

CLUSTERING UNDER PERTURBATION RESILIENCE*

MARIA FLORINA BALCAN[†] AND YINGYU LIANG[‡]

Abstract. Motivated by the fact that distances between data points in many real-world clustering instances are often based on heuristic measures, Bilu and Linial [12] proposed analyzing objective based clustering problems under the assumption that the optimum clustering to the objective is preserved under small multiplicative perturbations to distances between points. The hope is that by exploiting the structure in such instances, one can overcome worst case hardness results.

In this paper, we provide several strong results within this framework. For center-based objectives, we present an algorithm that can optimally cluster instances resilient to perturbations of factor $(1 + \sqrt{2})$, solving an open problem of Awasthi et al. [3]. For k -median, a center-based objective of special interest, we additionally give algorithms for a more relaxed assumption in which we allow the optimal solution to change in a small ϵ fraction of the points after perturbation. We give the first bounds known for k -median under this more realistic and more general assumption. We also provide positive results for min-sum clustering which is a generally much harder objective than center-based objectives. Our algorithms are based on new linkage criteria that may be of independent interest.

Additionally, we give sublinear-time algorithms, showing algorithms that can return an implicit clustering from only access to a small random sample.

Key words. clustering, perturbation resilience, k -median clustering, min-sum clustering

AMS subject classifications. 68Q25, 68Q32, 68T05, 68W25, 68W40

1. Introduction. Problems of clustering data from pairwise distance information are ubiquitous in science. A common approach for solving such problems is to view the data points as nodes in a weighted graph (with the weights based on the given pairwise information), and then to design algorithms to optimize various objective functions such as k -median or min-sum. For example, in the k -median clustering problem the goal is to partition the data into k clusters C_i , giving each a center c_i , in order to minimize the sum of the distances of all data points to the centers of their cluster. In the min-sum clustering approach the goal is to find k clusters C_i that minimize the sum of all intra-cluster pairwise distances. Yet unfortunately, for most natural clustering objectives, finding the optimal solution to the objective function is NP-hard. As a consequence, there has been substantial work on approximation algorithms [17, 13, 8, 14, 1] with both upper and lower bounds on the approximability of these objective functions on worst case instances.

Recently, Bilu and Linial [12] suggested an exciting, alternative approach aimed at understanding the complexity of clustering instances which arise in practice. Motivated by the fact that distances between data points in clustering instances are often based on a heuristic measure, they argue that interesting instances should be resilient to small perturbations in these distances. In particular, if small perturbations can cause the optimum clustering for a given objective to change drastically, then that probably is not a meaningful objective to be optimizing. Bilu and Linial [12] specifically define an instance to be α -perturbation resilient¹ for an objective Φ if perturbing pairwise distances by multiplicative factors in the range $[1, \alpha]$ does not change the optimum clustering under Φ . They consider in detail the case of Max-Cut clustering

*A preliminary version of this article appeared under the title *Clustering under Perturbation Resilience* in the Proceedings of the Thirty-Ninth International Colloquium on Automata, Languages and Programming, 2012.

[†]Carnegie Mellon University, Pittsburgh, PA 15213 (ninamf@cs.cmu.edu).

[‡]Georgia Institute of Technology, Atlanta, Georgia 30332 (yliang39@gatech.edu).

¹Bilu and Linial [12] refer to such instances as perturbation stable instances.

and give an efficient algorithm to recover the optimum when the instance is resilient to perturbations on the order of $\alpha > \min\{n/2, \sqrt{n\Delta}\}$ where Δ is the maximal degree of the graph. They also give an efficient algorithm for unweighted Max-Cut instances that are resilient to perturbations on the order of $\alpha \geq 4n/\delta$ where δ is the minimal degree of the graph.

Two important questions raised by the work of Bilu and Linial [12] are: (1) the degree of resilience needed for their algorithm to succeed is quite high: can one develop algorithms for important clustering objectives that require much less resilience? (2) the resilience definition requires the optimum solution to remain *exactly* the same after perturbation: can one succeed under weaker conditions? In the context of *center-based* clustering objectives such as k -median and k -center, Awasthi et al. [1] partially address the first of these questions and show that an algorithm based on the single-linkage heuristic can be used find the optimal clustering for α -perturbation-resilient instances for $\alpha = 3$. They also conjecture it to be NP-hard to beat 3 and prove beating 3 is NP-hard for a related notion.

In this work, we address both questions raised by [12] and additionally improve over [3]. First, for the center-based objectives we design a polynomial time algorithm for finding the optimum solution for instances resilient to perturbations of value $\alpha = 1 + \sqrt{2}$, thus beating the previously best known factor of 3 of Awasthi et al [3]. Second, for k -median (which is a specific center-based objective), we consider a weaker, relaxed, and more realistic notion of perturbation-resilience where we allow the optimal clustering of the perturbed instance to differ from the optimal of the original in a small ϵ fraction of the points. Compared to the original perturbation resilience assumption, this is arguably a more natural though also more difficult condition to deal with. We give positive results for this case as well, showing for somewhat larger values of α that we can still achieve a near-optimal clustering on the given instance (see Section 1.1 below for precise results). We additionally give positive results for min-sum clustering which is a generally much harder objective than center-based objectives. For example, the best known guarantee for min-sum clustering on worst-case instances is an $O(v^{-1} \log^{1+v} n)$ -approximation algorithm that runs in time $n^{O(1/v)}$ for any $v > 0$ due to Bartal et al. [8]; by contrast, the best guarantee known for k -median is factor $1 + \sqrt{3} + \epsilon$ [19].

Our results are achieved by carefully deriving structural properties of perturbation resilience. At a high level, all the algorithms we introduce work by first running appropriate linkage procedures to produce a hierarchical clustering, and then running dynamic programming to retrieve the best k -clustering present in the tree. To ensure that (under perturbation resilient instances) the hierarchy output in the first step has a pruning of low cost, we derive new linkage procedures (closure linkage and robust average linkage) which are of independent interest. While the overall analysis is quite involved, the clustering algorithms we devise are simple and robust. This simplicity and robustness allow us to show how our algorithms can be made sublinear-time by returning an implicit clustering from only a small random sample of the input.

From a learning theory perspective, the resilience parameter, α , can also be seen as an analog to a margin for clustering. In supervised learning, the margin of a data point is the distance, after scaling, between the data point and the decision boundary of its classifier, and many algorithms have stronger guarantees when the smallest margin over the entire data set is sufficiently large [26, 27]. The α parameter, similarly controls the magnitude of the perturbation the data can withstand before being clustered differently, which is, in essence, the data's distance to the decision

boundary for the given clustering objective. Hence, perturbation resilience is also a natural and interesting assumption to study from a learning theory perspective.

Our Results. In this paper, we greatly advance the line of work of [12] by solving a number of important problems of clustering perturbation-resilient instances under metric center-based and min-sum objectives.

In Section 3 we improve on the bounds of [3] for α -perturbation resilient instances for center-based objectives, giving an algorithm that efficiently² finds the optimum clustering for $\alpha = 1 + \sqrt{2}$. Most of the frequently used center-based objectives, such as k -median, are NP-hard to even approximate, yet we can recover the exact solution for perturbation resilient instances. Our algorithm is based on a new linkage procedure using a new notion of distance (closure distance) between sets that may be of independent interest.

In Section 4 we consider the more challenging and more general notion of (α, ϵ) -perturbation resilience for k -median, where we allow the optimal solution after perturbation to be ϵ -close to the original. We provide an efficient algorithm which for $\alpha > 2 + \sqrt{3}$ produces $(1 + O(\epsilon/\rho))$ -approximation to the optimum, where ρ is the fraction of the points in the smallest cluster. The key structural property we derive and exploit is that, except for ϵn bad points, most points are α times closer to their own center than to any other center. To eliminate the noise introduced by the bad points, we carefully partition the points into a list of sufficiently large blobs, each of which contains only good points from one optimal cluster. This then allows us to construct a tree on the blobs with a low-cost pruning that is a good approximation to the optimum.

In Section 5 we provide the first efficient algorithm for optimally clustering α -perturbation resilient min-sum instances. Our algorithm is based on an appropriate modification of average linkage that exploits the structure of min-sum perturbation resilient instances.

In Section 6, we show that for (α, ϵ) -perturbation resilient min-sum instances with α in the order of the ratio between the sizes of the largest and smallest clusters and $\epsilon = \tilde{O}(\rho)$, there exists a polynomial time algorithm that outputs a clustering that is both a $(1 + \tilde{O}(\epsilon/\rho))$ -approximation and $\tilde{O}(\epsilon)$ -close to the optimal clustering. The key structural property is that except for $\tilde{O}(\epsilon n)$ bad points, most points are $O(\alpha)$ times closer to their own optimal cluster than to any other optimal cluster. Similar to the case of k -median, we can partition the points into a list of sufficiently large blobs, each of which contains only good points from one optimal cluster. However, the properties of the good points are significantly weaker than those in the k -median case, and thus the linkage there does not guarantee a tree with a low-cost pruning. To utilize these properties, we introduce the notion of potential good points which can act as a proxy of the actual good points. We then design a robust average linkage algorithm based on the cost computed only on the potential good points, which constructs a tree with a pruning that assigns all good points correctly. The pruning can be found out efficiently, and after some processing it leads to a clustering that is both a good approximation and close to the optimal clustering.

We also provide sublinear-time algorithms both for the k -median and min-sum objectives (Sections 4.3 and 5.1), showing algorithms that can return an implicit clustering from only access to a small random sample.

²For clarity, in this paper efficient means polynomial in both n (the number of points) and k (the number of clusters).

Related Work. A subsequent work [11] of [12] by Bilu, Daniely, Linial and Saks studied the Max-Cut problem under Bilu and Linial stability, and showed how to solve in polynomial time $(1 + \epsilon)$ -stable instances of metric and dense Max-Cut, and $\Omega(\sqrt{n})$ -stable instances of general Max-Cut. The later bound is further improved by Makarychev, Makarychev and Vijayaraghavan [21]. They proposed a polynomial time exact algorithm for $\Omega(\sqrt{\log n} \log \log n)$ -stable Max-Cut instances based on semidefinite programming. They also proved that for Max k -Cut with $k \geq 3$, there is no polynomial-time algorithm that solves ∞ -stable instances of Max k -Cut unless $\mathbf{NP} = \mathbf{RP}$. Here an instance is ∞ -stable if it is α -stable for every α . In a recent work [22], they further proposed a beyond worst-case analysis model for Balanced-Cut, which is a planted model with random edges from permutation-invariant distributions. They achieved a constant factor approximation with respect to the cost of the planted cut when the number of random edges is $\Omega(n \text{polylog}(n))$.

In the context of objective based clustering, several recent papers have showed how to exploit other notions of stability for overcoming the existing hardness results on worst case instances. The ORSS stability notion of Ostrovsky, Rabani, Schulman and Swamy [25, 3] assumes that the cost of the optimal k -means solution is small compared to the cost of the optimal $(k - 1)$ -means solution. The BBG approximation stability condition of Balcan, Blum and Gupta [5] assumes that every nearly optimal solution is close to the target clustering. Awasthi, Sheffet and Blum [2] proposed a stability condition called weak-deletion stability, and showed that it is implied by both the ORSS stability and the BBG stability. Kumar and Kannan [18] proposed a proximity condition which assumes that in the target clustering, most data points satisfy that they are closer to their center than to any other center by an additive factor in the order of the maximal standard variance of their clusters in any direction. Their results are improved by Awasthi and Sheffet [4], which proposed a weaker version of the proximity condition called center separation, and designed algorithms achieving stronger guarantees under this weaker condition.

Several recent papers have showed how to exploit the structure of perturbation resilient instances in order to obtain better approximation guarantees (than those possible on worst case instances) for other difficult optimization problems. These include the game theoretic problem of finding Nash equilibria [6, 20] and the classic traveling salesman problem [23].

2. Notation and Preliminaries. In a clustering instance, we are given a set S of n points in a finite metric space, and we denote $d : S \times S \rightarrow \mathbb{R}_{\geq 0}$ as the distance function. Φ denotes the objective function over a partition of S into $k < n$ clusters which we want to optimize over the metric, ie. Φ assigns a score to every clustering. The optimal clustering with respect to Φ is denoted as $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$, and its cost is denoted as \mathcal{OPT} . The core concept we study in this paper is the perturbation resilience notion introduced by [12]. Formally:

DEFINITION 2.1. *A clustering instance (S, d) is α -perturbation resilient to a given objective Φ if for any function $d' : S \times S \rightarrow \mathbb{R}_{\geq 0}$ such that $\forall p, q \in S, d(p, q) \leq d'(p, q) \leq \alpha d(p, q)$, there is a unique optimal clustering \mathcal{C}' for Φ under d' and this clustering is equal to the optimal clustering \mathcal{C} for Φ under d .*

Note that in the definition, d' need not be a metric. Also note that the definition depends on the objective. In this paper, we focus on the center-based and min-sum objectives. For the *center-based objectives*, we consider separable center-based objectives defined by [3].

DEFINITION 2.2. *A clustering objective is center-based if the optimal solution*

can be defined by k points c_1, \dots, c_k in the metric space called centers such that every data point is assigned to its nearest center. Such a clustering objective is separable if it furthermore satisfies the following two conditions:

- (1) The objective function value of a given clustering is either a (weighted) sum or the maximum of the individual cluster scores.
- (2) Given a proposed single cluster, its score can be computed in polynomial time.

One particular center-based objective is the k -median objective. We partition S into k disjoint subsets $\mathcal{P} = \{P_1, P_2, \dots, P_k\}$ and assign a set of centers $\mathbf{p} = \{p_1, p_2, \dots, p_k\} \subseteq S$ for the subsets. The objective is $\Phi(\mathcal{P}, \mathbf{p}) = \sum_{i=1}^k \sum_{p \in P_i} d(p, p_i)$. The centers in the optimal clustering are denoted as $\mathbf{c} = \{c_1, \dots, c_k\}$. Clearly, in an optimal solution, each point is assigned to its nearest center. In such cases, the objective is denoted as $\Phi(\mathbf{c})$.

For the *min-sum objective*, we partition S into k disjoint subsets denoted as $\mathcal{P} = \{P_1, P_2, \dots, P_k\}$, and the goal is to minimize $\Phi(\mathcal{P}) = \sum_{i=1}^k \sum_{p \in P_i} \sum_{q \in P_i} d(p, q)$. Note that we sometimes denote Φ as Φ_S in the case where the distinction is necessary, such as in Section 4.3.

In Section 4 we consider a generalization of perturbation resilience where we allow a small difference between the original optimum and the new optimum after perturbation. Formally:

DEFINITION 2.3. *Let \mathcal{C} be the optimal k -clustering and \mathcal{C}' be another k -clustering of a set of n points. We say \mathcal{C}' is ϵ -close to \mathcal{C} if $\min_{\sigma \in \mathcal{S}_k} \sum_{i=1}^k |C_i \setminus C'_{\sigma(i)}| \leq \epsilon n$, where σ is a matching between indices of clusters of \mathcal{C}' and those of \mathcal{C} .*

DEFINITION 2.4. *A clustering instance (S, d) is (α, ϵ) -perturbation resilient to a given objective Φ if for any function $d' : S \times S \rightarrow \mathbb{R}_{\geq 0}$ s.t. $\forall p, q \in S, d(p, q) \leq d'(p, q) \leq \alpha d(p, q)$, the optimal clustering \mathcal{C}' for Φ under d' is ϵ -close to the optimal clustering \mathcal{C} for Φ under d .*

For simplicity, we assume ϵn is an integer and assume that $\min_i |C_i|$ is known (otherwise, we can simply search over the n possible different values).

For $A, B \subseteq S$ we define $d(A, B) := \sum_{p \in A} \sum_{q \in B} d(p, q)$ and $d(p, B) := d(\{p\}, B)$. Also, we define $d_a(A, B) := d(A, B)/(|A||B|)$ and $d_a(p, B) := d_a(\{p\}, B)$. Finally, if $A \cap B = \emptyset$, we sometimes write $A \cup B$ as $A + B$ to emphasize that they are disjoint.

3. α -Perturbation Resilience for Center-based Objectives. In this section we show that, for $\alpha \geq 1 + \sqrt{2}$, if the clustering instance is α -perturbation resilient for center-based objectives, then we can in polynomial time find the optimal clustering. This improves on the $\alpha \geq 3$ bound of [3] and stands in sharp contrast to the NP-Hardness results on worst-case instances. Our algorithm succeeds for an even weaker property, the α -center proximity, introduced in [3].

DEFINITION 3.1. *A clustering instance (S, d) satisfies the α -center proximity property if for any optimal cluster $C_i \in \mathcal{C}$ with center c_i , $C_j \in \mathcal{C} (j \neq i)$ with center c_j , any point $p \in C_i$ satisfies $\alpha d(p, c_i) < d(p, c_j)$.*

LEMMA 3.2. *Any clustering instance that is α -perturbation resilient to center-based objectives also satisfies the α -center proximity.*

The proof follows easily by constructing a specific perturbation that blows up all the pairwise distances within cluster C_i by a factor of α . By α -perturbation resilience, the optimal clustering remains the same after this perturbation. This then implies the desired result. The full proof appears in [3]. In the remainder of this section, we prove our results for α -center proximity, but because it is a weaker condition, our upper bounds also hold for α -perturbation resilience.

We begin with some key properties of α -center proximity instances.

LEMMA 3.3. *For any points $p \in C_i$ and $q \in C_j (j \neq i)$ in the optimal clustering of an α -center proximity instance, we have*

- (1) $d(c_i, q) > \frac{\alpha(\alpha-1)}{\alpha+1}d(c_i, p)$,
- (2) $d(p, q) > (\alpha - 1) \max\{d(p, c_i), d(q, c_j)\}$.

Consequently, when $\alpha \geq 1 + \sqrt{2}$, we have

- (1) $d(c_i, q) > d(c_i, p)$,
- (2) $d(p, q) > d(p, c_i)$.

Proof. (1) Lemma 3.2 gives us that $d(q, c_i) > \alpha d(q, c_j)$. By the triangle inequality, we have $d(c_i, c_j) \leq d(q, c_j) + d(q, c_i) < (1 + 1/\alpha)d(q, c_i)$. On the other hand, $d(p, c_j) > \alpha d(p, c_i)$ and therefore $d(c_i, c_j) \geq d(p, c_j) - d(p, c_i) > (\alpha - 1)d(p, c_i)$. Combining these inequalities, we get (1).

(2) The proof first appears in [3], and we include it for completeness. Without loss of generality, we can assume that $d(p, c_i) \geq d(q, c_j)$. By triangle inequality we have $d(p, q) \geq d(p, c_j) - d(q, c_j)$. From Lemma 3.2 we have $d(p, c_j) > \alpha d(p, c_i)$. Hence $d(p, q) > \alpha d(p, c_i) - d(q, c_j) \geq (\alpha - 1)d(p, c_i) \geq (\alpha - 1)d(q, c_j)$. \square

Lemma 3.3 implies for any optimal cluster C_i , the ball of radius $\max_{p \in C_i} d(c_i, p)$ around the center c_i contains only points from C_i , and moreover, points inside the ball are each closer to the center than to any point outside the ball. Inspired by this structural property, we define the notion of closure distance between two sets as the radius of the minimum ball that covers the sets and has some margin from points outside the ball. We show that any (strict) subset of an optimal cluster has smaller closure distance to another subset in the same cluster than to any subset of other clusters or to unions of other clusters. Using this, we will be able to define an appropriate linkage procedure that, when applied to the data, produces a tree on subsets that will all be laminar with respect to the clusters in the optimal solution. This will then allow us to extract the optimal solution using dynamic programming applied to the tree.

We now define the notion of closure distance and then present our algorithm for α -center proximity instances (Algorithm 1). Let $\mathbb{B}(p, r) := \{q : d(q, p) \leq r\}$ denote the ball around p with radius r .

DEFINITION 3.4. *The closure distance $d_S(A, A')$ between two disjoint non-empty subsets A and A' of point set S is the minimum $d \geq 0$ such that there is a point $c \in A \cup A'$ satisfying the following requirements:*

- (1) *coverage: the ball $\mathbb{B}(c, d)$ covers A and A' , that is, $A \cup A' \subseteq \mathbb{B}(c, d)$;*
- (2) *margin: points inside $\mathbb{B}(c, d)$ are closer to the center c than to points outside, that is, $\forall p \in \mathbb{B}(c, d), q \notin \mathbb{B}(c, d)$, we have $d(c, p) < d(p, q)$.*

Note that $d_S(A, A') = d_S(A', A) \leq \max_{p, q \in S} d(p, q)$ for any A and A' . Furthermore, it can be computed in polynomial time.

THEOREM 3.5. *For $(1 + \sqrt{2})$ -center proximity instances, Algorithm 1 outputs the optimal clustering in polynomial time.*

The proof follows immediately from the following key property of the Phase 1 of Algorithm 1. The details of dynamic programming are presented in Appendix A, and an efficient implementation of the algorithm is presented in Appendix B.

THEOREM 3.6. *For $(1 + \sqrt{2})$ -center proximity instances, Algorithm 1 constructs a binary tree \mathcal{T} such that the optimal clustering is a pruning of this tree.*

Proof. We prove correctness by induction. In particular, assume that our current clustering is laminar with respect to the optimal clustering. That is, for each cluster A in our current clustering and each C in the optimal clustering, we have either $A \subseteq C$,

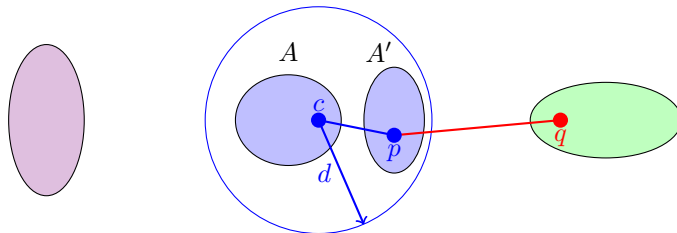


Fig. 1: Illustration for the closure distance.

Algorithm 1 Center-based objectives, α perturbation resilience

Input: Data set S , distance function $d(\cdot, \cdot)$ on S .

- 1: Begin with n singleton clusters.
- 2: Repeat till only one cluster remains:
merge clusters C, C' which minimize $d_S(C, C')$.
- 3: Let \mathcal{T} be the tree with single points as leaves and internal nodes corresponding to the merges performed.
- 4: Run dynamic programming on \mathcal{T} to get the minimum cost pruning $\tilde{\mathcal{C}}$.

Output: Clustering $\tilde{\mathcal{C}}$.

or $C \subseteq A$, or $A \cap C = \emptyset$. This is clearly true at the start. To prove that the merge steps keep the laminarity, we need to show the following: if A is a strict subset of an optimal cluster C_i , A' is a subset of another optimal cluster or the union of one or more other clusters, then there exists B from $C_i \setminus A$, such that $d_S(A, B) < d_S(A, A')$.

We first prove that there is a cluster $B \subseteq C_i \setminus A$ in the current cluster list such that $d_S(A, B) \leq \tilde{d} := \max_{p \in C_i} d(c_i, p)$. There are two cases. First, if $c_i \notin A$, then define B to be the cluster in the current cluster list that contains c_i . By induction, $B \subseteq C_i$ and thus $B \subseteq C_i \setminus A$. Then we have $d_S(B, A) \leq \tilde{d}$ since there is $c_i \in B$, and (1) for any $p \in A \cup B$, $d(c_i, p) \leq \tilde{d}$, (2) for any $p \in S$ satisfying $d(c_i, p) \leq \tilde{d}$, and any $q \in S$ satisfying $d(c_i, q) > \tilde{d}$, by Lemma 3.3 we know $p \in C_i$ and $q \notin C_i$, and thus $d(c_i, p) < d(p, q)$. In the second case when $c_i \in A$, we pick any $B \subseteq C_i \setminus A$ and a similar argument gives $d_S(A, B) \leq \tilde{d}$.

As a second step, we need to show that $\tilde{d} < \hat{d} := d_S(A, A')$. There are two cases: the center for $d_S(A, A')$ is in A or in A' . See Figure 2 for an illustration. In the first case, there is a point $c \in A$ such that c and \hat{d} satisfy the requirements of the closure distance. Pick a point $q \in A'$, and define C_j to be the cluster in the optimal clustering that contains q . As $d(c, q) \leq \hat{d}$, and by Lemma 3.3 we have $d(c_j, q) < d(c, q)$, then $d(c_j, c) \leq \hat{d}$ (otherwise it violates the second requirement of closure distance). Suppose $p = \arg \max_{p' \in C_i} d(c_i, p')$. Then we have $\tilde{d} = d(p, c_i) < d(p, c_j)/\alpha \leq (\tilde{d} + d(c_i, c) + d(c, c_j))/\alpha$ where the first inequality comes from Lemma 3.2 and the second from the triangle inequality. Since $d(c_i, c) < d(c, c_j)/\alpha$, we can combine the above inequalities and compare \tilde{d} and $d(c, c_j)$, and when $\alpha \geq 1 + \sqrt{2}$ we have $\tilde{d} < d(c, c_j) \leq \hat{d}$.

Now consider the second case, when there is a point $c \in A'$ such that c and \hat{d}

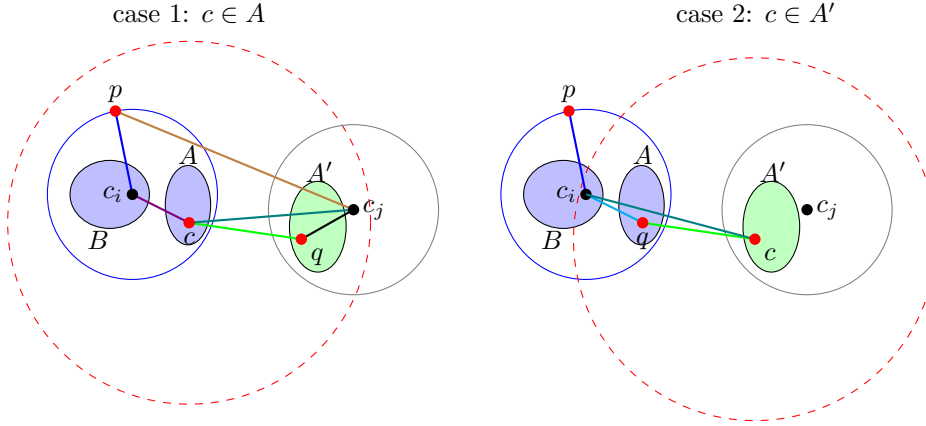


Fig. 2: Comparing \tilde{d} and $d_S(A, A')$ in closure linkage.

satisfy the requirements in the definition of the closure distance. Select an arbitrary point $q \in A$. We have $\hat{d} \geq d(c, q)$ from the first requirement, and $d(c, q) > d(c_i, q)$ by Lemma 3.3. Then from the second requirement of closure distance $d(c_i, c) \leq \hat{d}$. And by Lemma 3.3, $\tilde{d} = d(c_i, p) < d(c_i, c) \leq \hat{d}$. \square

NOTE 3.1. Our factor of $\alpha = 1 + \sqrt{2}$ beats the NP-hardness lower bound of $\alpha = 3$ of [3] for center-proximity instances. The reason is that the lower bound of [3] requires the addition of Steiner points that can act as centers but are not part of the data to be clustered (though the upper bound of [3] does not allow such Steiner points). One can also show a lower bound for center-proximity instances without Steiner points. In particular one can show that for any $\epsilon > 0$, the problem of solving $(2 - \epsilon)$ -center proximity k -median instances is NP-hard [9].

NOTE 3.2. The first condition in our definition of closure distance is similar to the minimax linkage criteria [10]. More precisely, our closure distance definition has two conditions: coverage condition and margin condition. If the margin condition is removed from the definition, then the closure distance reduces to the minimax linkage distance. For our purposes however, the margin condition is crucial — in particular, we can provably argue that when the center proximity condition is satisfied Algorithm 1 produces a tree such that the optimal clustering is a pruning of the tree (Theorem 3.5).

4. (α, ϵ) -Perturbation Resilience for the k -Median Objective. In this section we consider a natural relaxation of the α -perturbation resilience, the (α, ϵ) -perturbation resilience property, that requires the optimum after perturbation of up to a multiplicative factor α to be ϵ -close to the original (one should think of ϵ as sub-constant). We show that if the instance is (α, ϵ) -perturbation resilient, with $\alpha > 2 + \sqrt{3}$ and $\epsilon = O(\epsilon' \rho)$ where ρ is the fraction of the points in the smallest cluster, then we can in polynomial time output a clustering that provides a $(1 + \epsilon')$ -approximation to the optimum. Thus this improves over the best worst-case approximation guarantees known [19] when $\epsilon' \leq \sqrt{3}$ and also beats the lower bound of $(1 + 1/e)$ on the best approximation achievable on worst case instances for the metric k -median objective [16, 17] when $\epsilon' \leq 1/e$.

The key idea is to understand and leverage the structure implied by (α, ϵ) -perturbation resilience. We show that perturbation resilience implies that there exists only a small fraction of points that are bad in the sense that their distance to their own center is not α times smaller than their distance to any other centers in the optimal solution. We then use this bounded number of bad points in our clustering algorithm.

4.1. Structure of (α, ϵ) -Perturbation Resilience. To understand the structure of (α, ϵ) -perturbation resilience, we need to consider the difference between the optimal clustering \mathcal{C} under d and the optimal clustering \mathcal{C}' under a perturbation d' , defined as $\min_{\sigma \in \mathcal{S}_k} \sum_{i=1}^k |C_i \setminus C'_{\sigma(i)}|$. Throughout this section we will assume that $|C_i| > 2\epsilon n$ for all i .

Since $\sum_{i=1}^k |C_i \setminus C'_{\sigma(i)}| \leq \epsilon n$ by assumption, we clearly have separately for each i that $|C_i \setminus C'_{\sigma(i)}| \leq \epsilon n$. Since $|C_i| > 2\epsilon n$ this implies that $C'_{\sigma(i)}$ is the unique cluster in \mathcal{C}' such that $|C_i \cap C'_{\sigma(i)}| > \frac{1}{2}|C_i|$. Without loss of generality, let us index \mathcal{C}' so that σ is the identity. We denote by c'_i the center of C'_i .

In the following we introduce the notions of bad points and good points.

DEFINITION 4.1. *Define bad points for k -median to be those that are not α times closer to its own center than to any other center in the optimal clustering. That is,*

$$B := \cup_i B_i, \quad B_i := \{p \in C_i : \exists j \neq i, \alpha d(c_i, p) \geq d(c_j, p)\}.$$

The other points $G := S \setminus B$ are called good points. Let $G_i := G \cap C_i$ denote the good points in cluster C_i .

We show that under perturbation resilience we do not have too many bad points.

THEOREM 4.2. *Suppose the clustering instance is (α, ϵ) -perturbation resilient and $\min_i |C_i| > 6 \left(\frac{\alpha+1}{\alpha-1} \right) (\epsilon n + \alpha + 1)$. Then $|B| \leq \epsilon n$.*

Proof of Theorem 4.2. Assume for contradiction that $|B| > \epsilon n$. The main idea is to select a subset of $(\epsilon n + 1)$ bad points and then construct a specific perturbation so that in the new optimal clustering these (and only these) selected bad points move to new clusters, leading to a clustering that is ϵ far from the original optimal clustering. This is contradictory to the (α, ϵ) -perturbation resilience property, and thus there are at most ϵn bad points.

The selected bad points and the perturbation are defined as follows. Select an arbitrary subset \hat{B} of $(\epsilon n + 1)$ bad points from B , and let $\hat{B}_i = \hat{B} \cap C_i$ denote the selected bad points in C_i . Let $c(p)$ denote the second nearest center for $p \in \hat{B}_i$ and the nearest center for $p \notin \hat{B}_i$. The perturbation blows up all distances by a factor of α except for those distances between p and $c(p)$. Formally,

$$d'(p, q) = \begin{cases} d(p, q) & \text{if } p = c(q), \text{ or } q = c(p), \\ \alpha d(p, q) & \text{otherwise.} \end{cases}$$

The key challenge in showing the contradiction is to show that $c'_i = c_i$ for all i , that is, the optimal centers do not change after the perturbation. Once this is shown it is then immediate that in the optimum clustering under d' each point p is assigned to the center $c(p)$, and thus the selected bad points \hat{B} will move from their original optimal clusters and all others will not. Since all $|C_i| > 2|\hat{B}|$, the optimum σ remains the identity and so the distance between the new clustering and the original clustering is $|\hat{B}|$. This is contradictory to the (α, ϵ) -perturbation resilience property.

It will now be convenient to define a few quantities. Let $A_i = C'_i \setminus C_i$ (the points added when switching from C_i to C'_i), $M_i = C_i \setminus C'_i$ (the points missing), $W_i = (C_i \cap C'_i) \setminus \hat{B}_i$ (the common core), and $V_i = (C_i \cap C'_i) \cap \hat{B}_i$ (the selected bad points in common). So, $C_i = W_i + V_i + M_i$ and $C'_i = W_i + V_i + A_i$. See Figure 3. Note that $|A_i| \leq \epsilon n$, $|M_i| \leq \epsilon n$, and $|V_i| \leq \epsilon n + 1$, with the bulk of the points in W_i .

The intuition for the proof that $c'_i = c_i$ is the following. First, $d(c_i, c'_i)$ cannot be too large compared to the average distance between c_i and W_i , else by triangle inequality $d(c'_i, W_i)$ would also be large, violating the fact that $d'(c'_i, C'_i) \leq d'(c_i, C'_i)$. On the other hand, if $d(c_i, c'_i)$ is small then by triangle inequality $d(c'_i, W_i) \approx d(c_i, W_i)$. Since distances between c'_i and W_i are blown up by a factor of α in moving from d to d' but distance between c_i and W_i are not, this also will violate the fact that $d'(c'_i, C'_i) \leq d'(c_i, C'_i)$.

We begin with a convenient first step, showing that $c'_i \neq c_j$ for $j \neq i$.

FACT 4.1. *If $\min_i |C_i| > (\frac{2}{\alpha-1} + 3)\epsilon n + 1$, then $c'_i \neq c_j (\forall j \neq i)$.*

Proof Sketch. Assume for contradiction $c'_i = c_j$, and then consider the points in W_j . These points belong to C'_j , so under d' , they should be closer to their optimal center c'_j than to another center $c'_i = c_j$. By definition, $d'(c'_j, W_j) = \alpha d(c'_j, W_j)$, while $d'(c_j, W_j) = d(c_j, W_j)$. So, $\alpha d(c'_j, W_j) \leq d(c_j, W_j)$. This implies that the distance between c_j and c'_j cannot be too large compared to the average distance between c_j and W_j by triangle inequality. On the other hand, it also means c_j has much more cost than c'_j on W_j . Then c'_j should have much more cost on $C_j \setminus W_j$ since c_j is the optimal center for C_j . By triangle inequality, the distance between c_j and c'_j is much larger than the average distance between c_j and W_j . This is contradictory, and thus $c'_i \neq c_j$. The formal proof is presented in Appendix C. \square

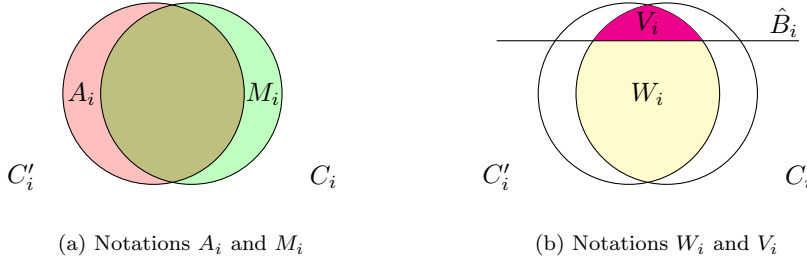


Fig. 3: Different types of points. (a) $A_i = C'_i \setminus C_i, M_i = C_i \setminus C'_i$. (b) $W_i = (C_i \cap C'_i) \setminus \hat{B}_i, V_i = (C_i \cap C'_i) \cap \hat{B}_i$. As a result, $C_i = W_i + V_i + M_i$ and $C'_i = W_i + V_i + A_i$.

We assume below for simplicity that $c(c'_i) \notin A_i$. For a complete proof, see Appendix C.

CLAIM 4.1. *For each i , $d(c_i, c'_i) \leq 3 \left(\frac{\alpha+1}{\alpha} \right) \frac{d(c_i, W_i)}{|C_i|}$.*

Proof Sketch. The key idea is that since $d'(c'_i, C'_i) \leq d'(c_i, C'_i)$ and $C'_i \setminus W_i$ is small, it must be the case that $d'(c'_i, W_i)$ is not too much larger than $d'(c_i, W_i)$. Now, since distances between c_i and W_i remained the same in moving from d to d' but distances between c'_i and W_i were blown up by a factor of α (except for the distance between c'_i and c_i itself), this means that $\alpha d(c'_i, W_i) - d(c_i, W_i)$ must be small. This is then used together with triangle inequality to get an upper bound on $d(c_i, c'_i)$.

First, if $c'_i = c_i$ the claim is trivially true so assume $c'_i \neq c_i$. As suggested above, we begin with the fact that $d'(c'_i, C'_i) \leq d'(c_i, C'_i)$ and then break C'_i into its three components W_i, V_i , and A_i . We move the W_i terms to one side (the cost more than c_i on W_i), the rest of the terms to another side (the cost saved on V_i and A_i):

$$(4.1) \quad d'(c'_i, W_i) - d'(c_i, W_i) \leq d'(c_i, A_i) - d'(c'_i, A_i) + d'(c_i, V_i) - d'(c'_i, V_i)$$

Beginning with the right hand side of (4.1), by the triangle inequality we have $d(c_i, V_i) \leq d(c'_i, V_i) + |V_i|d(c_i, c'_i)$. Thus, $d'(c_i, V_i) \leq d'(c'_i, V_i) + \alpha|V_i|d(c_i, c'_i)$. Similarly, under the assumption that $c(c'_i) \notin A_i$ we have $d'(c_i, A_i) \leq d'(c'_i, A_i) + \alpha|A_i|d(c_i, c'_i)$. So, the right-hand-side of (4.1) is at most $\alpha(|V_i| + |A_i|)d(c_i, c'_i)$. Now, examining the left hand side, this quantity is at least $\alpha d(c'_i, W_i \setminus \{c(c'_i)\}) - d(c_i, W_i)$. So, we have

$$(4.2) \quad \alpha d(c'_i, W_i \setminus \{c(c'_i)\}) - d(c_i, W_i) \leq \alpha(|V_i| + |A_i|)d(c_i, c'_i).$$

Using the fact that by triangle inequality, $\alpha(|W_i| - 1)d(c_i, c'_i) \leq \alpha d(c'_i, W_i \setminus \{c(c'_i)\}) + \alpha d(c_i, W_i \setminus \{c(c'_i)\})$, and subtracting $(\alpha + 1)d(c_i, W_i)$ from both sides, we get

$$(4.3) \quad \alpha d(c_i, c'_i)(|W_i| - 1) - (\alpha + 1)d(c_i, W_i) \leq \alpha d(c'_i, W \setminus \{c(c'_i)\}) - d(c_i, W_i).$$

Combining (4.2) and (4.3) we have:

$$\alpha d(c_i, c'_i)(|W_i| - 1) - (\alpha + 1)d(c_i, W_i) \leq \alpha d(c_i, c'_i) [|V_i| + |A_i|]$$

which implies the desired result. \square

CLAIM 4.2. *For each i , if $c'_i \neq c_i$ then $d(c_i, c'_i) \geq \left(\frac{\alpha-1}{2\alpha}\right) \frac{d(c_i, W_i)}{\epsilon n + \alpha + 1}$.*

Proof Sketch. Assume $c'_i \neq c_i$ and let $d_i = d(c_i, c'_i)$. We will first use the fact that by definition of c_i , $d(c_i, C_i) \leq d(c'_i, C_i)$. We then will move from C_i to C'_i and then from d to d' . Here we sketch the proof for the case when $c(c'_i) \notin A_i$; the complete proof is provided in the appendix.

First, since $d(c_i, C_i) \leq d(c'_i, C_i)$ and the difference between C'_i and C_i is small, $d(c_i, C'_i)$ cannot be too much larger than $d(c'_i, C'_i)$. In particular, since the set A_i of points added has $|A_i| \leq \epsilon n$ and the set M_i of points removed has $|M_i| \leq \epsilon n$ we have by triangle inequality:

$$(4.4) \quad d(c_i, A_i) \leq d(c'_i, A_i) + (\epsilon n)d_i,$$

$$(4.5) \quad d(c_i, M_i) \geq d(c'_i, M_i) - (\epsilon n)d_i.$$

So, $d(c_i, C'_i) \leq d(c'_i, C'_i) + 2(\epsilon n)d_i$.

Now, we switch from d to d' . We begin with A_i . By Fact 4.1 and assuming $c(c'_i) \notin A_i$, we have $d'(c'_i, A_i) = \alpha d(c'_i, A_i)$. For sure we have $d'(c_i, A_i) \leq \alpha d(c_i, A_i)$. Combining with (4.4) we have

$$(4.6) \quad d'(c_i, A_i) \leq d'(c'_i, A_i) + \alpha(\epsilon n)d_i.$$

But now c_i gets a saving from $C_i \setminus M_i$. Specifically, from (4.5) we have:

$$d(c_i, C_i \setminus M_i) \leq d(c'_i, C_i \setminus M_i) + (\epsilon n)d_i$$

so

$$(4.7) \quad \begin{aligned} \alpha d(c_i, C_i \setminus M_i) &\leq \alpha d(c'_i, C_i \setminus M_i) + \alpha(\epsilon n)d_i \\ &\leq d'(c'_i, C_i \setminus M_i) + (\alpha - 1)d_i + \alpha(\epsilon n)d_i \end{aligned}$$

where the second step is from the following fact: $d'(c'_i, C_i \setminus M_i) = \alpha d(c'_i, C_i \setminus M_i)$ if $c(c'_i) \neq c_i$, else $d'(c'_i, C_i \setminus M_i) = \alpha d(c'_i, C_i \setminus M_i) - (\alpha - 1)d_i$. Now, the left hand side above equals $d'(c_i, C_i \setminus M_i) + (\alpha - 1)d(c_i, W_i)$ because distances between c_i and W_i are *not* blown up by a factor of α .

Adding up (4.6) and (4.7) means that we get a contradiction if $(\alpha - 1)d(c_i, W_i) > [2\alpha\epsilon n + \alpha(\alpha + 1) + \alpha - 1]d_i$, in other words, if our savings in using c_i as center is greater than our extra cost. Therefore, $d_i \geq \left(\frac{\alpha-1}{2\alpha}\right) \frac{d(c_i, W_i)}{\epsilon n + \alpha + 1}$ as desired. \square

Combining the lower-bound of Claim 4.1 with the upper-bound of Claim 4.2 when $c'_i \neq c_i$, we get a contradiction for sufficiently large $|C_i|$ as given in the theorem statement, yielding $c'_i = c_i$. \square

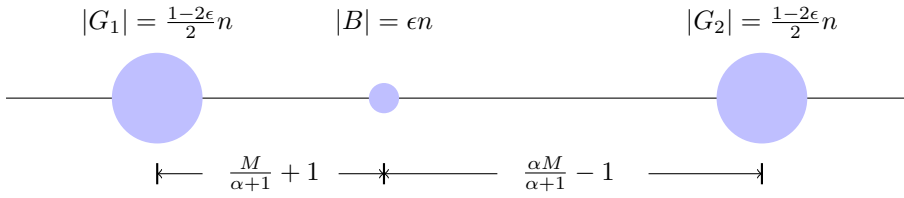


Fig. 4: An example showing the optimality of the bound on the number of bad points.

NOTE 4.1. The bound in Theorem 4.2 is optimal in the sense that for any $\alpha > 1$ and $0 < \epsilon < 1/5$, we can easily construct an (α, ϵ) -perturbation resilient 2-median instance which has ϵn bad points.

The instance is shown in Figure 4. It has 3 groups of points: G_1, G_2 , and B . Both G_1 and G_2 have $(1 - \epsilon)n/2$ points, and B has ϵn points. Let M be a sufficiently large constant, say, $M > n^2/\epsilon$. The distances within the same group are 1, while those between the points in G_1 and G_2 are M , those between the points in B and G_1 are $\frac{M}{\alpha+1} + 1$, and those between the points in B and G_2 are $\frac{\alpha M}{\alpha+1} - 1$. The instance satisfies the triangle inequality, which can be verified by a case-by-case study. The optimal clustering before perturbation has one center in G_1 and the other in G_2 . Then B are trivially bad points, and thus we have ϵn bad points in this instance.

Now we show that the instance is (α, ϵ) -perturbation resilient. To prove that the optimal clustering after perturbation \mathcal{C}' is ϵ -close to the original optimal clustering, it suffices to show that \mathcal{C}' has one center from $G_1 \cup B$ and the other center from G_2 . Assume for contradiction that this is not true. If both centers come from G_2 , the cost of points in G_1 is $\frac{(1-\epsilon)n}{2}M$. On the other hand, the optimal cost before perturbation is $(1 - \epsilon)n - 2 + \epsilon n(\frac{M}{\alpha+1} + 1)$, so the optimal cost after perturbation is no more than $\alpha[(1 - \epsilon)n - 2 + \epsilon n(\frac{M}{\alpha+1} + 1)]$. But this is smaller than $\frac{(1-\epsilon)n}{2}M$, which is a contradiction. Similarly, we get a contradiction if both centers come from $G_1 \cup B$.

NOTE 4.2. Theorem 4.2 requires that the sizes of the optimal clusters are sufficiently large. This makes sure that a majority of points in each optimal cluster stay after moving at most ϵn points, which means W_i is significantly larger than V_i, A_i and M_i . This fact is crucial for proving the theorem. In the following subsections, we assume $n > \alpha/\epsilon$, so that when $\alpha - 1 = O(1)$, the requirement in Theorem 4.2 is simplified as $\min_i |C_i| = \Omega(\epsilon n)$.

4.2. Approximating the Optimum Clustering. Now, we consider approximating the cost of the optimum clustering. We can see that after removing the bad points, the optimal clusters are far apart from each other. In order to get rid of the influence of the bad points, we generate a list of blobs, which form a partition of the data points, and each of which contains only good points from one optimal cluster. Then we construct a tree on the list of blobs with a pruning that assigns all good points correctly. We will show that this pruning has low cost, so the lowest cost pruning of the tree is a good approximation. The details are described in Algorithm 2.

A key step is to generate the list of almost “pure” blobs, which is described in Algorithm 3. Suppose for any i and any good point $p \in G_i$, its $\gamma|G_i|$ nearest neighbors contain no good points outside C_i . Also suppose the algorithm knows the value of γ . Informally, the algorithm maintains a threshold t . At each threshold, for each point p that has not been added to the list, the algorithm checks its γt nearest neighbors $N_{\gamma t}(p)$. It constructs a graph F_t by connecting any two points that have most neighbors in common. It then builds another graph H_t by connecting any two points that have sufficiently many neighbors in F_t , and adds sufficiently large components in H_t to the list. Finally, for each remaining point p , it checks if most of p 's neighbors are in the list and if there are blobs containing a significant amount of p 's neighbors. If so, it inserts p into such a blob with the smallest median distance. Then the threshold is increased and the above steps are repeated.

The intuition behind Algorithm 3 is as follows. As mentioned above, the algorithm works when for any i and any good point $p \in G_i$, the $\gamma|G_i|$ nearest neighbors of p contain no good points outside C_i ($\gamma = 1$ for the k -median instances considered in this section, as shown in Lemma 4.3; $\gamma = \frac{4}{5}$ for the min-sum instances considered in Section 6, as shown in Claim 6.3). Without loss of generality, assume $|C_1| \leq |C_2| \leq \dots \leq |C_k|$. When $t \leq |C_1|$, good points in different clusters do not have most neighbors in common and thus are not connected in F_t . However, they may be connected by a path of bad points. So we further build the graph H_t to disconnect such paths, which ensures that the blobs added into the list contain only good points from one optimal cluster. The final insert step (Step 6) makes sure that when $t = |C_1|$, all remaining good points in C_1 will be added to the list and will not affect the construction of blobs from other optimal clusters. We can show by induction that, at the end of the iteration $t = |C_i|$, all good points in $C_j (j \leq i)$ are added to the list. When t is large enough, any remaining bad points are inserted into the list, so the points are partitioned into a list of almost pure blobs. The formal guarantee for Algorithm 3 is stated in Lemma 4.4.

Another key step is to construct a tree on these blobs. Since good points are closer to good points in the same optimal cluster than to those in other clusters (Lemma 4.3), there exist algorithms that can build a tree with a pruning that assigns all good points correctly. In particular, we can use the robust linkage procedure in [7], which repeatedly merges the two blobs C, C' with the maximum score $\text{score}(C, C')$ defined as follows. For each $p \in C$, sort the other blobs in decreasing order of the median distance between p and points in the blob, and let $\text{rank}(p, C')$ denote the rank of C' . Then define $\text{rank}(C, C') = \text{median}_{x \in C}[\text{rank}(x, C')]$ and $\text{score}(C, C') = \min[\text{rank}(C, C'), \text{rank}(C', C)]$. Intuitively, for any blobs A, A' from the same optimal cluster and D from a different cluster, good points in A always rank A' later than D in the sorted list, so $\text{rank}(A, A') > \text{rank}(A, D)$. Similarly, $\text{rank}(A', A) > \text{rank}(A', D)$, and thus $\text{score}(A', A) > \text{score}(A, D)$. This means the algorithm will always merge blobs from the same cluster before merging them with blobs outside, so there is a

Algorithm 2 k -median, (α, ϵ) perturbation resilience

Input: Data set S , distance function $d(\cdot, \cdot)$ on S , $\min_i |C_i|$, $\epsilon > 0$

- 1: Run Algorithm 3 to generate a list \mathcal{L} of blobs with parameters $m_B = \epsilon n, \gamma = 1$.
- 2: Run the robust linkage procedure in [7] to get a cluster tree \mathcal{T} .
- 3: Run dynamic programming on \mathcal{T} to get the minimum cost pruning $\tilde{\mathcal{C}}$ and its centers $\tilde{\mathbf{c}}$.

Output: Clustering $\tilde{\mathcal{C}}$ and its centers $\tilde{\mathbf{c}}$.

Algorithm 3 Generating interesting blobs

Input: Data set S , distance function $d(\cdot, \cdot)$ on S , the size of the smallest optimal cluster $\min_i |C_i|$, the upper bound on the number of bad points m_B , a parameter $\gamma \in [4/5, 1]$

- 1: Let $N_r(p)$ denote the r nearest neighbors of p in S .
- 2: Let $\mathcal{L} = \emptyset, A_S = S$. Let the initial threshold $t = \min_i |C_i|$.
- 3: Construct a graph F_t by connecting $p, q \in A_S$ if $|N_{\gamma t}(p) \cap N_{\gamma t}(q)| > (2\gamma - 1)t - 2m_B$.
- 4: Construct a graph H_t by connecting points $p, q \in A_S$ if p, q share more than m_B neighbors in F_t .
- 5: Add to \mathcal{L} all the components C of H_t with $|C| \geq \frac{1}{2} \min_i |C_i|$ and remove them from A_S .
- 6: For each point $p \in A_S$, check if most of $N_{\gamma t}(p)$ are in \mathcal{L} and if there exists $C \in \mathcal{L}$ containing a significant number of points in $N_{\gamma t}(p)$. More precisely, check if
 - (1) $|N_{\gamma t}(p) \setminus \mathcal{L}| \leq \frac{1}{2} \min_i |C_i| + 2m_B$;
 - (2) $\mathcal{L}_p \neq \emptyset$ where $\mathcal{L}_p = \{C \in \mathcal{L} : |C \cap N_{\gamma t}(p)| \geq (\gamma - \frac{3}{5})|C|\}$.
 If so, assign p to the blob in \mathcal{L}_p of smallest median distance, remove p from A_S .
- 7: While $|A_S| > 0$, increase t by 1 and go to Step 3.

Output: The list \mathcal{L} .

pruning that assigns all good points correctly.

In the following, we prove that Algorithm 2 outputs a good approximation. We begin with a key property of (α, ϵ) -perturbation resilience that ensures the success of Algorithm 3.

LEMMA 4.3 (Theorem 8 in [9], Lemma 2.6 in [3]). *When $\alpha > 2 + \sqrt{3}$, for any good points $p_1, p_2 \in G_i, q \in G_j (j \neq i)$, we have $d(p_1, p_2) < d(p_1, q)$. Consequently, for any good point $p \in G_i$, all its $|G_i|$ nearest neighbors belong to $C_i \cup B$.*

Proof. The following proof is implicit in [3] and we include it for completeness. We rephrase it slightly so that it is more intuitive. By triangle inequality and the definition of good points,

$$d(p_2, p_1) + d(p_1, q) + d(q, c_j) \geq d(p_2, c_j) > \alpha d(p_2, c_i) \geq \alpha [d(p_1, p_2) - d(p_1, c_i)].$$

Rearranging terms leads to

$$(4.8) \quad d(p_1, q) + d(q, c_j) + \alpha d(p_1, c_i) > (\alpha - 1)d(p_1, p_2).$$

Now, to compare $d(p_1, q)$ and $d(p_1, p_2)$, we need to get rid of the extra terms $d(q, c_j)$ and $d(p_1, c_i)$. By the same proof in Lemma 3.3(2),

$$d(p_1, q) > (\alpha - 1)d(p_1, c_i), \quad \text{and} \quad d(p_1, q) > (\alpha - 1)d(q, c_j).$$

Plugging these into (4.8), we have

$$\left[1 + \frac{1}{\alpha - 1} + \frac{\alpha}{\alpha - 1}\right] d(p_1, q) > (\alpha - 1)d(p_1, p_2).$$

So when $\alpha > 2 + \sqrt{3}$, $d(p_1, q) > d(p_1, p_2)$. \square

LEMMA 4.4. *Suppose the number of bad points is bounded by m_B , and for any i and any good point $p \in G_i$, all its $\gamma|G_i|$ nearest neighbors in S are from $C_i \cup B$. If $\min_i |C_i| > 30m_B$, then Algorithm 3 generates a list \mathcal{L} of blobs each of size at least $\frac{1}{2} \min_i |C_i|$ such that:*

- (1) *The blobs in \mathcal{L} form a partition of S .*
- (2) *Each blob in \mathcal{L} contains good points from only one optimal cluster.*

Proof. We prove the following two claims by induction on $i \leq k$:

- (1) For any $t \leq |G_i|$, any blob in the list \mathcal{L} only contains good points from only one optimal cluster; all blobs have size at least $\frac{1}{2} \min_i |C_i|$.
- (2) At the beginning of the iteration $t = |G_i| + 1$, any good point $p \in G_j, j \leq i$ has already been assigned to a blob in the list that contains good points only from C_j .

The first two claims imply that each blob in the list contains good points from only one optimal cluster. Moreover, at the beginning of the iteration $t = |G_k| + 1$, all good points have been assigned to one of the blobs in \mathcal{L} , so there are only bad points left, the number of which is smaller than $\frac{1}{2} \min_i |C_i|$. These remaining points will eventually be assigned to the blobs before $\gamma t > n$, so the blobs form a partition of S .

The claims are clearly both true initially. We show now that as long as $t \leq |G_1|$, the graphs F_t and H_t have the following properties.

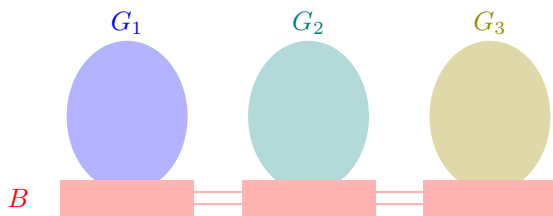


Fig. 5: A high level illustration of the graph F_t .

- No good point p_i in cluster C_i is connected in F_t to a good point p_j in a different cluster C_j . By assumption, p_i has no neighbors outside $C_i \cup B$ and p_j has no neighbors outside $C_j \cup B$, so they share at most $m_B < (2\gamma - 1)t - 2m_B$ neighbors.
- No point q is connected in F_t to both a good point p_i in C_i and a good point p_j in a different cluster C_j . If q is connected to p_i , then $|N_{\gamma t}(p_i) \cap N_{\gamma t}(q)| > (2\gamma - 1)t - 2m_B$. Since p_i has no neighbors outside $C_i \cup B$, $N_{\gamma t}(q)$ contains more than $(2\gamma - 1)t - 3m_B \geq \gamma t / 2$ points from G_i . Similarly, if q is connected to p_j , then $N_{\gamma t}(q)$ contains more than $\gamma t / 2$ points from G_j , which is contradictory. Thus, the graph F_t looks like the illustration in Figure 5.
- All the components in H_t of size at least $\frac{1}{2} \min_i |C_i|$ will only contain good points from one optimal cluster. As there are at most m_B bad points, any two points connected in H_t must be connected in F_t to at least one good point.

Then by the above two properties, points on a path in H_t must be connected in F_t to good points in the same cluster, so there is no path connecting good points from different clusters.

We can use the three properties to argue the first claim: as long as $t \leq |G_1|$, each blob in \mathcal{L} contains good points from at most one optimal cluster. This is true at the beginning and by the third property, for any $t \leq |G_1|$, anytime we insert a whole new blob in the list in Step 5, that blob must contain point from at most one optimal cluster. We now argue that this property is never violated as we assign points to blobs already in the list in Step 6. Suppose a good point $p \in C_i$ is inserted into $C \in \mathcal{L}$. Then $C \in \mathcal{L}_p$, which means $|N_{\gamma t}(p) \cap C| \geq |C|/5 > m_B$. So $N_{\gamma t}(p) \cap C$ contains at least one good point, which must be from C_i since $N_{\gamma t}(p)$ contains no good points outside C_i . Then by induction C must contain only good points from C_i , and thus adding p to C does not violate the first claim.

We now show the second claim: after the iteration $t = |G_1|$, all the good points in C_1 have already been assigned to a blob in the list that only contains good points from C_1 . There are two cases. First, if at the beginning of the iteration $t = |G_1|$, there are still at least $\frac{1}{2} \min_i |C_i|$ points from the good point set G_1 that do not belong to blobs in the list. Any such good point has all $\gamma|G_1|$ neighbors in $C_1 \cup B$. Then any two such good points share at least $2\gamma|G_1| - |C_1 \cup B| \geq (2\gamma - 1)|G_1| - |B| \geq (2\gamma - 1)t - 2m_B$ neighbors. So they will connect to each other in F_t and then in H_t , and thus we will add one blob to \mathcal{L} containing all these points. Second, it could be that at the beginning of the iteration $t = |G_1|$, all but less than $\frac{1}{2} \min_i |C_i|$ good points in G_1 have been assigned to a blob in the list. Denote the points that have not yet been assigned as E . Any point $p \in E$ has no neighbors outside $C_1 \cup B$. Then $|N_{\gamma t}(p) \setminus \mathcal{L}| \leq |E| + |B| \leq \frac{1}{2} \min_i |C_i| + 2m_B$. Also, there exists a blob C containing good points from C_1 such that $C \in \mathcal{L}_p$. Otherwise, $N_{\gamma t}(p)$ contains at most $(\gamma - \frac{3}{5})(|C_1 \cup B|) < \gamma|C_1| - \frac{1}{2}|C_1| - 2m_B$ points in $C_1 \cap \mathcal{L}$, while it contains at most $|E|$ good points in $C_1 \setminus \mathcal{L}$ and contains no points outside $C_1 \cup B$. In total, $N_{\gamma t}(p)$ has less than γt points, which is contradictory. So $\mathcal{L}_p \neq \emptyset$ and p will be added to the list in Step 6.

We then iterate the argument on the remaining set A_S . The key point is that for $t \geq |G_i|, i > 1$, we have that all the good points in C_1, C_2, \dots, C_i have already been assigned to blobs in \mathcal{L} . \square

Lemma 4.3 and 4.4 show that Algorithm 3 with parameters $m_B = \epsilon n$ and $\gamma = 1$ produces a list of sufficiently large, almost pure blobs. Then the robust linkage procedure in [7] can build a tree on these blobs with a pruning that assigns all good points correctly. Now it suffices to show that this pruning is a good approximation, for which we need to bound the cost increased by the bad points assigned incorrectly. The following property of these bad points turns out to be useful. Intuitively, Algorithm 3 is designed such that whenever a bad point is added to a blob containing good points from a different cluster, it must be closer to a significant number of points in that cluster than to a significant number of points in its own cluster. Then the cost increased by incorrectly assigning each such bad point is small, resulting in a good approximation.

LEMMA 4.5. *Suppose for any good point $p \in G_i$, all its $|G_i|$ nearest neighbors in S are from $C_i \cup B$, and $\min_i |C_i| > 30m_B$. When running Algorithm 3 with $\gamma = 1$, if a bad point $q \in B_i$ is assigned to a blob C containing good points from a different optimal clustering C_j , then there exist $m = \frac{1}{5} \min_i |C_i|$ points Z_i from C_i , and m points Z_j from C_j , such that $d(q, Z_i) \geq d(q, Z_j)$.*

Proof. There are two cases: q is added into C in (1) Step 5 or (2) Step 6.

Case 1. There must be a path in H_t connecting q to a good point in C_j at threshold t . For any edge (x, y) in H_t , since x, y share at least ϵn neighbors in F_t and there are at most ϵn bad points, they share at least one good point as neighbor in F_t . As shown in the proof of Lemma 4.4, no point can connect to good points from different clusters, so in F_t all points on the path must connect to good points in C_j . In particular, q is connected in F_t to a good point $p \in G_j$. Then $|N_t(p) \cap N_t(q)| > t - 2m_B$. Since p is still in A_S , $t \leq |G_j|$, and thus $N_t(p)$ contains no points outside $C_j \cup B$. This means that at least $t - 3m_B \geq m$ points in $N_t(q)$ are good points in C_j , then we can select m points Z_j from $N_t(q) \cap G_j$. We also have that at most $2m_B$ points in $N_t(q)$ are points in C_i , so we can select m points Z_i from $C_i \setminus N_t(q)$.

Case 2. There are three subcases when q is inserted into C at threshold t .

- (1) There is no good points from C_i in the list. Since $|N_t(q) \setminus \mathcal{L}| \leq \frac{1}{2} \min_i |C_i| + 2m_B$, $N_t(q)$ contains at most this number of good points in C_i . This means at least $\frac{1}{2} \min_i |C_i| - 2m_B > m$ good points in C_i are outside $N_t(q)$, from which we can select Z_i . On the other hand, we can select Z_j as follows. When inserting q into C , we have $|N_t(q) \cap C| \geq \frac{2}{5}|C| \geq m + m_B$. Since C contains only good points from C_j and some bad points, $N_t(q) \cap C$ contains at least m good points in C_j , from which we can select Z_j . Since Z_j are from $N_t(q)$ and Z_i are outside $N_t(q)$, we have $d(q, Z_i) \geq d(q, Z_j)$.
- (2) There exists $C' \in \mathcal{L}$ containing good points from C_i , but $C' \notin \mathcal{L}_p$. This means $|B(q, t) \cap C'| \leq \frac{2}{5}|C'|$, so there are at least $\frac{3}{5}|C'| \geq m + m_B$ points in C' are outside $N_t(q)$. At least m of these points are good points from C_i , since C' contains only good points from C_i and at most m_B bad points. So, we can select Z_i from them. On the other hand, we can select Z_j as in the first subcase.
- (3) There exists $C' \in \mathcal{L}_p$ containing good points from C_i . Since q is assigned to C rather than C' according to median distances, we know that at least half of the points Z'_j from C are closer to q than at least half of the points Z'_i from C' . Since there are at most m_B bad points, we can select m good points Z_j from Z'_j and select m good points Z_i from Z'_i . Note that Z_j are all from G_j and Z_i are all from G_i , so $d(q, Z_i) \geq d(q, Z_j)$.

Therefore, the statement is true in all cases. \square

THEOREM 4.6. *If the clustering instance is (α, ϵ) -perturbation resilient for $\alpha > 2 + \sqrt{3}$ and $\epsilon \leq \rho/30$ where $\rho = \frac{\min_i |C_i|}{n}$, then Algorithm 2 produces a clustering which is $(1 + \frac{5\epsilon}{\rho})$ -approximation to the optimal clustering with respect to the k -median objective in polynomial time.*

Proof. By Lemma 4.3 and 4.4, Algorithm 3 partition the points into a list of blobs, each of which has size at least $\frac{1}{2} \min_i |C_i|$ and contains only good points from one optimal cluster. Let B'_i denote the bad points that are assigned to blobs containing good points in C_i . By Lemma 4.3, Theorem 9 in [7] can be applied to \mathcal{L} , by which we know that $\{(C_i \cap G) \cup B'_i\}$ is a pruning of the tree. Suppose the cost of the optimum is \mathcal{OPT} . We now show that this pruning, using the original centers $\{c_i\}$, is a $(1 + \frac{5\epsilon}{\rho})$ -approximation to \mathcal{OPT} .

Suppose a bad point $q \in C_i$ is assigned to a blob C containing good points from a different optimal cluster C_j . By Lemma 4.5, there exist $m = \frac{1}{5} \min_i |C_i|$ points Z_i from C_i , and m points Z_j from C_j , such that $d(q, Z_i) \geq d(q, Z_j)$. Then the increase

in cost due to q is bounded as follows:

$$\begin{aligned} d(q, c_j) - d(q, c_i) &\leq \frac{d(q, Z_j) + d(c_j, Z_j)}{m} - \frac{d(q, Z_i) + d(c_i, Z_i)}{m} \\ &\leq \frac{1}{m}[d(c_j, Z_j) + d(c_i, Z_i)] \leq \frac{\mathcal{OPT}}{m}. \end{aligned}$$

As there are at most ϵn bad points and $m = \frac{\min_i |C_i|}{5}$, the increase of cost is at most $\frac{\epsilon n}{m} \mathcal{OPT} = \frac{5\epsilon}{\rho} \mathcal{OPT}$.

Running Time. In Algorithm 3, for each $p \in S$, we first sort all the other points in ascending order of distances in time $O(n^2 \log n)$. At each threshold t , think of a directed t -regular graph E_t , where, for each point q in the t nearest neighbors of a point p , there is a directed edge from p to q in E_t . Let A_E denote the adjacency matrix for E_t , and let $N = A_E A_E^T$. Then N_{pq} is the number of common neighbors between p and q , which can be used in constructing F_t . Computing N takes time $O(n^\omega)$, the state of the art for matrix multiplication. The same method can be used to compute the number of common neighbors in F_t and construct H_t . Since there are $O(n)$ thresholds, the total time for constructing F_t and H_t is $O(n^{\omega+1})$. For the other steps, adding a blob takes time $O(n^2)$ and inserting a point takes time $O(n^2)$. These steps can be performed at most $O(n)$ times, so they take $O(n^3)$ time. In total, Algorithm 3 takes time $O(n^{\omega+1})$. Since the robust linkage algorithm [7] takes time at most $O(n^{\omega+1})$, and the dynamic programming takes time $O(n^3)$ (Appendix A), the running time of Algorithm 2 is $O(n^{\omega+1})$. \square

We observe that for $\alpha > 2 + \sqrt{3}$ we can tolerate ϵ up to approximately ρ (recalling that ρ is the fraction of points in the smallest cluster), and beat the lower bound of $(1 + 1/e)$ on the best approximation achievable on worst case instances for the metric k -median objective [16, 17] when $\epsilon < \frac{\rho}{5e}$.

4.3. Sublinear Time Algorithm for the k -Median Objective. Consider a clustering instance (X, d) that is (α, ϵ) -perturbation resilient to k -median. For simplicity, suppose the distances are normalized such that $\max_{p, q} d(p, q) = 1$. Let $N = |X|$. Let $\rho = \min_i |C_i|/N$ denote the fraction of the points in the smallest cluster, $\zeta = \Phi_X(\mathbf{c})/N$ denote the average cost of the points in the optimum clustering.

Algorithm 4 k -median, (α, ϵ) perturbation resilience, sublinear

Input: Data set X , distance function $d(\cdot, \cdot)$ on X , $\min_i |C_i|$.

- 1: Draw a sample S of size $n = \Theta(\frac{k}{\epsilon^2 \zeta} \ln \frac{N}{\delta})$ i.i.d. from X .
- 2: Run Algorithm 2 on S to obtain $\tilde{\mathcal{C}}$ and $\tilde{\mathbf{c}}$.

Output: The implicit clustering obtained by assigning each point in X to its nearest neighbor in $\tilde{\mathbf{c}}$.

THEOREM 4.7. *Suppose (X, d) is (α, ϵ) -perturbation resilient for $\alpha > 2 + \sqrt{3}$, $\epsilon < \rho/100$. Then with probability $\geq 1 - \delta$, Algorithm 4 outputs an implicit clustering that is $2(1 + \frac{16\epsilon}{\rho})$ -approximation in time $\text{poly}(\log \frac{N}{\delta}, k, \frac{1}{\epsilon}, \frac{1}{\delta})$.*

Proof. It suffices to show that $\Phi_S(\tilde{\mathbf{c}})$ is close to $\Phi_S(\mathbf{c})$ where \mathbf{c} are the optimal centers for X . Note that Algorithm 2 builds a tree with a pruning \mathcal{P}' that assigns all good points correctly. The key is to use the cost of this pruning as a bridge for $\Phi_S(\tilde{\mathbf{c}})$ and $\Phi_S(\mathbf{c})$: on one hand, $\Phi_S(\tilde{\mathbf{c}})$ is no more than the cost of \mathcal{P}' since $\tilde{\mathbf{c}}$ is the centers in the minimum cost pruning; on the other hand, the cost of \mathcal{P}' is roughly bounded by twice $\Phi_S(\mathbf{c})$ by triangle inequality.

Formally, recall the following notations. If we partition A into \mathcal{P} , the cost using centers \mathbf{p} is denoted as $\Phi_A(\mathcal{P}, \mathbf{p})$. If we partition A by assigning points to nearest centers in \mathbf{p} , the cost is denoted as $\Phi_A(\mathbf{p})$. We will show that the cost of the implicit clustering $\Phi_X(\tilde{\mathbf{c}})$ approximates the optimum $\Phi_X(\mathbf{c})$.

First, we will prove that when n is sufficiently large, with high probability, $\Phi_X(\tilde{\mathbf{c}})/N \approx \Phi_S(\tilde{\mathbf{c}})/n$ and $\Phi_X(\mathbf{c})/N \approx \Phi_S(\mathbf{c})/n$. For every set of centers \mathbf{p} , if $n = \Theta(\frac{k}{v^2\zeta} \log \frac{N}{\delta})$ where $0 < v < 1$, then by Chernoff bound,

$$\Pr \left[\left| \frac{\Phi_S(\mathbf{p})}{n} - \frac{\Phi_X(\mathbf{p})}{N} \right| > v \frac{\Phi_X(\mathbf{p})}{N} \right] \leq 2 \exp \left\{ -\frac{v^2}{3} \frac{\Phi_X(\mathbf{p})}{N} n \right\} \leq 2 \exp \left\{ -\frac{v^2}{3} \zeta n \right\} \leq \frac{\delta}{4N^k}.$$

By union bound, we have with probability at least $1 - \delta/4$,

$$(1-v) \frac{\Phi_X(\tilde{\mathbf{c}})}{N} \leq \frac{\Phi_S(\tilde{\mathbf{c}})}{n}, \text{ and } \frac{\Phi_S(\mathbf{c})}{n} \leq (1+v) \frac{\Phi_X(\mathbf{c})}{N}.$$

We can choose $v = \epsilon/20$, then it is sufficient to show $\Phi_S(\tilde{\mathbf{c}}) \leq 2(1 + 12\epsilon/\rho)\Phi_S(\mathbf{c})$.

Next, since $\tilde{\mathcal{C}}$ may be different from $\mathcal{C} \cap S$, we need to find a bridge for comparing $\Phi_S(\tilde{\mathbf{c}})$ and $\Phi_S(\mathbf{c})$. Now, we turn to analyze Algorithm 2 on S to find such a bridge. First, we know that X has at most ϵN bad points. Since n is sufficiently large, with probability at least $1 - \delta/4$, S has at most $2\epsilon n$ bad points. Similarly, with probability at least $1 - \delta/4$, for any i , $|C_i \cap S| > 60\epsilon n$. These ensure that Algorithm 2 can successfully produce a tree with a pruning \mathcal{P}' that assigns all good points in S correctly, as shown in Theorem 4.6. Suppose in S , \mathbf{c}' are the optimal centers for \mathcal{P}' . Then we can use $\Phi_S(\mathcal{P}', \mathbf{c}')$ as a bridge for comparing $\Phi_S(\tilde{\mathbf{c}})$ and $\Phi_S(\mathbf{c})$.

On one hand, $\Phi_S(\tilde{\mathbf{c}}) \leq \Phi_S(\tilde{\mathcal{C}}, \tilde{\mathbf{c}}) \leq \Phi_S(\mathcal{P}', \mathbf{c}')$. The first inequality comes from the fact that in $\Phi_S(\tilde{\mathbf{c}})$ each point is assigned to its nearest center and the second comes from that $\tilde{\mathcal{C}}$ is the minimum cost pruning.

On the other hand, $\Phi_S(\mathcal{P}', \mathbf{c}') \leq 2\Phi_S(\mathcal{P}', \mathbf{c}) \leq 2(1 + 12\epsilon/\rho)\Phi_S(\mathbf{c})$. The second inequality comes from an argument similar to that in Theorem 4.6 and the fact that $\Phi_S(\mathcal{P}', \mathbf{c})$ is different from $\Phi_S(\mathbf{c})$ only on the bad points. The first inequality comes from the triangle inequality. More precisely, for any $N'_i \in \mathcal{P}'$,

$$\begin{aligned} 2|N'_i| \sum_{p \in N'_i} d(p, c_i) &= \sum_{q \in N'_i} \sum_{p \in N'_i} [d(p, c_i) + d(q, c_i)] \\ &\geq \sum_{p \in N'_i} \sum_{q \in N'_i} d(p, q) \geq \sum_{p \in N'_i} \sum_{q \in N'_i} d(q, c'_i) = |N'_i| \sum_{q \in N'_i} d(q, c'_i). \end{aligned}$$

This completes the proof. \square

NOTE 4.3. If we have an oracle that given a set of points C'_i finds the best center *in* X for that set, then we can save a factor of 2 in the bound.

5. α -Perturbation Resilience for the Min-Sum Objective. In this section we provide an efficient algorithm for clustering α -perturbation resilient instances for the metric min-sum k -clustering problem (Algorithm 5). Recall that $d(A, B) = \sum_{p \in A} \sum_{q \in B} d(p, q)$ and $d_a(A, B) = d(A, B)/(|A||B|)$.

THEOREM 5.1. For $(3 \frac{\max_i |C_i|}{\min_i |C_i|})$ -perturbation resilient instances, Algorithm 5 outputs the optimal min-sum k -clustering in polynomial time.

Algorithm 5 min-sum, α perturbation resilience**Input:** Data set S , distance function $d(\cdot, \cdot)$ on S , $\min_i |C_i|$.

- 1: Connect each point with its $\frac{1}{2} \min_i |C_i|$ nearest neighbors.
- 2: Initialize the clustering \mathcal{C}' with each connected component being a cluster.
- 3: Repeat until only one cluster remains in \mathcal{C}' :
merge clusters C, C' in \mathcal{C}' which minimize $d_a(C, C')$.
- 4: Let \mathcal{T} be the tree with components as leaves and internal nodes corresponding to the merges performed.
- 5: Run dynamic programming on \mathcal{T} to get the minimum min-sum cost pruning $\tilde{\mathcal{C}}$.

Output: $\tilde{\mathcal{C}}$.

Intuition. To prove the theorem, first we show that the α -perturbation resilience property implies the following (Lemma 5.2): for any two different optimal clusters C_i and C_j and any $A \subseteq C_i$, we have $\alpha d(A, C_i \setminus A) < d(A, C_j)$. This follows by considering the perturbation where $d'(p, q) = \alpha d(p, q)$ if $p \in A, q \in C_i \setminus A$ and $d'(p, q) = d(p, q)$ otherwise, and using the fact that the optimum does not change after the perturbation. This can be used to show that when $\alpha > 3 \frac{\max_{i'} |C_{i'}|}{\min_{i'} |C_{i'}|}$, we have the following (Lemma 5.3): (1) for any optimal clusters C_i and C_j and any $A_i \subseteq C_i, A_j \subseteq C_j$ such that $\min(|C_i \setminus A_i|, |C_j \setminus A_j|) > \min_i |C_i|/2$ we have $d_a(A_i, A_j) > \min\{d_a(A_i, C_i \setminus A_i), d_a(A_j, C_j \setminus A_j)\}$; (2) for any point p in the optimal cluster C_i , twice its average distance to points in $C_i \setminus \{p\}$ is smaller than the distance to any point in other optimal cluster C_j . Claim (2) implies that for any point $p \in C_i$ its $|C_i|/2$ nearest neighbors are in the same optimal cluster, so the leaves of the tree \mathcal{T} are laminar to the optimum clustering. Claim (1) can be used to show that the merge steps preserve the laminarity with the optimal clustering, so the minimum cost pruning of \mathcal{T} will be the optimal clustering, as desired.

We now prove the key lemmas mentioned above, and then present the detailed proof for the theorem.

FACT 5.1. *For any nonempty sets A, C, D such that $D \subseteq C$ we have*

$$|D|d(A, C) \leq |C|d(A, D) + |A|d(D, C \setminus D).$$

Proof. By the triangle inequality, for any $p \in A, q \in C \setminus D$ and $z \in D$,

$$d(p, q) \leq d(p, z) + d(z, q).$$

Summing over all $p \in A, q \in C \setminus D$ and $z \in D$, we have

$$|D|d(A, C \setminus D) \leq |C \setminus D|d(A, D) + |A|d(D, C \setminus D).$$

Plugging in $d(A, C \setminus D) = d(A, C) - d(A, D)$ leads to the statement. \square

LEMMA 5.2. *Suppose the clustering instance is α -perturbation resilient to the min-sum objective. For any two different optimal clusters C_i and C_j and any $A \subseteq C_i$, we have $\alpha d(A, C_i \setminus A) < d(A, C_j)$.*

Proof. This follows by considering a specific perturbation and using the fact that the optimum does not change after the perturbation. The perturbation is defined as:

$$d'(p, q) = \begin{cases} \alpha d(p, q) & \text{if } p \in A, q \in C_i \setminus A \text{ or } q \in A, p \in C_i \setminus A, \\ d(p, q) & \text{otherwise.} \end{cases}$$

Since d' is a valid α -perturbation of d , the optimal clustering after perturbation should remain the same. In particular, its cost should be smaller than that of the clustering

obtained by replacing C_i and C_j with $C_i \setminus A$ and $A \cup C_j$. After canceling the terms common in the two costs, we have $2d'(A, C_i \setminus A) < 2d'(A, C_j)$, which then leads to $\alpha d(A, C_i \setminus A) < d(A, C_j)$. \square

LEMMA 5.3. *Suppose the clustering instance is α -perturbation resilient to minimum for $\alpha > 3 \frac{\max_{i'} |C_{i'}|}{\min_{i'} |C_{i'}|}$. Then the following statements are true:*

- (1) *For any two different optimal clusters C_i and C_j and any $A_i \subseteq C_i, A_j \subseteq C_j$, if $|C_i \setminus A_i|$ and $|C_j \setminus A_j|$ are larger than $\min_{i'} |C_{i'}|/2$, then*

$$d_a(A_i, A_j) > \min[d_a(A_i, C_i \setminus A_i), d_a(A_j, C_j \setminus A_j)].$$

- (2) *For any point p , all its $\min_{i'} |C_{i'}|/2$ nearest neighbors are in the same optimal cluster.*

Proof. (1) Let $\bar{A}_i := C_i \setminus A_i$ and $\bar{A}_j := C_j \setminus A_j$. By Lemma 5.2 and Fact 5.1,

$$(5.1) \quad \alpha d(A_i, \bar{A}_i) < d(A_i, C_j) \leq \frac{1}{|A_j|} \left[|C_j| d(A_i, A_j) + |A_i| d(A_j, \bar{A}_j) \right],$$

$$(5.2) \quad \alpha d(A_j, \bar{A}_j) < d(A_j, C_i) \leq \frac{1}{|A_i|} \left[|C_i| d(A_j, A_i) + |A_j| d(A_i, \bar{A}_i) \right].$$

Divide Inequality (5.1) by $|A_i|$, divide Inequality (5.2) by $|A_j|$, add them up, and move the $d(A_j, \bar{A}_j)$ and $d(A_i, \bar{A}_i)$ terms to the left-hand-side:

$$(\alpha - 1)|\bar{A}_j| d_a(A_j, \bar{A}_j) + (\alpha - 1)|\bar{A}_i| d_a(A_i, \bar{A}_i) < (|C_i| + |C_j|) d_a(A_i, A_j).$$

Since α , $|\bar{A}_j|$ and $|\bar{A}_i|$ are large enough, $(\alpha - 1)|\bar{A}_i| > |C_i|$ and $(\alpha - 1)|\bar{A}_j| > |C_j|$. So,

$$|C_j| d_a(A_j, \bar{A}_j) + |C_i| d_a(A_i, \bar{A}_i) < (|C_i| + |C_j|) d_a(A_i, A_j).$$

Dividing by $(|C_i| + |C_j|)$ we see that $d_a(A_i, A_j)$ is greater than a weighted average of $d_a(A_j, \bar{A}_j)$ and $d_a(A_i, \bar{A}_i)$ and so is certainly greater than the minimum as desired.

- (2) Suppose p comes from the optimal cluster C_i . Let $q = \arg \min_{p' \notin C_i} d(p, p')$, and suppose $q \in C_j$.

If $d_a(p, C_i) \geq d_a(q, C_j)$, then by Inequality (5.1),

$$\alpha d(p, C_i) \leq |C_j| d(p, q) + d(q, C_j) = |C_j| [d(p, q) + d_a(q, C_j)] \leq |C_j| [d(p, q) + d_a(p, C_i)]$$

which leads to $d_a(p, C_i) < d(p, q)/2$ since α is sufficiently large.

If $d_a(p, C_i) < d_a(q, C_j)$, then we have $d_a(q, C_j) < d(p, q)/2$ by a similar argument.

In conclusion, we always have $d_a(p, C_i) < d(p, q)/2$, which means that more than $|C_i|/2$ points in C_i are within distance less than $d(p, q) = \min_{p' \notin C_i} d(p, p')$. \square

We are now ready to use the lemmas to prove our theorem.

Proof of Theorem 5.1. It is sufficient to show that in Algorithm 5:

- (1) Initially each $A \in \mathcal{C}'$ satisfies $A \subseteq C$ for some $C \in \mathcal{C}$;
- (2) \mathcal{C}' is always laminar to the optimal clustering \mathcal{C} , that is, for any $A \in \mathcal{C}'$ and $C \in \mathcal{C}$, we have either $A \subseteq C$, or $C \subseteq A$, or $A \cap C = \emptyset$.

Then the minimum cost pruning of \mathcal{T} will be the optimal clustering, which can be obtained by dynamic programming.

Lemma 5.3(2) implies that initially each $A \in \mathcal{C}'$ satisfies $A \subseteq C$ for some $C \in \mathcal{C}$. This means that \mathcal{C}' is laminar initially. Then Lemma 5.3(1) can be used to show that the merge steps preserve the laminarity, so \mathcal{C}' is always laminar to the optimal clustering.

More precisely, we prove the laminarity by induction. By Lemma 5.3(2), \mathcal{C}' is laminar initially. It is sufficient to prove that if the current clustering is laminar, then the merge step keeps the laminarity. Assume that our current clustering \mathcal{C}' is laminar to the optimal clustering. Consider a merge of two clusters A and A' . There are two cases when laminarity could fail to be satisfied after the merge:

- (1) They are strict subsets from different optimal clusters, that is, $A \subsetneq C_i, A' \subsetneq C_j \neq C_i$;
- (2) A is a strict subset of an optimal cluster C_i and A' is the union of one or several other optimal cluster(s).

In the first case, $C_i \setminus A_i$ ($C_j \setminus A'$ respectively) contains at least one cluster in \mathcal{C}' . By construction, each cluster in \mathcal{C}' is of size at least $\min_{i'} |C_{i'}|/2$, and thus $C_i \setminus A_i$ and $C_j \setminus A'$ are of size at least $\min_{i'} |C_{i'}|/2$. Then by Lemma 5.3(1), A and A' cannot be merged. In the second case, for any E that is a subset of $C_i \setminus A$ in the current clustering, we have $d_a(A, E) \geq d_a(A, A')$. We know that $d_a(A, C_i \setminus A)$ is a weighted average of the average distances between A and the clusters that are subsets of $C_i \setminus A$ in the current clustering, so $d_a(A, C_i \setminus A) \geq d_a(A, A')$. Also, $d_a(A, A')$ is a weighted average of the average distances between A and the optimal clusters in A' , so there must exist an optimal cluster $C_j \subseteq A'$ such that $d_a(A, C_j) \leq d_a(A, A') \leq d_a(A, C_i \setminus A)$. This means

$$d(A, C_j) \leq \frac{|C_j|}{|C_i \setminus A|} d(A, C_i \setminus A) \leq \alpha d(A, C_i \setminus A)$$

where the last inequality comes from $\alpha \geq 3 \frac{\max_{i'} |C_{i'}|}{\min_{i'} |C_{i'}|}$ and $|C_i \setminus A| \geq \min_{i'} |C_{i'}|/2$. This contradicts Lemma 5.2. So the merge of the two clusters A and A' will preserve the laminarity.

Running Time. Finding the nearest neighbors for each point takes $O(n \log n)$ time, so the step of constructing components takes $O(n^2 \log n)$ time. To compute average distances between clusters, we can record the size of each cluster, and $d(C'_i, C'_j)$ for any C'_i, C'_j in the current clustering, and update $d(C'_i \cup C'_j, C'_l) = d(C'_i, C'_l) + d(C'_j, C'_l)$ for any other cluster C'_l when merging C'_i and C'_j . So the merge steps take $O(n^3)$ time. As dynamic programming takes $O(n^3)$ time, we can find the optimum clustering in $O(n^3)$ time. \square

5.1. Sublinear Algorithm for the Min-Sum Objective. Here we provide a sublinear algorithm for a clustering instance (X, d) that is α -perturbation resilient to the min-sum objective. For simplicity, suppose the distances are normalized such that $\max_{p,q} d(p, q) = 1$. Let $N = |X|$. Let $\rho = \min_i |C_i|/N$ denote the fraction of the points in the smallest optimal cluster, and $\eta = \min_j \min_{p \notin C_j} d_a(p, C_j)$ denote the minimum average distance between an optimal cluster and a point outside.

Algorithm 6 min-sum, α perturbation resilience, sublinear

Input: Data set X , distance function $d(\cdot, \cdot)$ on X , $\min_i |C_i|$.

- 1: Draw a sample S of size $n = \Theta(\frac{1}{\rho\eta} \ln \frac{Nk}{\delta})$ i.i.d. from X .
- 2: Run Algorithm 5 on S to obtain $\tilde{\mathcal{C}}$.

Output: The implicit clustering of X obtained by assigning each point $p \in X$ to $\tilde{C}_i \in \tilde{\mathcal{C}}$ such that $d(p, \tilde{C}_i)$ is minimized.

Our main result in this subsection is the following.

THEOREM 5.4. *Suppose the clustering instance (X, d) is α -perturbation resilient*

to the min-sum objective where $\alpha \geq 6 \frac{\max_i |C_i|}{\min_i |C_i|}$. Then with probability at least $1 - \delta$, Algorithm 6 outputs an implicit optimum clustering in time $\text{poly}(\log \frac{Nk}{\delta}, \frac{1}{\rho}, \frac{1}{\eta})$.

Proof. To prove the theorem, we first show the following (Lemma 5.5): with high probability, \mathcal{C}' in Algorithm 5 is always laminar to $\mathcal{C} \cap S$. The key idea is that when the sample is sufficiently large, we have that for any $p \in C_i$ and $C_j (j \neq i)$,

$$3 \frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S|} d(p, C_i \cap S) < d(p, C_j \cap S)$$

since $\frac{d(p, C_i \cap S)}{n} \approx \frac{d(p, C_i)}{N}$, $\frac{d(p, C_j \cap S)}{n} \approx \frac{d(p, C_j)}{N}$ and $\frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S|} \approx \frac{\max_i |C_i|}{\min_i |C_i|}$. Then $\mathcal{C} \cap S$ satisfies the properties for the linkage in Algorithm 5 to succeed, and thus \mathcal{C}' in Algorithm 5 is always laminar to $\mathcal{C} \cap S$. Then $\mathcal{C} \cap S$ is a pruning of the tree.

Then we show that $\mathcal{C} \cap S$ is actually the minimum cost pruning $\tilde{\mathcal{C}}$ (Lemma 5.6). The key idea is that clusters in $\mathcal{C} \cap S$ are far apart, the cost increased by joining different clusters in it is larger than that saved by splitting clusters, so any other pruning has larger cost than $\mathcal{C} \cap S$. It immediately follows from the two lemmas that the implicit clustering obtained is the optimum clustering \mathcal{C} . \square

We now present the proofs of the lemmas for the correctness of the theorem.

LEMMA 5.5. *Suppose the clustering instance is α -perturbation resilient to the min-sum objective for $\alpha \geq 6 \frac{\max_i |C_i|}{\min_i |C_i|}$. When $n = O(\frac{1}{\rho\eta} \ln \frac{Nk}{\delta})$, with probability at least $1 - \delta$, \mathcal{C}' in Algorithm 5 is always laminar to $\mathcal{C} \cap S$.*

Proof. The intuition is that on X , for any $i \neq j$, any $p \in C_i$, we have $\alpha d(p, C_i) < d(p, C_j)$. When n is sufficiently large, we can show $d(p, C_i \cap S) \approx \frac{n}{N} d(p, C_i)$ for any i and $\frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S|} \approx \frac{\max_i |C_i|}{\min_i |C_i|}$, and thus we have a similar claim on S . Then \mathcal{C}' in Algorithm 5 is always laminar to $\mathcal{C} \cap S$.

First, since $n = O(\frac{1}{\rho\eta} \ln \frac{Nk}{\delta})$, by Chernoff bound we have

$$\Pr \left[\left| \frac{|C_i \cap S|}{n} - \frac{|C_i|}{N} \right| \geq v \frac{|C_i|}{N} \right] \leq 2 \exp \left\{ -\frac{2v^2}{2+v} \frac{|C_i|}{N} n \right\} \leq 2 \exp \{ -\frac{2v^2}{2+v} \rho n \} \leq \frac{\delta}{4k}.$$

Set $v = 1/20$. By union bound, with probability at least $1 - \delta/4$, for any $1 \leq i \leq k$,

$$(5.3) \quad \left(1 - \frac{1}{20}\right) \frac{n}{N} |C_i| \leq |C_i \cap S| \leq \left(1 + \frac{1}{20}\right) \frac{n}{N} |C_i|.$$

Similarly, with probability at least $1 - \delta/4$, for any $i \neq j$ and $p \in C_i$,

$$(5.4) \quad d(p, C_j \cap S) \geq \left(1 - \frac{1}{20}\right) \frac{n}{N} d(p, C_j).$$

Now, fix any i and $p \in C_i$. We have

$$\Pr \left[\frac{d(p, C_i \cap S)}{n} - \frac{d(p, C_i)}{N} \geq v \frac{d(p, C_i)}{N} \right] \leq \exp \left\{ -\frac{2v^2}{2+v} \frac{d(p, C_i)}{N} n \right\}.$$

Let $v = \frac{\min_{j \neq i} d(p, C_j)}{10d(p, C_i)}$. Since $v \geq 1/10$,

$$\begin{aligned} & \Pr \left[\frac{d(p, C_i \cap S)}{n} - \frac{d(p, C_i)}{N} \geq \frac{\min_{j \neq i} d(p, C_j)}{10N} \right] \leq \exp \left\{ -\frac{2v}{21} \frac{d(p, C_i)}{N} n \right\} \\ & \leq \exp \left\{ -\frac{2}{210} \frac{\min_{j \neq i} d(p, C_j)}{N} n \right\} \leq \exp \left\{ -\frac{2}{210} \min_{j \neq i} d_a(p, C_j) \rho n \right\} \leq \frac{\delta}{4Nk}. \end{aligned}$$

By union bound, with probability at least $1 - \delta/4$, for any $j \neq i$ and $p \in C_i$,

$$(5.5) \quad d(p, C_i \cap S) \leq \frac{n}{N}d(p, C_i) + \frac{n}{10N}d(p, C_j).$$

Now, by (5.3), we have $\max_i |C_i \cap S| \leq (1 + v)\frac{n}{N} \max_i |C_i|$, $\min_i |C_i \cap S| \geq (1 - v)\frac{n}{N} \min_i |C_i|$. Combined these with (5.4) (5.5), we have that with probability at least $1 - \delta$, for any $i \neq j$ and any $p \in C_i$,

$$3 \frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S|} d(p, C_i \cap S) < d(p, C_j \cap S),$$

which guarantees the success of Algorithm 5. \square

LEMMA 5.6. *Suppose the clustering instance is α -perturbation resilient to the min-sum objective for $\alpha \geq 6 \frac{\max_i |C_i|}{\min_i |C_i|}$. When $n = O(\frac{1}{\rho\eta} \ln \frac{Nk}{\delta})$, with probability at least $1 - \delta$, the minimum min-sum cost pruning of the tree in Algorithm 5 is $\mathcal{C} \cap S$.*

Proof. Since the tree is laminar to $\mathcal{C} \cap S$, we know that $\mathcal{C} \cap S$ is a pruning of the tree, and any other pruning can be obtained by splitting some clusters in $\mathcal{C} \cap S$ and joining some others into unions. Intuitively, the clusters in $\mathcal{C} \cap S$ are far apart, so the cost increased by joining different clusters is larger than the cost saved by splitting clusters. This claim then implies $\mathcal{C} \cap S$ is the minimum cost pruning. We first prove a similar claim for \mathcal{C} by the α -perturbation resilience, that is, for any three different clusters $C_i, C_j, C_l \in \mathcal{C}$, any $A_X \subseteq C_i$, $\alpha d(A_X, C_i \setminus A_X) < d(C_j, C_l)$. Then we prove the claim for $\mathcal{C} \cap S$: for any $A \subseteq C_i \cap S$, $2d(A, C_i \cap S \setminus A) < d(C_j \cap S, C_l \cap S)$. Finally we use it to prove $\mathcal{C} \cap S$ is the minimum cost pruning.

CLAIM 5.1. *For any three different optimal clusters $C_i, C_j, C_l \in \mathcal{C}$, any $A_X \subseteq C_i$, $\alpha d(A_X, C_i \setminus A_X) < d(C_j, C_l)$.*

Proof. For any $A_X \subseteq C_i$, we define a perturbation as follows: blow up the distances between the points in A_X and those in $C_i \setminus A_X$ by a factor of α , and keep all the other pairwise distances unchanged. By the α -perturbation resilience, we know that \mathcal{C} is still the optimum clustering after perturbation. Therefore, it has lower cost than the clustering obtained by replacing C_i with A_X and $C_i \setminus A_X$, and replacing C_j and C_l with $C_j \cup C_l$. After canceling the common terms in the costs of the two clusterings, we have $2d'(A_X, C_i \setminus A_X) < 2d'(C_j, C_l)$, which leads to the claim. \square

CLAIM 5.2. *With probability at least $1 - \delta$, for any $A \subseteq C_i \cap S$,*

$$2d(A, C_i \cap S \setminus A) \leq 2d(C_i \cap S, C_i \cap S) < d(C_j \cap S, C_l \cap S).$$

Proof. On one hand, for $A_X \subseteq C_i$, $\alpha d(A_X, C_i \setminus A_X) < d(C_j, C_l)$ by Claim 5.1. Since $\sum_{A_X \subseteq C_i} d(A_X, C_i \setminus A_X) = 2^{|C_i|} d(C_i, C_i)/2$, we have $\frac{\alpha}{2} d(C_i, C_i) < d(C_j, C_l)$. On the other hand, a similar argument as that of Lemma 5.5 shows with probability at least $1 - \delta/2$, for any C_i and any $p \in C_i$, $d(p, C_i \cap S) \leq (1 + v)\frac{n}{N}d(p, C_i)$ for $v = 1/20$. So

$$\begin{aligned} d(C_i \cap S, C_i \cap S) &= \sum_{p \in C_i \cap S} d(p, C_i \cap S) \leq (1 + v)\frac{n}{N} \sum_{p \in C_i \cap S} d(p, C_i) \\ &= (1 + v)\frac{n}{N} \sum_{q \in C_i} d(C_i \cap S, q) \leq (1 + v)^2 \frac{n^2}{N^2} d(C_i, C_i). \end{aligned}$$

Similarly, with probability at least $1 - \delta/2$, for any C_j and C_l , $d(C_j \cap S, C_l \cap S) \geq (1 - v)^2 \frac{n^2}{N^2} d(C_j, C_l)$. The claim then follows by combining the three inequalities and noting $v = 1/20$. \square

Now, we use Claim 5.2 to prove the optimality of $\mathcal{C} \cap S$. Suppose a pruning \mathcal{P}^* is obtained by splitting h clusters in $\mathcal{C} \cap S$ and at the same time joining some other clusters into g unions. Specifically, for $1 \leq i \leq h$, split $C_i \cap S$ into $m_i \geq 2$ clusters $S_{i,1}, \dots, S_{i,m_i}$; after that, merge $C_{h+1} \cap S, \dots, C_{h+l_g} \cap S$ into g unions, that is, for $1 \leq j \leq g$ and $l_0 = 0$, merge $l_j - l_{j-1} \geq 2$ clusters $C_{h+l_{j-1}+1} \cap S, \dots, C_{h+l_j} \cap S$ into a union U_j ; the other clusters in $\mathcal{C} \cap S$ remain the same in \mathcal{P}^* . Since the number of clusters is still k , we have $\sum_i m_i - h = l_g - g$. The cost saved by splitting clusters is

$$(5.6) \quad \sum_{1 \leq i \leq h} \sum_{1 \leq p \neq q \leq m_i} 2d(S_{i,p}, S_{i,q}) = \sum_{1 \leq i \leq h} \sum_{1 \leq p \leq m_i} 2d(S_{i,p}, C_i \cap S \setminus S_{i,p}).$$

The cost increased by joining clusters is

$$(5.7) \quad \sum_{1 \leq j \leq g} \sum_{h+l_{j-1} < p \neq q \leq h+l_j} 2d(C_p \cap S, C_q \cap S).$$

To prove $\mathcal{C} \cap S$ is the minimum cost pruning, we need to show that the saved cost (5.6) is less than the increased cost (5.7). Since each term in (5.7) is twice larger than any term in (5.6), it suffices to show that the number of the terms in (5.7) is at least half the number of the terms in (5.6). Formally, we need to show

$$2 \sum_{1 \leq j \leq g} \binom{l_j - l_{j-1}}{2} \geq \sum_{1 \leq i \leq h} m_i.$$

We have $2 \sum_j \binom{l_j - l_{j-1}}{2} = \sum_j (l_j - l_{j-1})(l_j - l_{j-1} - 1) \geq 2 \sum_j (l_j - l_{j-1} - 1) = 2(l_g - g)$, where the inequality comes from $l_j - l_{j-1} \geq 2$. Since $l_g - g = \sum_i m_i - h$, it is sufficient to show $l_g - g \geq h$. This comes from $l_g - g = \sum_i m_i - h = \sum_i (m_i - 1) \geq \sum_i 1 = h$ since $m_i \geq 2$. \square

6. (α, ϵ) -Perturbation Resilience for the Min-Sum Objective. For (α, ϵ) -perturbation resilient min-sum instances, we will show that if $\alpha \geq \frac{8 \max_i |C_i|}{\min_i |C_i|}$, $\epsilon = \tilde{O}(\rho)$ where $\rho = \frac{\min_i |C_i|}{n}$, there exists a polynomial time algorithm that outputs a clustering that is both a $(1 + \tilde{O}(\epsilon/\rho))$ -approximation and also $\tilde{O}(\epsilon)$ -close to the optimal clustering. Formally,

THEOREM 6.1. *Suppose the instance is (α, ϵ) -perturbation resilient to the min-sum objective for $\alpha > \frac{8 \max_i |C_i|}{\min_i |C_i|}$ and $\epsilon < \frac{\min_i |C_i|}{600n \log n}$. There exists an algorithm that outputs a clustering which is a $\left(1 + \frac{40\epsilon n \log n}{\min_i |C_i|}\right)$ -approximation to the optimal clustering in polynomial time. Furthermore, the output clustering is also $(6\epsilon \log n)$ -close to the optimal clustering.*

Since $\epsilon = O\left(\frac{\min_i |C_i|}{n \log n}\right)$, the approximation factor is always $O(1)$ and gets better if ϵ gets smaller. To prove the theorem, we first derive new useful structural properties implied by (α, ϵ) -perturbation resilience for min-sum, and then use them to design our algorithm achieving the guarantees in the theorem. Throughout this section, we assume $\alpha > \frac{8 \max_i |C_i|}{\min_i |C_i|}$ and $\epsilon < \frac{\min_i |C_i|}{600n \log n}$, except for where their values are explicitly specified. Also, since $\max_i |C_i| / \min_i |C_i| < n$ we may assume without loss of generality that $\alpha < 8n$.

6.1. Structural Properties of (α, ϵ) -Perturbation Resilience. In this subsection, we introduce the notion of bad points for the min-sum objective, and then show that there are just a few bad points while the other points have useful properties. More precisely,

DEFINITION 6.2. *Define bad points for min-sum to be those that are not β times closer to its own cluster than to other clusters, where $\beta = \frac{4}{5}\alpha$. That is,*

$$B := \cup_i B_i, \text{ where } B_i := \{p \in C_i : \exists j \neq i, d(p, C_j) \leq \beta d(p, C_i)\}, \beta := \frac{4}{5}\alpha.$$

The other points $G = S \setminus B$ are called good points.

Here β is chosen to be $\frac{4}{5}\alpha$ for reasons that will become clear in the proof of bounding the number of bad points. Informally, we will show that when α is sufficiently large and ϵ is sufficiently small, the number of bad points are bounded by $\tilde{O}(\epsilon n)$ (Theorem 6.4 in Section 6.1.1). The good points in different optimal clusters, by definition, are far from each other. It is then possible to design approximation algorithms if the influence of the few bad points can be eliminated.

However, we do not know the actual bad points or the actual good points. The key is that we can introduce a proxy called potential bad points (potential good points respectively), which can be easily computed. Formally,

DEFINITION 6.3.

- (1) Define $m_B := 6\epsilon \log n$.
- (2) For a set A with $|A| > 2m_B$, define the potential bad points $F(A)$ to be the $2m_B$ points in A that are farthest from A . That is, $F(A) \subseteq A$, $|F(A)| = 2m_B$, and for any $p \in F(A)$, $q \in S \setminus F(A)$, $d(p, A) \geq d(q, A)$. The other points $P(A) := A \setminus F(A)$ are called potential good points.
- (3) For a cluster A , define its robust min-sum cost as $d_{\text{rs}}(A) := d(P(A), P(A))$, where $P(A)$ are the potential good points in A . For a clustering \mathcal{C} , define its robust min-sum cost as $\sum_{C \in \mathcal{C}} d_{\text{rs}}(C)$.

We show that the robust min-sum cost computed after removing the potential bad points approximates the min-sum cost computed after removing the actual bad points (see Section 6.1.2). In other words, we can design approximation algorithms using the potential bad points as if we knew the actual bad points.

6.1.1. Bounding the Number of Bad Points. Here we bound the number of bad points by $\tilde{O}(\epsilon n)$ when α is sufficiently large and ϵ is sufficiently small.

THEOREM 6.4. *Suppose the clustering instance is (α, ϵ) -perturbation resilient to the min-sum objective for $\alpha > 4$ and $\epsilon < \frac{\min_i |C_i|}{200n}$. Then $|B| \leq m_B = 6\epsilon n \log n$.*

Proof. Assume for contradiction $|B| > 2\eta\epsilon n$ where $\eta = \left\lceil \log \frac{\max_i \max_{p \in B_i} d(p, C_i)}{\min_i \min_{p \in B_i} d(p, C_i)} \right\rceil$. We will first construct a perturbation which leads to a contradiction, and then show that $\eta \leq 3 \log n$, completing the proof.

We begin constructing the perturbation by introducing some notations. Consider the η intervals as follows: $[2^{t-1}v, 2^t v]$ where $v = \min_i \min_{p \in B_i} d(p, C_i)$, $1 \leq t \leq \eta$. At least one of the intervals, say $[r, 2r]$, will contain the costs of more than $2\epsilon n$ bad points. Let \hat{B} denote an arbitrary subset of $2\epsilon n$ bad points in this interval. Let $\hat{B}_i = \hat{B} \cap C_i$ denote the selected bad points in the optimal cluster C_i . Let $K_i = C_i \setminus \hat{B}_i$ denote the other points in C_i , and set $K = \cup_i K_i$. Denote as D_i all those selected bad points whose second nearest cluster is C_i , that is, $D_i = \{p : \exists j \text{ such that } p \in \hat{B}_j \text{ and } i = \arg \min_{\ell \neq j} d(p, C_\ell)\}$. Note that by definition we have $\cup_i D_i = \hat{B}$. Finally, let $\tilde{C}_i = K_i \cup D_i$. See Figure 6 for an illustration.

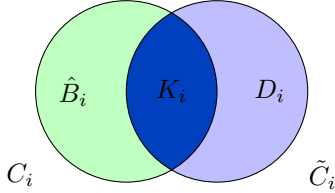


Fig. 6: Perturbation construction.

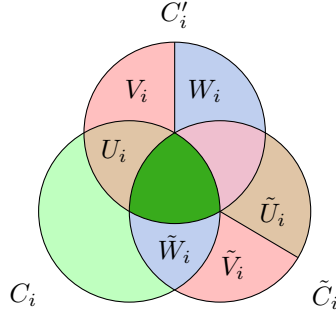


Fig. 7: Different types of points.

Now we are ready to construct the perturbation, which tries to make the selected bad points move to their second nearest clusters and keep the other points in their original clusters. More precisely, the perturbation is constructed as follows: blow up all distances by a factor of α except those within $\tilde{C}_i, 1 \leq i \leq k$. Intuitively, this perturbation favors the clustering \tilde{C}_i .

To derive a contradiction, consider the optimal clustering after perturbation, denoted as $\{C'_i\}$. Since there are more than ϵn bad points in \tilde{B} , by (α, ϵ) -perturbation resilience, not all of them move to new clusters in $\{C'_i\}$, and thus $\{C'_i\}$ is different from $\{\tilde{C}_i\}$. In fact, we will show that the clustering $\{\tilde{C}_i\}$ has a lower cost than $\{C'_i\}$, which is a contradiction. To do so, we consider changing $\{C'_i\}$ to $\{\tilde{C}_i\}$ by moving points. It is sufficient to show that by moving these points, the cost saved is larger than the cost added (summing over all clusters i).

To bound the costs, we first divide the points into different types. See Figure 7 for an illustration. First, we need to move out $C'_i \setminus \tilde{C}_i$ from each C'_i . These points can be divided into three types:

- (1) $U_i = C'_i \cap \hat{B}_i$ are the selected bad points in C_i that need to be moved out.
- (2) $V_i = (C'_i \setminus \tilde{C}_i) \cap (\cup_{j \neq i} \hat{B}_j) = (C'_i \setminus D_i) \cap (\cup_{j \neq i} \hat{B}_j) = (\hat{B} \setminus \hat{B}_i) \cap (C'_i \setminus D_i)$ are the selected bad points that are from other optimal clusters. They are in C'_i but their second nearest cluster is not C_i .
- (3) $W_i = (C'_i \setminus \tilde{C}_i) \cap (\cup_{j \neq i} K_j) = \cup_{j \neq i} (K_j \cap C'_i)$ are points that are from K_j for some $j \neq i$ and are in C'_i . But they are not from K_i and thus are not in \tilde{C}_i .

Second, we need to move in $\tilde{C}_i \setminus C'_i$ for each \tilde{C}_i . Similarly, these points can also be divided into three types:

- (1) $\tilde{W}_i = K_i \setminus C'_i = \cup_{j \neq i} (K_i \cap C'_j)$ are those points in K_i and in C'_j for some $j \neq i$. This means that they are points in $\cup_j W_j$. More specifically, we have $\cup_i \tilde{W}_i = \cup_j W_j$.
- (2) $\tilde{V}_i = (D_i \setminus C'_i) \cap [\cup_{\ell \neq j} (\hat{B}_\ell \cap C'_j)]$ are part of the selected bad points whose second nearest cluster is C_i but not in C'_i . They are originally in \hat{B}_ℓ for some ℓ but are in C'_j for some $j \neq \ell$. In other words, they are points from V_j for

some j , and we have $\cup_i \tilde{V}_i = \cup_j V_j$. Formally,

$$\begin{aligned}\tilde{V}_i &= (D_i \setminus C'_i) \cap \left[\cup_j ((\hat{B} \setminus \hat{B}_j) \cap C'_j) \right] = \cup_{j \neq i} \left[(\hat{B} \setminus \hat{B}_j) \cap C'_j \cap D_i \right], \\ \cup_i \tilde{V}_i &= \cup_j \left[(\hat{B} \setminus \hat{B}_j) \cap C'_j \cap (\cup_{i \neq j} D_i) \right] = \cup_j \left[(\hat{B} \setminus \hat{B}_j) \cap C'_j \setminus D_j \right] = \cup_j V_j.\end{aligned}$$

(3) $\tilde{U}_i = (D_i \setminus C'_i) \cap \left[\cup_j (\hat{B}_j \cap C'_j) \right]$ are also part of the selected bad points whose second nearest cluster is C_i . They are originally in \hat{B}_j for some j and are also in C'_j . In other words, they are points from U_j for some j , and we have $\cup_i \tilde{U}_i = \cup_j U_j$.

In summary, C'_i and \tilde{C}_i are partitioned into four parts respectively:

$$C'_i = C'_i \cap \tilde{C}_i + U_i + V_i + W_i, \quad \tilde{C}_i = C'_i \cap \tilde{C}_i + \tilde{U}_i + \tilde{V}_i + \tilde{W}_i.$$

These different types of points have the following relations:

$$\cup_i \tilde{U}_i = \cup_j U_j, \quad \cup_i \tilde{V}_i = \cup_j V_j, \quad \cup_i \tilde{W}_i = \cup_j W_j.$$

We now consider the costs saved and added when moving these points for all clusters i . Suppose we first move out $\{W_i\}_{i=1}^k$, then $\{V_i\}_{i=1}^k$, and finally $\{U_i\}_{i=1}^k$. The cost saved by moving out $\{W_i\}$ is at least $2 \sum_i d'(W_i, C'_i \cap C_i)$, that saved by moving out $\{V_i\}$ is at least $2 \sum_i d'(V_i, C'_i \cap C_i)$, and that saved by moving out $\{U_i\}$ is at least $2 \sum_i d'(U_i, C'_i \cap K_i)$. Next, we move in $\{\tilde{W}_i\}_{i=1}^k$, then $\{\tilde{V}_i\}_{i=1}^k$, and finally $\{\tilde{U}_i\}_{i=1}^k$. The cost added by moving in $\{\tilde{W}_i\}$ is at most $2 \sum_i d'(\tilde{W}_i, \tilde{W}_i + C'_i \cap \tilde{C}_i)$, that added by moving in $\{\tilde{V}_i\}$ is at most $2 \sum_i d'(\tilde{V}_i, \tilde{C}_i)$, and that added by moving in $\{\tilde{U}_i\}$ is at most $2 \sum_i d'(\tilde{U}_i, \tilde{C}_i)$.

We are now ready to show that the cost saved is greater than the cost added. The high level idea is that a significant amount of cost is saved by moving U_i to the correct clusters, while the costs added by moving V_i and W_i are generally small since the number of points moved is bounded by $3\epsilon n$ and the cost of the selected bad points moved is bounded by $2r$. Formally, we have the following claim, whose proof is presented in Appendix D.1.

CLAIM 6.1. *The costs saved and added by moving $\{U_i\}$, $\{V_i\}$ and $\{W_i\}$ satisfy:*

$$\begin{aligned}(1) \quad & 2 \sum_i d'(U_i, C'_i \cap K_i) - 2 \sum_i d'(\tilde{U}_i, \tilde{C}_i) \\ & \geq \frac{3}{10} \alpha \sum_i d(U_i, C_i) - \frac{2\alpha}{100} \sum_i d(W_i, C_i) - \frac{8\alpha + 16}{100} r\epsilon n, \\ (2) \quad & 2 \sum_i d'(V_i, C'_i \cap C_i) - 2 \sum_i d'(\tilde{V}_i, \tilde{C}_i) \\ & \geq \frac{99}{50} (\alpha - 2) \sum_i d(V_i, C_i) - \frac{2\alpha}{100} \sum_i d(W_i, C_i) - \frac{4\alpha + 8\beta}{100} r\epsilon n, \\ (3) \quad & 2 \sum_i d'(W_i, C'_i \cap C_i) - 2 \sum_i d'(\tilde{W}_i, \tilde{W}_i + C'_i \cap \tilde{C}_i) \\ & \geq \frac{98}{50} (\alpha - 2) \sum_i d(W_i, C_i) - \frac{4\alpha + 4\beta}{100} r\epsilon n.\end{aligned}$$

After adding up all the inequalities in the claim, the right hand side is a lower bound on the difference between the cost saved and the cost added, which we now show must be positive when $\alpha > 4$ and $\beta \leq \frac{4}{5}\alpha$. The terms about $d(W_i, C_i)$ and $d(V_i, C_i)$ are non-negative, so it suffices to show that

$$\frac{3}{10}\alpha \sum_i d(U_i, C_i) - \frac{8\alpha + 16}{100}r\epsilon n - \frac{4\alpha + 8\beta}{100}r\epsilon n - \frac{4\alpha + 4\beta}{100}r\epsilon n > 0.$$

Since $30\alpha > 16\alpha + 12\beta + 16$, what remains is to show $\sum_i d(U_i, C_i) \geq r\epsilon n$. First, $d(p, C_i) \geq r$ for any $p \in U_i$. Second, $|\cup_i U_i| \geq \epsilon n$ since there are $2\epsilon n$ selected bad points but no more than ϵn of them move from their original clusters in $\{C_i\}$ to a different cluster in $\{C'_i\}$. Then we have $\sum_i d(U_i, C_i) \geq \sum_i r|U_i| = r \sum_i |U_i| \geq r\epsilon n$. Hence, the difference between the cost saved and the cost added is positive. This means the cost of $\{\tilde{C}_i\}$ is smaller than the cost of $\{C'_i\}$, which contradicts the assumption that $\{C_i\}$ is the optimal clustering under d' . Therefore, there can be at most $2\eta\epsilon n$ bad points.

Finally, what is left is to show that $\eta \leq 3 \log n$. Suppose p_1 is the point that achieves $\max_i \max_{p \in B_i} d(p, C_i)$ and p_2 is the point that achieves $\min_i \min_{p \in B_i} d(p, C_i)$. Without loss of generality, suppose $p_1 \in C_1$ and $p_2 \in C_2$. By definition of bad points, there exists $C_i \neq C_2$ such that $d(p_2, C_i) \leq \beta d(p_2, C_2)$. If $C_i \neq C_1$, we have $d(p_1, C_1) \leq d(C_2, C_i)$, since otherwise we can get lower cost by splitting C_1 into p_1 and $C_1 \setminus \{p_1\}$ while merging C_2 and C_i . If $C_i = C_1$, we also have $d(p_1, C_1) \leq d(C_2, C_i)$, since otherwise we can get lower cost by splitting C_1 into p_1 and $C_1 \setminus \{p_1\}$ and then merging C_2 and $C_1 \setminus \{p_1\}$. In both cases, we have

$$\begin{aligned} d(p_1, C_1) &\leq d(C_2, C_i) \leq |C_i|d(p_2, C_2) + |C_2|d(p_2, C_i) \\ &\leq |C_i|d(p_2, C_2) + \beta|C_2|d(p_2, C_2) \\ &\leq 8n^2d(p_2, C_2) \end{aligned}$$

where the last inequality follows from $\beta \leq 8n$. Then we have $\eta \leq 3 \log n$. \square

6.1.2. Properties of Actual and Potential Good Points. Since there are just a few bad points and the good points in different clusters are far apart, the cost between sufficiently large subsets of their good points accounts for most of the cost between the two clusters. This means that the min-sum cost can be approximately computed on the good points, and the min-sum clustering can be approximately solved if we knew the actual good points.

LEMMA 6.5. *Suppose $W_i \subseteq G_i, W_j \subseteq G_j$. When $|C_i| \geq 50|C_i \setminus W_i|$ and $|C_j| \geq 50|C_j \setminus W_j|$, we have $d(C_i, C_j) \leq \frac{3}{2}d(W_i, W_j)$.*

Proof Sketch. Since $|W_i| \approx |C_i|$ and $|W_j| \approx |C_j|$, it suffices to show that $d_a(C_i, C_j) \leq d_a(W_i, W_j)$ approximately. By the triangle inequality,

$$d_a(C_i, C_j) \leq d_a(C_i, W_i) + d_a(W_i, W_j) + d_a(W_j, C_j),$$

so we only need to bound $d_a(W_i, C_i)$ and $d_a(W_j, C_j)$ by $d_a(W_i, W_j)$.

By definition of good points, $d_a(W_i, C_i)$ is much less than $d_a(W_i, C_j)$, which is approximately $d_a(W_i, W_j)$ since W_j takes up a majority of points in C_j . A similar argument bounds $d_a(W_j, C_j)$, which then leads to the lemma. The complete proof is provided in Appendix D. \square

Furthermore, the good points in different optimal clusters are far apart in the following sense: the good points from two different clusters have cost much larger than those in a third cluster have. Formally,

LEMMA 6.6. *For any three different optimal clusters C_i, C_j , and C_l , and any $A \subset G_i$, $\frac{18}{5}d(A, G_i \setminus A) < d(G_j, G_l)$. Consequently, $\frac{9}{5}d(G_i, G_i) < d(G_j, G_l)$.*

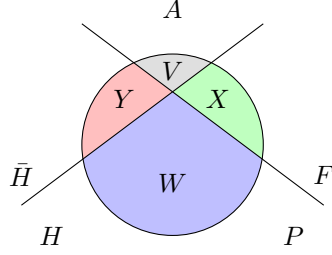


Fig. 8: Notations in Lemma 6.7.

The proof of the lemma is provided in Appendix D. This lemma means that if we can construct a tree on the good points with a pruning that assigns all good points correctly, then this property is useful in finding the pruning.

Now we turn to analyze the potential good points. A key property of the potential good points is the following: for any point p and any sufficiently large set A , the cost between p and the potential good points in A is roughly bounded by the cost between p and any sufficiently large subset H of A . See Lemma 6.7 for details. A specific case is when H is the actual good points in A . In this case, the property says that the cost between p and the potential good points is roughly bounded by the cost between p and the actual good points. This means that in suitable situations, we can regard potential good points as actual good points.

LEMMA 6.7. *Suppose $H \subseteq A$ such that $|A \setminus H| \leq m_B$. Let $F = F(A)$, $P = P(A)$, $\bar{H} = A \setminus H$. Let $W = H \cap P$, $V = \bar{H} \cap F$, $X = F \cap H$, $Y = \bar{H} \cap P$. See Figure 8 for an illustration. If $|A| \geq 20m_B$, then for any p , $d(p, P) \leq \frac{|W|+|Y|}{|W|-|X|}d(p, H)$.*

Proof. Since $d(p, H) = d(p, X) + d(p, W)$ and $d(p, P) = d(p, Y) + d(p, W)$, the lemma is true if $Y = \emptyset$. Otherwise, we need to compare $d(p, X)$ and $d(p, Y)$. By the triangle inequality, we have

$$d_a(W, X) \leq d_a(W, p) + d_a(p, X), \quad d_a(p, Y) \leq d_a(p, W) + d_a(W, Y)$$

which then lead to

$$d(p, X) \geq \frac{d(W, X)}{|W|} - \frac{|X|}{|W|}d(p, W), \quad d(p, Y) \leq \frac{d(W, Y)}{|W|} + \frac{|Y|}{|W|}d(p, W).$$

From these bounds on $d(p, X)$ and $d(p, Y)$, we have

$$d(p, H) \geq \frac{d(W, X)}{|W|} + \frac{|W| - |X|}{|W|}d(p, W), \quad d(p, P) \leq \frac{d(W, Y)}{|W|} + \frac{|W| + |Y|}{|W|}d(p, W).$$

The lemma then follows from these two inequalities and the following claim.

CLAIM 6.2. $d(X, W) \geq d(Y, W + Y)$.

Intuitively, the points in $X \subseteq F$ are among those farthest away from A , so $d_a(X, A) \geq d_a(Y, A)$. Since $|A \setminus H| \leq m_B$ and $|F| = 2m_B$, $|X| \geq 2|Y|$. Then the cost of Y cannot be too large compared to that of X . The complete proof of the claim is provided in Appendix D. \square

Algorithm 7 Robust Average Linkage

Input: Data set S , distance function $d(\cdot, \cdot)$ on S , $\min_i |C_i|$, $\epsilon > 0$.

- 1: Use Algorithm 3 with $m_B = 6\epsilon n \log n$ and $\gamma = \frac{4}{5}$ to get a list \mathcal{L}_0 of blobs.
- 2: Initialize the clustering \mathcal{L} with each blob being a cluster.
- 3: Repeat till only one cluster remains:
 - merge clusters C, C' which minimize $d_{\text{ra}}(C, C')$.
- 4: Let \mathcal{T} be the tree with blobs as leaves and internal nodes corresponding to the merges performed.

Output: The tree \mathcal{T} .

6.2. Approximating the Optimal Clustering. In this subsection, we design an approximation algorithm and prove our final result Theorem 6.1 by utilizing the properties of the (α, ϵ) -perturbation resilience. Note that we can generate a list of sufficiently large almost “pure” blobs using Algorithm 3. However, unlike for (α, ϵ) -perturbation resilient k -median instances, it is not guaranteed that the robust linkage procedure in [7] can link these blobs into a tree so that a pruning of the tree assigns all but bad points correctly. In Section 6.2.1, we design a robust average linkage algorithm to achieve this goal, and in Section 6.2.2, we show that this pruning can be found in polynomial time. Although this pruning may not be a good approximation to the optimum, in Section 6.2.3 we show that by reassigning points in it, a good approximation can be computed, which leads to Theorem 6.1.

6.2.1. Constructing A Tree with A Pruning Close to the Optimum.

Here we show that by utilizing the bound on the number of bad points, we can construct a tree with a pruning that assigns all good points correctly. As described in Algorithm 7, we first use Algorithm 3 to generate a list of blobs, and then use a robust version of average linkage to link them into a tree: repeatedly merge the two blobs with the minimum robust average distance defined as follows.

DEFINITION 6.8. *The robust average distance $d_{\text{ra}}(A_1, A_2)$ between two sets A_1, A_2 is defined as the average distance between their potential good points. Formally,*

$$d_{\text{ra}}(A_1, A_2) := \frac{d(P(A_1), P(A_2))}{|P(A_1)||P(A_2)|}.$$

We now prove that the tree output by Algorithm 7 has a pruning that correctly assigns all good points.

LEMMA 6.9. *The tree output in Algorithm 7 has a pruning \mathcal{C}' that assigns all good points correctly.*

Proof. To analyze the algorithm, we begin with the following property of good points. When combined with the property of Algorithm 3 (Lemma 4.4), it immediately shows that each blob in the list \mathcal{L}_0 has size at least $\frac{1}{2} \min_i |C_i|$, and contains good points from only one optimal cluster.

CLAIM 6.3. *For any $p \in G_i$, all its $\frac{4|C_i|}{5}$ nearest neighbors belong to $C_i \cup B$.*

Proof. We need to show that for any $j \neq i$ and any good point $q \in G_j$, $d(p, q)$ is sufficiently large compared to $d_a(p, C_i)$. Intuitively, p is much farther away from C_j than from C_i , that is, $\beta d(p, C_i) \leq d(p, C_j)$. It suffices to bound $d(p, C_j)$ by $d(p, q)$ and $d(p, C_i)$. By the triangle inequality,

$$d(p, C_j) \leq |C_j|d(p, q) + d(q, C_j) \quad \text{and} \quad d(q, C_j) \leq \frac{1}{\beta}d(q, C_i) \leq \frac{|C_i|}{\beta}d(p, q) + \frac{1}{\beta}d(p, C_i).$$

Combining these inequalities, we have $(\beta - \frac{1}{\beta})d(p, C_i) \leq (|C_j| + \frac{|C_i|}{\beta})d(p, q)$. When $\alpha > 8 \frac{\max_i |C_i|}{\min_i |C_i|}$, $5d_a(p, C_i) < d(p, q)$, which then leads to the conclusion. \square

It now suffices to prove by induction that the clustering $\mathcal{L} \cap G$ is always laminar to $\mathcal{C} \cap G$. It is true at the beginning by the property of Algorithm 3. Assume for contradiction that the laminarity is first violated after merging A and D . There are two cases:

- (1) A and D are strict subsets of different optimal clusters;
- (2) A is a strict subset of G_i while D is the union of the good points in several optimal clusters.

We have the following claims for the two cases respectively.

CLAIM 6.4.

- (1) Suppose $A \in \mathcal{L}, A \cap G \subsetneq G_i$, and $D \in \mathcal{L}, D \cap G \subsetneq G_j (j \neq i)$. Then there exists $A' \neq A$ in \mathcal{L} such that $A' \cap G \subsetneq G_i$ and $d_{\text{ra}}(A, A') < d_{\text{ra}}(A, D)$.
- (2) Suppose $A \in \mathcal{L}, A \cap G \subsetneq G_i$, and $D \in \mathcal{L}, D \cap G$ is the union of good points in several optimal clusters. Then there exists $A' \neq A$ in \mathcal{L} such that $A' \cap G \subsetneq G_i$ and $d_{\text{ra}}(A, A') < d_{\text{ra}}(A, D)$.

The intuition is that since good points in different optimal clusters are far away, we can find a blob A' from $C_i \cup B$ such that good points in A' have smaller average distance to good points in A than those in D have, that is,

$$(6.1) \quad d_a(A \cap G, A' \cap G) < \frac{1}{2}d_a(A \cap G, D \cap G).$$

Then by using Lemma 6.7, we show that the potential good points approximates the actual good points: $d_{\text{ra}}(A, A')$ is approximately less than $d_a(A \cap G, A' \cap G)$, and $d_{\text{ra}}(A, D)$ is approximately greater than $d_a(A \cap G, D \cap G)$. These together with Inequality 6.1 lead to the claims. Their complete proofs are presented in Appendix E.1.

By these two claims, we should first merge A with A' rather than with D , which is contradictory. So the laminarity is always preserved, which completes the proof. \square

6.2.2. Getting A Pruning Close to the Optimal Clustering. We have shown that a tree can be constructed such that there is a pruning, denoted as \mathcal{C}' , that assigns all good points correctly. Here we show how to find this pruning from the tree. Suppose we can remove the actual bad points and compute the cost between the good points. Since the good points from different clusters are far apart, the good point cost increased by joining different clusters in \mathcal{C}' is larger than that saved by splitting clusters in \mathcal{C}' (Lemma 6.6). Then any other pruning has larger cost than \mathcal{C}' . Unfortunately, we do not know the actual good points. Therefore, we consider the potential good points and compute the robust min-sum cost. It turns out that \mathcal{C}' indeed is the pruning with the minimum robust min-sum cost.

LEMMA 6.10. *Suppose the pruning $\mathcal{C}' = \{C'_1, \dots, C'_k\}$ in tree \mathcal{T} assigns all good points correctly. Then \mathcal{C}' is the minimum robust min-sum cost pruning in the tree.*

Proof Sketch. Computing the robust min-sum cost will eliminate the effect of the bad points and work as if we knew the actual good points: the robust min-sum cost saved by splitting a node is at most the good point cost saved (Claim 6.5), and the robust min-sum cost increased by merging two nodes is in the same order of the good point cost increased (Claim 6.6).

CLAIM 6.5. *If $|C'_i| \geq 20m_B$, then $d_{\text{rs}}(C'_i) \leq d(G_i, G_i)$.*

Proof. The claim follows from Claim 6.2 (See Figure 8 for an illustration of the notations) by setting $A = C'_i$ and $H = G_i$. In particular, we have $d_{\text{rs}}(A) =$

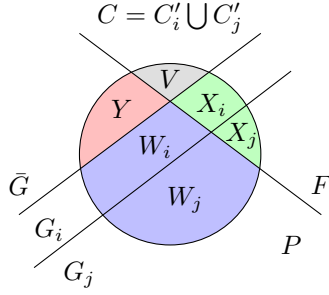


Fig. 9: Notations in Claim 6.6.

$d(P, P) \leq d(W, W) + 2d(Y, W + Y)$ and $d(H, H) \geq d(W, W) + 2d(X, W)$. By Claim 6.2, $d(Y, W + Y) \leq d(X, W)$, which completes the proof. \square

CLAIM 6.6. For $t \in \{i, j\}$, $|C_t| \geq 100m_B$, and C'_t contains all good points in C_t but no good points in other optimal clusters. Then $d_{\text{rs}}(C'_i \cup C'_j) - d(G_i, G_i) - d(G_j, G_j) \geq (\frac{4}{3} - \frac{4}{\beta})d(G_i, G_j)$.

Proof. Let $C = C'_i \cup C'_j$, $F = F(C)$, $P = P(C)$, and $G = G_i + G_j$, $\bar{G} = C \setminus G$. Define $W_i = G_i \cap P$, $X_i = G_i \cap F$; define W_j, X_j similarly. Also, define $Y = P \cap \bar{G}$, $V = F \cap \bar{G}$. See Figure 9. Then the left hand side of the statement is

$$(6.2) \quad d(P, P) - d(G_i, G_i) - d(G_j, G_j) \geq 2[d(W_i, W_j) - d(X_i, G_i) - d(X_j, G_j)].$$

By Lemma 6.5, we have $d(C_i, C_j) \leq \frac{3}{2}d(W_i, W_j)$. By the definition of good points,

$$d(X_i, C_i) + d(X_j, C_j) \leq \frac{d(X_i, C_j) + d(X_j, C_i)}{\beta} \leq \frac{2}{\beta}d(C_i, C_j).$$

Plugging these into (6.2), we have that the left hand side of the statement is at least $(\frac{4}{3} - \frac{4}{\beta})d(C_i, C_j) \geq (\frac{4}{3} - \frac{4}{\beta})d(G_i, G_j)$. \square

By these claims and Lemma 6.6, when splitting some node in the pruning while merging some other two, the cost increased is larger than the cost saved. Since any pruning of size k can be obtained from C' by splitting some nodes while merging some others, we can show that C' has the minimum robust min-sum cost. See Appendix E.2 for the complete proof. \square

6.2.3. Getting a Good Approximation. We have showed that the pruning C' that assigns all good points correctly has the minimum robust min-sum cost, so we can use dynamic programming on the tree to get the pruning. However, this pruning may not be a good approximation. For example, consider an instance consisting of two unbalanced clusters. Assume that there is only one bad point, belonging to the small cluster. Further assume the distances between the good points in each cluster are negligible, then assigning the bad point incorrectly to the large cluster will lead to an $O\left(\frac{\max_i |C_i|}{\min_i |C_i|}\right)$ -approximation. So the pruning C' may not be a constant approximation.

Here we show that a constant factor approximation can be computed by reassigning the points in C' . As described in Algorithm 8, we reassign each point p to the index i that minimizes the cost between p and the potential good points in the cluster C'_i . To analyze Algorithm 8, we first prove that after reassignment all good points are still assigned correctly (Lemma 6.11), and then bound the cost.

LEMMA 6.11. For any $p \in G_i$, any $j \neq i$, $d(p, P(C'_j)) > d(p, P(C'_i))$.

Algorithm 8 Getting a good approximation

Input: A clustering $\mathcal{C}' = \{C'_1, \dots, C'_k\}$, where $G_i \subseteq C'_i \subseteq C_i \cup B$.

- 1: **for** each point p **do**
- 2: Assign p to the index i such that $d(p, P(C'_i))$ is minimized.
- 3: **end for**
- 4: Let C''_i be the set of points assigned to the index i .

Output: The clustering $\mathcal{C}'' = \{C''_1, \dots, C''_k\}$.

Proof Sketch. By Lemma 6.7, $d(p, P(C'_i)) \approx d(p, C_i)$. By definition of good points, $\beta d(p, C_i) \leq d(p, C_j)$, so it suffices to show that $d(p, P(C'_j))$ is not so small compared to $d(p, C_j)$. Let $W_j = G_j \cap P(C'_j)$ denote the good points that are also potential good points, and let $Z_j = C_j \setminus W_j$ denote all other points in C_j . Since $W_j \subseteq P(C'_j)$, we only need to prove that $d(p, W_j)$ is large compared to $d(p, Z_j)$. Intuitively, this is true since p is far from good points in G_j , and W_j contains most of G_j . See Appendix E.3 for the complete proof. \square

We are now ready to prove our final result.

Proof of Theorem 6.1. By Lemma 6.11, all the good points in C_i are assigned correctly to C''_i . Let $A_i = C''_i \setminus G_i$ denote all the bad points assigned to C''_i . The cost of the output clustering \mathcal{C}'' can be written as follows.

$$(6.3) \quad \begin{aligned} \sum_i d(C''_i, C''_i) &= \sum_i d(G_i + A_i, G_i + A_i) \\ &= \sum_i d(G_i, G_i) + 2 \sum_i d(G_i, A_i) + \sum_i d(A_i, A_i). \end{aligned}$$

We need to bound the last two terms.

Let $r = \frac{\min_i |C_i|}{m_B}$. By the triangle inequality, we have $d_a(A_i, A_i) \leq 2d_a(A_i, G_i)$, leading to

$$(6.4) \quad d(A_i, A_i) \leq \frac{2|A_i|}{|G_i|} d(A_i, G_i) \leq \frac{2m_B}{|C_i| - m_B} d(A_i, G_i) \leq \frac{2}{r-5} d(A_i, G_i).$$

So it suffices to bound $d(A_i, G_i)$. In Appendix E.3, we prove

$$\text{CLAIM 6.7. } \sum_i d(A_i, G_i) \leq \frac{r^2}{(r-5)^2} \sum_i d(C_i, C_i) - \frac{r^2-1}{(r-5)^2} \sum_i d(G_i, G_i).$$

Proof Sketch. The intuition is as follows. Suppose $p \in A_i$ comes from C_j . Then $d(p, G_i) \approx d(p, P(C'_i))$, which is smaller than $d(p, P(C'_j))$ since p is assigned to i . By Lemma 6.7, we have $d(p, P(C'_j)) \approx d(p, G_j)$. Applying this argument for all $p \in A_i$ and all i , and noting that $\cup_i A_i = \cup_j B_j$, we have $\sum_i d(A_i, G_i)$ is approximately less than $\sum_j d(B_j, G_j) \leq \sum_j [d(C_j, C_j) - d(G_j, G_j)]$. \square

The proof of correctness is completed by combining Claim 6.7 and (6.3)(6.4).

Running Time. Algorithm 3 takes time $O(n^{\omega+1})$ (as shown in the proof of Theorem 4.6), and the rest steps of Algorithm 7 take time $O(n^3)$. Finding the minimum robust min-sum cost pruning in the tree output by Algorithm 7 takes time $O(n^3)$, and Algorithm 8 takes time $O(n^3)$. So the total running time is $O(n^{\omega+1})$. \square

7. Discussion and Open Questions. We advance the line of research on clustering under perturbation resilience in multiple ways. For α -perturbation resilient instances, we improve on the known guarantees for center-based objectives and give the first analysis for min-sum. Furthermore, for k -median and min-sum, we analyze

and give the first algorithmic guarantees known for a relaxed but more challenging condition of (α, ϵ) -perturbation resilience, where an ϵ fraction of points are allowed to move after perturbation. We also give sublinear-time algorithms for k -median and min-sum under perturbation resilience.

A natural direction for future investigation is to explore whether one can take advantage of smaller perturbation factors for perturbation resilient instances in Euclidian spaces³. More broadly, it would be interesting to explore other ways in which perturbation resilient instances behave better than worst case instances (e.g., natural algorithms converge faster).

Another interesting direction is to design private clustering algorithms under perturbation resilience. Some stability notions are useful for differential private analysis [24, 15]. It would be interesting to explore the perturbation resilience property and design efficient and private clustering algorithms.

Acknowledgments. We thank Avrim Blum for numerous useful discussions. This work was supported in part by NSF grants CCF-0953192 and CCF-1101283, ONR N00014-09-1-0751, AFOSR grant FA9550-09-1-0538, a Microsoft Faculty Fellowship, and a Google Research Award.

REFERENCES

- [1] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k -median and facility location problems. *SIAM Journal of Computing*, 2004.
- [2] P. Awasthi, A. Blum, and O. Sheffet. Stability yields a ptas for k -median and k -means clustering. In *Proceedings of the IEEE Annual Symposium on Foundations of Computer Science*, 2010.
- [3] P. Awasthi, A. Blum, and O. Sheffet. Center-based clustering under perturbation stability. *Information Processing Letters*, 2012.
- [4] P. Awasthi and O. Sheffet. Improved spectral-norm bounds for clustering. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*. 2012.
- [5] M.-F. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *Journal of ACM*, 2013.
- [6] M. F. Balcan and M. Braverman. Approximate nash equilibria under stability conditions. *CoRR*, abs/1008.1827, 2010.
- [7] M. F. Balcan and P. Gupta. Robust hierarchical clustering. In *Proceedings of the Annual Conference on Learning Theory*, 2010.
- [8] Y. Bartal, M. Charikar, and D. Raz. Approximating min-sum k -clustering in metric spaces. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2001.
- [9] L. Ben-David, Shalev and Reyzin. Data stability in clustering: A closer look. In *To appear in the ALT 2012 Special Issue of Theoretical Computer Science*, 2014.
- [10] J. Bien and R. Tibshirani. Hierarchical clustering with prototypes via minimax linkage. *Journal of the American Statistical Association*, 106(495), 2011.
- [11] Y. Bilu, A. Daniely, N. Linial, and M. Saks. On the practically interesting instances of max-cut. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science*, 2013.
- [12] Y. Bilu and N. Linial. Are stable instances easy? In *Proceedings of the Symposium on Innovations in Computer Science*, 2010.
- [13] M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k -median problem. *Journal of Computer and System Sciences*, 2002.
- [14] W. F. de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani. Approximation schemes for clustering problems. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2003.

³That is, where d is a Euclidean metric, though as in Definitions 2.1 and 2.4, d' need not be. Alternatively, one could also consider a natural version of Definitions 2.1 and 2.4 in which d' must be Euclidean as well, and in fact implemented via a perturbation of coordinate values.

- [15] C. Dwork and J. Lei. Differential privacy and robust statistics. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2009.
- [16] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 1999.
- [17] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2002.
- [18] A. Kumar and R. Kannan. Clustering with spectral norm and the k-means algorithm. In *Proceedings of the IEEE Annual Symposium on Foundations of Computer Science*, 2010.
- [19] S. Li and O. Svensson. Approximating k-median via pseudo-approximation. In *Proceedings of the ACM Symposium on the Theory of Computing*, 2013.
- [20] R. J. Lipton, E. Markakis, and A. Mehta. On stability properties of economic solution concepts. Manuscript, 2006.
- [21] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Bilu-linial stable instances of max cut and minimum multiway cut. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2014.
- [22] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Constant factor approximation for balanced cut in the pie model. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2014.
- [23] M. Mihalák, M. Schöngens, R. Šrámek, and P. Widmayer. On the complexity of the metric tsp under stability considerations. In *Proceedings of the International Conference on Current Trends in Theory and Practice of Computer Science*, 2011.
- [24] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2007.
- [25] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science*, 2006.
- [26] B. Schölkopf and A. J. Smola. *Learning with Kernels*. 2002.
- [27] V. N. Vapnik. *Statistical learning theory*. 1998.

Appendix A. Finding the Minimum Cost k -Cluster Pruning.

The idea of using dynamic programming to find the optimal k -clustering in a tree of clusters is proposed in [3]. We can find the optimal clustering by examining the entire tree of clusters produced. Denote the cost of the optimal m -clustering of a tree node p as $\text{cost}(p, m)$. The optimal m -clustering of a tree node p is either the entire subtree as one cluster ($m = 1$), or the minimum over all choices of m_1 -clustering over its left subtree and m_2 -clustering over its right subtree ($1 < m \leq k$), where m_1, m_2 are positive integers such that $m_1 + m_2 = m$. Therefore, we can traverse the tree bottom up, recursively solving the m -clustering problem for $1 \leq m \leq k$ for each tree node. The algorithm is presented in Algorithm 9. Suppose that computing the cost of a cluster takes time $O(t)$ ($O(n^2)$ for k -median, k -means and min-sum). Since there are $O(n)$ nodes, and on each node p , computing $\text{cost}(p, 1)$ takes $O(t)$ time, computing $\text{cost}(p, m)$ ($1 < m \leq k$) takes $O(k^2)$, in total the algorithm takes time $O(nt + nk^2)$.

Note that when \mathcal{T} is a multi-branch tree and not suitable for dynamic programming, we need to turn it into a 2-branch tree \mathcal{T}' as follows. For each node with more than 2 children, for example, the node R with children R_1, R_2, \dots, R_t ($t > 2$), we first merge R_1 and R_2 into one node, then merge this node with R_3 ; repeat until we merge all nodes R_1, R_2, \dots, R_t into R . In this way, we get a 2-branch tree \mathcal{T}' and can run dynamic programming on it. Note that each pruning in \mathcal{T} has a corresponding pruning in \mathcal{T}' , so the minimum cost pruning of \mathcal{T}' has no greater cost than the minimum cost pruning of \mathcal{T} . Also note that when the cost function is center-based, such as k -median, the algorithm essentially computes a center for the node p when computing $\text{cost}(p, 1)$. So it can output the centers together with the pruning.

Appendix B. An Efficient Implementation of Algorithm 1.

Here we show an efficient implementation of Algorithm 1, namely Algorithm 10. This implementation takes time only $O(n^3)$.

Algorithm 9 Dynamic Programming in Tree of Clusters

Input: A tree of clusters \mathcal{T} on a data set S , distance function $d(\cdot, \cdot)$ on S , k .

- 1: Traverse \mathcal{T} bottom up.
- 2: **for** each node $R \in \mathcal{T}$ **do**
- 3: Calculate $\text{cost}(R, 1)$. For $1 < m \leq k$, calculate $\text{cost}(R, m)$ as follows.
- 4: **if** R is a leaf **then**
- 5: $\text{cost}(R, m) = \text{cost}(R, 1)$.
- 6: **else**
- 7: $\text{cost}(R, m) = \min[\text{cost}(R_1, m_1) + \text{cost}(R_2, m_2)]$, where R_1, R_2 are R 's children and $m_1 + m_2 = m$.
- 8: **end if**
- 9: **end for**
- 10: Traverse backwards to get the k -clustering \mathcal{C} that achieves $\text{cost}(r, k)$ where r is the root.

Output: The k -clustering \mathcal{C} .

Algorithm 10 Efficient Implementation of Algorithm 1

Input: Data set S , distance function $d(\cdot, \cdot)$ on S .

- 1: Sort all the pairwise distances in ascending order.
- 2: **for** each $p \in S$ and $1 \leq i \leq n$ **do**
- 3: Compute L^p , $\chi(p, i)$. Then compute $\chi^*(p, i)$ by Equation (B.1).
- 4: **end for**
- 5: Let the current clustering be n singleton clusters.
- 6: **for** $d(p, q)$ in ascending order **do**
- 7: Suppose $q = L_i^p$. Check if $d(p, q)$ satisfies the three claims in Fact B.2, where the third claim can be checked by verifying if $\chi^*(p, i) = -1$.
- 8: If so, merge all the clusters covered by $\mathbb{B}(p, d(p, q))$.
- 9: **end for**
- 10: Construct the tree \mathcal{T} with points as leaves and internal nodes corresponding to the merges performed.
- 11: Run dynamic programming on \mathcal{T} to get the minimum cost pruning $\tilde{\mathcal{C}}$.

Output: The clustering $\tilde{\mathcal{C}}$.

Note that at each merge step in Algorithm 1, we only need to find the two clusters with the minimum closure distance. So we hope to compute the minimum closure distance without computing all the distances between any two current clusters. First we notice the following facts.

FACT B.1. *In the execution of Algorithm 1, if d is the minimum closure distance for the current clustering, then*

- (1) *there exist $c, p \in S$ such that $d = d(c, p)$;*
- (2) *d is no less than the minimum closure distances in previous clusterings.*

Proof. For the first claim, let c be the center of the ball in the definition of closure distance, and p be the farthest point from the center in the ball, then $d = d(c, p)$. The second claim comes from the fact that the clusters in the current clustering are supersets of those in previous clusterings. \square

Fact B.1 implies that we can check in ascending order the pairwise distances no less than the minimum closure distance in the last clustering, and determine if the

checked pairwise distance is the minimum closure distance in the current clustering. More specifically, suppose we have some black-box method for checking if a pairwise distance is the minimum closure distance in the current clustering, we can perform the closure linkage as follows: sort the pairwise distances in a list in ascending order; start from the first distance in the list; check if the current distance is the minimum closure distance in the current clustering; if it is, merge clusters covered by the ball defined by the checked distance; continue to check the next distance in the list. So it is sufficient to design a method to determine if a pairwise distance is the minimum closure distance in the current clustering. Our method is based on the following facts.

FACT B.2. *In Algorithm 1, if $d(c, p)$ is the minimum closure distance for the current clustering, then*

- (1) *at least 2 clusters intersects $\mathbb{B}(c, d(c, p))$;*
- (2) *all the clusters intersecting $\mathbb{B}(c, d(c, p))$ are covered by $\mathbb{B}(c, d(c, p))$;*
- (3) *for any $p' \in \mathbb{B}(c, d(c, p))$, $q \notin \mathbb{B}(c, d(c, p))$, $d(c, p') < d(p', q)$.*

Proof. The first claim and the third claim follow from the definition. We can prove the second claim by induction. This is trivial at the beginning. Suppose it is true up to any previous clustering, we prove it for the current clustering C' . We need to show that for any $C' \in \mathcal{C}'$ such that $C' \cap \mathbb{B}(c, d(c, p)) \neq \emptyset$, $C' \subseteq \mathbb{B}(c, d(c, p))$. If $c \in C'$, then by definition, $C' \subseteq \mathbb{B}(c, d(c, p))$. If C' is a single point set $\{c_1\}$, then trivially $C' \subseteq \mathbb{B}(c, d(c, p))$. What is left is the case when $c \notin C'$ and C' is generated by merging clusters in a previous step. Suppose when C' is formed, the closure distance between those clusters is defined by $c_1 \in C'$ and p_1 . By induction, if $c \in \mathbb{B}(c_1, d(c_1, p_1))$, c would have been merged into C' when C' is merged, which is contradictory to $c \notin C'$. So we have $c \notin \mathbb{B}(c_1, d(c_1, p_1))$, that is, $d(c, c_1) > d(c_1, p_1)$. Then by the margin requirement of $\mathbb{B}(c_1, d(c_1, p_1))$, $d(c, q) > d(c_1, q)$ for any $q \in \mathbb{B}(c, d(c, p)) \cap C'$. This further leads to $c_1 \in \mathbb{B}(c, d(c, p))$, since otherwise by the margin requirement of $\mathbb{B}(c, d(c, p))$ and $q \in \mathbb{B}(c, d(c, p))$, we would have $d(c, q) < d(c_1, q)$. So for any point $q' \in C'$, since $d(c_1, q') \leq d(c_1, p_1) < d(c, c_1)$, we have $q' \in \mathbb{B}(c, d(c, p))$ from the margin requirement, so $C' \subseteq \mathbb{B}(c, d(c, p))$. \square

Notice if a pairwise distance satisfies the three claims, then it defines a closure distance for the clusters covered. So if we check the pairwise distances in ascending order, then the first one that satisfies the three claims must be the minimum closure distance in the current clustering. So we have a method to determine if a pairwise distance is the minimum closure distance.

However, naively checking the third claim in Fact B.2 takes $O(n^2)$, which is still not good enough. We can refine this step since intuitively, for every c , if $d(c, p)$ comes after $d(c, p)$ in the distance list, then when checking $d(c, q)$, we can utilize the information obtained from checking $d(c, p)$. Specifically, for every $p \in S$, define $L^p = (L_1^p, \dots, L_n^p)$ to be a sorted list of points in S , according to their distances to p in ascending order. Let $\chi^*(p, i)$ denote the index of the farthest point in L^p , which makes $d(p, L_i^p)$ fail the third claim in Fact B.2. Formally, define $\chi^*(p, i)$ to be the maximum $j > i$ such that there exists $s \leq i$ satisfying $d(p, L_s^p) \geq d(L_s^p, L_j^p)$. If no such point L_j^p exists, let $\chi^*(p, i) = -1$. Then $d(p, L_i^p)$ satisfies the third claim if and only if $\chi^*(p, i) = -1$, thus we turn the task of checking the claim into computing $\chi^*(p, i)$. In order to use the information obtained when previously checking $d(p, L_{i-1}^p)$, we compute $\chi^*(p, i)$ from $\chi^*(p, i-1)$. By the definition of χ^* , what is new of $\chi^*(p, i)$ compared to $\chi^*(p, i-1)$ is just the maximum $j > i$ such that $d(p, L_i^p) \geq d(L_i^p, L_j^p)$. Define $\chi(p, i)$ to be the maximum $j > i$ such that $d(p, L_i^p) \geq d(L_i^p, L_j^p)$; if no such j

exists, let $\chi(p, i) = -1$. Then it is easy to verify that

$$(B.1) \quad \chi^*(p, i) = \max\{\chi^*(p, i-1), \chi(p, i)\}.$$

It takes $O(n)$ time to compute $\chi(p, i)$, thus we can compute $\chi^*(p, i)$ for all $p \in S, 1 \leq i \leq n$ in $O(n^3)$ time. The implementation is finally summarized in Algorithm 10.

Appendix C. Bounding the Number of Bad Points for k -Median.

Here we present the complete proof. Recall the definitions of the selected bad points and the constructed perturbation in Section 4.1 and recall the following notations. $\{C'_i\}$ denote the new optimal clustering after perturbation, where C'_i corresponds to C_i . Let $A_i = C'_i \setminus C_i$ (the points added when switching from C_i to C'_i), $M_i = C_i \setminus C'_i$ (the points missing), $W_i = (C_i \cap C'_i) \setminus \hat{B}_i$ (the common core), and $V_i = (C_i \cap C'_i) \cap \hat{B}_i$ (the selected bad points in common). So, $C_i = W_i + V_i + M_i$ and $C'_i = W_i + V_i + A_i$. See Figure 3 in Section 4.1 for an illustration. Note that $|A_i| \leq \epsilon n$, $|M_i| \leq \epsilon n$, and $|V_i| \leq \epsilon n + 1$, with the bulk of the points in W_i .

We begin with the proof of the fact that $c'_i \neq c_j$ for $j \neq i$.

FACT 4.1. *If $\min_i |C_i| > (\frac{2}{\alpha-1} + 3)\epsilon n + 1$, then $c'_i \neq c_j (\forall j \neq i)$.*

Proof. The intuition is that if $c'_i = c_j$, then under d' , points in W_j should be closer to c'_j than to $c'_i = c_j$. So under d , these points are α time closer to c'_j than to c_j . This means that the distance between c_j and c'_j is no so large compared to the average distance between c_j and W_j by the triangle inequality. On the other hand, this also means that c_j has $(1 - 1/\alpha)d(c_j, W_j)$ more cost than c'_j on W_j . Then c_j should save this cost on other parts of C_j . In other words, c'_j has at least $(1 - 1/\alpha)d(c_j, W_j)$ more cost on these points than c_j has. By the triangle inequality, the distance between c_j and c'_j is much larger than the average distance between c_j and W_j , which is contradictory.

Formally, assume for contradiction that $c'_i = c_j$. To apply the intuition described above, we first need to show $c'_j \neq c_l (\forall l)$. Clearly, $c'_j \neq c_j$, since otherwise, moving all the points in C'_j to C_j will not increase the cost, which violates (α, ϵ) -perturbation resilience. We also know that $c'_j \neq c_l (l \neq j)$ since otherwise, there is $p \in W_j$, $d(c_l, p) = d(c'_j, p) \leq d'(c'_j, p) < d'(c'_i, p) = d(c_j, p)$, which contradicts the fact that $p \in C_j$.

Now we can apply the intuition described above to show that $c'_i = c_j$ and $c'_j \neq c_l (\forall l)$ lead to an contradiction. Note that points in $W_j \cup V_j = C_j \cap C'_j$ are closer to c'_j than to $c'_i = c_j$ under d' . Then back to d , for any $p \in W_j$, since $c'_j \neq c_l (\forall l)$, $\alpha d(c'_j, p) = d'(c'_j, p) \leq d'(c'_i, p) = d(c_j, p)$, resulting in $d(c'_j, W_j) \leq d(c_j, W_j)/\alpha$. Similarly, for any $p \in V_j$, $\alpha d(c'_j, p) = d'(c'_j, p) \leq d'(c'_i, p) = \alpha d(c_j, p)$, resulting in $d(c'_j, V_j) \leq d(c_j, V_j)$. These facts have two consequences.

First, since points in W_j are α time closer to c'_j than to c_j , the distance between c'_j and c_j is small:

$$(C.1) \quad d(c'_j, c_j) \leq \frac{d(c'_j, W_j)}{|W_j|} + \frac{d(c_j, W_j)}{|W_j|} \leq (1 + \frac{1}{\alpha})d(c_j, W_j).$$

Second, since c_j is the optimal center for $C_j = W_j \cup V_j \cup M_j$, it should save a lot of cost on M_j compared to c'_j , which suggests c_j and c'_j would be far apart. Formally,

$$d(c'_j, C_j) = d(c'_j, W_j \cup V_j \cup M_j) \geq d(c_j, C_j) = d(c_j, W_j \cup V_j \cup M_j).$$

Since $d(c'_j, W_j) \leq d(c_j, W_j)/\alpha$ and $d(c'_j, V_j) \leq d(c_j, V_j)$, we have

$$(C.2) \quad \begin{aligned} d(c'_j, M_j) - d(c_j, M_j) &\geq d(c_j, W_j) - \frac{1}{\alpha}d(c_j, W_j), \\ |M_i|d(c'_j, c_j) &\geq (1 - \frac{1}{\alpha})d(c_j, W_j). \end{aligned}$$

When $|C_j| > (\frac{2}{\alpha-1} + 3)\epsilon n + 1$, we have $(1 - 1/\alpha)|W_j| > (1 + 1/\alpha)|M_j|$. Then Inequalities C.2 and C.1 lead to $d(c_j, c'_j) = 0$. This means $c_j = c'_j$ which is a contradiction to the assumptions. \square

We next present a fact that is useful in the proofs of the two claims.

FACT C.1. *Suppose $\min_i |C_i| > (\frac{2}{\alpha-1} + 3)\epsilon n + 1$. If $c_i \neq c'_i$, then we have*

- (1) $d'(c'_i, A_i) \geq \alpha d(c'_i, A_i \setminus \{c(c'_i)\})$,
- (2) $d'(c_i, A_i) \leq \alpha d(c_i, A_i \setminus \{c(c'_i)\}) + \alpha(1 + \alpha)d(c'_i, c_i)$.

Proof. These translations from d' to d can be verified by the definition of d' . In most cases, $d'(\cdot, \cdot) = \alpha d(\cdot, \cdot)$; the only exceptions are the distances between p and $c(p)$. The detailed verification is presented below.

(1) Since $c'_i \neq c_i$, and by Fact 4.1, we know $c'_i \neq c_j (\forall j)$. So we only need to check if $c(c'_i) \in A_i$. We have

$$d'(c'_i, A_i) \geq d'(c'_i, A_i \setminus \{c(c'_i)\}) = \alpha d(c'_i, A_i \setminus \{c(c'_i)\}).$$

(2) If $c(c'_i) \notin A_i$, then the inequality is trivial. If $c(c'_i) \in A_i$, then

$$d'(c_i, A_i) = d'(c_i, A_i \setminus \{c(c'_i)\}) + d'(c_i, c(c'_i)) \leq \alpha d(c_i, A_i \setminus \{c(c'_i)\}) + \alpha d(c_i, c(c'_i)).$$

We have $d(c_i, c(c'_i)) \leq d(c_i, c'_i) + d(c'_i, c(c'_i))$. If c'_i is a selected bad point, then $d(c'_i, c(c'_i)) \leq \alpha d(c'_i, c_i)$. Otherwise, $c(c'_i)$ is the nearest center for c'_i , then $d(c'_i, c(c'_i)) \leq d(c'_i, c_i)$. In any case, the inequality for $d'(c_i, A_i)$ follows. \square

We are now ready to present the complete proofs for the two claims. These proofs are almost the same as the proofs under the assumption $c(c'_i) \notin A_i$, except that when translating $d'(c_i, A_i)$ and $d'(c'_i, A_i)$ to costs under d , we use Fact C.1 instead of the assumption $c(c'_i) \notin A_i$.

CLAIM 4.1. *For each i , $d(c_i, c'_i) \leq 3 \frac{(\alpha+1)}{\alpha} \frac{d(c_i, W_i)}{|C_i|}$.*

Proof. First, if $c'_i = c_i$ the claim is trivially true so assume $c'_i \neq c_i$. We begin with the fact that $d'(c'_i, C'_i) \leq d'(c_i, C'_i)$ and then break C'_i into its three components W_i , V_i , and A_i . We move the W_i terms to one side (the cost more than c_i on W_i), the rest of the terms to another side (the cost saved on V_i and A_i):

$$(C.3) \quad d'(c'_i, W_i) - d'(c_i, W_i) \leq d'(c_i, A_i) - d'(c'_i, A_i) + d'(c_i, V_i) - d'(c'_i, V_i).$$

Beginning with the right hand side of (C.3), by the triangle inequality we have $d(c_i, V_i) \leq d(c'_i, V_i) + |V_i|d(c_i, c'_i)$. Thus, $d'(c_i, V_i) \leq d'(c'_i, V_i) + \alpha|V_i|d(c_i, c'_i)$. Similarly, by Fact C.1 we have $d'(c_i, A_i) \leq d'(c'_i, A_i) + \alpha|A_i|d(c_i, c'_i) + \alpha(\alpha + 1)d(c_i, c'_i)$. So, the right hand side of (C.3) is at most $\alpha(|V_i| + |A_i| + \alpha + 1)d(c_i, c'_i)$. Now, examining the left hand side, this quantity is at least $\alpha d(c'_i, W_i \setminus \{c(c'_i)\}) - d(c_i, W_i)$. So, we have

$$(C.4) \quad \alpha d(c'_i, W_i \setminus \{c(c'_i)\}) - d(c_i, W_i) \leq \alpha(|V_i| + |A_i| + \alpha + 1)d(c_i, c'_i).$$

Using the fact that by triangle inequality, $\alpha(|W_i| - 1)d(c_i, c'_i) \leq \alpha d(c'_i, W_i \setminus \{c(c'_i)\}) + \alpha d(c_i, W_i \setminus \{c(c'_i)\})$, and subtracting $(\alpha + 1)d(c_i, W_i)$ from both sides, we get

$$(C.5) \quad \alpha d(c_i, c'_i)(|W_i| - 1) - (\alpha + 1)d(c_i, W_i) \leq \alpha d(c'_i, W \setminus \{c(c'_i)\}) - d(c_i, W_i).$$

Combining (C.4) and (C.5) we have:

$$\alpha d(c_i, c'_i)(|W_i| - 1) - (\alpha + 1)d(c_i, W_i) \leq \alpha d(c_i, c'_i) [|V_i| + |A_i| + \alpha + 1]$$

which implies the desired result when $|C_i| > 5\epsilon n + 2\alpha + 6$. \square

CLAIM 4.2. *For each i , if $c'_i \neq c_i$ then $d(c_i, c'_i) \geq \left(\frac{\alpha-1}{2\alpha}\right) \frac{d(c_i, W_i)}{\epsilon n + \alpha + 1}$.*

Proof. Assume $c'_i \neq c_i$ and let $d_i = d(c_i, c'_i)$. We will first use the fact that by definition of c_i , $d(c_i, C_i) \leq d(c'_i, C_i)$. We then will move from C_i to C'_i and then from d to d' .

First, since $d(c_i, C_i) \leq d(c'_i, C_i)$ and the difference between C'_i and C_i is small, $d(c_i, C'_i)$ cannot be too much larger than $d(c'_i, C'_i)$. In particular, since the set A_i of points added has $|A_i| \leq \epsilon n$ and the set M_i of points removed has $|M_i| \leq \epsilon n$ we have by triangle inequality:

$$(C.6) \quad d(c_i, A_i) \leq d(c'_i, A_i) + (\epsilon n)d_i,$$

$$(C.7) \quad d(c_i, M_i) \geq d(c'_i, M_i) - (\epsilon n)d_i.$$

So, $d(c_i, C'_i) \leq d(c'_i, C'_i) + 2(\epsilon n)d_i$.

Now, we switch from d to d' . We begin with A_i . By Fact C.1 we have

$$(C.8) \quad d'(c_i, A_i) \leq d'(c'_i, A_i) + \alpha \epsilon n d_i + \alpha(\alpha + 1)d_i.$$

But now c_i gets a saving from $C_i \setminus M_i$. Specifically, from (C.7) we have:

$$d(c_i, C_i \setminus M_i) \leq d(c'_i, C_i \setminus M_i) + (\epsilon n)d_i$$

so

$$(C.9) \quad \begin{aligned} \alpha d(c_i, C_i \setminus M_i) &\leq \alpha d(c'_i, C_i \setminus M_i) + \alpha(\epsilon n)d_i \\ &\leq d'(c'_i, C_i \setminus M_i) + (\alpha - 1)d_i + \alpha(\epsilon n)d_i \end{aligned}$$

where the second step is from the following fact: $d'(c'_i, C_i \setminus M_i) = \alpha d(c'_i, C_i \setminus M_i)$ if $c(c'_i) \neq c_i$, else $d'(c'_i, C_i \setminus M_i) = \alpha d(c'_i, C_i \setminus M_i) - (\alpha - 1)d_i$. Now, the left hand side above equals $d'(c_i, C_i \setminus M_i) + (\alpha - 1)d(c_i, W_i)$ because distances between c_i and W_i are *not* blown up by a factor of α .

Adding up (C.8) and (C.9) means that we get a contradiction if $(\alpha - 1)d(c_i, W_i) > [2\alpha \epsilon n + \alpha(\alpha + 1) + \alpha - 1]d_i$, in other words, if our savings in using c_i as center is greater than our extra cost. Therefore, $d_i \geq \left(\frac{\alpha-1}{2\alpha}\right) \frac{d(c_i, W_i)}{\epsilon n + \alpha + 1}$ as desired. \square

Combining the lower bound of Claim 4.1 with the upper bound of Claim 4.2 when $c'_i \neq c_i$, we get a contradiction for $|C_i| > 6 \left(\frac{\alpha+1}{\alpha-1}\right) (\epsilon n + \alpha + 1)$ as given in the theorem statement, yielding $c'_i = c_i$.

Appendix D. (α, ϵ) -Perturbation Resilient Min-Sum Instances.

D.1. Proofs for Bounding the Number of Bad Points for Min-Sum.

Recall the definitions of the bad points and the perturbation constructed to bound the number of bad points in Section 6.1.1. Suppose an intervals $[r, 2r]$ contains the costs of more than $2\epsilon n$ bad points. Let \hat{B} denote a subset of $2\epsilon n$ bad points in this interval, and let $\hat{B}_i = \hat{B} \cap C_i$, $K_i = C_i \setminus \hat{B}_i$. Suppose for a bad point $p \in \hat{B}_i$, C_j is its second nearest cluster, that is, $j = \arg \min_{\ell \neq i} d(p, C_\ell)$. Denote this as $p \in D_j$. Let $\tilde{C}_i = K_i \cup D_i$. The perturbation is constructed as follows: blow up all distances by a factor of α except those within \tilde{C}_i , $1 \leq i \leq k$. Let $\{C'_i\}$ denote the optimal clustering

after perturbation. Recall the definitions of U_i, V_i, W_i and $\tilde{U}_i, \tilde{V}_i, \tilde{W}_i$, and see Figure 7 for an illustration. The following facts come from their definitions.

FACT D.1. *We have $\cup_i U_i = \cup_i \tilde{U}_i$, $\cup_i V_i = \cup_i \tilde{V}_i$ and $\cup_i W_i = \cup_i \tilde{W}_i$. Furthermore,*

$$\begin{aligned} \sum_i d(\tilde{U}_i, C_i) &\leq \beta \sum_i d(U_i, C_i) \\ \sum_i d(\tilde{V}_i, C_i) &\leq \sum_i d(V_i, C_i), \\ \sum_i d(\tilde{W}_i, C_i) &\leq \sum_i d(W_i, C_i). \end{aligned}$$

We are ready to prove the claim needed for bounding the number of bad points.

CLAIM 6.1.(1). *The costs saved and added by moving $\{U_i, 1 \leq i \leq k\}$ satisfy*

$$\begin{aligned} &2 \sum_i d'(U_i, C'_i \cap K_i) - 2 \sum_i d'(\tilde{U}_i, \tilde{C}_i) \\ &\geq \frac{3}{10} \alpha \sum_i d(U_i, C_i) - \frac{2\alpha}{100} \sum_i d(W_i, C_i) - \frac{8\alpha + 16}{100} ren. \end{aligned}$$

Proof. Intuitively, we have that $\sum_i d'(U_i, C'_i \cap K_i) \approx \alpha \sum_i d(U_i, C_i)$. Similarly, $\sum_i d'(\tilde{U}_i, \tilde{C}_i) \approx \sum_i d(\tilde{U}_i, C_i)$. Their difference is then roughly $(\alpha - \beta) \sum_i d(U_i, C_i)$, since $\sum_i d(\tilde{U}_i, C_i) \leq \beta \sum_i d(U_i, C_i)$.

Formally, we have

$$\begin{aligned} \text{(D.1)} \quad d'(U_i, C'_i \cap K_i) &= \alpha d(U_i, C'_i \cap K_i) = \alpha d(U_i, C_i) - \alpha d(U_i, \tilde{W}_i + \hat{B}_i), \\ \text{(D.2)} \quad d'(\tilde{U}_i, \tilde{C}_i) &= d(\tilde{U}_i, \tilde{C}_i) = d(\tilde{U}_i, K_i) + d(\tilde{U}_i, D_i) \leq d(\tilde{U}_i, C_i) + d(\tilde{U}_i, D_i). \end{aligned}$$

Then it suffices to bound the approximation error $d(U_i, \tilde{W}_i + \hat{B}_i)$ and $d(\tilde{U}_i, D_i)$. First, for $d(U_i, \tilde{W}_i + \hat{B}_i)$ we have

$$\begin{aligned} d(U_i, \tilde{W}_i + \hat{B}_i) &\leq \frac{|\tilde{W}_i + \hat{B}_i|}{|C_i|} d(U_i, C_i) + \frac{|U_i|}{|C_i|} d(C_i, \tilde{W}_i + \hat{B}_i) \\ &\leq \frac{3}{100} d(U_i, C_i) + \frac{1}{100} d(C_i, \tilde{W}_i + \hat{B}_i) \end{aligned}$$

where the first inequality is by Fact D.2, and the second is from the fact that $|\tilde{W}_i| \leq \epsilon n$, $|\hat{B}_i| \leq 2\epsilon n$, $|U_i| \leq \epsilon n$ and $|C_i| \geq 100\epsilon n$. For the second term on the right hand side, we have $\sum_i d(C_i, \tilde{W}_i) \leq \sum_i d(W_i, C_i)$, Furthermore, the points in \hat{B}_i has cost at most $2r$ and $\sum_i |\hat{B}_i| \leq 2\epsilon n$. So

$$\sum_i d(U_i, \tilde{W}_i + \hat{B}_i) \leq \frac{3}{100} \sum_i d(U_i, C_i) + \frac{1}{100} \sum_i d(C_i, W_i) + \frac{4ren}{100}.$$

Similarly, for $d(\tilde{U}_i, D_i)$ we have

$$\sum_i d(\tilde{U}_i, D_i) \leq \sum_i \left[\frac{|D_i|}{|C_i|} d(\tilde{U}_i, C_i) + \frac{|\tilde{U}_i|}{|C_i|} d(C_i, D_i) \right] \leq \frac{2\beta}{100} \sum_i d(U_i, C_i) + \frac{8ren}{100}.$$

The claim follows by summing (D.1) and (D.2) over $1 \leq i \leq k$ and plugging in the last two inequalities. \square

CLAIM 6.1.(2). *The costs saved and added by moving $\{V_i, 1 \leq i \leq k\}$ satisfy*

$$\begin{aligned} & 2 \sum_i d'(V_i, C'_i \cap C_i) - 2 \sum_i d'(\tilde{V}_i, \tilde{C}_i) \\ & \geq \frac{99}{50}(\alpha - 2) \sum_i d(V_i, C_i) - \frac{2\alpha}{100} \sum_i d(W_i, C_i) - \frac{4\alpha + 8\beta}{100} r\epsilon n. \end{aligned}$$

Proof. The intuition is similar to that of Claim 6.1.(a): $\sum_i d'(V_i, C'_i \cap C_i) \approx \alpha \sum_i d(V_i, C_i)$ and $\sum_i d'(\tilde{V}_i, \tilde{C}_i) \approx \sum_i d(\tilde{V}_i, \tilde{C}_i)$. Since $\sum_i d(V_i, C_i) \geq \sum_i d(\tilde{V}_i, \tilde{C}_i)$, their difference is roughly $(\alpha - 1) \sum_i d(V_i, C_i)$.

Formally, we have

$$(D.3) \quad d'(V_i, C'_i \cap C_i) = \alpha d(V_i, C'_i \cap C_i) = \alpha d(V_i, C_i) - \alpha d(V_i, C_i \setminus C'_i),$$

$$(D.4) \quad d'(\tilde{V}_i, \tilde{C}_i) = d(\tilde{V}_i, \tilde{C}_i) \leq d(\tilde{V}_i, C_i) + d(\tilde{V}_i, D_i).$$

Then it suffices to bound the approximation error $d(V_i, C_i \setminus C'_i)$ and $d(\tilde{V}_i, D_i)$. First,

$$\begin{aligned} d(V_i, C_i \setminus C'_i) & \leq \frac{|C_i \setminus C'_i|}{|C_i|} d(V_i, C_i) + \frac{|V_i|}{|C_i|} d(C_i \setminus C'_i, C_i) \\ & \leq \frac{1}{100} d(V_i, C_i) + \frac{1}{100} d(C_i \setminus C'_i, C_i) \end{aligned}$$

where the first inequality is by Fact D.2 and the second is from the fact that $|C_i \setminus C'_i| \leq \epsilon n$, $|V_i| \leq \epsilon n$ and $|C_i| \geq 100\epsilon n$. For the second term on the right hand side, we have $d(C_i \setminus C'_i, C_i) \leq d(\tilde{W}_i, C_i) + d(\hat{B}_i \setminus C'_i, C_i)$. Note that $\sum_i d(\tilde{W}_i, C_i) \leq \sum_i d(W_i, C_i)$. Furthermore, the points in \hat{B}_i have cost at most $2r$ and $\sum_i |\hat{B}_i \setminus C'_i| \leq \epsilon n$ by perturbation resilience. So

$$\sum_i d(V_i, C_i \setminus C'_i) \leq \frac{1}{100} \sum_i d(V_i, C_i) + \frac{1}{100} \sum_i d(W_i, C_i) + \frac{2r\epsilon n}{100}.$$

Similarly, for $d(\tilde{V}_i, D_i)$ we have

$$\sum_i d(\tilde{V}_i, D_i) \leq \sum_i \left[\frac{|D_i|}{|C_i|} d(\tilde{V}_i, C_i) + \frac{|\tilde{V}_i|}{|C_i|} d(C_i, D_i) \right] \leq \frac{2}{100} \sum_i d(V_i, C_i) + \frac{4\beta r\epsilon n}{100}.$$

The claim follows by summing (D.3) and (D.4) over $1 \leq i \leq k$ and plugging the last two inequalities. \square

CLAIM 6.1.(3). *The costs saved and added by moving $\{W_i, 1 \leq i \leq k\}$ satisfy*

$$\begin{aligned} & 2 \sum_i d'(W_i, C'_i \cap C_i) - 2 \sum_i d'(\tilde{W}_i, \tilde{W}_i + C'_i \cap \tilde{C}_i) \\ & \geq \frac{98}{50}(\alpha - 2) \sum_i d(W_i, C_i) - \frac{4\alpha + 4\beta}{100} r\epsilon n. \end{aligned}$$

Proof. The intuition is similar to that of Claim 6.1.(a): $\sum_i d'(W_i, C'_i \cap C_i) \approx \alpha \sum_i d(W_i, C_i)$ and $\sum_i d'(\tilde{W}_i, \tilde{W}_i + C'_i \cap \tilde{C}_i) \approx \sum_i d(\tilde{W}_i, C_i)$. Since $\sum_i d(W_i, C_i) \geq \sum_i d(\tilde{W}_i, C_i)$, their difference is roughly $(\alpha - 1) \sum_i d(W_i, C_i)$.

Formally, we have

$$(D.5) \quad d'(W_i, C'_i \cap C_i) = \alpha d(W_i, C'_i \cap C_i) = \alpha d(W_i, C_i) - \alpha d(W_i, C_i \setminus C'_i).$$

$$(D.6) \quad d'(\tilde{W}_i, \tilde{W}_i + C'_i \cap \tilde{C}_i) = d(\tilde{W}_i, C'_i \cap D_i + C_i \cap \tilde{C}_i) \\ \leq d(\tilde{W}_i, C'_i \cap D_i) + d(\tilde{W}_i, C_i).$$

Then it suffices to bound the approximation error $d(W_i, C_i \setminus C'_i)$ and $d(\tilde{W}_i, C'_i \cap D_i)$. First, for $d(W_i, C_i \setminus C'_i)$ we have

$$d(W_i, C_i \setminus C'_i) \leq \frac{|C_i \setminus C'_i|}{|C_i|} d(W_i, C_i) + \frac{|W_i|}{|C_i|} d(C_i \setminus C'_i, C_i) \\ \leq \frac{1}{100} d(W_i, C_i) + \frac{1}{100} d(C_i \setminus C'_i, C_i)$$

where the first inequality is by Fact D.2 and the second from the fact that $|C_i \setminus C'_i| \leq \epsilon n$, $|W_i| \leq \epsilon n$ and $|C_i| \geq 100\epsilon n$. For the second term on the right hand side, we have $d(C_i \setminus C'_i, C_i) = d(\tilde{W}_i, C_i) + d(\hat{B}_i \setminus C'_i, C_i)$. Note that $\sum_i d(\tilde{W}_i, C_i) \leq \sum_i d(W_i, C_i)$. Furthermore, the points in \hat{B}_i have cost at most $2r$ and $\sum_i |\hat{B}_i \setminus C'_i| \leq \epsilon n$ by perturbation resilience. So

$$\sum_i d(W_i, C_i \setminus C'_i) \leq \frac{2}{100} \sum_i d(W_i, C_i) + \frac{2\epsilon n}{100}.$$

Similarly, for $d(\tilde{W}_i, C'_i \cap D_i)$ we have

$$\sum_i d(\tilde{W}_i, C'_i \cap D_i) \leq \sum_i \left[\frac{|C'_i \cap D_i|}{|C_i|} d(\tilde{W}_i, C_i) + \frac{|\tilde{W}_i|}{|C_i|} d(C'_i \cap D_i, C_i) \right] \\ \leq \frac{1}{100} \sum_i d(W_i, C_i) + \frac{2\beta\epsilon n}{100}.$$

The claim follows by summing (D.5) and (D.6) over $1 \leq i \leq k$ and plugging the last two inequalities. \square

D.2. Properties of Good Points in Min-Sum. The first property is that the cost between two optimal clusters is roughly that between sufficiently large subsets of their good points (Lemma 6.5). To prove this, we begin with the following useful fact.

FACT D.2. *For any non-empty sets A, B and C , we have $d_a(A, B) \leq d_a(A, C) + d_a(C, B)$, and thus $d(A, B) \leq \frac{|B|}{|C|} d(A, C) + \frac{|A|}{|C|} d(C, B)$.*

Then we have the following useful property for good points, which shows that good points are much closer to its own cluster than to good points in any other cluster. This property then leads to Lemma 6.5.

LEMMA D.1. *For $G_A \subseteq G_i, G_B \subseteq G_j, j \neq i$, we have*

$$d_a(G_A, C_i) \leq \gamma_{ji} d_a(G_A, G_B), \text{ where } \gamma_{ji} = \frac{|C_j|}{(\beta - 1/\beta)|C_i|} + \frac{1}{\beta^2 - 1}.$$

Consequently, if $\alpha > \frac{8 \max_i |C_i|}{\min_i |C_i|}$, we have $d_a(G_A, C_i) \leq \frac{11}{50} d_a(G_A, G_B)$.

Proof. For any $p \in G_A$, we have $\beta d(p, C_i) < d(p, C_j)$. It follows from Fact D.2 that

$$\begin{aligned} \beta d(G_A, C_i) < d(G_A, C_j) &\leq \frac{|C_j|}{|G_B|} d(G_A, G_B) + \frac{|G_A|}{|G_B|} d(C_j, G_B) \\ \beta d(G_B, C_j) < d(G_B, C_i) &\leq \frac{|C_i|}{|G_A|} d(G_B, G_A) + \frac{|G_B|}{|G_A|} d(C_i, G_A). \end{aligned}$$

Plug the second inequality into the first inequality, then the lemma follows. \square

LEMMA 6.5. *Suppose $\alpha > \frac{8 \max_i |C_i|}{\min_i |C_i|}$ and $W_i \subseteq G_i, W_j \subseteq G_j$. When $|C_i| \geq 50|C_i \setminus W_i|$ and $|C_j| \geq 50|C_j \setminus W_j|$, we have $d(C_i, C_j) \leq \frac{3}{2}d(W_i, W_j)$.*

Proof. By Fact D.2 and Lemma D.1, we have

$$d_a(C_i, C_j) \leq d_a(C_i, W_i) + d_a(W_i, W_j) + d_a(W_j, C_j) \leq \left(\frac{11}{50} + 1 + \frac{11}{50}\right) d_a(W_i, W_j)$$

which leads to $d(C_i, C_j) \leq \frac{36}{25} \frac{|C_i||C_j|}{|W_i||W_j|} d(W_i, W_j) \leq \frac{3}{2}d(W_i, W_j)$. \square

Another property of good points is that the good points from two different clusters have cost much larger than those in a third cluster have (Lemma 6.6). To prove this, we need to prove Lemma D.2, which bounds the cost of the optimal clustering $\mathcal{C}' = \{C'_t\}$ under the perturbed distance function. Recall the definitions of the perturbation and \mathcal{C}' in Section 6.1.2. The perturbation blows up all pairwise distances by a factor of α except the intra-cluster distances in $\tilde{\mathcal{C}}$, where $\tilde{\mathcal{C}}$ is the clustering obtained from the optimal clustering by splitting C_i into A and $C_i \setminus A$ and merging C_j and C_l . Let $\mathcal{C}' = \{C'_i\}$ denote the optimal clustering under the perturbed distance function d' , where the clusters are indexed so that C'_i corresponds to C_i and the distance between the two clustering is $\sum_i |C_i \setminus C'_i|$.

To bound the cost of $\mathcal{C}' = \{C'_t\}$, we compare it to the cost of the optimal clustering $\mathcal{C} = \{C_t\}$ before perturbation. If $\mathcal{C}' = \mathcal{C}$, then the cost is only increased by blowing up the distances between A and $C_i \setminus A$ (Claim D.1). However, the optimal clustering may change after the perturbation, so we need to consider how much cost is saved by the change (Claim D.2).

Intuitively, the cost saved should be small. To see this, consider a point p moved from C_s to C'_t . Then we need to pay $d'(p, C'_t)$ instead of $d(p, C_s)$. Note that p is in C_s but not C_t , so $d(p, C_s) \leq d(p, C_t)$. Also, C'_t and C_t differ only on at most ϵn points, then $d'(p, C'_t)$ is larger or comparable to $d(p, C_t)$ and thus $d(p, C_s)$.

There are two technical details in the above description. The first is to translate $d'(p, C'_t)$ to $d(p, C'_t)$. We consider two cases (as in the proof of Claim D.2). If p is moved between C_j and C_l , then $d'(p, C'_t)$ is roughly $d(p, C'_t)$ since the distances between C_j, C_l are not blown up. Otherwise, $d'(p, C'_t)$ is roughly $\alpha d(p, C'_t)$. Another technical detail is to show that $d(p, C'_t)$ roughly equals $d(p, C_t)$. Since $d(p, C'_t) \geq d(p, C'_t \cap C_t)$, it suffices to show that $d(p, C'_t \cap C_t)$ is comparable to $d(p, C_t)$, where Fact 5.1 turns out to be useful.

LEMMA D.2. *Suppose $\alpha > \frac{6 \max_i |C_i|}{\min_i |C_i|}$ and $\min_i |C_i| \geq 100m_B$. We have*

$$\sum_{t=1}^k d'(C'_t, C'_t) - \sum_{t=1}^k d(C_t, C_t) \geq 2(\alpha - 1)d(A, G_i \setminus A) - \frac{4\alpha + 8}{100}d(C_j, C_l).$$

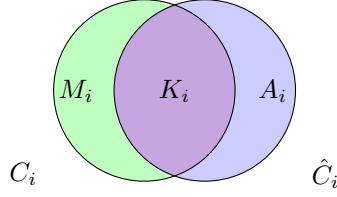


Fig. 10: Illustration of the notations in Lemma D.2.

Proof. Let $K_t = C_t \cap C'_t$, $A_t = C'_t \setminus C_t$, $M_t = C_t \setminus C'_t$. See Figure 10 for an illustration. We have

$$\begin{aligned}
& \sum_{t=1}^k d'(C'_t, C'_t) - \sum_{t=1}^k d(C_t, C_t) \\
& \geq \sum_{t=1}^k [d'(K_t, K_t) + 2d'(A_t, K_t)] - \sum_{t=1}^k [d(K_t, K_t) + 2d(M_t, C_t)] \\
& = \left[\sum_{t=1}^k d'(K_t, K_t) - \sum_{t=1}^k d(K_t, K_t) \right] + 2 \left[\sum_{t=1}^k d'(A_t, K_t) - \sum_{t=1}^k d(M_t, C_t) \right].
\end{aligned}$$

The first term on the right hand side corresponds to the cost increased by blowing up the distances within the clusters, the second term corresponds to the cost increased by moving points away. We will bound the two terms respectively in the following two claims, which then lead to the lemma.

Let $l(p)$ denote the index of the optimal cluster in \mathcal{C} that p falls in: if $p \in C_t$, then $l(p) = t$. Similarly, let $l'(p)$ denote the optimal cluster in \mathcal{C}' that p falls in after perturbation: if $p \in C'_t$, then $l'(p) = t$.

The first term is roughly the cost increased by blowing the distances between A_i and $C_i \setminus A_i$, which is about $2(\alpha - 1)d(A, C_i \setminus A)$. However, some points in C_i may move away, so we need to exclude the cost of these points. More precisely, we only consider good points, and also exclude the cost of the good points moved away ($G_i \cap M_i$).

CLAIM D.1.

$$\begin{aligned}
& \sum_{t=1}^k d'(K_t, K_t) - \sum_{t=1}^k d(K_t, K_t) \\
\text{(D.7)} \quad & \geq 2(\alpha - 1)d(A, G_i \setminus A) - \frac{2(\alpha - 1)}{\beta} \sum_{p \in G_i \cap M_i} d(p, C_{l'(p)}).
\end{aligned}$$

Proof. By the definition of the perturbation, we have

$$\begin{aligned}
& \sum_{t=1}^k d'(K_t, K_t) - \sum_{t=1}^k d(K_t, K_t) \\
& \geq 2d'(A \cap K_i, (G_i \setminus A) \cap K_i) - 2d(A \cap K_i, (G_i \setminus A) \cap K_i) \\
& \geq 2(\alpha - 1)d(A \cap K_i, (G_i \setminus A) \cap K_i) \\
& \geq 2(\alpha - 1)[d(A, G_i \setminus A) - d(A \cap M_i, G_i \setminus A) - d((G_i \setminus A) \cap M_i, A)] \\
& \geq 2(\alpha - 1)[d(A, G_i \setminus A) - d(G_i \cap M_i, C_i)].
\end{aligned}$$

The claim then follows from that for any $p \in G_i \cap M_i$, $d(p, C_i) \leq \frac{1}{\beta}d(p, C_{l'(p)})$. \square

The second term is roughly the cost increased by moving points away. Consider a point $p \in C_1$ that moves to C'_2 . The new cost is $d'(p, C'_2) \approx d'(p, C_2) = \alpha d(p, C_2)$, and the old cost is $d(p, C_1) \leq d(p, C_2)$, so the cost increased is roughly $(\alpha - 1)d(p, C_2)$. Note that C'_2 only approximately equals C_2 . Also, the above intuition does not hold for points that move between C_j and C_l since the distances between them are not blown up. These facts only decrease the bound slightly, as shown in the following claim.

CLAIM D.2. Let $X = (\cup_t A_t) \setminus (A_l \cap C_j) \setminus (A_j \cap C_l)$.

$$\begin{aligned}
& \sum_{t=1}^k d'(A_t, K_t) - \sum_{t=1}^k d(M_t, C_t) \\
\text{(D.8)} \quad & \geq \left(\frac{98\alpha}{100} - 1 \right) \sum_{p \in X} d(p, C_{l'(p)}) - \frac{2\alpha + 4}{100} d(C_j, C_l).
\end{aligned}$$

Proof. We have

$$\text{(D.9)} \quad \sum_{t=1}^k d'(A_t, K_t) - \sum_{t=1}^k d(M_t, C_t) \geq \sum_{t=1}^k \sum_{p \in A_t} [d'(p, K_t) - d(p, C_{l(p)})].$$

Intuitively, $d'(p, K_{l'(p)})$ should be larger or comparable to $d(p, C_{l(p)})$. On one hand, $d(p, C_{l(p)}) \leq d(p, C_{l'(p)})$ since p is assigned to $C_{l(p)}$ instead of $C_{l'(p)}$ in the optimal clustering under d . On the other hand, we also know that $d(p, K_{l'(p)})$ is comparable to $d(p, C_{l'(p)})$ by Fact 5.1.

Before using this intuition, we first need to translate $d'(p, K_{l'(p)})$ to $d(p, K_{l'(p)})$. Since the distances between C_j and C_l is not blown up, we need to consider separately the case when p is moved between C_j and C_l . Equivalently, we divide $\cup_t A_t$ into two parts: $V = (A_j \cap C_l) \cup (A_l \cap C_j)$ and $X = (\cup_t A_t) \setminus (A_l \cap C_j) \setminus (A_j \cap C_l)$. Now we consider the two parts respectively.

Case 1. Suppose $p \in A_j \cap C_l$. By Fact 5.1, we have $d'(p, K_{l'(p)}) = d(p, K_j) \geq \frac{|K_j|}{|C_j|}d(p, C_j) - \frac{1}{|C_j|}d(M_j, C_j)$. Since $d(p, C_{l(p)}) = d(p, C_l) \leq d(p, C_j)$, and $d(M_j, C_j) \leq d(M_j, C_l) \leq d(C_j, C_l)$,

$$\begin{aligned}
& d'(p, K_{l'(p)}) - d(p, C_{l(p)}) \geq -\frac{|M_j|}{|C_j|}d(p, C_j) - \frac{1}{|C_j|}d(C_j, C_l), \\
& \sum_{p \in A_j \cap C_l} [d'(p, K_{l'(p)}) - d(p, C_{l(p)})] \geq -\left[\frac{|M_j|}{|C_j|} + \frac{|A_j \cap C_l|}{|C_j|} \right] d(C_j, C_l).
\end{aligned}$$

Since $|M_j| \leq \epsilon n$, $|A_j| \leq \epsilon n$, this is bounded by $-\frac{2}{100}d(C_j, C_l)$. A similar argument holds for $A_j \cap C_l$. So

$$(D.10) \quad \sum_{p \in V} [d'(p, K_{l'(p)}) - d(p, C_{l(p)})] \geq -\frac{4}{100}d(C_j, C_l).$$

Case 2. For $p \in X$, we have by Fact 5.1

$$d'(p, K_{l'(p)}) = \alpha d(p, K_{l'(p)}) \geq \alpha \left[\frac{|K_{l'(p)}|}{|C_{l'(p)}|} d(p, C_{l'(p)}) - \frac{1}{|C_{l'(p)}|} d(M_{l'(p)}, C_{l'(p)}) \right].$$

Then for X , since $d(p, C_{l(p)}) \leq d(p, C_{l'(p)})$ and $\frac{|K_{l'(p)}|}{|C_{l'(p)}|} \geq \frac{99}{100}$, we have

$$(D.11) \quad \begin{aligned} & \sum_{p \in X} [d'(p, K_{l'(p)}) - d(p, C_{l(p)})] \\ & \geq \left(\frac{99\alpha}{100} - 1 \right) \sum_{p \in X} d(p, C_{l'(p)}) - \sum_{p \in X} \frac{\alpha}{|C_{l'(p)}|} d(M_{l'(p)}, C_{l'(p)}). \end{aligned}$$

Since $X \subseteq \cup_t A_t$, and $|C_t| \geq 100|A_t|$, the second term on the right hand side is bounded by

$$(D.12) \quad \begin{aligned} \sum_t \frac{\alpha|A_t|}{|C_t|} d(M_t, C_t) & \leq \frac{\alpha}{100} \sum_t d(M_t, C_t) = \frac{\alpha}{100} \sum_{p \in \cup_t A_t} d(p, C_{l(p)}) \\ & = \frac{\alpha}{100} \left[\sum_{p \in V} d(p, C_{l(p)}) + \sum_{p \in V} d(p, C_{l(p)}) \right] \\ & \leq \frac{\alpha}{100} \left[\sum_{p \in V} d(p, C_{l'(p)}) + 2d(C_i, C_j) \right]. \end{aligned}$$

The claim follows from the inequalities (D.10), (D.11), and (D.12). \square

The proof is completed by combining the two claims. \square

LEMMA 6.6. *Suppose $\alpha > \frac{8 \max_i |C_i|}{\min_i |C_i|}$ and $\epsilon < \frac{\min_i |C_i|}{600n}$. For any three different optimal clusters C_i, C_j , and C_l , and any $A \subset G_i$, $\frac{18}{5}d(A, G_i \setminus A) < d(G_j, G_l)$. Consequently, $\frac{9}{5}d(G_i, G_i) < d(G_j, G_l)$.*

Proof. The key idea is as follows. Let $\tilde{\mathcal{C}}$ denote the clustering obtained from the optimal clustering by splitting C_i into A and $C_i \setminus A$ and merging C_j and C_l , that is, $\tilde{\mathcal{C}} = \{A, C_i \setminus A, C_j \cup C_l\} \cup \{C_t, t \neq i, j, l\}$. Suppose we construct a perturbation that favors the clustering $\tilde{\mathcal{C}}$: blow up all pairwise distances by a factor of α except the intra-cluster distances in $\tilde{\mathcal{C}}$. Let $\mathcal{C}' = \{C'_i\}$ denote the optimal clustering under the perturbed distance function d' , where the clusters are indexed so that C'_i corresponds to C_i and the distance between the two clustering is $\sum_i |C_i \setminus C'_i|$. By (α, ϵ) -perturbation resilience, we know that \mathcal{C}' is different from $\tilde{\mathcal{C}}$ and has no greater cost than $\tilde{\mathcal{C}}$. We then show that compared to the optimal cost under the original distances, the cost of $\tilde{\mathcal{C}}$ under perturbed distances d' is larger by at most $O(d(C_j, C_l)) = O(d(G_j, G_l))$, while the cost of \mathcal{C}' under perturbed distances d' is larger by roughly $O(\alpha)d(A, G_i \setminus A)$. These then leads to the first statement.

More precisely, the cost of $\tilde{\mathcal{C}}$ under d' is larger than that of \mathcal{C} under d by at most $2d(C_j, C_l)$. For \mathcal{C}' , we have $\sum_{t=1}^k d'(C'_t, C'_t) - \sum_{t=1}^k d(C_t, C_t) \geq 2(\alpha - 1)d(A, G_i \setminus A) - \frac{4\alpha + 8}{100}d(C_j, C_l)$ by Lemma D.2. Since \mathcal{C}' has smaller cost than $\tilde{\mathcal{C}}$, we have

$$2(\alpha - 1)d(A, G_i \setminus A) - \frac{4\alpha + 8}{100}d(C_j, C_l) \leq 2d(C_j, C_l).$$

When $\alpha > \frac{8 \max_i |C_i|}{\min_i |C_i|}$, we have $\frac{27}{5}d(A, G_i \setminus A) \leq d(C_j, C_l)$. By Lemma 6.5, $d(C_j, C_l) \leq \frac{3}{2}d(G_j, G_l)$, which then leads to the first part of the lemma.

The second part of the lemma follows from the fact that $\sum_{A \subseteq G_i} d(A, G_i \setminus A) = \frac{2^{|G_i|}}{2}d(G_i, G_i)$. \square

D.3. Properties of Potential Good Points in Min-Sum. A key property of the potential good points is that the cost between any point p and the potential good points in a sufficiently large set A is roughly the cost between p and any sufficiently large subset H of A (Lemma 6.7). In its proof in Section 6.1.2, we need the following claim.

CLAIM 6.2. *Suppose $H \subseteq A$ such that $|A \setminus H| \leq m_B$. Let $F = F(A), P = P(A), \bar{H} = A \setminus H$. Let $W = H \cap P, V = \bar{H} \cap F, X = F \cap H, Y = \bar{H} \cap P$. See Figure 8 for illustration. If $|A| \geq 20m_B$, then $d(W, X) \geq d(Y, W + Y)$.*

Proof. The claim is true if $Y = \emptyset$. Otherwise, by definition of the potential bad points $F = F(A)$, we have $d_a(X, A) \geq d_a(Y, A)$. By definition,

$$(D.13) \quad d_a(X, A) = |W|d_a(X, W) + |V|d_a(X, V) + |Y|d_a(X, Y) + |X|d_a(X, X),$$

$$(D.14) \quad |A|d_a(Y, A) = |W + Y|d_a(Y, W + Y) + |V|d_a(Y, V) + |X|d_a(Y, X).$$

To compare $d(X, W)$ and $d(Y, W + Y)$, we need to bound the other terms in (D.13) and (D.14). By Fact D.2,

$$d_a(X, V) \leq d_a(X, W) + d_a(W, Y) + d_a(Y, V),$$

$$d_a(X, Y) \leq d_a(X, W) + d_a(Y, W), \quad d_a(X, X) \leq 2d_a(X, W).$$

Now we plug these into (D.13), and then plug (D.13) and (D.14) into $d_a(X, A) \geq d_a(Y, A)$. Since $d(W, Y) \leq d(Y, W + Y)$ and $d_a(Y, X) \geq 0$, we have

$$[|W| - |Y| - |V|]d(Y, W + Y) \leq [|W| + 2|X| + |Y| + |V|] \frac{|Y|}{|X|}d(X, W).$$

Since $|X + V| = |F| = 2m_B$ and $|Y + V| = |A \setminus H| \leq m_B$, we have $\frac{|Y|}{|X|} \leq 1/2$. Then the lemma follows from the fact that $|A| \geq 20m_B, |F| = 2m_B$ and $|A \setminus H| \leq m_B$. \square

Appendix E. The Algorithm for (α, ϵ) -Perturbation Resilient Min-Sum.

E.1. Proofs for Tree Construction for Min-Sum. To show that the robust average linkage algorithm keeps the laminarity of the clustering \mathcal{L} with respect to the optimal clustering (after removing the bad points), we need to show the following claim.

CLAIM 6.4.(1) *Suppose $A \in \mathcal{L}, A \cap G \subsetneq G_i$, and $D \in \mathcal{L}, D \cap G \subsetneq G_j (j \neq i)$. Then there exists $A' \neq A$ in \mathcal{L} such that $A' \cap G \subsetneq G_i$ and $d_{\text{ra}}(A, A') < d_{\text{ra}}(A, D)$.*

Proof. The claim follows from the following three statements:

- (a) $d_a(A \cap G, A' \cap G) < \frac{1}{2}d_a(A \cap G, D \cap G)$;
- (b) $d_{ra}(A, A') \leq \frac{7}{5}d_a(A \cap G, A' \cap G)$;
- (c) $\frac{9}{10}d_a(A \cap G, D \cap G) \leq d_{ra}(A, D)$.

We now prove the statements respectively.

- (a) For simplicity, let $G_A = A \cap G, G_D = D \cap G$. From Lemma D.1, we have

$$d_a(G_A, C_i) \leq \gamma_{ji} d_a(G_A, G_D), \text{ where } \gamma_{ji} = \frac{|C_j|}{(\beta - 1/\beta)|C_i|} + \frac{1}{\beta^2 - 1}.$$

Since $d(G_A, G_i \setminus G_A) \leq d(G_A, C_i)$, we have

$$d_a(G_A, G_i \setminus G_A) \leq \frac{|C_i|}{|G_i \setminus G_A|} d_a(G_A, C_i) \leq \gamma_{ji} \frac{|C_i|}{|G_i \setminus G_A|} d_a(G_A, G_D) \leq \frac{1}{2} d_a(G_A, G_D)$$

where the last step follows from $\alpha \geq 6 \frac{\max_i |C_i|}{\min_i |C_i|} + 2$, $|G_i \setminus A|$ is at least $\frac{1}{2} \min_i |C_i| - m_B$.

- (b) $d_{ra}(A, A') \leq \frac{7}{5}d_a(A \cap G, A' \cap G)$. By Lemma 6.7 and the fact that $|A| \geq \frac{1}{2} \min_i |C_i|, |A'| \geq \frac{1}{2} \min_i |C_i|$ and $\min_i |C_i| > 100m_B$, we have

$$d(P(A), P(A')) \leq \frac{10}{9}d(P(A), A' \cap G) \leq \frac{100}{81}d(A \cap G, A' \cap G).$$

Then the claim follows from the fact that $|P(A)| \geq \frac{48}{50}|A|, |P(A')| \geq \frac{48}{50}|A'|$.

- (c) For simplicity, let $G_A = A \cap G, G_D = D \cap G$. Divide G_A into two parts: $W_A = G_A \cap P(A)$ and $X_A = G_A \cap F(A)$. Define W_D and X_D similarly. See Figure 11 for an illustration.

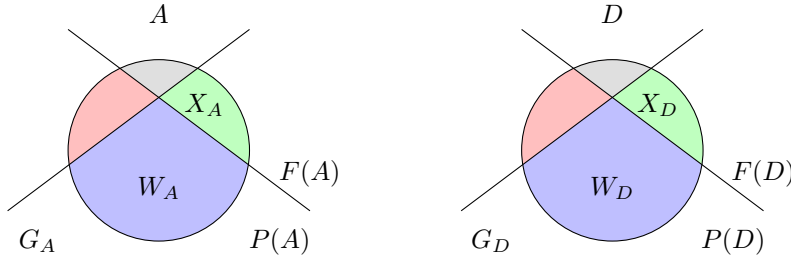


Fig. 11: Illustration of the notations in Claim 6.4.

To show $d(G_A, G_D) \leq O(1)d(P(A), P(D))$, it suffices to show $d(G_A, G_D) \leq O(1)d(W_A, W_D)$. Since $d(G_A, G_D) \leq d(W_A, W_D) + d(G_A, X_D) + d(G_D, X_A)$, we only need to bound $d(G_A, X_D)$ and $d(G_D, X_A)$. By Fact D.2 and Lemma D.1 we have

$$d_a(G_A, X_D) \leq d_a(G_A, G_D) + d_a(G_D, X_D) \leq \left(1 + \frac{11}{50}\right) d_a(G_A, G_D).$$

Since $|X_D| \leq 2m_B$ and $|D| \geq \frac{1}{2} \min_i |C_i| \geq 50m_B$, we have

$$d(G_A, X_D) \leq \frac{61}{50} \frac{|X_D|}{|G_D|} d(G_A, G_D) \leq \frac{1}{20} d(G_A, G_D).$$

Similarly, $d(G_D, X_A) \leq \frac{1}{20}d(G_A, G_D)$. Therefore,

$$d(G_A, G_D) \leq d(W_A, W_D) + d(G_A, X_D) + d(G_D, X_A) \leq d(W_A, W_D) + \frac{1}{10}d(G_A, G_D)$$

which leads to $\frac{9}{10}d(G_A, G_D) \leq d(W_A, W_D) \leq d(P(A), P(D))$. Then the claim follows from the fact that $|G_A| \geq |P(A)|$, $|G_D| \geq |P(D)|$. \square

CLAIM 6.4.(2) *Suppose $A \in \mathcal{L}$, $A \cap G \subsetneq G_i$, and $D \in \mathcal{L}$, $D \cap G$ is the union of good points in several optimal clusters. Then there exists $A' \neq A$ in \mathcal{L} such that $A' \cap G \subsetneq G_i$ and $d_{\text{ra}}(A, A') < d_{\text{ra}}(A, D)$.*

Proof. The proof idea is similar to that for Claim 6.4.(1). The only difference is the proof for $d_a(A \cap G, A' \cap G) < \frac{1}{2}d_a(A \cap G, D \cap G)$. Since $D \cap G = \cup_{j \in I_D} G_j$, it suffices to show that $d_a(A \cap G, A' \cap G) < \frac{1}{2}d_a(A \cap G, G_j)$ for any $j \in I_D$, which can be proved by the same argument in Claim 6.4.(1). \square

E.2. Proofs for Getting the Pruning Close to the Optimal Clustering.

The same argument as that for Claim 6.6 leads to a corollary for the general case when multiple clusters are merged.

COROLLARY E.1. *Let $I \subseteq [k]$. Suppose for any $t \in I$, $|C_t| \geq 100m_B$, and C'_t contains all good points in C_t but no good points in other optimal clusters. Then*

$$d_{\text{rs}}(\cup_{t \in I} C'_t) - \sum_{t \in I} d(G_t, G_t) \geq \left(\frac{4}{3} - \frac{4}{\beta}\right) \sum_{s \neq t \in I} d(G_t, G_s).$$

LEMMA 6.10. *Suppose the pruning $\mathcal{C}' = \{C'_1, \dots, C'_k\}$ in the tree \mathcal{T} assigns all good points correctly. Then \mathcal{C}' is the minimum robust min-sum cost pruning in the tree.*

Proof. First, by Lemma 6.6, good points from different clusters are far apart while good points in the same cluster are close. Second, by Claim 6.5 and Corollary E.1, the cost of good points can be approximated by the cost of the potential good points (the robust min-sum cost). We now use the above lemmas to show that \mathcal{C}' has minimum robust min-sum cost, so that we can use dynamic programming on the tree to get the pruning.

Suppose a pruning \mathcal{P} is obtained by splitting h clusters in \mathcal{C}' and at the same time joining some other clusters into g unions. Specifically, for $1 \leq i \leq h$, split C'_i into $m_i \geq 2$ clusters $S_{i,1}, \dots, S_{i,m_i}$; after that, merge $C'_{h+1}, \dots, C'_{h+l_g}$ into g unions, that is, for $1 \leq j \leq g$, $l_0 = 0$, merge $l_j - l_{j-1} \geq 2$ clusters $C'_{h+l_{j-1}+1}, \dots, C'_{h+l_j}$ into a union U_j ; the other clusters in \mathcal{C}' remain the same in \mathcal{P} . Since the number of clusters is still k , we have $\sum_i m_i - h = l_g - g$.

By Claim 6.5, the cost saved by splitting the h clusters is

$$(E.1) \quad \sum_{1 \leq i \leq h} d_{\text{rs}}(C'_i) - \sum_{1 \leq i \leq h} \sum_{1 \leq p \leq m_i} d_{\text{rs}}(S_{i,p}) \leq \sum_{1 \leq i \leq h} d_{\text{rs}}(C'_i) \leq \sum_{1 \leq i \leq h} d(G_i, G_i).$$

The cost increased by joining clusters is

$$\begin{aligned}
& \sum_{1 \leq j \leq g} \left[d_{\text{rs}}(U_j) - \sum_{h+l_{j-1} < t \leq h+l_j} d_{\text{rs}}(C'_t) \right] \\
\geq & \sum_{1 \leq j \leq g} \left[d_{\text{rs}}(U_j) - \sum_{h+l_{j-1} < t \leq h+l_j} d(G_t, G_t) \right] \\
\text{(E.2)} \quad & \geq \sum_{1 \leq j \leq g} \left[\sum_{h+l_{j-1} < t \neq s \leq h+l_j} \left(\frac{4}{3} - \frac{4}{\beta} \right) d(G_t, G_s) \right]
\end{aligned}$$

where the first inequality follows from Claim 6.5, and the second inequality follows from Corollary E.1. To prove \mathcal{C}' is the minimum cost pruning, we need to show that the saved cost (E.1) is less than the increased cost (E.2). Since by Lemma 6.6, each term in (E.2) is larger than any term in (E.1), it is sufficient to show that the number of the terms in (E.2) is no less than the number of the terms in (E.1), that is $\sum_{1 \leq j \leq g} \binom{l_j - l_{j-1}}{2} \geq h$. We have $\sum_j \binom{l_j - l_{j-1}}{2} = \frac{1}{2} \sum_j (l_j - l_{j-1})(l_j - l_{j-1} - 1) \geq \sum_j (l_j - l_{j-1} - 1) = l_g - g$, where the inequality is from $l_j - l_{j-1} \geq 2$. Since $m_i \geq 2$, $l_g - g = \sum_{i=1}^h m_i - h \geq h$, which completes the proof. \square

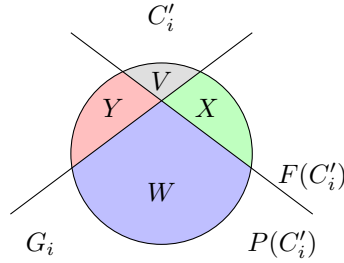


Fig. 12: Notations in Lemma 6.11 and Claim 6.7.

E.3. Getting A Constant Factor Approximation. To get a good approximation from a clustering \mathcal{C}' that assigns all good points correctly, Algorithm 8 reassigns each point p to the index i that minimizes the cost between p and the potential good points in C'_i . The following lemma shows that after this reassignment, all good points are still assigned correctly.

LEMMA 6.11. *For any $p \in G_i$, any $j \neq i$, $d(p, P(C'_j)) > d(p, P(C'_i))$.*

Proof. Let $W_i = G_i \cap P(C'_i)$ denote the good points that are also potential good points, and let $Z_i = C_i \setminus W_i$ denote all other points in C_i . See Figure 12 for an illustration. By Lemma 6.7, $d(p, P(C'_i)) \approx d(p, C_i)$. By definition of good points, $\beta d(p, C_i) \leq d(p, C_j)$. So it suffices to show that $d(p, P(C'_j))$ is not so small compared to $d(p, C_j)$. Since $W_j \subseteq P(C'_j)$, it suffices to prove that $d(p, W_j)$ is large compared to $d(p, Z_j)$.

First, by triangle inequality, $d(p, Z_j) \leq \frac{|Z_j|}{|W_j|} d(p, W_j) + \frac{1}{|W_j|} d(Z_j, W_j)$. Also, $d(Z_j, W_j) \leq d(C_j, W_j) \leq \frac{1}{\beta} d(C_i, W_j)$ by definition of good points. Furthermore,

$d(C_i, W_j) \leq |W_j|d(p, C_i) + |C_i|d(p, W_j)$. So

$$\begin{aligned} d(p, Z_j) &\leq \left(\frac{|Z_j|}{|W_j|} + \frac{|C_i|}{\beta|W_j|} \right) d(p, W_j) + \frac{1}{\beta} d(p, C_i) \\ &\leq \left(\frac{|Z_j|}{|W_j|} + \frac{|C_i|}{\beta|W_j|} \right) d(p, W_j) + \frac{1}{\beta^2} [d(p, Z_j) + d(p, W_j)]. \end{aligned}$$

Therefore, we have $d(p, Z_j) \leq \frac{1}{3}d(p, W_j)$, since $|Z_j| \leq 4m_B$, $|W_j| \geq \frac{95}{100}|C_j| \geq 95m_B$. This leads to $d(p, W_j) \geq \frac{3}{4}d(p, C_j)$. Then the lemma follows from

$$d(p, P(C'_j)) \geq d(p, W_j) \geq \frac{3}{4}d(p, C_j) \geq \frac{3}{4}\beta d(p, C_i) \geq \frac{3}{4}\beta d(p, G_i) \geq \frac{30}{44}\beta d(p, P(C'_i))$$

where the last step follows from Lemma 6.7. \square

Recall that A_i are all the bad points reassigned to index i by Algorithm 8. To bound the cost of the clustering after reassignment, we need to bound $d(A_i, G_i)$ as follows.

CLAIM 6.7. $\sum_i d(A_i, G_i) \leq \frac{r^2}{(r-5)^2} \sum_i d(C_i, C_i) - \frac{r^2-1}{(r-5)^2} \sum_i d(G_i, G_i)$, where $r = \frac{\min_i |C_i|}{m_B}$.

Proof. Let $W_i = P(C'_i) \cap G_i$. See Figure 12 for an illustration. By Fact 5.2,

$$\begin{aligned} \text{(E.3)} \quad d(A_i, G_i) &\leq \frac{|G_i|}{|W_i|} d(A_i, W_i) + \frac{|A_i|}{|W_i|} d(G_i, W_i) \\ &\leq \frac{|G_i|}{|W_i|} d(A_i, P(C'_i)) + \frac{|A_i|}{|W_i|} d(G_i, G_i). \end{aligned}$$

So it suffices to bound $d(A_i, P(C'_i))$. Fix $p \in A_i$, and suppose $p \in C_j$. We have

$$\begin{aligned} d(p, P(C'_i)) &\leq d(p, P(C'_j)) \leq \frac{|W_j| + |Y_j|}{|W_j| - |X_j|} d(p, G_j) \\ &\leq \frac{|C_j|}{|C_j| - 2|X_j| - |Y_j|} d(p, G_j) = \frac{r}{r-5} d(p, G_j) \end{aligned}$$

where the second step follows from Lemma 6.7 and the last from $|X_j| \leq 2m_B$ and $|Y_j| \leq m_B$. Then

$$\begin{aligned} \sum_{i=1}^k d(A_i, P(C'_i)) &\leq \frac{r}{r-5} \sum_j \sum_{p \in (\cup_i A_i) \cap C_j} d(p, G_j) = \frac{r}{r-5} \sum_{j=1}^k d(B_j, G_j) \\ \text{(E.4)} \quad &\leq \frac{r}{r-5} \sum_{j=1}^k [d(C_j, C_j) - d(G_j, G_j)]. \end{aligned}$$

The claim follows from the inequalities (E.3)(E.4) and $|X_i| \leq 2m_B$, $|A_i| \leq m_B$. \square