even for $k = 2$ [25] in the metric case, where the only available information about the points is their pairwise dissimilarity. In fact, for general k in the metric case, it is NP-hard to approximate the problem within a 1.415-multiplicative factor [32, 18]. However, no such hardness of approximation is known for the Euclidean case, i.e., ℓ_2^2 min-sum, where the selected cost function is based on the geometry of the underlying space; the only known lower bound is the NP-hardness of the problem [2, 10, 3]. Thus a fundamental open question is:

▶ **Question 1.** *Is ℓ_2^2 min-sum k-clustering APX-hard? That is, does there exist a natural hardness-of-approximation barrier for polynomial time algorithms?*

Due to existing APX-hardness results for centroid-based clustering such as k-means and k-median [50, 17, 19], it is widely believed that ℓ_2^2 min-sum clustering is indeed APX-hard. Thus, there has been a line of work preemptively seeking to overcome such limitations. Indeed, on the positive side, [36] first showed that min-sum k-clustering in the d-dimensional ℓ_2^2 case can be solved in polynomial time if both d and k are constants. For general graphs and fixed constant k, [33] gave a 2-approximation algorithm using runtime $n^{\mathcal{O}(k)}$. The approximation guarantees were improved by a line of work [37, 58, 63], culminating in polynomial-time approximation schemes by [24] for both the ℓ_2^2 case and the metric case. Without any assumptions on d and k, [11] introduced a polynomial algorithm that achieves an $\mathcal{O}\left(\frac{1}{\varepsilon}\log^{1+\varepsilon} n\right)$-multiplicative approximation. Therefore, a long-standing direction in the study of ℓ_2^2 min-sum clustering is:

▶ **Question 2.** *How can we algorithmically bridge the gap between the NP-hardness of solving the ℓ_2^2 min-sum clustering and the large multiplicative guarantees of existing approximation algorithms?*

A standard approach to circumvent poor dependencies on the size of the input dataset is to sparsify the problem. Informally, we would like to reduce the search space by considering fewer candidate solutions and reduce the dependency on the number of input points by aggregating them. For min-sum clustering this is a particular challenge, as a candidate solution is a partition and the cost of that partition depends on all pairwise distances between all the points. While sparsification algorithms exist for graph clustering [40, 51] and k-means clustering [21, 20], where the output is typically called a coreset, similar constructions are not known to exist for min-sum clustering.

Another standard approach to overcome limitations inherent in worst-case impossibility barriers is to consider beyond worst case analysis. To that end, recent works have observed that in many applications, auxiliary information is often available and can potentially form the foundation upon which machine learning models are built. For example, previous datasets with potentially similar behavior can be used as training data for models to label future datasets. However, these heuristics lack provable guarantees and can produce embarrassingly inaccurate predictions when generalizing to unfamiliar inputs [65]. Nevertheless, *learning-augmented algorithms* [60] have been shown to achieved both good algorithmic performance when the oracle is accurate, i.e., consistency, and standard algorithmic performance when the oracle is inaccurate, i.e., robustness for a wide range of settings, such as data structure design [48, 59, 55], algorithms with faster runtime [26, 15, 23], online algorithms with better competitive ratio [62, 30, 49, 68, 69, 9, 35, 56, 1, 4, 8, 31, 45, 41, 5, 64], and streaming algorithms that are more space-efficient [34, 38, 43, 14, 13, 54]. In particular, [28, 61] introduce algorithms for k-means and k-median clustering that can achieve approximation guarantees beyond the known APX-hardness limits.

1.1 Our Contributions

In this paper, we perform a comprehensive study on the approximability of the ℓ_2^2 min-sum k-clustering by answering Question 1 and Question 2.

Hardness-of-approximation of min-sum k-clustering. We first answer Question 1 in the affirmative, by not only showing that the ℓ_2^2 min-sum k-clustering is APX-hard but further giving an explicit constant NP-hardness of approximation result for the problem.

▶ **Theorem 3** (Hardness of approximation of ℓ_2^2 min-sum k-clustering)**.** *It is NP-hard to approximate ℓ_2^2 min-sum k-clustering to a factor better than* 1.056*. Moreover, assuming the Dense and Balanced Johnson Coverage Hypothesis (*$\mathsf{Balanced-JCH}^*$*), we have that the ℓ_2^2 min-sum k-clustering is NP-hard to approximate to a factor better than* 1.327*.*

We remark that $\mathsf{Balanced-JCH}^*$ in the theorem statement above is simply a balanced formulation of the recently introduced Johnson Coverage Hypothesis [19].

Fast polynomial-time approximation scheme. In light of Theorem 3, a natural question would be to closely examine alternative conditions in which we can achieve a $(1+\varepsilon)$-approximation to min-sum k-clustering, i.e., Question 2. To that end, there are a number of existing polynomial-time approximation schemes (PTAS) [37, 58, 63, 24], the best of which uses runtime $n^{\mathcal{O}\left(k/\varepsilon^2\right)}$ for the ℓ_2^2 case. However, as noted by [22], even algorithms with runtime quadratic in the size n of the input dataset are generally not sufficiently scalable to handle large datasets. In this paper, we present an algorithm with a running time that is nearly nearly linear. Specifically, we show

▶ **Theorem 4.** *There exists an algorithm running in time $O\left(n^{1+o(1)}d\cdot 2^{\eta\cdot k^2\cdot\varepsilon^{-12}\log^2(k/(\varepsilon\delta))}\right)$, for some absolute constant η, that computes a $(1+\varepsilon)$-approximate solution to ℓ_2^2 k-MinSum Clustering with probability $1-\delta$.*

We again emphasize that the runtime of 4 is linear in the size n of the input dataset, though it has exponential dependencies in both the number k of clusters and the approximation parameter $\varepsilon > 0$. By contrast, the best previous PTAS uses runtime $n^{\mathcal{O}\left(k/\varepsilon^2\right)}$, which has substantially worse dependency on the size n of the input dataset.

Learning-augmented algorithms. Unfortunately, exponential dependencies on the number k of clusters can still be prohibitive for moderate values of k. To that end, we turn our attention to learning-augmented papers. We consider the standard *label oracle* model for clustering, where the algorithm has access to an oracle that provides a label for each input point. Formally, for each point x of the n input points, the oracle outputs a label $i\in[k]$ for x, so that the labels implicitly partition the input dataset into k clusters that induce an approximately optimal solution. However, the oracle also has some amount of adversarial error that respects the precision and recall of each cluster; we defer the formal definition to Definition 25. One of the reasons label oracles have been used for learning-augmented algorithms for clustering is their relative ease of acquisition via machine learning models that are trained on a similar distribution of data. For example, a smaller separate dataset can be observed and used as a "training" data, an input to some heuristic to cluster the initial data, which we can then use to form a predictor for the actual input dataset. Indeed, implementations of label oracles have been shown to perform well in practice [28, 61].

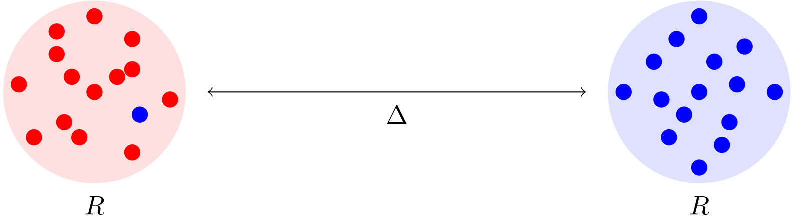


Figure 2 Note that with arbitrarily small error rate, i.e., $\frac{1}{n}$, a single mislabeled point among the n input points causes the resulting clustering to be arbitrarily bad for $\Delta \gg n^2 \cdot R$.

We also remark that perhaps counter-intuitively, a label oracle with arbitrarily high accuracy does not trivialize the problem. In particular, the naïve algorithm of outputting the clustering induced by the labels does not work. As a simple example, consider an input dataset where half of the n points are at $x = 0$ and the other half of the points are at $x = 1$. Then for $k = 2$, the clear optimal clustering is to cluster the points at the origin together, and cluster the points at $x = 1$ together, which induces the optimal cost of zero. However, if even one of the n points is incorrect, then the clustering output by the labels has cost at least 1. Therefore, even with error rate as small as $\frac{1}{n}$, the multiplicative approximation of the naïve algorithm can be arbitrarily bad. See Figure 2 for an illustration of this example. Of course, this example does not rule out more complex algorithms that combines the labels with structural properties of optimal clustering and indeed, our algorithm utilizes such properties.

We give a polynomial-time algorithm for the ℓ_2^2 min-sum k-clustering that can provide guarantees beyond the computational limits of Theorem 3, given a sufficiently accurate oracle.

▶ **Theorem 5.** *There exists a polynomial-time algorithm that uses a label predictor with error rate* $\alpha \in \left[0, \frac{1}{2}\right)$ *and outputs a* $\frac{1+\gamma\alpha}{(1-\alpha)^2}$*-approximation to the* ℓ_2^2 *min-sum* k*-clustering problem, where* $\gamma = 7.7$ *for* $\alpha \in \left[0, \frac{1}{7}\right)$ *or* $\gamma = \frac{5\alpha - 2\alpha^2}{(1-2\alpha)(1-\alpha)}$ *for* $\alpha \in \left[0, \frac{1}{2}\right)$.

We remark that Theorem 5 does not require the true error rate α as an input parameter. Because we are in an offline setting, where can run Theorem 5 multiple times with guesses for the true error rate α, in decreasing powers of $\frac{1}{\lambda}$ for any constant $\lambda > 1$. We can then compare the resulting clustering output by each guess for α and take the output the best clustering.

1.2 Technical Overview

Hardness of approximation. Recently, the authors of [19] put forth the Johnson Coverage Hypothesis (JCH) and introduced a framework to obtain (optimal) hardness of approximation results for k-median and k-means in ℓ_p-metrics. The proof of Theorem 3 builds on this framework. JCH roughly asserts that for large enough constant z, given as input an integer k and a collection of z-sets (i.e., sets each of size z) over some universe, it is NP-hard to distinguish the completeness case where there is a collection C of k many $(z-1)$-sets such that every input set is covered[1] by some set in C, from the soundness case where every collection C of k many $(z-1)$-sets does not cover much more than $1-\frac{1}{e}$ fraction of the input sets (see Hypothesis 7 for a formal statement).

In this paper, we consider a natural generalization of JCH, called $\mathsf{Balanced}-\mathsf{JCH}^*$, where we assume that the number of input sets is "dense", i.e., $\omega(k)$, and more importantly that in the completeness case, the collection C covers the input z-sets in a balanced manner, i.e., we can partition the input to k equal parts such that each part is completely covered by a single set in C (see Hypothesis 9 for a formal statement).

We now sketch the proof of Theorem 3 assuming $\mathsf{Balanced}-\mathsf{JCH}^*$. Given a collection of m many z-sets over a universe $[n]$ as input, we create a point for each input set, which is simply the characteristic vector of the set as a subset of $[n]$, i.e., the points are all n-dimensional Boolean vectors of Hamming weight z. In the completeness case, from the guarantees of $\mathsf{Balanced}-\mathsf{JCH}^*$, it is easy to see that the points created can be divided into k equal clusters of size m/k such that all the z-sets of a cluster are completely covered by a single $(z-1)$-set. This implies that the squared Euclidean distance between a pair of points within a cluster is exactly 2 and thus the ℓ_2^2 min-sum k-clustering cost is $k\cdot 2\cdot (m/k)(m/k-1)\approx 2m^2/k$.

On the other hand, in the soundness case, we first use the density guarantees of $\mathsf{Balanced}-\mathsf{JCH}^*$ to argue that most clusters are not small. Then suppose that we had a low cost ℓ_2^2 min-sum k-clustering, we look at a typical cluster and observe that the squared distance of any two points in the cluster must be a positive even integer, and it is exactly 2 only when the two input sets corresponding to the points intersect on a $(z-1)$-set. Thus, if the cost of the clustering is close to $\alpha\cdot 2m^2/k$ (for some $\alpha\geq 1$), then we argue (using convexity) that for a typical cluster that there must be a $(z-1)$-set that covers $(1-\alpha')m/k$ many z-sets in that cluster, where α' depends on α. Thus, from this we decode k-many $(z-1)$-sets which cover a large fraction of the input z-sets. In order to obtain the unconditional NP-hardness result, much like in [19], we need to extend the above reduction to a more general problem. This is indeed established in Theorem 12, and after this we prove a special case of a generalization of $\mathsf{Balanced}-\mathsf{JCH}^*$ (when $z=3$) which is done in Theorem 11 and this involved proving additional properties of the reduction of [19] from the multilayered PCPs of [27, 46] to 3-Hypergraph Vertex Coverage.

Nearly Linear Time PTAS. An important feature of ℓ_2^2 Min-Sum Clustering is that we can use assignments of clusters to their mean to obtain the cost of the points in the cluster, an idea previously used in [37, 58, 63, 24]. We show how to reduce the number of candidate means to a constant (depending only on k and ε. The idea here is to use D^2 sampling methods akin to k-means++ [6]. Unfortunately, by itself, it is not sufficient as there may exist clusters that have significant Min-Sum clustering cost, but are not detectable by D^2 sampling. To this end, we augment D^2 sampling via a careful pruning strategy that removes high costing points, increasing the relative cost of clusters of high density. Thereafter, we show that given sufficiently many samples, we can find a small set of suitable candidate means that are induced by a nearly optimal clustering.

What remains to be shown is how to find an assignment of points to these centers with similar cost. For this, we could use a flow-based approach, but this results in a n^3 running time. Instead, we employ a discretization and bucketing strategy that allows us to sparsify the point set while preserving the Min-Sum clustering cost, akin to coresets.

[1] A $(z-1)$-set covers a z-set if the former is a subset of the latter.

Learning-augmented algorithm. Our starting point for our learning-augmented algorithm for min-sum k-clustering is the learning-augmented algorithms for k-means clustering by [28, 61]. We can rewrite the Min-Sum clustering cost in terms of weighted squared distances to the means or centroids. Our goal is therefore to quickly identify suitable centroids c_i. The algorithms note that the clustering objective can be decomposed across the points that are given each label $i \in [k]$. Thus we consider the subset P_i of points of the input dataset X that are given label i by the oracle.

The cluster P_i can have an α fraction of incorrect points. The main observation is that there can be two cases. Either P_i includes a number of "bad" points that are far from the true mean and thus easy to identify, or P_i includes a number of "bad" points that are difficult to identify but also are close to the true mean and thus do not largely affect the overall clustering cost. Thus the algorithm simply needs to prune away the points that are far away, which can be achieved by selecting the interval of $(1 - \mathcal{O}(\alpha))$ points that has the best clustering cost. Therefore, we have a set of centers for which there exists an assignment that obtains a good approximation of the cost of the optimal min-sum k-clustering; it remains to identify the actual clusters.

To find an assignment of points to candidate means, we use min-cost flow approach, similar to [24]. The constrained min-cost flow problem can be written as a linear program. Therefore to identify the overall clusters, we run any standard polynomial-time algorithm for solving linear programs [44, 66, 67, 52, 53, 16, 42]. It then follows by that well-known integrality theorems for min-cost flow, the resulting solution is integral and thus provides a valid clustering with approximately optimal ℓ_2^2 min-sum k-clustering objective.

2 Hardness of Approximation of ℓ_2^2 Min-Sum k-Clustering

In this section, we show the hardness of approximation of ℓ_2^2 min-sum k-clustering, i.e., Theorem 3. We first define the relevant formulations of Johnson Coverage Hypothesis in Section 2.1. Next, in Section 2.2 we provide the main reduction from the Johnson coverage problem to the ℓ_2^2 min-sum k-clustering problem. Finally, we prove a special case of a generalization of $\mathsf{Balanced-JCH}^*$ which yields the unconditional NP-hardness factor claimed in Theorem 3.

2.1 Johnson Coverage Hypothesis

In this section, we recall the Johnson Coverage problem, followed by the Johnson Coverage hypothesis [19]. Let $n, z, y \in \mathbb{N}$ such that $n \geq z > y$. Let $E \subseteq \binom{[n]}{z}$ and $S \in \binom{[n]}{y}$. We define the coverage of S w.r.t. E, denoted by $\mathsf{cov}(S, E)$ as follows: $\mathsf{cov}(S, E) = \{T \in E \mid S \subset T\}$.

▶ **Definition 6** (Johnson Coverage Problem). *In the (α, z, y)-Johnson Coverage problem with $z > y \geq 1$, we are given a universe $U := [n]$, a collection of subsets of U, denoted by $E \subseteq \binom{[n]}{z}$, and a parameter k as input. We would like to distinguish between the following two cases:*

- **Completeness***: There exists $\mathcal{C} := \{S_1, \ldots, S_k\} \subseteq \binom{[n]}{y}$ such that*

$$\mathsf{cov}(\mathcal{C}) := \underset{i\in[k]}{\cup} \mathsf{cov}(S_i, E) = E.$$

- **Soundness***: For every $\mathcal{C} := \{S_1, \ldots, S_k\} \subseteq \binom{[n]}{y}$ we have $|\mathsf{cov}(\mathcal{C})| \leq \alpha \cdot |E|$.*

We call $(\alpha, z, z-1)$-Johnson Coverage as (α, z)-Johnson Coverage.

Notice that $(\alpha, 2)$-Johnson Coverage Problem is simply the well-studied vertex coverage problem (with gap α). Also, notice that if instead of picking the collection $\mathcal{C}$ from $\binom{[n]}{y}$, we replace it with picking the collection $\mathcal{C}$ from $\binom{[n]}{1}$ with a similar notion of coverage, then we simply obtain the Hypergraph Vertex Coverage problem (which is equivalent to the Max k-Coverage problem for unbounded z). In Figure 3 we provide a few examples of instances of the Johnson coverage problem.

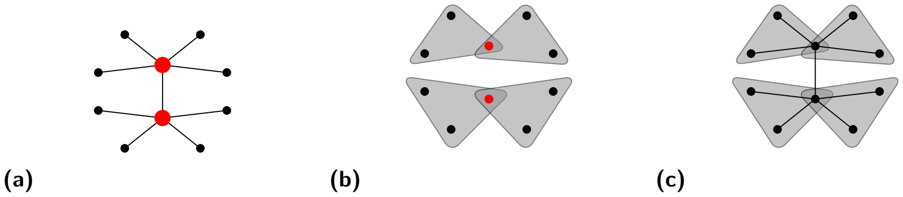


Figure 3 Examples of input instances of the Johnson Coverage Hypothesis for $k = 2$. Figure 3a shows an example of a completeness instance of $(0.7, 2, 1)$, since all subsets of size 2, i.e., all edges, can be covered by $k = 2$ choices of subset of size 1, i.e., two vertices. Figure 3b shows an example of a completeness instance of $(0.7, 3, 1)$, since all subsets of size 3 can be covered by $k = 2$ vertices. Figure 3c shows an example of a soundness instance of $(0.7, 3, 2)$, since at most $2 \leq 0.7 \cdot 4$ subsets of size 3 can be covered by any choice of $k = 2$ edges.

▶ **Hypothesis 7** (Johnson Coverage Hypothesis (JCH) [19]). *For every constant $\varepsilon > 0$, there exists a constant $z := z(\varepsilon) \in \mathbb{N}$ such that deciding the $\left(1 - \frac{1}{e} + \varepsilon, z\right)$-Johnson Coverage Problem is* NP*-Hard.*

Note that since Vertex Coverage problem is a special case of the Johnson Coverage problem, we have that the NP-Hardness of (α, z)-Johnson Coverage problem is already known for $\alpha = 0.944$ [7] (under unique games conjecture).

On the other hand, if we replace picking the collection $\mathcal{C}$ from $\binom{[n]}{z-1}$ by picking from $\binom{[n]}{1}$, then for the Hypergraph Vertex Coverage problem, we do know that for every $\varepsilon > 0$ there is some constant z such that the Hypergraph Vertex Coverage problem is NP-Hard to decide for a factor of $\left(1 - \frac{1}{e} + \varepsilon\right)$ [29]. For continuous clustering objectives, a dense version of JCH is sometimes needed to prove inapproximability results (see [19] for a discussion on this). Thus, we state:

▶ **Hypothesis 8** (Dense Johnson Coverage Hypothesis (JCH*) [19]). JCH *holds for instances (U, E, k) of Johnson Coverage problem where $|E| = \omega(k)$.*

More generally, let (α, z, y)-Johnson Coverage* problem be the special case of the (α, z, y)-Johnson Coverage problem where the instances satisfy $|E| = \omega(k \cdot |U|^{z-y-1})$. Then JCH* states that for any $\varepsilon > 0$, there exists $z = z(\varepsilon)$ such that $(1 - 1/e + \varepsilon, z, z - 1)$-Johnson Coverage* is NP-Hard. This additional property has always been obtained in literature by looking at the hard instances that were constructed. In [17], where the authors proved the previous best inapproximability results for continuous case k-means and k-median, it was observed that hard instances of $(0.94, 2, 1)$-Johnson Coverage constructed in [7] can be made to satisfy the above property. Now we are ready to define the variant of JCH needed for proving inapproximability of ℓ_2^2 min-sum k-clustering. For any two non-empty finite sets A, B, and a constant $\delta \in [0, 1]$, we say a function $f : A \to B$ is δ-balanced if for all $b \in B$ we have $|\{a \in A : f(a) = b\}| \leq (1 + \delta) \cdot \frac{|A|}{|B|}$. We then put forth the following hypothesis.

▶ **Hypothesis 9** (Dense and Balanced Johnson Coverage Hypothesis ($\mathsf{Balanced-JCH}^*$))**.** JCH *holds for instances* (U, E, k) *of Johnson Coverage problem where* $|E| = \omega(k)$ *and in the completeness case there exists* $\mathcal{C} := \{S_1, \ldots, S_k\} \subseteq \binom{[n]}{z-1}$ *and a* **0-balanced** *function* $\psi : E \to [k]$ *such that for all* $T \in E$ *we have* $S_{\psi(T)} \subset T$*.*

More generally, let (α, z, y, δ)-Balanced Johnson Coverage* problem be the special case of the (α, z, y)-Johnson Coverage* problem where the instances admit a δ-balanced function $\psi : E \to [k]$ in the completeness case which partitions E to k parts, say $E_1 \dot\cup \cdots \dot\cup E_k$ such that for all $i \in [k]$ we have $\mathsf{cov}(S_i, E_i) = E_i$ and $|E_i| \leq \frac{|E|}{k} \cdot (1 + \delta)$. Then $\mathsf{Balanced-JCH}^*$ states that for any $\varepsilon > 0$, there exists $z = z(\varepsilon)$ such that $(1 - 1/e + \varepsilon, z, z-1, 0)$-Balanced Johnson Coverage* is NP-Hard. As with the case of JCH^*, the balanced addition to JCH^* is also quite natural and candidate constructions typically give this property for free. To support this point, we will prove some special case of this. In [19] the authors had proved the following special case of JCH^*.

▶ **Theorem 10** ([19])**.** *For any* $\varepsilon > 0$*, given a simple 3-hypergraph* $\mathcal{H} = (V, H)$ *with* $n = |V|$*, it is NP-hard to distinguish between the following two cases:*

- **Completeness:** *There exists* $S \subseteq V$ *with* $|S| = n/2$ *that intersects every hyperedge.*
- **Soundness:** *Any subset* $S \subseteq V$ *with* $|S| \leq n/2$ *intersects at most a* $(7/8 + \varepsilon)$ *fraction of hyperedges.*

Furthermore, under randomized reductions, the above hardness holds when $|H| = \omega(n^2)$*.*

▶ **Theorem 11.** *Theorem 10 holds even with the following additional completeness guarantee for all* $\delta > 0$*: there exists* $S := \{v_1, \ldots, v_k\} \subseteq V$ *and a* δ*-balanced function* $\psi : H \to [k]$ *such that for all* $e \in H$ *we have* $v_{\psi(e)} \in e$*.*

This result will be used to prove the unconditional NP-hardness of approximating ℓ_2^2 min-sum k-clustering problem.

2.2 Inapproximability of ℓ_2^2 min-sum k-clustering

▶ **Theorem 12.** *Assume* (α, z, y, δ)*-Balanced Johnson Coverage** *is* NP*-Hard. For every constant* $\varepsilon > 0$*, given a point-set* $P \subset \mathbb{R}^d$ *of size* n *(and* $d = \mathcal{O}(\log n)$*) and a parameter* k *as input, it is* NP*-Hard to distinguish between the following two cases:*

- **Completeness***: There exists partition* $P_1^* \dot\cup \cdots \dot\cup P_k^* := P$ *such that*

$$\sum_{i \in [k]} \sum_{p,q \in P_i^*} \|p - q\|_2^2 \leq (1 + 3\delta) \cdot (z - y) \cdot \rho n^2 / k,$$

- **Soundness***: For every partition* $P_1 \dot\cup \cdots \dot\cup P_k := P$ *we have*

$$\sum_{i \in [k]} \sum_{p,q \in P_i} \|p - q\|_2^2 \geq (1 - o(1)) \cdot \left(\alpha \cdot \sqrt{z - y} + (1 - \alpha) \cdot \sqrt{z - y + 1}\right)^2 \cdot \rho n^2 / k,$$

for some constant $\rho > 0$*.*

Putting together the above theorem with Theorem 11 (i.e., NP-hardness of $(7/8+\varepsilon, 3, 1, \delta)$-Balanced Johnson Coverage* problem for all $\varepsilon, \delta > 0$), we obtain the NP-hardness of approximating ℓ_2^2 min-sum k-clustering. The above theorem also immediately yields the hardness of approximating ℓ_2^2 min-sum k-clustering under $\mathsf{Balanced-JCH}^*$ (i.e., conditional NP-hardness of $(1 - 1/e + \varepsilon, z, z - 1, 0)$-Balanced Johnson Coverage* problem for all $\varepsilon > 0$ and some $z = z(\varepsilon) \in \mathbb{N}$). This completes the proof of Theorem 3.

3 PTAS based on D^2 Sampling

For a set $A \subset \mathbb{R}^d$, let $\mu(A) := \frac{1}{|A|}\sum_{p\in A} p$ denote its mean. Let $\mathcal{C} = \{C_1, \ldots C_k\}$ be an optimal k-MinSum clustering of a point set A. We use $\mu_i = \mu(C_i)$ to denote the mean of C_i and we use $\Delta_i = \frac{\sum_{p\in C_i}\|p-\mu_i\|^2}{|C_i|}$ to denote the average mean squared distance of C_i to μ_i. We further use C_i^β to denote the subset of C_i with $\|p-\mu_i\|^2 \le \beta \cdot \Delta_i$. Finally, let OPT denote the cost of an optimal solution. So, $\mathsf{OPT} = \sum_{i=1}^k |C_i|^2 \cdot \Delta_i$.

▶ **Definition 13.** *We say that m is an ε-approximate mean of C_i if $\|m-\mu_i\|^2 \le \varepsilon \cdot \Delta_i$. We say that a set $S \subset A$ is an (ε,β)-mean seeding set for $C_i \in \mathcal{C}$, if there exists a subset $S' \cup \{s\} \subset S$ with $\|s-\mu_i\|^2 \le \beta \cdot \Delta_i$ and a weight assignment $w : S' \to \mathbb{R}_{\ge 0}$ such that $\left\|\frac{1}{\sum_{p\in S'} w(p)}\sum_{p\in S'} w(p)\cdot p - \mu_i\right\|^2 \le \varepsilon \cdot \Delta_i$.*

We will use the following well-known identities for Euclidean means.

▶ **Lemma 14** ([36]). *Let $A \subset \mathbb{R}^d$ be a set of points. Then for any $c \in \mathbb{R}^d$:*
- $\sum_{p\in A}\|p-c\|^2 = \sum_{p\in A}\|p-\mu(A)\|^2 + |A|\cdot\|\mu(A)-c\|^2$.
- $\sum_{p,q\in A}\|p-q\|^2 = 2\cdot|A|\cdot\sum_{p\in A}\|p-\mu(A)\|^2$.

We also show that we only have to consider seeding sets with $\beta \in \Theta(\varepsilon^{-2})$.

▶ **Lemma 15.** *For any cluster C_i, $\varepsilon \in (0,1)$ and $\beta \ge 12\varepsilon^{-2}$, we have that $\mu_i(C_i^\beta) = \frac{1}{|C_i^\beta|}\sum_{p\in C_i^\beta} p$ is a ε-approximate mean of C_i.*

Finally, we also show how to efficiently extract a mean from a mean seeding set, while being oblivious to Δ_i.

▶ **Lemma 16.** *Let S be an $(\varepsilon/4,\beta)$-mean seeding set of a cluster C_j with mean μ_j. Then we can compute $\left(\frac{10\beta\cdot|S|}{\varepsilon}+1\right)^{|S|}$ choices of weights in time linear in the size of choices such that at least one of the computed choices satisfies $\left\|\frac{1}{\sum_{p\in S} w(p)}\sum_{p\in S} w(p)\cdot p - \mu_j\right\|^2 \le \varepsilon \cdot \Delta_j$.*

Computing a Mean-Seeding Set via Uniform Sampling

▶ **Lemma 17.** *Let $\varepsilon \in (0,1)$ and $\beta > 48\varepsilon^2$. With probability at least $1-\delta$, a set of $32k\varepsilon^{-1}\log\delta^{-1}$ points S sampled uniformly at random with replacement from A contains is a (ε,β)-mean seeding set of any C_i with $|C_i| \ge \frac{n}{k}$.*

D^2 Subsampling

We now define an algorithm for sampling points that induce means from the target clusters. The high level idea is as follows. We construct a rooted tree in which every node is labeled by a set of candidate cluster means. For a parent and child pair of nodes, the parent's set is a subset of the child's set. The construction is iterative. Given an interior node, we construct its children by adding a candidate mean to the parent's set. The candidantes are generated using points sampled at random from a distribution that will be defined later. The goal is to have, eventually, an ε-approximate mean for every optimal cluster. This will be achieved with high probability at one of the leaves of the tree. The root of the tree is labeled with the empty set, and its children are constructed via uniform sampling. Subsequently, we refine the sampling distribution to account for various costs and densities of the clusters.

We now go into more detail for the various sampling stages of the algorithm.

Preprocessing: We ensure that all points are not too far from each other.

Initialization: We initialize the set of means via uniform sampling. Due to Lemma 17, we can enumerate over potential sets of ε-approximate means for all clusters of size $\frac{n}{k}$. Each candidate mean defines a child of the root.

Sampling Stage: Consider a node of the tree labeled with a non-empty set of candidate means M. We put $\Gamma_i = 2^{-i} \cdot \sum_{q \in A} \min_{m \in M} \|q - m\|^2$ for $i \in \{0, 1, \ldots, 13 \log(nk/\varepsilon)\}$, where η is an absolute constant to be defined later. Let $A_{i,M} = \{q \in A \colon \min_{m \in M} \|q - m\|^2 \leq \Gamma_i\}$. (Note that $A_{0,M}$ includes all the points.) Let $\mathbb{P}_i$ denote the probability distribution on $A_{i,M}$ induced by setting, for each $p \in A_{i,M}$, $\mathbb{P}_i[p] = \frac{\min_{m \in M} \|p-m\|^2}{\sum_{p \in A_{i,M}} \min_{m \in M} \|p-m\|^2}$ We'll use $\mathbb{P}$ to denote $\mathbb{P}_0$. For each i, we sample a sufficient (polynomial in k and ε, but independent of n) number of points independently from the distribution $\mathbb{P}_i$. Let S denote the set of sampled points.

Mean Extraction Stage: We enumerate over combinations of points in $M \cup S$, using some non-uniform weighing to fix a mean to add to M, see Lemma 16. Each choice of mean is added to M to create a child of the node labeled M.

Throughout this section we will use the following definition. Given a set of centers M, we say that a cluster C_i is ε-covered by M if $|C_i|^2 \cdot \min_{m \in M} \|\mu_i - m\|^2 \leq \frac{\varepsilon}{2} \cdot \left(\frac{1}{k} \cdot OPT + |C_i|^2 \Delta_i\right)$. Our goal will be to prove the following lemma.

▶ **Lemma 18.** *Let $\mathcal{C} = \{C_1, \ldots C_k\}$ be the clusters of an optimal Min-Sum k-clustering and let η be an absolute constant. For every $\delta, \epsilon > 0$, there is a randomized algorithm that outputs a collection of at most $n^{o(1)} \cdot 2^{\eta \cdot k^2 \cdot \varepsilon^{-12} \log^2(k/(\varepsilon\delta))}$ sets of at most k centers M, such that with probability $1 - \delta$ at least one of them that ε-covers every $C_i \in \mathcal{C}$. The algorithm runs in time $n^{1+o(1)} \cdot d \cdot 2^{\eta \cdot k^2 \cdot \varepsilon^{-12} \log^2(k/(\varepsilon\delta))}$.*

Note that if all clusters of $\mathcal{C}$ are ε-covered, then there exists an assignment of points to centers, such that Min-Sum clustering cost of the resulting clustering is at most $(1 + \varepsilon) \cdot \mathsf{OPT}$.

Preprocessing

The first lemma allows us to assume that all points are in some sense close to each other.

▶ **Lemma 19.** *Suppose $n > 20$. Given an set of n points $A \subset \mathbb{R}^d$, we can partition a point set into subsets $A_1, \ldots A_k$, such that $\|p - q\|^2 \leq n^{10} \cdot \mathsf{OPT}$ for any two points $p, q \in A_i$ and such that any cluster C_j is fully contained in one of the A_i. The partitioning takes time $\tilde{O}(nd + k^2)$.*

Computing a Mean-Seeding Set via D^2 Sampling

We now consider a slight modification of Lemma 17 to account for sampling points from a cluster non-uniformly. We introduce the notion of a distorted core as follows. Given a cluster C_j, a set of centers M, and parameters α, β, we say that a subset of $C_j^\beta \cup M$ is a (C_j, β, α, M)-distorted core (denoted $core(C_j, \beta, \alpha, M)$) iff it is the image of a mapping $\pi_{\alpha,M} : C_j^\beta \to C_j^\beta \cup M$ such that for any point $p \in C_j^\beta$, we have

$$\pi_{\alpha,M}(p) = \begin{cases} p & \text{if } \min_{m \in M} \|p - m\|^2 \geq \alpha \cdot \Delta_j \\ \underset{m \in M}{\operatorname{argmin}} \|p - m\|^2 & \text{if } \min_{m \in M} \|p - m\|^2 < \alpha \cdot \Delta_j \end{cases}.$$

We use $D(C_j, \beta, \alpha, M)$ to denote the set of points in C_j^β such that $\min_{m \in M} \|p - m\|^2 < \alpha \cdot \Delta_j$.

The following lemmas relate the goodness of a mean computed on an α-distorted core to the mean on the entire set of points when sampling points proportionate to squared distances. We start by proving an analogue of Lemma 15.

▶ **Lemma 20.** *Let $\alpha \leq \frac{\varepsilon}{4}$ and let $\beta \geq \frac{144}{\varepsilon^2}$. Given a set of centers M and a cluster C_j, let $\hat{\mu_j} = \frac{1}{|C_j^\beta|}\sum_{p \in C_j^\beta} \pi_{\alpha,M}(p)$. Then, $\|\hat{\mu_j} - \mu_j\|^2 \leq \varepsilon \cdot \Delta_j$.*

We now characterize when M either covers a cluster C_j, or when M is a suitable seeding set for C_j. The following lemma says that if M is not a seeding set of C_j, then there exist many points in the core C_j^β of C_j that are far from M.

▶ **Lemma 21.** *Given $\alpha \leq \frac{\varepsilon}{16}$, $\beta \geq \frac{2400}{\varepsilon^2}$, and $\gamma \leq \sqrt{\frac{\varepsilon}{16(\beta+\alpha)}}$, and a set of centers M, let C_j be a cluster for which $|D(C_j, \beta, \alpha, M)| \geq (1-\gamma) \cdot |C_j^\beta|$. Then M is an (ε, β)-mean seeding set of C_j.*

Next, we show that the marginal probability of picking a point from an uncovered cluster C_j cannot be significantly smaller than the marginal probability of picking a point from the union of covered clusters with larger cardinality than C_j.

▶ **Lemma 22.** *Let M be a set of centers, and let $\mathcal{C}$ denote a set of clusters that are ε-covered by M. Let $\mathcal{H}$ denote the set of points in all the clusters in $\mathcal{C}$. Let $\beta > \frac{2400}{\varepsilon^2}$. Consider a cluster $C_j \notin \mathcal{C}$. Let i be the largest index such that $C_i \in \mathcal{C}$. Suppose that M is not an (ε, β)-mean seeding set of C_j, and that $i < j$. Then $\mathbb{P}[p \in C_j^\beta \mid p \in \mathcal{H} \cup C_j] \geq \frac{\varepsilon^4 \cdot \beta^{-3/2}}{1088k}$.*

We now consider a cluster C_j that is small compared to the union of the clusters C_j' with $j' > j$. In this case, we show that one of the distance-proportional distributions that we use guarantees that the probability of sampling points from the core of C_j is large.

▶ **Lemma 23.** *Let M be a set of centers. Let $\beta > \frac{2400}{\varepsilon^2}$. Let j be the smallest index such that C_j is not ε-covered by M. If M is not an (ε, β)-mean seeding set for C_j, then there exists $i \in \{0, 1, \ldots, \eta \log(nk/\varepsilon)\}$ such that $C_j^\beta \in A_{i,M}$ and $\mathbb{P}_i[p \in C_j^\beta] \geq \frac{1}{4352 \cdot k} \cdot \left(\frac{\varepsilon}{\beta^{5/8}}\right)^4$*

Finally, we show how to account for the sampling bias when estimating the means.

▶ **Lemma 24.** *Let M be a set of centers. Let j be the smallest index such that C_j is not ε-covered by M. Suppose that M is not an $(\varepsilon/4, \beta)$-mean seeding set for C_j. Consider a set of points S' sampled iid from $\mathbb{P}_i$, and let $S = S' \cap C_j^\beta$. If $\beta \geq 2400\varepsilon^{-2}$ and $S > 17825792 \cdot k \left(\frac{\beta^{7/12}}{\varepsilon}\right)^6 \log(2/\delta)$, then with probability at least $1-\delta$, we have that $S' \cup M$ is an $(\varepsilon/4, \beta)$-mean seeding set of C_j.*

The proof of Lemma 18 now argues that, given a set of points M, that we can find a seeding set for the largest uncovered cluster via D^2 sampling with respect to a suitable distribution $\mathbb{P}_i$. The existence of such a distribution, as well as the number of samples is given via Lemma 24 and extracting a suitable mean can be done via Lemma 16. The overall number of candidate solutions is now exponential in the number of points sampled over the course of the procedure, which is bounded by $\text{poly}(k, \varepsilon^{-1}, \log \delta^{-1})$.

Obtaining the Parameterized PTAS

We complete this section by explaining how to funnel the mean-seeding procedure into a PTAS. This yields Theorem 4. For every candidate solution given by Lemma 18, we bucket points by squared distance to the respective centers. This number of buckets depends on a guess of the optimum, which is inexpensive to obtain, as well as a discretization over possible different distances to centers, of which we show there cannot exist too many. Since the number of buckets are very small, we can efficiently find an assignment. Then we show that we can extract a clustering from the bucketed assignment in linear time.

4 Learning-Augmented ℓ_2^2 Min-Sum k-Clustering

In this section, we describe and analyze our learning-augmented algorithm, corresponding to Theorem 5. We first formally define the precision and recall guarantees of a label predictor.

▶ **Definition 25** (Label predictor). *Suppose that there is an oracle that produces a label $i \in [k]$ for each $x \in X$, so that the labeling partitions $X = P_1 \dot\cup \ldots \dot\cup P_k$ into k clusters $P_1, \ldots, P_k$, where all points in P_i have the same label $i \in [k]$. We say the oracle is a* label predictor with error rate α *if there exists some fixed optimal min-sum clustering $P_1^*, \ldots, P_k^*$ such that for all $i \in [k]$, $|P_i \cap P_i^*| \ge (1-\alpha)\max(|P_i|, |P_i^*|)$. We say that $P^* = \{P_1^*, \ldots, P_k^*\}$ is the clustering consistent with the label oracle.*

We also recall the following guarantees of previous work on learning-augmented k-means clustering for a label predictor with error rate $\alpha \in \left[0, \frac{1}{2}\right)$.

▶ **Theorem 26** ([61]). *Given a label predictor with error rate $\alpha < \frac{1}{2}$ consistent with some clustering $P^* = \{P_1^*, \ldots, P_k^*\}$ with centers $\{c_1^*, \ldots, c_k^*\}$, there exists a polynomial-time algorithm* LearnedCenters *that outputs a set of centers $\{c_1, \ldots, c_k\}$, so that for each $i \in [k]$, $\sum_{x \in P_i^*} \|x - c_i\|_2^2 \le (1 + \gamma_\alpha \alpha) \sum_{x \in P_i^*} \|x - c_i^*\|_2^2$, where $\gamma_\alpha = 7.7$ for $\alpha \in \left[0, \frac{1}{7}\right)$ or $\gamma_\alpha = \frac{5\alpha - 2\alpha^2}{(1-2\alpha)(1-\alpha)}$ for $\alpha \in \left[0, \frac{1}{2}\right)$.*

Unfortunately, although the centers $\{c_1, \ldots, c_k\}$ returned by LearnedCenters are good centers for the clustering induced by a near-optimal ℓ_2^2 min-cost k-clustering, it is not clear what the resulting assignment should be. In fact, we emphasize that unlike k-means clustering, the optimal ℓ_2^2 min-cost k-clustering may not assign each point to its closest center.

Constrained min-cost flow. To that end, we now create a constrained min-cost flow problem as follows. We first create a source node s and a sink node t and require that $n = |X|$ flow must be pushed from s to t. We create a node u_x for each point $x \in X$ and create a directed edge from s to each node u_x with capacity 1 and cost 0. There are no more outgoing edges from s or incoming edges to each u_x. This ensures that to achieve n flow from s to t, a unit of flow must be pushed across each node u_x.

Algorithm 1 Learning-augmented min-sum k-clustering.

```
Input: Dataset $X$ with partition $P_1, \ldots, P_k$ induced by label predictor with error rate $\alpha$
Output: Labels for all points consistent with a $(1 + \mathcal{O}(\alpha))$-optimal min-sum $k$-clustering
Let $c_1, \ldots, c_k$ be the output centers of LearnedCenters on $P_1, \ldots, P_k$
Create a min-cost flow problem $\mathcal{F}$ with required flow $n$
Solve the flow problem $\mathcal{F}$
For each $x \in X$, let the flow from $u_x$ be sent to the node $v_{\ell_x}$, so that $\ell_x \in [k]$
Label $x$ with $\ell_x$
```

We then adjust an integrality theorem to handle capacitated edges, thereby showing that the resulting solution for the min-cost flow problem is integral, and show that since the constraint matrix is totally unimodular, i.e., all submatrices have determinant -1, 0, or 1, then a valid clustering can be recovered by using the output of a linear program solver. Thus, we have the following guarantees for our learning-augmented algorithm.

▶ **Theorem 27.** *There exists a polynomial-time algorithm that uses a label predictor with error rate α and outputs a $\frac{1+\gamma_\alpha \alpha}{(1-\alpha)^2}$-approximation to min-sum k-clustering, where γ_α is the fixed constant from Theorem 26.*

References

1 Anders Aamand, Justin Y. Chen, and Piotr Indyk. (optimal) online bipartite matching with degree information. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2022.

2 Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Mach. Learn.*, 75(2):245–248, 2009. doi:10.1007/S10994-009-5103-0.

3 Enver Aman, Karthik C. S., and Sharath Punna. On connections between k-coloring and Euclidean k-means. In *32nd Annual European Symposium on Algorithms, ESA 2024*, 2024. To appear.

4 Keerti Anand, Rong Ge, Amit Kumar, and Debmalya Panigrahi. Online algorithms with multiple predictions. In *International Conference on Machine Learning, ICML*, pages 582–598, 2022. URL: https://proceedings.mlr.press/v162/anand22a.html.

5 Antonios Antoniadis, Christian Coester, Marek Eliás, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. *ACM Trans. Algorithms*, 19(2):19:1–19:34, 2023. doi:10.1145/3582689.

6 David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In Nikhil Bansal, Kirk Pruhs, and Clifford Stein, editors, *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 1027–1035. SIAM, 2007. URL: http://dl.acm.org/citation.cfm?id=1283383.1283494.

7 Per Austrin, Subhash Khot, and Muli Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. *Theory Comput.*, 7(1):27–43, 2011. doi:10.4086/TOC.2011.V007A003.

8 Yossi Azar, Debmalya Panigrahi, and Noam Touitou. Online graph algorithms with predictions. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 35–66, 2022. doi:10.1137/1.9781611977073.3.

9 Étienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

10 Sandip Banerjee, Rafail Ostrovsky, and Yuval Rabani. Min-sum clustering (with outliers). In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 16:1–16:16, 2021. doi:10.4230/LIPICS.APPROX/RANDOM.2021.16.

11 Yair Bartal, Moses Charikar, and Danny Raz. Approximating min-sum k-clustering in metric spaces. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing*, pages 11–20, 2001. doi:10.1145/380752.380754.

12 Babak Behsaz, Zachary Friggstad, Mohammad R. Salavatipour, and Rohit Sivakumar. Approximation algorithms for min-sum k-clustering and balanced k-median. *Algorithmica*, 81(3):1006–1030, 2019. doi:10.1007/S00453-018-0454-1.

13 Justin Y. Chen, Talya Eden, Piotr Indyk, Honghao Lin, Shyam Narayanan, Ronitt Rubinfeld, Sandeep Silwal, Tal Wagner, David P. Woodruff, and Michael Zhang. Triangle and four cycle counting with predictions in graph streams. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

14 Justin Y. Chen, Piotr Indyk, and Tal Wagner. Streaming algorithms for support-aware histograms. In *International Conference on Machine Learning, ICML*, pages 3184–3203, 2022. URL: https://proceedings.mlr.press/v162/chen22g.html.

15 Justin Y. Chen, Sandeep Silwal, Ali Vakilian, and Fred Zhang. Faster fundamental graph algorithms via learned predictions. In *International Conference on Machine Learning, ICML*, pages 3583–3602, 2022. URL: https://proceedings.mlr.press/v162/chen22v.html.

16 Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *J. ACM*, 68(1):3:1–3:39, 2021. doi:10.1145/3424305.

17 Vincent Cohen-Addad and Karthik C. S. Inapproximability of clustering in lp metrics. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 519–539, 2019. doi:10.1109/FOCS.2019.00040.

18 Vincent Cohen-Addad, Karthik C. S., and Euiwoong Lee. On approximability of clustering problems without candidate centers. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2635–2648, 2021. doi:10.1137/1.9781611976465.156.

19 Vincent Cohen-Addad, Karthik C. S., and Euiwoong Lee. Johnson coverage hypothesis: Inapproximability of k-means and k-median in ℓ_p-metrics. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1493–1530, 2022. doi:10.1137/1.9781611977073.63.

20 Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, Chris Schwiegelshohn, and Omar Ali Sheikh-Omar. Improved coresets for euclidean k-means. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2022.

21 Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. A new coreset framework for clustering. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 169–182, 2021. doi:10.1145/3406325.3451022.

22 Artur Czumaj and Christian Sohler. Sublinear-time approximation algorithms for clustering via random sampling. *Random Struct. Algorithms*, 30(1-2):226–256, 2007. doi:10.1002/RSA.20157.

23 Sami Davies, Benjamin Moseley, Sergei Vassilvitskii, and Yuyan Wang. Predictive flows for faster ford-fulkerson. In *International Conference on Machine Learning, ICML*, volume 202, pages 7231–7248, 2023. URL: https://proceedings.mlr.press/v202/davies23b.html.

24 Wenceslas Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation schemes for clustering problems. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 50–58, 2003. doi:10.1145/780542.780550.

25 Wenceslas Fernandez de la Vega and Claire Kenyon. A randomized approximation scheme for metric MAX-CUT. *J. Comput. Syst. Sci.*, 63(4):531–541, 2001. doi:10.1006/JCSS.2001.1772.

26 Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Faster matchings via learned duals. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 10393–10406, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/5616060fb8ae85d93f334e7267307664-Abstract.html.

27 Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. *SIAM J. Comput.*, 34(5):1129–1146, 2005. doi:10.1137/S0097539704443057.

28 Jon C. Ergun, Zhili Feng, Sandeep Silwal, David P. Woodruff, and Samson Zhou. Learning-augmented k-means clustering. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

29 Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.

30 Sreenivas Gollapudi and Debmalya Panigrahi. Online algorithms for rent-or-buy with expert advice. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 2319–2327, 2019. URL: http://proceedings.mlr.press/v97/gollapudi19a.html.

31 Elena Grigorescu, Young-San Lin, Sandeep Silwal, Maoyuan Song, and Samson Zhou. Learning-augmented algorithms for online linear and semidefinite programming. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2022.

32 Venkatesan Guruswami and Piotr Indyk. Embeddings and non-approximability of geometric problems. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 537–538, 2003. URL: `http://dl.acm.org/citation.cfm?id=644108.644198`.

33 Nili Guttmann-Beck and Refael Hassin. Approximation algorithms for min-sum p-clustering. *Discret. Appl. Math.*, 89(1-3):125–142, 1998. `doi:10.1016/S0166-218X(98)00100-0`.

34 Chen-Yu Hsu, Piotr Indyk, Dina Katabi, and Ali Vakilian. Learning-based frequency estimation algorithms. In *7th International Conference on Learning Representations, ICLR*, 2019.

35 Sungjin Im, Ravi Kumar, Mahshid Montazer Qaem, and Manish Purohit. Online knapsack with frequency predictions. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 2733–2743, 2021. URL: `https://proceedings.neurips.cc/paper/2021/hash/161c5c5ad51fcc884157890511b3c8b0-Abstract.html`.

36 Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering (extended abstract). In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, pages 332–339, 1994. `doi:10.1145/177424.178042`.

37 Piotr Indyk. A sublinear time approximation scheme for clustering in metric spaces. In *40th Annual Symposium on Foundations of Computer Science, FOCS*, pages 154–159, 1999. `doi:10.1109/SFFCS.1999.814587`.

38 Piotr Indyk, Ali Vakilian, and Yang Yuan. Learning-based low-rank approximations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS*, pages 7400–7410, 2019. URL: `https://proceedings.neurips.cc/paper/2019/hash/1625abb8e458a79765c62009235e9d5b-Abstract.html`.

39 Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999. `doi:10.1145/331499.331504`.

40 Arun Jambulapati, Yang P. Liu, and Aaron Sidford. Chaining, group leverage score overestimates, and fast spectral hypergraph sparsification. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 196–206. ACM, 2023. `doi:10.1145/3564246.3585136`.

41 Shaofeng H.-C. Jiang, Erzhi Liu, You Lyu, Zhihao Gavin Tang, and Yubo Zhang. Online facility location with predictions. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

42 Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. A faster algorithm for solving general lps. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 823–832, 2021. `doi:10.1145/3406325.3451058`.

43 Tanqiu Jiang, Yi Li, Honghao Lin, Yisong Ruan, and David P. Woodruff. Learning-augmented data stream algorithms. In *8th International Conference on Learning Representations, ICLR*, 2020.

44 Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Comb.*, 4(4):373–396, 1984. `doi:10.1007/BF02579150`.

45 Misha Khodak, Maria-Florina Balcan, Ameet Talwalkar, and Sergei Vassilvitskii. Learning predictions for algorithms with predictions. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2022.

46 Subhash Khot. Hardness results for coloring 3 -colorable 3 -uniform hypergraphs. In *43rd Symposium on Foundations of Computer Science (FOCS), Proceedings*, pages 23–32, 2002. `doi:10.1109/SFCS.2002.1181879`.

47 Jon M. Kleinberg. An impossibility theorem for clustering. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS*, pages 446–453, 2002. URL: `https://proceedings.neurips.cc/paper/2002/hash/43e4e6a6f341e00671e123714de019a8-Abstract.html`.

48 Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference*, pages 489–504, 2018. doi:10.1145/3183713.3196909.

49 Silvio Lattanzi, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Online scheduling via learned weights. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1859–1877, 2020. doi:10.1137/1.9781611975994.114.

50 Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. *Inf. Process. Lett.*, 120:40–43, 2017. doi:10.1016/J.IPL.2016.11.009.

51 James R. Lee. Spectral hypergraph sparsification via chaining. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 207–218. ACM, 2023. doi:10.1145/3564246.3585165.

52 Yin Tat Lee and Aaron Sidford. Efficient inverse maintenance and faster algorithms for linear programming. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS*, pages 230–249, 2015. doi:10.1109/FOCS.2015.23.

53 Yin Tat Lee, Zhao Song, and Qiuyi Zhang. Solving empirical risk minimization in the current matrix multiplication time. In *Conference on Learning Theory, COLT*, pages 2140–2157, 2019. URL: http://proceedings.mlr.press/v99/lee19a.html.

54 Yi Li, Honghao Lin, Simin Liu, Ali Vakilian, and David P. Woodruff. Learning the positions in countsketch. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

55 Honghao Lin, Tian Luo, and David P. Woodruff. Learning augmented binary search trees. In *International Conference on Machine Learning, ICML*, pages 13431–13440, 2022. URL: https://proceedings.mlr.press/v162/lin22f.html.

56 Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *J. ACM*, 68(4):24:1–24:25, 2021. doi:10.1145/3447579.

57 Meena Mahajan, Prajakta Nimbhorkar, and Kasturi R. Varadarajan. The planar k-means problem is np-hard. *Theor. Comput. Sci.*, 442:13–21, 2012. doi:10.1016/J.TCS.2010.05.034.

58 Jirí Matousek. On approximate geometric k-clustering. *Discret. Comput. Geom.*, 24(1):61–84, 2000. doi:10.1007/S004540010019.

59 Michael Mitzenmacher. A model for learned bloom filters and optimizing by sandwiching. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 462–471, 2018. URL: https://proceedings.neurips.cc/paper/2018/hash/0f49c89d1e7298bb9930789c8ed59d48-Abstract.html.

60 Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, pages 646–662. Cambridge University Press, 2020. doi:10.1017/9781108637435.037.

61 Thy Dinh Nguyen, Anamay Chaturvedi, and Huy L. Nguyen. Improved learning-augmented algorithms for k-means and k-medians clustering. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

62 Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ML predictions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS*, pages 9684–9693, 2018. URL: https://proceedings.neurips.cc/paper/2018/hash/73a427badebe0e32caa2e1fc7530b7f3-Abstract.html.

63 Leonard J. Schulman. Clustering for edge-cost minimization (extended abstract). In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 547–555, 2000. doi:10.1145/335305.335373.

64 Yongho Shin, Changyeol Lee, Gukryeol Lee, and Hyung-Chan An. Improved learning-augmented algorithms for the multi-option ski rental problem via best-possible competitive analysis. In *International Conference on Machine Learning, ICML*, pages 31539–31561, 2023. URL: https://proceedings.mlr.press/v202/shin23c.html.

65 Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR, Conference Track Proceedings*, 2014.

66 Pravin M. Vaidya. Speeding-up linear programming using fast matrix multiplication. In *30th Annual Symposium on Foundations of Computer Science*, pages 332–337, 1989.

67 Pravin M. Vaidya. An algorithm for linear programming which requires o(((m+n)n^2 + (m+n)$^{1.5}$n)l) arithmetic operations. *Math. Program.*, 47:175–201, 1990. doi:10.1007/BF01580859.

68 Shufan Wang, Jian Li, and Shiqiang Wang. Online algorithms for multi-shop ski rental with machine learned advice. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

69 Alexander Wei and Fred Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

70 Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005. doi:10.1109/TNN.2005.845141.