Clustering under Perturbation Resilience

Maria Florina Balcan[*] Yingyu Liang[†]

November 16, 2012

Abstract

Recently, Bilu and Linial [8] formalized an implicit assumption often made when choosing a clustering objective: that the optimum clustering to the objective should be preserved under small multiplicative perturbations to distances between points. They showed that for max-cut clustering it is possible to circumvent NP-hardness and obtain polynomial-time algorithms for instances resilient to large (factor $O(\sqrt{n})$) perturbations, and subsequently Awasthi et al. [2] considered center-based objectives, giving algorithms for instances resilient to $O(1)$ factor perturbations.

In this paper, we greatly advance this line of work. For center-based objectives, we present an algorithm that can optimally cluster instances resilient to $(1+\sqrt{2})$-factor perturbations, solving an open problem of Awasthi et al.[2]. 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.

[*]Georgia Institute of Technology, `ninamf@cc.gatech.edu`
[†]Georgia Institute of Technology, `yliang39@gatech.edu`

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 [12, 9, 7, 10, 1] with both upper and lower bounds on the approximability of these objective functions on worst case instances.

Recently, Bilu and Linial [8] 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 [8] specifically define an instance to be α-perturbation resilient[1] for an objective Φ if perturbing pairwise distances by multiplicative factors in the range $[1, \alpha]$ does not change the optimum clustering under Φ.[2] They consider in detail the case of max-cut clustering and give an efficient algorithm to recover the optimum when the instance is resilient to perturbations on the order of $\alpha = O(\sqrt{n})$.

Two important questions raised by the work of Bilu and Linial [8] 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 [8] and additionally improve over [2]. 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 [2]. 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 $O(\delta^{-1} \log^{1+\delta} n)$-approximation algorithm that runs in time $n^{O(1/\delta)}$ due to Bartal et al. [7]; by contrast, the best guarantee known for k-median is factor $3 + \epsilon$.

Our results are achieved by carefully deriving structural properties of perturbation-resilience. At a high

[1] Bilu and Linial [8] refer to such instances as perturbation stable instances.

[2] Of course, the *score* of the optimum solution will change; what the definition requires is that the partitioning induced by the optimum remains the same.

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 approximate closure 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 [17, 18]. 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 [8] 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 [2] for α-perturbation resilient instances for center-based objectives, giving an algorithm that efficiently[3] 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 $\alpha > 2+\sqrt{7}$ 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 closer to their own center than to any other center. Using this fact, we then design an approximate version of the closure linkage criterion that allows us to carefully eliminate the noise introduced by the bad points and construct a tree that has 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-min-sum perturbation resilient instances. Our algorithm is based on an appropriate modification of average linkage that exploits the structure of min-sum perturbation resilient instances.

We also provide sublinear-time algorithms both for the k-median and min-sum objectives (Sections 4.3 and 5), showing algorithms that can return an implicit clustering from only access to a small random sample.

Related Work: In the context of objective based clustering, several recent papers have showed how to exploit various notions of stability for overcoming the existing hardness results on worst case instances. These include the stability notion of Ostrovsky et al. [15, 2] that assumes that the instance has the property that the cost of the optimal k-means solution is small compared to the cost of the optimal $(k-1)$-means solution and the approximation stability condition of Balcan et al. [3] that assumes that every nearly optimal solution is close to the target clustering.

Even closer to our work, 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 [5, 13] and the classic traveling salesman problem [14].

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

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 $d : S \times S \to \mathbb{R}_{\geq 0}$ 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 w.r.t. Φ is denoted as $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, and its cost is denoted as $\mathcal{OPT}$. The core concept we study in this paper is the perturbation resilience notion introduced by [8]. Formally:

Definition 1. *A clustering instance (S, d) is α-**perturbation resilient** to a given objective Φ if for any function $d' : S \times S \to \mathbb{R}_{\geq 0}$ s.t. $\forall p, q \in S, d(p, q) \leq d'(p, q) \leq \alpha d(p, q)$, 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.*

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 [2].

Definition 2. *A clustering objective is center-based if the optimal solution can be defined by k points $c_1, \cdots, c_k$ 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:*

- *The objective function value of a given clustering is either a (weighted) sum or the maximum of the individual cluster scores.*
- *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 $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ and assign a set of centers $\mathbf{p} = \{p_1, p_2, \ldots, p_k\} \subseteq S$ for the subsets. The goal is to minimize $\Phi(\mathcal{P}, \mathbf{p}) = \sum_{i=1}^{k} \sum_{p \in P_i} d(p, p_i)$. The centers in the optimal clustering are denoted as $\mathbf{c} = \{c_1, \ldots, c_k\}$. 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 $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$, and the goal is to minimize $\Phi(\mathcal{P}) = \sum_{i=1}^{k} \sum_{p,q \in P_i} d(p, q)$. 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 Definition 1 where i.e. we allow a small difference between the original optimum and the new optimum after perturbation. Formally:

Definition 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 4. *A clustering instance (S, d) is (α, ϵ)-**perturbation resilient** to a given objective Φ if for any function $d' : S \times S \to \mathbb{R}_{\geq 0}$ 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 $A, B \subseteq S$ we define $d_{sum}(A, B) \doteq \sum_{p \in A} \sum_{q \in B} d(p, q)$ and $d_{sum}(p, B) \doteq d_{sum}(\{p\}, B)$. For simplicity, we will sometimes assume that $\min_i |C_i|$ is known. (Otherwise, we can simply search over the n possible different values.)

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 [2] 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 [2].

Definition 5. *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 1. *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 [2]. 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 2. *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, when $\alpha \geq 1 + \sqrt{2}$, we have: (1) $d(c_i, q) > d(c_i, p)$, (2) $d(p, c_i) < d(p, q)$.*

Proof. (1) Lemma 1 gives us that $d(q, c_i) > \alpha d(q, c_j)$. By the triangle inequality, we have $d(c_i, c_j) \leq d(q, c_j) + d(q, c_i) < (1 + \frac{1}{\alpha}) d(q, c_i)$. 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) Also by triangle inequality, $d(p, q) > (\alpha - 1) \max(d(p, c_i), d(q, c_j))$. (Its proof appears in [2].) $\square$

Lemma 2 implies that 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 $\mathbb{B}(p, r) = \{q : d(q, p) \leq r\}$.

Definition 6. *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', i.e. $A \cup A' \subseteq \mathbb{B}(c, d)$;
(2) margin: points inside $\mathbb{B}(c, d)$ are closer to the center c than to points outside,
i.e. $\forall p \in \mathbb{B}(c, d), q \notin \mathbb{B}(c, d)$, we have $d(c, p) < d(p, q)$.

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

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

Algorithm 1 Center-based objectives, α perturbation resilience

Input: Data set S, distance function $d(\cdot,\cdot)$ on S.
Phase 1: Begin with n singleton clusters.
• Repeat till only one cluster remains: merge clusters C, C' which minimize $d_S(C, C')$.
• Let T be the tree with single points as leaves and internal nodes corresponding to the merges performed.
Phase 2: Apply dynamic programming on T to get the minimum cost pruning $\tilde{\mathcal{C}}$.
Output: Clustering $\tilde{\mathcal{C}}$.

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 F.1, and an efficient implementation of the algorithm is presented in Appendix F.2.

Theorem 2. *For $(1+\sqrt{2})$-center proximity instances, Phase 1 of Algorithm 1 (the closure linkage phase) constructs a binary tree s. t. 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 $A \subseteq C$, or $C \subseteq A$ or $A \cap C = \varnothing$. 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 $d_S(A, B) < d_S(A, A') = d_S(A', A)$.

We first prove that there is a cluster $B \subseteq C_i \setminus A$ in the current cluster list such that $d_S(A, B) \leq d = \max_{p \in C_i} d(c_i, p)$. 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 $d_S(B, A) \leq d$ since there is $c_i \in B$, and (1) for any $p \in A \cup B$, $d(c_i, p) \leq d$, (2) for any $p \in S$ satisfying $d(c_i, p) \leq d$, and any $q \in S$ satisfying $d(c_i, q) > d$, by Lemma 2 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 $d_S(A, B) \leq d$.

As a second step, we need to show that $d < \hat{d} = d_S(A, A')$. There are two cases: the center for $d_S(A, A')$ is in A or in A'. 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 2 $d(c_j, q) < d(c, q)$, we must have $d(c_j, c) \leq \hat{d}$ (otherwise it violates the second requirement of closure distance). Suppose $p = \arg\max_{p' \in C_i} d(c_i, p')$. Then we have $d = d(p, c_i) < d(p, c_j)/\alpha \leq (d + d(c_i, c) + d(c, c_j))/\alpha$ where the first inequality comes from Lemma 1 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 d and $d(c, c_j)$, and when $\alpha \geq 1 + \sqrt{2}$ we have $d < d(c, c_j) \leq \hat{d}$.

Now consider the second case, when there is a point $c \in A'$ such that c and $\hat{d}$ 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 2. Then from the second requirement of closure distance $d(c_i, c) \leq \hat{d}$. And by Lemma 2, $d = d(c_i, p) < d(c_i, c)$, we have $d < d(c_i, c) \leq \hat{d}$. □

Note: Our factor of $\alpha = 1 + \sqrt{2}$ beats the NP-hardness *lower bound* of $\alpha = 3$ of [2] for center-proximity instances. The reason is that the lower bound of [2] 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 [2] 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 [16].

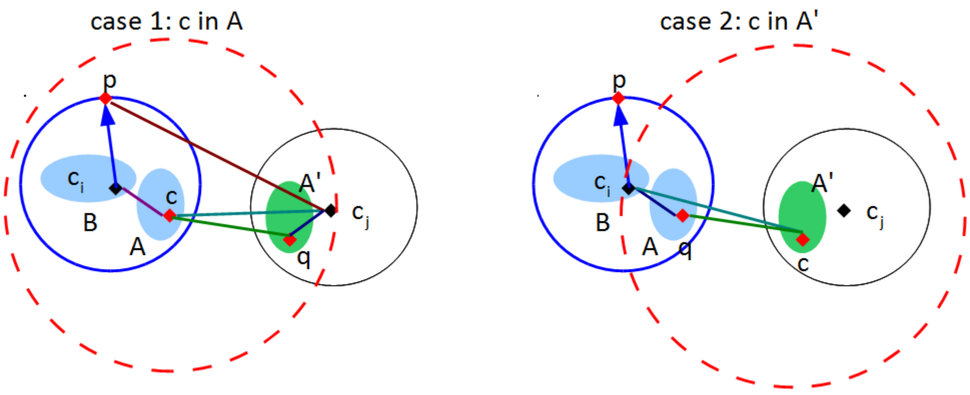

Figure 1: Illustration for comparing d and $d_S(A, A')$ in Theorem 2

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 $\alpha > 2 + \sqrt{7}$ 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 when $\epsilon' \leq 2$ 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 [11, 12] 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

For understanding the (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 d', defined as $\min_{\sigma \in \mathcal{S}_k} \sum_{i=1}^{k} |C_i \setminus C'_{\sigma(i)}|$. Without loss of generality, we assume in this subsection that $\mathcal{C}'$ is indexed so that the argmin σ is the identity, and the distance between $\mathcal{C}$ and $\mathcal{C}'$ is $\sum_{i=1}^{k} |C_i \setminus C'_i|$. We will denote by c'_i the center of C'_i.

In the following we call a point *good* if it is α times closer to its own center than to any other center in the optimal clustering; otherwise we call it *bad*. Let B_i be the set of bad points in C_i. That is, $B_i = \{p : p \in C_i, \exists j \neq i, \alpha d(c_i, p) > d(c_j, p)\}$. Let $G_i = C_i \setminus B_i$ be the good points in cluster i. Let $B = \cup_i B_i$ and $G = \cup_i G_i$. We show that under perturbation resilience we do not have too many bad points. Formally:

Theorem 3. *Suppose the clustering instance is (α, ϵ)-perturbation resilient and $\min_i |C_i| > (2 + \frac{2\alpha}{\alpha-1})\epsilon n + \frac{2\alpha(\alpha+1)}{\alpha-1}$. Then $|B| \leq \epsilon n$.*

Here we provide a proof sketch of the theorem, and the full proof can be found in Appendix A. At the end of Appendix A, we also point out that the bound in Theorem 3 is an optimal bound for the bad points in the sense that for any $\alpha > 1$ and $\epsilon < \frac{1}{5}$, we can easily construct an (α, ϵ)-perturbation resilient 2-median instance which has ϵn bad points.

*Proof Sketch of [*Theorem 3] The main idea is to construct a specific perturbation that forces certain selected bad points to move from their original optimal clusters. Then the (α, ϵ)-perturbation resilience leads to a bound on the number of selected bad points, which can also be proved to be a bound on all the bad points. The selected bad points $\hat{B}_i$ in cluster i are defined by arbitrarily selecting $\min(\epsilon n + 1, |B_i|)$ points from B_i. Let $\hat{B} = \cup_i \hat{B}_i$. For $p \in \hat{B}_i$, let $c(p) = \arg\min_{c_j, j \neq i} d(p, c_j)$ denote its second nearest center; for $p \in C_i \setminus \hat{B}_i$, $c(p) = c_i$. The perturbation we consider blows up all distances by a factor of α except for those distances between p and $c(p)$. Formally, we define d' as $d'(p,q) = d(p,q)$ if $p = c(q)$ or $q = c(p)$, and $d'(p,q) = \alpha d(p,q)$ otherwise.

The key challenge in proving a bound on the selected bad points is to show that $c'_i = c_i$ for all i, i.e., the optimal centers do not change after the perturbation. Then in the optimum under d' each point p is assigned to the center $c(p)$, and therefore the selected bad points ($\hat{B}$) will move from their original optimal clusters. By (α, ϵ)-perturbation resilience property we get an upper bound on the number of selected bad points.

Suppose C'_i is obtained by adding point set A_i and removing point set M_i from C_i, i.e. $A_i = C'_i \setminus C_i, M_i = C_i \setminus C'_i$. Let $A = \cup_i A_i,\ M = \cup_i M_i$. At a high level, we prove that $c_i = c'_i$ for all i as follows. We first show that for each cluster, its new center is close to its old center, roughly speaking since the new and old cluster have a lot in common (Claim 1). We then show if $c_i \neq c'_i$ for some i, then the weighted sum of the distances $\sum_{1 \le i \le k} (|A_i| + \alpha + 1 + |M_i|) d(c_i, c'_i)$ should be large (Claim 2). However, this contradicts Claim 1, so the centers do not move after the perturbation.

In proving Claims 1 and 2, we need to translate d' to d, and for doing so, a key fact we show is that for any i, $c'_i \neq c_j$ for any $j \neq i$. The intuition is that if $c'_i = c_j$, then $c'_i = c_j$ should be farther to the points in $C'_j \cap C_j$ than c'_j, thus c_j should save a lot of cost on other parts of C_j than c'_j, and by the triangle inequality, c_j should be far from c'_j; on the other hand, they are both close to points in $C'_j \cap C_j$, which is contradictory.

Claim 1. *For each i, $d_{sum}(c_i, (C_i \cap C'_i) \setminus \hat{B}_i) \ge \frac{\alpha}{\alpha+1}(|(C_i \cap C'_i) \setminus \hat{B}_i| - |\hat{B}_i \setminus M_i| - |A_i| - (\alpha+1)) d(c_i, c'_i)$, i.e., $d(c_i, c'_i)$ can be bounded approximately by the average distance between c_i and a large portion of C_i.*

Proof Sketch: The key idea is that under d', c'_i is the optimal center for C'_i, so it has no more cost than c_i on C'_i. Since $\hat{B}_i \setminus M_i$ and A_i are small compared to $(C_i \cap C'_i) \setminus \hat{B}_i$, c'_i cannot save much cost on $\hat{B}_i \setminus M_i$ and A_i, thus it cannot have much more cost on $(C_i \cap C'_i) \setminus \hat{B}_i$ than c_i. Then c'_i is close to $(C_i \cap C'_i) \setminus \hat{B}_i$, and so is c_i, then c'_i is close to c_i. Formally, we have $d'_{sum}(c'_i, C'_i) \le d'_{sum}(c_i, C'_i)$. We divide C'_i into three parts $(C_i \cap C'_i) \setminus \hat{B}_i, \hat{B}_i \setminus M_i$ and A_i, and move terms on $(C_i \cap C'_i) \setminus \hat{B}_i$ to one side (the cost more than c_i on $(C_i \cap C'_i) \setminus \hat{B}_i$), the rest terms to another side (the cost saved on $\hat{B}_i \setminus M_i$ and A_i). After translating the terms from d' to d, we apply the triangle inequality and obtain the desired result. □

Claim 2. *Let $I_i = 1$ if $c_i \neq c'_i$ and $I_i = 0$ otherwise. Then* $(\alpha - 1) \sum_{1 \le i \le k} I_i d_{sum}(c_i, (C_i \cap C'_i) \setminus \hat{B}_i) \le \alpha \sum_{1 \le i \le k} (|A_i| + \alpha + 1 + |M_i|) d(c_i, c'_i)$.

Proof Sketch: We have $\sum_i d_{sum}(c_i, C_i) \le \sum_i d_{sum}(c'_i, C_i)$ by using the fact that c_i are the optimal centers for C_i under d; and $\sum_i d'_{sum}(c'_i, C'_i) \le \sum_i [d'_{sum}(c_i, C'_i - \hat{B}_i) + \sum_{p \in \hat{B}_i \setminus M_i} d'(c(p), p)]$ by using the fact that c'_i are the optimal centers for C'_i under d'. We multiply the first inequality by α and add it to the second inequality. Then we divide C_i into three parts $M_i, \hat{B}_i \setminus M_i$ and $(C_i \cap C'_i) \setminus \hat{B}_i$, C'_i into $A_i, \hat{B}_i \setminus M_i$ and $(C_i \cap C'_i) \setminus \hat{B}_i$, and $C'_i \setminus \hat{B}_i$ into A_i and $(C_i \cap C'_i) \setminus \hat{B}_i$. After translating the terms from d' to d, we notice that the clustering that under d' assigns points in $C'_i \setminus \hat{B}_i$ to c_i and points p in $\hat{B}_i \setminus M_i$ to $c(p)$ (corresponding to the right hand side of the second inequality) saves as much cost as $(\alpha - 1) \sum_i d_{sum}(c_i, (C_i \cap C'_i) \setminus \hat{B}_i)$ on $(C_i \cap C'_i) \setminus \hat{B}_i$ compared to the optimum clustering under d'. So, the optimum clustering under d' must save this cost on other parts, i.e. A_i and $\hat{B}_i \setminus M_i$. By triangle inequality, we obtain the desired result. □

Combining Claims 1 and 2, we get

$$\sum_{1\le i\le k} \alpha d(c_i,c_i')\big[(|A_i|+\alpha+1+|M_i|) - \frac{\alpha-1}{\alpha+1}(|(C_i\cap C_i')\setminus \hat{B}_i| - |\hat{B}_i\setminus M_i| - |A_i| - (\alpha+1))I_i\big] \ge 0.$$

If $I_i = 0$, we have $d(c_i, c_i') = 0$; if $I_i = 1$, since $|C_i| > (2+\frac{2\alpha}{\alpha-1})\epsilon n + \frac{2\alpha(\alpha+1)}{\alpha-1}$, the coefficient of $d(c_i, c_i')$ is negative. So the left hand side is no greater than 0. Therefore, all terms are equal to 0, i.e. for all $1 \le i \le k$, $d(c_i, c_i') = 0$. Then points in $\hat{B}_i$ will move to other clusters after perturbation, which means that $\hat{B}_i \subseteq M_i$, thus $\hat{B} \subseteq M$. Then $|\hat{B}| \le |M| \le \epsilon n$. Specially, $|\hat{B}_i| \le \epsilon n$ for any i. Then $|B_i| \le \epsilon n$, otherwise $|\hat{B}_i|$ would be $\epsilon n + 1$. So $\hat{B}_i = B_i$, and $\hat{B} = B$ and $|B| = |\hat{B}| \le \epsilon n$. □

4.2 Approximating the Optimum Clustering

Since (α,ϵ)-perturbation resilient instances have at most ϵn bad points, we can show that for $\alpha > 4$ such instances satisfy the ϵ-strict separation property (the property that after eliminating an ϵ fraction of the points, the remaining points are closer to points in their own cluster than to other points in different clusters). Therefore, we could use the clustering algorithms in [4, 6] to output a hierarchy such that the optimal clustering is ϵ-close pruning of this tree. However, this pruning might not have a small cost and it is not clear how to retrieve a small cost clustering from the tree constructed by these generic algorithms. In this section, we design a new algorithm for obtaining a good approximation to the optimum for (α,ϵ)-perturbation resilient instances. This algorithm uses a novel linkage procedure to first construct a tree that we further process to output a desired clustering. This linkage based procedure uses an approximate version of the closure condition discussed in Section 3. We begin with the definition of approximate closure condition.

Definition 7. *Let $p, q \in S$ and assume $\mathcal{C}'$ is a clustering of S. Let $U_{p,q} = \{C | C \in \mathcal{C}', |C \setminus \mathbb{B}(p, d(p,q))| \le \epsilon n, C \cap \mathbb{B}(p, d(p,q)) \ne \emptyset\}$. The ball $\mathbb{B}(p, d(p,q))$ satisfies the approximate closure condition with respect to $\mathcal{C}'$ if $|\cup_{C\in U_{p,q}} C| \ge \min_i |C_i| - \epsilon n$ and the following conditions are satisfied:*
(1) approximate coverage: the ball covers most of $U_{p,q}$, i.e. $|\cup_{C\in U_{p,q}} C_i \setminus \mathbb{B}(p, d(p,q))| \le \epsilon n$;[4]
(2) approximate margin: after removing a few points outside the ball, points inside are closer to each other than to points outside, i.e. $\exists E \subseteq S \setminus \mathbb{B}(p, d(p,q)), |E| \le \epsilon n$, such that for any $p_1, p_2 \in \mathbb{B}(p, d(p,q))$, $q_1 \in S \setminus \mathbb{B}(p, d(p,q)) \setminus E$, we have $d(p_1,p_2) < d(p_1,q_1)$.

We are now ready to present our main algorithm for the (α,ϵ)-perturbation resilient instances, Algorithm 2. Informally, it starts with singleton points in their own clusters. It then checks in increasing order of $d(p,q)$ whether the approximate closure condition is satisfied for $\mathbb{B}(p, d(p,q))$, and if so it merges all the clusters in the current clustering nearly contained within the ball $\mathbb{B}(p, d(p,q))$. As we show below, the tree produced has a pruning that respects the optimal clustering. However, this pruning may contain more than k-clusters, so in the second phase, we clean the tree so that we can ensure there is a pruning with k-clusters that coincides with the optimal clustering on the good points. Finally we run dynamic programming to get the minimum cost pruning, which as we prove provides a good approximation to the optimal clustering.

Our main result in this section is the following.

Theorem 4. *Suppose the clustering instance is (α,ϵ)-perturbation resilient to k-median. If $\alpha > 2+\sqrt{7}$ and $\epsilon \le \rho/5$ where $\rho = (\min_i |C_i| - 15)/n$, then in polynomial time, Algorithm 2 outputs a tree $\tilde{T}$ that contains a pruning that is ϵ-close to the optimum clustering. Moreover, if $\epsilon \le \rho\epsilon'/5$ where $\epsilon' \le 1$, the clustering produced is a $(1+\epsilon')$-approximation to the optimum.*

[4]Note that in the definition of $U_{p,q}$, each cluster in it has at most ϵn points outside the ball $\mathbb{B}(p, d(p,q))$. But the approximate coverage is stronger: $U_{p,q}$, as a whole, can have at most ϵn outside.

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

Input: Data set S, distance function $d(\cdot,\cdot)$ on S, $\min_i |C_i|$, $\epsilon > 0$
Phase 1: Sort all the pairwise distances $d(p,q)$ in ascending order.
• Initialize $\mathcal{C}'$ to be the clustering with each singleton point being a cluster.
• For $d(p,q)$ in ascending order,
•   If $\mathbb{B}(p, d(p,q))$ satisfies approximate closure condition and $|U_{p,q}| > 1$ then merge $U_{p,q}$.
• Construct the tree T with points as leaves and internal nodes corresponding to the merges performed.
Phase 2: If a node has only singleton points as children, delete his children from T; get T'.
• For any remaining singleton node p in T', assign p to the non-singleton leaf of smallest median distance.
Phase 3: Apply dynamic programming on the cleaned $\tilde{T}$ to get the minimum k-median cost pruning $\tilde{\mathcal{C}}$.
Output: Clustering $\tilde{\mathcal{C}}$, (optional) tree $\tilde{T}$.

Theorem 4 follows immediately from the following lemmas. The details can be found in Appendix B.

Lemma 3. *If $\alpha > 2+\sqrt{7}$, $\epsilon \leq \rho/5$, then the tree T contains nodes $N_i (1 \leq i \leq k)$ such that $N_i \setminus B = C_i \setminus B$.*

Proof Sketch: For each i, we let $q_i^* = \arg\max_{q_i \in C_i \setminus B} d(c_i, q_i)$. The proof follows from two key facts: (1) If $\mathcal{C}' \setminus B$ is laminar to $\mathcal{C} \setminus B$ right before checking some $d(p,q)$, then for any i, j, $i \neq j$ such that either $d(p,q)$ is checked before $d(c_i, q_i^*)$ or $d(p,q)$ is checked before $d(c_j, q_j^*)$ or $p = c_i, q = q_i^*$ or $p = c_j, q = q_j^*$, we have that $U_{p,q}$ cannot contain both good points from C_i and C_j. (2) If $\mathcal{C}' \setminus B$ is laminar to $\mathcal{C} \setminus B$ right before checking $d(c_i, q_i^*)$, we have that right after checking $d(c_i, q_i^*)$ there is a cluster containing all the good points in cluster i and no other good points.

Consider any merge step s.t. $U_{p,q}$ contains good points from both C_i and $C_j (j \neq i)$. Fact (1) implies both $d(c_i, q_i^*)$ and $d(c_j, q_j^*)$ must have been checked, so fact (2) implies all good points in C_i and C_j respectively have already been merged. So the laminarity is always satisfied. Then the lemma follows from fact (2).

We now prove fact (1). Suppose for contradiction that there exist good points from C_i and C_j in $U_{p,q}$. From the laminarity assumption, the fact that clusters in $U_{p,q}$ have only ϵn points outside $\mathbb{B}(p, d(p,q))$ and $|B| \leq \epsilon n$, we can show there exist good points $p_i \in C_i$ and $p_j \in C_j$ in $\mathbb{B}(p, d(p,q))$. When $\alpha > 2+\sqrt{7}$ we can show $d(c_i, q_i^*) < d(p_i, p_j)/2$, and by triangle inequality $d(p_i, p_j)/2 \leq d(p,q)$, so $d(p,q) > d(c_i, q_i^*)$. The same argument leads to $d(p,q) > d(c_j, q_j^*)$. This is a contradiction to the assumption that $d(p,q)$ is checked before $d(c_i, q_i^*)$ or before $d(c_j, q_j^*)$ or $p = c_i, q = q_i^*$ or $p = c_j, q = q_j^*$.

We now prove fact (2). It is sufficient to show that $\cup_{C \in U_{c_i, q_i^*}} C \setminus B = C_i \setminus B$ and U_{c_i, q_i^*} satisfies the approximate closure condition. First, U_{c_i, q_i^*} contains no good points outside C_i by fact (1). Second, any C containing good points from C_i is in U_{c_i, q_i^*}. By fact (1), C has no good points outside C_i. Since $\mathbb{B}(c_i, d(c_i, q_i^*))$ contains all good points in C_i, C has only bad points outside the ball, so $C \in U_{c_i, q_i^*}$. We finally show U_{c_i, q_i^*} satisfies the approximate closure condition. Since in addition to all good points in C_i, $\cup_{C \in U_{c_i, q_i^*}} C$ can only contain bad points, it has at most ϵn points outside $\mathbb{B}(c_i, d(c_i, q_i^*))$, so approximate coverage condition is satisfied. And we can show for $\alpha > 2+\sqrt{7}$, $2d(c_i, q_i^*)$ is smaller than the distance between any point in $\mathbb{B}(c_i, d(c_i, q_i^*))$ and any good point outside C_i. Then let $E = B \setminus \mathbb{B}(c_i, d(c_i, q_i^*))$, approximate margin condition is satisfied. We also have $|\cup_{C \in U_{c_i, q_i^*}} C| \geq |C_i \setminus B| \geq \min_i |C_i| - \epsilon n$. □

Lemma 4. *If $\alpha > 2+\sqrt{7}$, $\epsilon \leq \epsilon' \rho/5$ where $\epsilon' \leq 1$, then $\tilde{\mathcal{C}}$ is a $(1+\epsilon')$-approximation to the optimum.*

Proof Sketch: By Lemma 3, T has a pruning $\mathcal{P}$ that contains $N_i (1 \leq i \leq k)$ and possibly some bad points, such that $N_i \setminus B = C_i \setminus B$. Therefore, each non-singleton leaf in T' has only good points from one optimal cluster and has more good points than bad points. This then implies that each singleton good points in T' is assigned to a leaf that has good points from its own optimal cluster.

So $\mathcal{P}$ becomes $\mathcal{P}' = \{N_i'\}$ such that $N_i' \backslash B = C_i \backslash B$. It is sufficient to prove the cost of $\mathcal{P}'$ approximates $\mathcal{OPT}$, i.e. to bound the increase of cost caused by a bad point $p_j \in C_j$ ending up in $N_i' (i \neq j)$. There are two cases: p_j belongs to a non-singleton leaf node in T' or p_j is a singleton in T'. In either case, the leaf in which p_j ends up in $\tilde{T}$ contains at least $K = (\min_i |C_i| - \epsilon n)/2 - \epsilon n$ good points p_{it} from C_i such that for any other leaf containing only good points from C_j we can find at least K good points p_{js} from C_j satisfying $d(p_j, p_{it}) \leq d(p_j, p_{js})$. Then the increase of cost due to p_j can be bounded as follows.

$$d(p_j, c_i) - d(p_j, c_j) \leq \frac{1}{K}\left\{\sum_{1 \leq t \leq K}\left[d(p_j, p_{it}) + d(p_{it}, c_i)\right] - \sum_{1 \leq s \leq K}\left[d(p_j, p_{js}) - d(p_{js}, c_j)\right]\right\} \leq \frac{1}{K}\mathcal{OPT}.$$

As $|B| \leq \epsilon n$, the cost of $\mathcal{P}'$ is $\leq (1 + \frac{\epsilon n}{K})\mathcal{OPT}$, so the minimum cost pruning $\tilde{\mathcal{C}}$ is a $(1 + \frac{2\epsilon n}{\min_i |C_i| - 3\epsilon n})$-approximation to the optimum. By setting $\epsilon' \geq \frac{2\epsilon n}{\min_i |C_i| - 3\epsilon n}$, we get the desired result. □

We note that approximate margin condition in the Definition 7 can be verified in $O(n^3)$ time by enumerating $p_1, p_2 \in \mathbb{B}(p, d(p, q)), q_1 \notin \mathbb{B}(p, d(p, q))$, and checking if there are no more than ϵn such q_1 that there exist p_1, p_2 violating the condition. So Algorithm 2 runs in polynomial time.

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. Let $N = |X|$, $\rho = \min_i |C_i|/N$ denote the fraction of the points in the smallest cluster, $D = \max_{p,q \in X} d(p, q)$ denote the diameter of X, $\zeta = \Phi_X(\mathbf{c})/N$ denote the average cost of the points in the optimum clustering. We have:

Theorem 5. *Suppose (X, d) is (α, ϵ)-perturbation resilient for $\alpha > 8$, $\epsilon < \rho/20$. Let $0 < \lambda < 1$. Then w.p. $\geq 1 - \delta$, we can get an implicit clustering that is $2\frac{1+\lambda}{1-\lambda}(1 + \frac{8\epsilon}{\rho - 12\epsilon})$-approximation in time $O((\frac{kD^2}{\lambda^2\epsilon^2\zeta^2}\ln\frac{N}{\delta})^5)$.*

Proof Sketch: We sample a set S of size $n = \Theta(\frac{kD^2}{\lambda^2\epsilon^2\zeta^2}\ln\frac{N}{\delta})$ and run Algorithm 2 on S to obtain the minimum cost pruning $\tilde{\mathcal{C}}$ and its centers $\tilde{\mathbf{c}}$. We then output the implicit clustering of the whole space X that assigns each point in X to its nearest neighbor in $\tilde{\mathbf{c}}$. Here we describe a proof sketch that shows $\Phi_X(\tilde{\mathbf{c}}) \approx \Phi_X(\mathbf{c})$, and the full proof can be found in Appendix C.

Since when n is sufficiently large, w.h.p. $\Phi_X(\tilde{\mathbf{c}})/N \approx \Phi_S(\tilde{\mathbf{c}})/n$ and $\Phi_X(\mathbf{c})/N \approx \Phi_S(\mathbf{c})/n$, so it is sufficient to show $\Phi_S(\tilde{\mathbf{c}})$ is not much larger than $\Phi_S(\mathbf{c})$. However, $\tilde{\mathcal{C}}$ may be different from $\mathcal{C} \cap S$, so we need a bridge for the two.

Notice w.h.p. S has only $2\epsilon n$ bad points, and each cluster $C_i \cap S$ is large. Moreover, when $\alpha > 8$, any good point is 3 times closer to those in the same cluster than to those in other clusters. Then even if $\mathbf{c}$ are not sampled, we can choose an arbitrary good point from each cluster $C_i \cap S$ to be its center, so that we can still prove Algorithm 2 forms nodes N_i approximating the clusters, and $\tilde{T}$ has a pruning $\mathcal{P}'$, which is different from $\mathcal{C} \cap S$ only on the bad points. 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, $\Phi_S(\tilde{\mathbf{c}}) \leq \Phi_S(\mathcal{P}', \mathbf{c}')$. This is because (1) since $\tilde{\mathcal{C}}$ is the minimum cost pruning, $\Phi_S(\tilde{\mathcal{C}}, \tilde{\mathbf{c}}) \leq \Phi_S(\mathcal{P}', \mathbf{c}')$; (2) since in $\Phi_S(\tilde{\mathbf{c}})$ each point is assigned to its nearest center but in $\Phi_S(\tilde{\mathcal{C}}, \tilde{\mathbf{c}})$ this may not be true, $\Phi_S(\tilde{\mathbf{c}}) \leq \Phi_S(\tilde{\mathcal{C}}, \tilde{\mathbf{c}})$. On the other hand, $\Phi_S(\mathcal{P}', \mathbf{c}')$ is not much larger than $\Phi_S(\mathbf{c})$. This is because (1) $\Phi_S(\mathcal{P}', \mathbf{c})$ is different from $\Phi_S(\mathbf{c})$ only on the bad points, so by the approach similar to that in Lemma 4, we can show the increase of cost is limited; (2) by the triangle inequality we have $\Phi_S(\mathcal{P}', \mathbf{c}') \leq 2\Phi_S(\mathcal{P}', \mathbf{c})$. □

Note: 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 min-sum k-clustering problem (Algorithm 3). We use the following notations: $d_{avg}(A, B) \doteq d_{sum}(A, B)/(|A||B|)$ and $d_{avg}(p, B) = d_{avg}(\{p\}, B)$.

Algorithm 3 min-sum, α perturbation resilience

Input: Data set S, distance function $d(\cdot, \cdot)$ on S, $\min_i |C_i|$.
Phase 1: Connect each point with its $\frac{1}{2}\min_i |C_i|$ nearest neighbors.
- Initialize the clustering $\mathcal{C}'$ with each connected component being a cluster.
- Repeat till only one cluster remains in $\mathcal{C}'$: merge clusters C, C' in $\mathcal{C}'$ which minimize $d_{avg}(C, C')$.
- Let T be the tree with components as leaves and internal nodes corresponding to the merges performed.

Phase 2: Apply dynamic programming on T to get the minimum min-sum cost pruning $\tilde{\mathcal{C}}$.
Output: Output $\tilde{\mathcal{C}}$.

Theorem 6. *For $(3\frac{\max_i |C_i|}{\min_i |C_i|-1})$-perturbation resilient instances, Algorithm 3 outputs the optimal min-sum k-clustering in polynomial time.*

Proof Sketch: First we show that the α-perturbation resilience property implies that for any two different optimal clusters C_i and C_j and any $A \subseteq C_i$, we have $\alpha d_{sum}(A, C_i \setminus A) < d_{sum}(A, C_j)$. 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 $\alpha > 3\frac{\max_i |C_i|}{\min_i |C_i|-1}$ we have: (1) for any optimal clusters C_i and C_j and any $A \subseteq C_i$, $A' \subseteq C_j$ s.t. $\min(|C_i \setminus A|, |C_j \setminus A'|) > \min_i |C_i|/2$ we have $d_{avg}(A, A') > \min\{d_{avg}(A, C_i \setminus A), d_{avg}(A', C_j \setminus A')\}$; (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. Fact (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 T are laminar to the optimum clustering. Fact (1) can be used to show that the merge steps preserve the laminarity with the optimal clustering, so the minimum cost pruning of T will be the optimal clustering, as desired. The full proof can be found in Appendix D. □

Theorem 7. *Suppose the clustering instance (X, d) is α-perturbation resilient to the min-sum objective where $\alpha \geq 6\frac{\max_i |C_i|}{\min_i |C_i|-1}$. Then w.p. $\geq 1-\delta$, we can get an implicit optimum clustering in time $O((\frac{D^2}{\rho^2\eta^2}\ln\frac{Nk}{\delta})^3)$ where $\eta = \min_{p\in X, 1\leq i\leq k} d_{avg}(p, C_i)$ is the minimum average distance between points and optimal clusters.*

Proof Sketch: We sample a set S of size $n = \Theta(\frac{D^2}{\rho^2\eta^2}\ln\frac{Nk}{\delta})$ and run Algorithm 3 on S. We then output the implicit clustering of the whole space X that assigns each point $p \in X$ to $\tilde{C}_i \in \tilde{\mathcal{C}}$ s. t. $d_{sum}(p, \tilde{C}_i)$ is minimized. We have that for any $p \in C_i$ and $C_j (j \neq i)$,$3\frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S|-1} d_{sum}(p, C_i \cap S) < d_{sum}(p, C_j \cap S)$, since when n is sufficiently large, $d_{sum}(p, C_i \cap S) \approx d_{sum}(p, C_i)|S|/|X|$, $d_{sum}(p, C_j \cap S) \approx d_{sum}(p, C_j)|S|/|X|$ and $\frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S|-1} \approx \frac{\max_i |C_i|}{\min_i |C_i|-1}$. So the tree T is laminar to $\mathcal{C} \cap S$. Since 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 $\mathcal{C} \cap S$ is the minimum cost pruning, so Algorithm 3 on S outputs $\mathcal{C} \cap S$, and the theorem follows. The full proof can be found in Appendix E. □

6 Discussion and Open Questions

In this work, we advance the line of research on perturbation resilience in clustering 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, 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 spaces[5]. 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).

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*, 33(3), 2004.

[2] P. Awasthi, A. Blum, and O. Sheffet. Center-based clustering under perturbation stability. *Inf. Process. Lett.*, 112:49–54, Jan. 2012.

[3] M. F. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2009.

[4] M. F. Balcan, A. Blum, and S. Vempala. A discriminative framework for clustering via similarity functions. In *STOC*, 2008.

[5] M. F. Balcan and M. Braverman. Approximate nash equilibria under stability conditions. *CoRR*, abs/1008.1827, 2010.

[6] M. F. Balcan and P. Gupta. Robust hierarchical clustering. In *COLT*, 2010.

[7] Y. Bartal, M. Charikar, and D. Raz. Approximating min-sum -clustering in metric spaces. In *STOC*, 2001.

[8] Y. Bilu and N. Linial. Are stable instances easy? In *Innovations in Computer Science*, 2010.

[9] M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *J. Comput. Syst. Sci.*, 65(1), 2002.

[10] W. F. de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani. Approximation schemes for clustering problems. In *STOC*, 2003.

[11] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *J. Algorithms*, 31(1):228–248, 1999.

[12] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *STOC*, 2002.

[13] R. J. Lipton, E. Markakis, and A. Mehta. On stability properties of economic solution concepts. Manuscript, 2006.

[5]That is, where d is a Euclidean metric, though as in Definitions 1 and 4, d' need not be. Alternatively, one could also consider a natural version of Definitions 1 and 4 in which d' must be Euclidean as well, and in fact implemented via a perturbation of coordinate values.

[14] M. Mihalák, M. Schöngens, R. Srámek, and P. Widmayer. On the complexity of the metric tsp under stability considerations. In *SOFSEM*, pages 382–393, 2011.

[15] R. Ostrovsky, Y. Rabani, L. Schulman, and C. Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *FOCS*, 2006.

[16] L. Reyzin. Personal communication. 2011.

[17] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[18] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., 1998.

A Full Proof of Theorem 3

The main idea for proving the theorem is to construct a specific perturbation that forces certain selected bad points to move from their original optimal clusters. Then the (α, ϵ)-perturbation resilience leads to a bound on the number of the selected bad points, which can also be proved to be a bound on the number of all the bad points.

Specifically, the selected bad points $\hat{B}_i$ in cluster i are defined by arbitrarily selecting $\min(\epsilon n + 1, |B_i|)$ points from B_i. Let $\hat{B} = \cup_i \hat{B}_i$. For $p \in \hat{B}_i$, let $c(p) = \arg\min_{c_j, j \neq i} d(p, c_j)$ denote its second nearest center; for $p \in C_i \setminus \hat{B}_i$, $c(p) = c_i$. The perturbation we consider blows up all distances by a factor of α except for those distances between p and $c(p)$. Formally,

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

Suppose after the perturbation, C'_i is obtained by adding point set A_i and removing point set M_i from C_i, i.e. $A_i = C'_i \backslash C_i$, $M_i = C_i \backslash C'_i$. Let $A = \cup_i A_i$, $M = \cup_i M_i$. For convenience, we use $W_i = (C_i \cap C'_i) \backslash \hat{B}_i$, $V_i = \hat{B}_i \setminus M_i$, and thus we have $C_i = W_i \cup V_i \cup M_i$, $C'_i = W_i \cup V_i \cup A_i$.

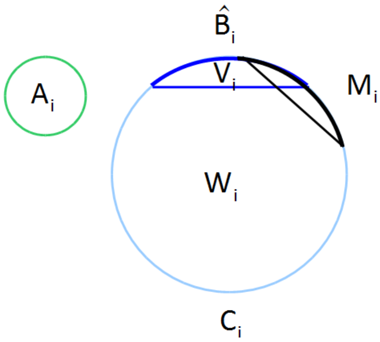

Figure 2: Illustration for notations. Note that $W_i = (C_i \cap C'_i) \setminus \hat{B}_i$, $V_i = \hat{B}_i \setminus M_i$ and $C'_i = W_i \cup V_i \cup A_i$.

Given the perturbation, we prove that all the selected bad points move by showing that $c_i = c'_i$ for all i. We first show that for each cluster, its new center is close to its old center, roughly speaking since the new and old cluster have a lot in common (Claim 1). We then show if $c_i \neq c'_i$ for some i, then the weighted sum of the distances $\sum_{1 \leq i \leq k} (|A_i| + \alpha + 1 + |M_i|) d(c_i, c'_i)$ should be large (Claim 2). However, this contradicts Claim 1, so the centers do not move after the perturbation.

In the proofs of Claim 1 and 2, we make use of the triangle inequality frequently. However, d' is not a metric, so we need to translate d' to d. We begin with some useful facts and use them to prove a summarization of the translation from d' to d.

Fact 1. *Suppose the clustering instance is (α, ϵ)-perturbation resilient.*

(1) If $c(c'_i) \in A_i$, then $d(c_i, c(c'_i)) \leq (1 + \alpha) d(c_i, c'_i)$.

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

Proof. (1) Let $c_l = c(c'_i)$. There are three cases: 1. c'_i is in C_i; 2. c'_i is outside C_i and is a selected bad point; 3. c'_i is outside C_i and is not a selected bad point.

Case 1: c'_i is in C_i. Since $c'_i \in C'_i$, we know either $c'_i \in W_i$ or $c'_i \in V_i$. If $c'_i \in W_i$, then $c(p) = c_i \notin A_i$ which is contradictory to the assumption that $c(c'_i) \in A_i$, so it must be that $c'_i \in V_i$. We have

$$d(c_i, c_l) \leq d(c_i, c'_i) + d(c'_i, c_l) \leq (1+\alpha)d(c_i, c'_i)$$

where the last inequality comes from the fact that c'_i is a selected bad point and $c(c'_i) = c_l$.
Case 2: In this case, c'_i is from $\hat{B}_j$ for some $j (j \neq i, j \neq l)$ and move to C'_i after perturbation, i.e. $c'_i \in \hat{B}_j \cap A_i$. Then we have

$$d(c_i, c_l) \leq d(c_i, c'_i) + d(c'_i, c_l) \leq d(c_i, c'_i) + \alpha d(c'_i, c_j) \leq (1+\alpha)d(c_i, c'_i)$$

where the second inequality comes from the fact that c'_i is a selected bad point and $c(c'_i) = c_l$, and the last inequality comes from $c'_i \in C_j$.
Case 3: In this case, c'_i is from $C_l \setminus \hat{B}_l$ and move to C'_i after perturbation, i.e. $c'_i \in (C_l \setminus \hat{B}_l) \cap A_i$, we have

$$d(c_i, c_l) \leq d(c_i, c'_i) + d(c'_i, c_l) \leq 2d(c_i, c'_i) \leq (1+\alpha)d(c_i, c'_i)$$

where the second inequality comes from $c'_i \in C_l$ and the last inequality comes from $\alpha \geq 1$.
(2) Assume $c'_i = c_j$. Then we would have the following: $c'_j \neq c_l (\forall l)$. First $c'_j \neq c_j$, since otherwise, moving all the points in C'_j to C'_i 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 need to show that $c'_i = c_j$ and $c'_j \neq c_l (\forall l)$ lead to an contradiction. First we can lower bound $d(c_j, c'_j)$. Due to (α, ϵ)-perturbation resilience, $W_j \cup V_j = C_j \cap C'_j$ satisfies $|W_j \cup V_j| \geq |C_j| - \epsilon n$. These many points are closer to c'_j than to $c'_i = c_j$ under d'. However, back in d, c_j is the optimal center for $C_j = W_j \cup V_j \cup M_j$, so 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, by the fact that c_j is the optimal center for C_j, we have

$$d_{sum}(c'_j, C_j) = d_{sum}(c'_j, W_j \cup V_j \cup M_j) \geq d_{sum}(c_j, C_j) = d_{sum}(c_j, W_j \cup V_j \cup M_j).$$

To simplify the inequality, notice for any $p \in W_j$, $\alpha d(c'_j, p) = d'(c'_j, p) \leq d'(c'_i, p) = d(c_j, p)$, resulting in $d_{sum}(c'_j, W_j) \leq d_{sum}(c_j, W_j)/\alpha$. 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 $d_{sum}(c'_j, V_j) \leq d_{sum}(c_j, V_j)$. So the inequality becomes

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

Second, we can upper bound $d(c_j, c'_j)$ since points in W_j are both close to c_j and c'_j. Let $p^* = \arg\min_{p \in W_j} d(p, c_j)$, then

$$d(c_j, c'_j) \leq d(c_j, p^*) + d(c'_j, p^*) \leq 2d(c_j, p^*) \leq 2d_{sum}(c_j, W_j)/|W_j|. \tag{2}$$

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

Fact 2. *Suppose the clustering instance is* (α, ϵ)*-perturbation resilient and* $\min_i |C_i| > (\frac{2\alpha}{\alpha-1} + 2)\epsilon n + 1$. *If* $c_i \neq c_i'$*, then we have*

$$\begin{aligned} d'_{sum}(c_i', W_i) &\geq \alpha d_{sum}(c_i', W_i \setminus \{c(c_i')\}), \\ d'_{sum}(c_i', V_i) &= \alpha d(c_i', V_i), \\ d'_{sum}(c_i', A_i) &\geq \alpha d_{sum}(c_i', A_i \setminus \{c(c_i')\}), \\ d'_{sum}(c_i, W_i) &= d_{sum}(c_i, W_i), \\ d'_{sum}(c_i, V_i) &= \alpha d(c_i, V_i), \\ d'_{sum}(c_i, A_i) &\leq \alpha d_{sum}(c_i, A_i \setminus \{c(c_i')\}) + \alpha(1+\alpha) d(c_i', c_i). \end{aligned}$$

Proof. These can be easily 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.

Since $c_i' \neq c_i$, and by Fact 1.(2), we know $c_i' \neq c_j (\forall j)$. So when translating $d'_{sum}(c_i', C)$(C is W_i, V_i or A_i), we only need to check if $c(c_i') \in C$. For W_i,

$$d'_{sum}(c_i', W_i) \geq d'_{sum}(c_i', W_i \setminus \{c(c_i')\}) = \alpha d_{sum}(c_i', W_i \setminus \{c(c_i')\}).$$

For V_i, since there is no center in V_i, $d'_{sum}(c_i', V_i) = \alpha d(c_i', V_i)$. For A_i,

$$d'_{sum}(c_i', A_i) \geq d'_{sum}(c_i', A_i \setminus \{c(c_i')\}) = \alpha d_{sum}(c_i', A_i \setminus \{c(c_i')\}).$$

Now consider the sum of distances concerning c_i. For $d'_{sum}(c_i, W_i)$ and $d'_{sum}(c_i, V_i)$, they follow from the definition of d'. For A_i, if $c(c_i') \in A_i$, then

$$\begin{aligned} d'_{sum}(c_i, A_i) &= d'_{sum}(c_i, A_i \setminus \{c(c_i')\}) + d'(c_i, c(c_i')) \\ &\leq \alpha d_{sum}(c_i, A_i \setminus \{c(c_i')\}) + \alpha d(c_i, c(c_i')) \\ &\leq \alpha d_{sum}(c_i, A_i \setminus \{c(c_i')\}) + \alpha(1+\alpha) d(c_i', c_i). \end{aligned}$$

where the last inequality comes from Fact 1.(1). If $c(c_i') \notin A_i$, then

$$d'_{sum}(c_i, A_i) = d'_{sum}(c_i, A_i \setminus \{c(c_i')\}) \leq \alpha d_{sum}(c_i, A_i \setminus \{c(c_i')\})$$

where the inequality comes from the definition of d'. □

Claim 1. *Suppose the clustering instance is* (α, ϵ)*-perturbation resilient, and* $\min_i |C_i| > (\frac{2\alpha}{\alpha-1} + 2)\epsilon n + 1$. *Then for each* i*,*

$$d_{sum}(c_i, W_i) \geq \frac{\alpha}{\alpha+1}(|W_i| - |V_i| - |A_i| - (\alpha+1)) d(c_i, c_i')$$

where $W_i = (C_i \cap C_i') \setminus \hat{B}_i, V_i = \hat{B}_i \setminus M_i$. *This means* $d(c_i, c_i')$ *can be bounded approximately by the average distance between* c_i *and a large portion of* C_i.

Proof. Here we will prove an upper bound for the distance between the new center and the old center. The high level intuition is that under d', c_i' is the optimal center for C_i', so it has no more cost than c_i on C_i'. Since W_i is much larger than V_i and A_i, if c_i' has much more cost on W_i than c_i, then c_i' cannot compensate the cost on V_i and A_i. So in order not to have much more cost on W_i, by the triangle inequality c_i' should be close to c_i. In the following we give the detailed proof.

If $c'_i = c_i$, $d(c'_i, c_i) = 0$, which immediately implies the bound. Otherwise, we need to use the fact that c'_i has smaller cost than c_i on C'_i under d'. Formally, we have $d'_{sum}(c'_i, C'_i) \leq d'_{sum}(c_i, C'_i)$, i.e.

$$d'_{sum}(c'_i, W_i) - d'_{sum}(c_i, W_i) \leq d'_{sum}(c_i, V_i) + d'_{sum}(c_i, A_i) - d'_{sum}(c'_i, V_i) - d'_{sum}(c'_i, A_i)$$

Translating d' to d by Fact 2, we have

$$\begin{aligned}\alpha d_{sum}(c'_i, W_i \setminus \{c(c'_i)\}) - d_{sum}(c_i, W_i) \quad \leq \quad & \alpha d_{sum}(c_i, V_i) + \alpha d_{sum}(c_i, A_i \setminus \{c(c'_i)\}) + \alpha(1+\alpha)d(c'_i, c_i) \\ & - \alpha d_{sum}(c'_i, V_i) - \alpha d_{sum}(c'_i, A_i \setminus \{c(c'_i)\}).\end{aligned}$$

By the triangle inequality,

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

which implies the desired result. $\square$

Claim 2. *Suppose the clustering instance is* (α, ϵ)*-perturbation resilient, and* $\min_i |C_i| > (2 + \frac{2\alpha}{\alpha - 1})\epsilon n + 1$. *Let* $I_i = 1$ *if* $c_i \neq c'_i$ *and* $I_i = 0$ *otherwise. Then*

$$(\alpha - 1) \sum_{1 \leq i \leq k} I_i d_{sum}(c_i, W_i) \leq \alpha \sum_{1 \leq i \leq k} (|A_i| + \alpha + 1 + |M_i|) d(c_i, c'_i)$$

where $W_i = (C_i \cap C'_i) \setminus \hat{B}_i$.

Proof. Here we will prove a lower bound for the weighted sum of the distances between the new centers and the old centers. The high level intuition is as follows: the clustering that under d' assigns points in $C'_i \setminus \hat{B}_i = W_i \cup A_i$ to c_i and points p in V_i to $c(p)$, saves as much cost as $d'_{sum}(c'_i, W_i) - d'_{sum}(c_i, W_i) \approx (\alpha - 1)\sum_i d_{sum}(c_i, W_i)$ on W_i compared to the optimum clustering under d', if $c'_i \neq c_i$. So the optimum clustering under d' must save this cost on other parts of points. If the distances between $\{c'_i\}$ and $\{c_i\}$ are all small, then $\{c'_i\}$ could not save much cost compared to $\{c_i\}$. So the weighted sum of the distances between $\{c'_i\}$ and $\{c_i\}$ should be large. Formally, we have the following inequality from the fact that $\{c'_i\}$ are the optimal centers under d', thus have no more cost than the clustering that under d' assigns points in $W_i \cup A_i$ to c_i and points p in V_i to $c(p)$:

$$\sum_i d'_{sum}(c'_i, C'_i) \leq \sum_i [d'_{sum}(c_i, C'_i \setminus V_i) + \sum_{p \in V_i} d'(c(p), p)]. \tag{3}$$

In the following we will show the detailed proof. In the proof, to estimate $d'_{sum}(c'_i, W_i) - d'_{sum}(c_i, W_i) = \alpha d_{sum}(c'_i, W_i) - d_{sum}(c_i, W_i)$ when $c'_i \neq c_i$, we will need the following fact: $\{c_i\}$ are the optimal centers under d, so $d_{sum}(c_i, W_i)$ could not be much larger than $d_{sum}(c'_i, W_i)$. Formally, we need Inequality 4 which comes from the fact that $\{c_i\}$ are the optimal centers under d:

$$\sum_i d_{sum}(c_i, C_i) \leq \sum_i d_{sum}(c'_i, C_i). \tag{4}$$

Now, as a first step, we combine the two inequalities. For Inequality 3, we need to divide C'_i into A_i, V_i and W_i, and divide $C'_i \setminus V_i$ into A_i and W_i. For Inequality 4, we need to multiply it by α and divide C_i into three parts M_i, V_i and W_i. Add them up and keep the terms on the same part of points in the same group:

$$\begin{aligned}
\textstyle\sum_i \quad & [d'_{sum}(c_i, A_i) - d'_{sum}(c'_i, A_i) \\
+ \quad & \alpha d_{sum}(c'_i, M_i) - \alpha d_{sum}(c_i, M_i) \\
+ \quad & \alpha d_{sum}(c'_i, V_i) - d'_{sum}(c'_i, V_i) + \sum_{p \in V_i} d(p, c(p)) - \alpha d_{sum}(c_i, V_i) \\
+ \quad & \alpha d_{sum}(c'_i, W_i) - d'_{sum}(c'_i, W_i) + d'_{sum}(c_i, W_i) - \alpha d_{sum}(c_i, W_i)] \\
\geq \quad & 0.
\end{aligned}$$

Notice the last line is essentially the terms needed in estimating $d'_{sum}(c'_i, W_i) - d'_{sum}(c_i, W_i)$. And the rest terms approximate the cost compensated on A_i, M_i and V_i, which can be bounded in terms of $d(c_i, c'_i)$ by the triangle inequality. Then we can compare $d(c_i, c'_i)$ to the estimated $d'_{sum}(c'_i, W_i) - d'_{sum}(c_i, W_i)$. Formally, as a second step, we need to first translate d' into d, and then apply the triangle inequality to the terms on A_i, M_i and V_i. Rewrite it as $\sum_i T_i \geq 0$ where T_i are all the terms related to i. In the easy case when $c'_i = c_i$, most terms cancel out, and we get $T_i = \sum_{p \in V_i} d(p, c(p)) - \alpha d_{sum}(c_i, V_i)$. Then $T_i \leq (\alpha - \alpha) d_{sum}(c_i, V_i) = 0$ since V_i are the selected bad points. If $c'_i \neq c_i$, Fact 2 leads to

$$\begin{aligned}
T_i \quad \leq \quad & \alpha d_{sum}(c_i, A_i \setminus \{c(c'_i)\}) + \alpha(1+\alpha) d(c'_i, c_i) - \alpha d_{sum}(c'_i, A_i \setminus \{c(c'_i)\}) \\
+ \quad & \alpha d_{sum}(c'_i, M_i) - \alpha d_{sum}(c_i, M_i) \\
+ \quad & \alpha d(c'_i, V_i) - \alpha d(c'_i, V_i) + \sum_{p \in V_i} d(p, c(p)) - \alpha d_{sum}(c_i, V_i) \\
+ \quad & \alpha d_{sum}(c'_i, W_i) - \alpha d_{sum}(c'_i, W_i \setminus \{c(c'_i)\}) + (1-\alpha) d_{sum}(c_i, W_i).
\end{aligned}$$

The first line is bounded by $\alpha(\alpha + |A_i|) d(c_i, c'_i)$. The second line is bounded by $\alpha d(c_i, c'_i)|M_i|$. The third line is bounded by 0, since V_i are the selected bad points and thus $\sum_{p \in V_i} d(p, c(p)) \leq \alpha d_{sum}(c_i, V_i)$. For the fourth line, if $c(c'_i) \notin W_i$, then it is $0 + (1-\alpha) d_{sum}(c_i, W_i)$; otherwise, $c(c'_i) = c_i$, then it is $\alpha d(c_i, c'_i) + (1-\alpha) d_{sum}(c_i, W_i)$. In conclusion, in any case we have the following bound for T_i:

$$T_i \quad \leq \quad \alpha(|A_i| + |M_i| + \alpha + 1) d(c_i, c'_i) + (1-\alpha) I_i d_{sum}(c_i, W_i).$$

Then $\sum_i T_i \geq 0$ implies the desired result. □

Theorem 3. *Suppose the clustering instance is (α, ϵ)-perturbation resilient and $\min_i |C_i| > (2 + \frac{2\alpha}{\alpha-1})\epsilon n + \frac{2\alpha(\alpha+1)}{\alpha-1}$. Then $|B| \leq \epsilon n$.*

Proof. Claim 1 shows an upper bound for the distance between the new center and the old center of each optimal cluster, and Claim 2 shows a lower bound for the weighted sum of the distances between the new centers and the old centers when $c'_i \neq c_i$ for some i. However, the two bounds lead to a contradiction when $\min_i |C_i|$ is sufficiently large, so the centers should not move after the perturbation. This then implies that in the optimal clustering under d' each point p is assigned to the center $c(p)$, and therefore the selected bad points ($\hat{B}$) will move from their original optimal clusters. Using this and the (α, ϵ)-perturbation resilience property we get an upper bound on the number of the selected bad points. This can also be proved to be a bound on the number of all the bad points due to the way we construct $\hat{B}$.

Formally, combining Claims 1 and 2, we get

$$\sum_{1 \leq i \leq k} \alpha d(c_i, c'_i) \big[(|A_i| + \alpha + 1 + |M_i|) - \frac{\alpha - 1}{\alpha + 1} (|(C_i \cap C'_i) \setminus \hat{B}_i| - |\hat{B}_i \setminus M_i| - |A_i| - (\alpha + 1)) I_i \big] \geq 0.$$

If $I_i = 0$, we have $d(c_i, c'_i) = 0$; if $I_i = 1$, since $|C_i| > (2 + \frac{2\alpha}{\alpha-1})\epsilon n + \frac{2\alpha(\alpha+1)}{\alpha-1}$, the coefficient of $d(c_i, c'_i)$ is negative. So the left hand side is no greater than 0. Therefore, all terms are equal to 0, i.e. for all $1 \le i \le k$, $d(c_i, c'_i) = 0$. Then points in $\hat{B}_i$ will move to other clusters after perturbation, which means that $\hat{B}_i \subseteq M_i$, thus $\hat{B} \subseteq M$. Then $|\hat{B}| \le |M| \le \epsilon n$. Specially, $|\hat{B}_i| \le \epsilon n$ for any i. Then $|B_i| \le \epsilon n$, otherwise $|\hat{B}_i|$ would be $\epsilon n + 1$. So $\hat{B}_i = B_i$, and $\hat{B} = B$ and $|B| = |\hat{B}| \le \epsilon n$. □

Note: Notice that the bound in Theorem 3 is an optimal bound for the bad points in the sense that for any $\alpha > 1, \epsilon < \frac{1}{5}$, we can easily construct an (α, ϵ)-perturbation resilient 2-median instance which has ϵn bad points. The construction is as follows. Construct 3 groups of points: G_1, G_2 and B. Each of G_1 and G_2 has $(1-\epsilon)n/2$ points, and B has ϵn points. 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 $M/(\alpha+1)+1$, and those between the points in B and G_2 are $\alpha M/(\alpha+1) - 1$. The instance satisfies the triangle inequality, which can be verified by a case-by-case study. When M is sufficiently large, the optimal clustering before perturbation has one center in G_1 and the other in G_2, then B are trivially bad points. So it is sufficient to show that the instance is (α, ϵ)-perturbation resilient. Notice the optimal cost before perturbation is $(1-\epsilon)n + \epsilon n(\frac{M}{\alpha+1}+1)$, so the optimal cost after perturbation is no more than $\alpha[(1-\epsilon)n + \epsilon n(\frac{M}{\alpha+1}+1)]$. Suppose after perturbation, both of the two centers come from $G_1 \cup B$ or both come from G_2, then the cost is larger than $\frac{(1-\epsilon)n}{2}(\frac{\alpha M}{\alpha+1} - 1) > \alpha[(1-\epsilon)n + \epsilon n(\frac{M}{\alpha+1}+1)]$ when M is sufficiently large. So in the optimal clustering after perturbation, it must be that one center is from $G_1 \cup B$ and the other is from G_2, then G_1 and G_2 remain in different clusters. This means the instance is (α, ϵ)-perturbation resilient.

B Full proof of Theorem 4

To prove the theorem, we first show the properties of the linkage and cleaning phases in Lemma 3 and Lemma 4 respectively. Before that we present some facts that are used in the analysis. For each i, let $q_i^* = \arg\max_{q_i \in C_i \setminus B} d(c_i, q_i)$.

Fact 3. *Suppose $\alpha > 2 + \sqrt{7}, \epsilon \le \rho/5$. Let $C_i, C_j (j \ne i)$ be two different optimal clusters.*

(1) For any good point $p_j \in C_j \setminus B$, $d(c_i, p_j) > 3d(c_i, q_i^*)$;
(2) For any point $p_i \in \mathbb{B}(c_i, d(c_i, q_i^*))$, any good point $p_j \in C_j \setminus B$, $d(p_i, p_j) > 2d(c_i, q_i^*)$;
(3) If $\mathcal{C}' \setminus B$ is laminar to $\mathcal{C} \setminus B$ before checking $d(p,q)$, and $C \in U_{p,q}$ contains good points from C_i, then $C \cap \mathbb{B}(p, d(p,q))$ contains good points from C_i;

Proof. (1) First we have

$$d(c_i, q_i^*) \le \frac{1}{\alpha - 1} d(c_i, c_j).$$

Since $\alpha d(c_j, p_j) \le d(c_i, p_j)$, we have

$$\frac{\alpha+1}{\alpha} d(c_i, p_j) \ge d(c_i, p_j) + d(c_j, p_j) \ge d(c_i, c_j).$$

When $\alpha > 2 + \sqrt{7}$, we have

$$d(c_i, p_j) \ge \frac{\alpha}{\alpha+1} d(c_i, c_j) > \frac{3}{\alpha - 1} d(c_i, c_j) \ge 3d(c_i, q_i^*).$$

(2) This follows from (1) since

$$d(p_i, p_j) \ge d(c_i, p_j) - d(c_i, p_i) > 3d(c_i, q_i^*) - d(c_i, q_i^*) = 2d(c_i, q_i^*).$$

(3) If $|C| = 1$, then $C \subseteq \mathbb{B}(p, d(p,q))$, and we are done. Otherwise, C must be generated from some merge step, and thus $|C| > 2\epsilon n$. We have $|B| \leq \epsilon n$, so C contains more than ϵn good points. If C contains good points only from C_i, then C contains more than ϵn good points from C_i; this is also true when C contains good points from another optimal cluster, since in this case we have $C_i \setminus B \subseteq C \setminus B$ from the laminarity assumption. In either case, since there are at most ϵn points outside the ball $\mathbb{B}(p, d(p,q))$, there must be at least one good point in $C \cap \mathbb{B}(p, d(p,q))$ from C_i. □

Lemma 3. *If $\alpha > 2+\sqrt{7}$, $\epsilon \leq \rho/5$, then the tree T contains nodes $N_i (1 \leq i \leq k)$ such that $N_i \setminus B = C_i \setminus B$.*

Proof. The proof follows from two key facts:

(1) If $\mathcal{C}' \setminus B$ is laminar to $\mathcal{C} \setminus B$ right before checking some $d(p,q)$, then for any i, j, $i \neq j$ such that either $d(p,q)$ is checked before $d(c_i, q_i^*)$ or $d(p,q)$ is checked before $d(c_j, q_j^*)$ or $p = c_i, q = q_i^*$ or $p = c_j, q = q_j^*$, we have that $U_{p,q}$ cannot contain both good points from C_i and C_j.

(2) If $\mathcal{C}' \setminus B$ is laminar to $\mathcal{C} \setminus B$ right before checking $d(c_i, q_i^*)$, we have that right after checking $d(c_i, q_i^*)$ there is a cluster containing all the good points in cluster i and no other good points.

We have $\mathcal{C}' \setminus B$ is laminar to $\mathcal{C} \setminus B$ initially. Consider any merge step corresponding to some $d(p,q)$ such that $U_{p,q}$ contains good points from both C_i and $C_j (j \neq i)$. Fact (1) implies that both $d(c_i, q_i^*)$ and $d(c_j, q_j^*)$ must have been checked, then fact (2) implies that all the good points in C_i and C_j respectively have already been merged. So $\mathcal{C}' \setminus B$ is always laminar to $\mathcal{C} \setminus B$. Then the lemma follows from fact (2).

We now prove fact (1). Suppose for contradiction that there exist good points from C_i and C_j in $U_{p,q}$. By Fact 3.(3), there exist good points $p_i \in C_i$ and $p_j \in C_j$ in $\mathbb{B}(p, d(p,q))$. By Fact 3.(2), $2d(c_i, q_i^*) < d(p_i, p_j)$, and by triangle inequality $d(p_i, p_j) \leq d(p, p_i) + d(p, p_j) \leq 2d(p,q)$, so $d(p,q) > d(c_i, q_i^*)$. The same argument leads to $d(p,q) > d(c_j, q_j^*)$. This is a contradiction to the assumption that $d(p,q)$ is checked before $d(c_i, q_i^*)$ or before $d(c_j, q_j^*)$ or $p = c_i, q = q_i^*$ or $p = c_j, q = q_j^*$.

We now prove fact (2). It is sufficient to show that $\cup_{C \in U_{c_i, q_i^*}} C \setminus B = C_i \setminus B$ and U_{c_i, q_i^*} satisfies the approximate closure condition. First, U_{c_i, q_i^*} contains no good points outside C_i by fact (1). Second, any C containing good points from C_i is in U_{c_i, q_i^*}. If $|C| = 1$, then this is trivial. Otherwise, C must be formed by a merge step, and then by fact (1), C has no good points outside C_i. Since $\mathbb{B}(c_i, d(c_i, q_i^*))$ contains all good points in C_i, C has only bad points outside the ball, we have $C \in U_{c_i, q_i^*}$.

We finally show that U_{c_i, q_i^*} satisfies the approximate closure condition. The approximate coverage condition is satisfied since in addition to all good points in C_i, $\cup_{C \in U_{c_i, q_i^*}} C$ can only contain bad points, so it has at most ϵn points outside $\mathbb{B}(c_i, d(c_i, q_i^*))$. To show the approximate margin condition, we use Fact 3.(2): for $\alpha > 2 + \sqrt{7}$, $2d(c_i, q_i^*)$ is smaller than the distance between any point in $\mathbb{B}(c_i, d(c_i, q_i^*))$ and any good point outside C_i. Then by eliminating all bad points outside $\mathbb{B}(c_i, d(c_i, q_i^*))$, we get the approximate margin condition. We also have $| \cup_{C \in U_{c_i, q_i^*}} C| \geq |C_i \setminus B| \geq \min_i |C_i| - \epsilon n$. □

Lemma 4. *If $\alpha > 2 + \sqrt{7}$, $\epsilon \leq \epsilon' \rho/5$ where $\epsilon' \leq 1$, then $\tilde{\mathcal{C}}$ is a $(1+\epsilon')$-approximation to the optimum.*

Proof. It is sufficient to first prove that there is a pruning $\mathcal{P}'$ in $\tilde{T}$ containing only k nodes $N_i' (1 \leq i \leq k)$ such that $N_i' \setminus B = C_i \setminus B$, and then prove its cost is approximately $\mathcal{OPT}$.

By Lemma 3, the tree T has a pruning $\mathcal{P}$ which contains $N_i (1 \leq i \leq k)$ and possibly some bad points, such that $N_i \setminus B = C_i \setminus B$. We now show that any singleton good point p in N_i in T' remains in N_i after cleaning, so $\mathcal{P}$ becomes the desired $\mathcal{P}'$. For any non-singleton leaf L_i under N_i, L_j under $N_j (j \neq i)$, since they have more good points than bad points, we can find good points $p_i \in L_i$ and $p_j \in L_j$, such that

$$\text{median}\{d(p,q), q \in L_j\} \geq d(p, p_j) > d(p, p_i) \geq \text{median}\{d(p,q), q \in L_i\}.$$

So p is assigned to some leaf node under N_i.

Therefore, $\mathcal{P}$ becomes $\mathcal{P}' = \{N_i', 1 \le i \le k\}$ such that $N_i' \setminus B = C_i \setminus B$. It is sufficient to prove the cost of $\mathcal{P}'$ approximates $\mathcal{OPT}$, i.e. to bound the increase of cost caused by a bad point $p_j \in C_j$ ending up in $N_i'(i \ne j)$. There are two cases: p_j belongs to a non-singleton leaf node in T' or p_j is a singleton in T'. In the first case, suppose p_j is merged into the leaf node when checking $d(p,q)$. Then according to the approximate margin condition, there are at least $\min_i |C_i| - 3\epsilon n$ good points p_{it} from C_i inside $\mathbb{B}(p, d(p,q))$, and the same number of good points p_{js} from C_j outside $\mathbb{B}(p, d(p,q))$, such that $d(p_j, p_{it}) \le d(p_j, p_{js})$. In the second case, since p_j is assigned to the nearest non-singleton leaf node according to median distance, there are at least $K = (\min_i |C_i| - \epsilon n)/2 - \epsilon n$ good points p_{it} from C_i inside the leaf node, and the same number of good points p_{js} from C_j outside the leaf node, such that $d(p_j, p_{it}) \le d(p_j, p_{js})$. In either case, the increase of cost due to p_j can be bounded as follows.

$$
\begin{aligned}
d(p_j, c_i) - d(p_j, c_j) &\le \frac{1}{K}\Big\{\sum_{1\le t\le K}\big[d(p_j, p_{it}) + d(p_{it}, c_i)\big] - \sum_{1\le s\le K}\big[d(p_j, p_{js}) - d(p_{js}, c_j)\big]\Big\} \\
&\le \frac{1}{K}\Big\{\sum_{1\le t\le K} d(p_{it}, c_i) + \sum_{1\le s\le K} d(p_{js}, c_j)\Big\} \le \frac{1}{K}\mathcal{OPT}.
\end{aligned}
$$

As $|B| \le \epsilon n$, the cost of $\mathcal{P}'$ is at most $(1 + \frac{\epsilon n}{K})\mathcal{OPT}$, so the minimum cost pruning $\tilde{\mathcal{C}}$ is a $(1 + \frac{2\epsilon n}{\min_i |C_i| - 3\epsilon n})$-approximation to the optimum. By setting $\epsilon' \ge \frac{2\epsilon n}{\min_i |C_i| - 3\epsilon n}$, we get the desired result. □

Theorem 4. *Suppose the clustering instance is (α, ϵ)-perturbation resilient to k-median. If $\alpha > 2 + \sqrt{7}$ and $\epsilon \le \rho/5$ where $\rho = (\min_i |C_i| - 15)/n$, then in polynomial time, Algorithm 2 outputs a tree $\tilde{T}$ that contains a pruning that is ϵ-close to the optimum clustering. Moreover, if $\epsilon \le \rho\epsilon'/5$ where $\epsilon' \le 1$, the clustering produced is a $(1+\epsilon')$-approximation to the optimum.*

Proof. The theorem follows immediately from Lemma 3 and 4.
Running Time Sorting the distances takes $O(n^2)$. Computing $\mathbb{B}(p, d(p,q))$ and $U_{p,q}$ takes $O(n^2)$. The step of checking if $\mathbb{B}(p, d(p,q))$ satisfies the approximate closure condition needs more careful analysis. The approximate coverage condition can be verified in $O(n)$. The approximate margin condition can be verified in $O(n^3)$ as follows: for each $q_1 \in S \setminus \mathbb{B}(p, d(p,q))$, enumerate all $p_1, p_2 \in \mathbb{B}(p, d(p,q))$ to check if there exist p_1, p_2 such that they violate the requirement $d(p_1, p_2) < d(p_1, q_1)$, and if so, mark q_1; then the approximate margin condition is satisfied if and only if the number of marked points is no more than ϵn. So in total, the linkage phase takes $O(n^5)$. The cleaning phase takes time $O(n^3)$, and the dynamic programming takes time $O(n^3)$. Therefore, the running time to get the approximation is $O(n^5)$. □

C Full Proof of Theorem 5

Theorem 5. *Suppose (X, d) is (α, ϵ)-perturbation resilient for $\alpha > 8$, $\epsilon < \rho/20$. Let $0 < \lambda < 1$. Then with probability at least $1 - \delta$, we can get an implicit clustering that is $2\frac{1+\lambda}{1-\lambda}(1 + \frac{8\epsilon}{\rho - 12\epsilon})$-approximation in time $O((\frac{kD^2}{\lambda^2\epsilon^2\zeta^2}\ln\frac{N}{\delta})^5)$.*

Proof. We sample a set S of size $n = \Theta(\frac{kD^2}{\lambda^2\epsilon^2\zeta^2}\ln\frac{N}{\delta})$ and run Algorithm 2 on S to obtain the minimum cost pruning $\tilde{\mathcal{C}}$ and its centers $\tilde{\mathbf{c}}$. We then output the implicit clustering of the whole space X that assigns each point in X to its nearest neighbor in $\tilde{\mathbf{c}}$. We will show that the cost of this clustering is close to the optimum, i.e. $\Phi_X(\tilde{\mathbf{c}}) \le 2\frac{1+\lambda}{1-\lambda}(1 + \frac{8\epsilon}{\rho - 12\epsilon})\Phi_X(\mathbf{c})$.

To compare the two, since $\tilde{\mathbf{c}}$ is computed on S, we first approximate them using costs on S. Formally, for every set of centers $\bar{\mathbf{c}}$, $\mathbf{E}(\Phi_S(\bar{\mathbf{c}})/n) = \Phi_X(\bar{\mathbf{c}})/N$, so if $n \geq \frac{k\ln(8N/\delta)D^2}{2\lambda^2\zeta^2}$, the Chernoff bound leads to

$$\begin{aligned} Pr\left[\left|\frac{\Phi_S(\bar{\mathbf{c}})}{n} - \frac{\Phi_X(\bar{\mathbf{c}})}{N}\right| > \lambda\frac{\Phi_X(\bar{\mathbf{c}})}{N}\right] &= Pr\left[\left|\frac{\Phi_S(\bar{\mathbf{c}})/n - \Phi_X(\bar{\mathbf{c}})/N}{D}\right| > \lambda\frac{\Phi_X(\bar{\mathbf{c}})/N}{D}\right] \\ &\leq 2e^{-2(\frac{\lambda\zeta}{D})^2 n} \leq \frac{\delta}{4N^k}. \end{aligned}$$

By the union bound, we have with probability at least $1-\delta/4$, for any k-median centers $\bar{\mathbf{c}}$, $\Phi_S(\bar{\mathbf{c}})/n \approx \Phi_X(\bar{\mathbf{c}})/N$. Specifically, $(1-\lambda)\Phi_X(\tilde{\mathbf{c}})/N \leq \Phi_S(\tilde{\mathbf{c}})/n$ and $\Phi_S(\mathbf{c})/n \leq (1+\lambda)\Phi_X(\mathbf{c})/N$. So it is sufficient to show $\Phi_S(\tilde{\mathbf{c}}) \leq 2(1+\frac{8\epsilon}{\rho-12\epsilon})\Phi_S(\mathbf{c})$. However, $\tilde{\mathcal{C}}$ may be different from $\mathcal{C}\cap S$, so we need a "bridge" for the two.

Now we turn to analyze Algorithm 2 on S to find such a bridge. First we have that X has at most ϵN bad points, so when $n \geq \frac{\ln(8/\delta)}{2\epsilon^2}$, with probability at least $1-\delta/4$, S has at most $2\epsilon n$ bad points. Similarly, when $n \geq \frac{30\ln(8k/\delta)}{\epsilon^2}$, we have with probability at least $1-\delta/4$, for any i, $|C_i \cap S| > 5\times(2\epsilon n)+15$. Moreover, for any good points $p_1, p_2 \in C_i \cap S \setminus B, q_1 \in C_j \cap S \setminus B (j \neq i)$, we have

$$d(p_1,p_2) \leq d(c_i,p_1) + d(c_i,p_2) \leq \frac{d(p_1,q_1)+d(p_2,q_1)}{\alpha-1} \leq \frac{2d(p_1,q_1)+d(p_1,p_2)}{\alpha-1},$$

and thus $3d(p_1,p_2) < d(p_1,q_1)$ when $\alpha > 8$. Then we can choose an arbitrary good point $\tilde{c}_i$ from $C_i\cap S$ to be the center of $C_i \cap S$, so that even if c_i is not sampled, we still have that the tree T in Algorithm 2 has nodes N_i such that $N_i \setminus B = C_i \cap S \setminus B$. This is because $\cup_{C\in U_{\tilde{c}_i,\tilde{q}_i^*}} \setminus B = C_i \cap S \setminus B$ where $\tilde{q}_i^* = \arg\min_{q\in C_i\cap S\setminus B} d(\tilde{c}_i, q)$, and $U_{\tilde{c}_i,\tilde{q}_i^*}$ satisfies approximate coverage and margin conditions since the distance between any good point outside C_i and any point in the ball $\mathbb{B}(\tilde{c}_i,\tilde{q}_i^*)$ is 2 times larger than $d(\tilde{c}_i,\tilde{q}_i^*)$. So Algorithm 2 can successfully produce a tree $\tilde{T}$ with a pruning $\mathcal{P}' = \{N_1', N_2', \ldots, N_k'\}$, such that $N_i' \setminus B = C_i \cap S \setminus B$. Suppose in S, $\mathbf{c}' = \{c_1', c_2', \ldots, c_k'\}$ 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, we have $\Phi_S(\tilde{\mathbf{c}}) \leq \Phi_S(\mathcal{P}',\mathbf{c}')$. First, since $\tilde{\mathcal{C}}$ is the minimum cost pruning, $\Phi_S(\tilde{\mathcal{C}},\tilde{\mathbf{c}}) \leq \Phi_S(\mathcal{P}',\mathbf{c}')$. Second, since in $\Phi_S(\tilde{\mathbf{c}})$ each point is assigned to its nearest center but in $\Phi_S(\tilde{\mathcal{C}},\tilde{\mathbf{c}})$ this may not be true, $\Phi_S(\tilde{\mathbf{c}}) \leq \Phi_S(\tilde{\mathcal{C}},\tilde{\mathbf{c}})$.

On the other hand, we have $\Phi_S(\mathcal{P}',\mathbf{c}') \leq 2(1+\frac{8\epsilon}{\rho-12\epsilon})\Phi_S(\mathbf{c})$. First, $\Phi_S(\mathcal{P}',\mathbf{c})$ is different from $\Phi_S(\mathbf{c})$ only on the bad points. Using the approach similar to that in Lemma 4, we can show that for any bad point $p_j \in C_j \cap S$ which ends up in the wrong cluster $N_i'(i\neq j)$, it must be closer to $(\min_i |C_i\cap S| - 6\epsilon n)/2 \geq (\min_i |C_i|/(2N) - 6\epsilon)n/2$ good points in N_i' than to the same number of good points in C_j, and thus we have $\Phi_S(\mathcal{P}',\mathbf{c}) \leq (1+\frac{8\epsilon}{\rho-12\epsilon})\Phi_S(\mathbf{c})$. Second, $\Phi_S(\mathcal{P}',\mathbf{c})$ may be smaller than $\Phi_S(\mathcal{P}',\mathbf{c}')$, since $\mathbf{c}$ are chosen from X while $\mathbf{c}'$ are only chosen from S. But by the triangle inequality, for any $N_i' \in \mathcal{P}'$,

$$2|N_i'|\sum_{p\in N_i'} d(p,c_i) = \sum_{q\in N_i'}\sum_{p\in N_i'}[d(p,c_i)+d(q,c_i)] \geq \sum_{p\in N_i'}\sum_{q\in N_i'} d(p,q) \geq \sum_{p\in N_i'}\sum_{q\in N_i'} d(q,c_i') = |N_i'|\sum_{q\in N_i'} d(q,c_i'),$$

so we have $\Phi_S(\mathcal{P}',\mathbf{c}') \leq 2\Phi_S(\mathcal{P}',\mathbf{c})$. □

D Full Proof of Theorem 6

To prove the theorem, first we show that the α-perturbation resilience property implies that for any two different optimal clusters C_i and C_j and any $A \subseteq C_i$, we have $\alpha d_{sum}(A, C_i \setminus A) < d_{sum}(A, C_j)$. This follows by constructing a specific perturbation and using the fact that the optimum does not change after the perturbation. This can be used to show that when $\alpha > 3\frac{\max_i |C_i|}{\min_i |C_i|-1}$ we have: (1) for any two different optimal clusters C_i and C_j and any $A \subseteq C_i$, $A' \subseteq C_j$ such that $\min(|C_i \setminus A|, |C_j \setminus A'|) > \min_i |C_i|/2$, $d_{avg}(A, A') > \min(d_{avg}(A, C_i \setminus A), d_{avg}(A', C_j \setminus A'))$; (2) for any point $p \in C_i$, its $|C_i|/2$ nearest neighbors are in the same optimal cluster. These two facts then imply that $\mathcal{C}'$ in Algorithm 3 is always laminar to the optimal clustering, so the minimum cost pruning of T will be the optimal clustering, as desired.

We now formally present a detailed proof and first start with a few useful facts.

Fact 4. *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 $A \subseteq C_i, A' \subseteq C_j$, we have the following claims.*

(1) $\alpha d_{sum}(A, C_i \setminus A) < d_{sum}(A, C_j)$.

(2) $\alpha d_{sum}(A, C_i \setminus A) < \frac{|C_j|}{|A'|} d_{sum}(A, A') + \frac{|A|}{|A'|} d_{sum}(A', C_j \setminus A')$.

(3) When $\alpha > 3\frac{\max_i |C_i|}{\min_i |C_i|-1}$, and $|C_i \setminus A|$ and $|C_j \setminus A'|$ are larger than $\min_i |C_i|/2$, we have $d_{avg}(A, A') > \min(d_{avg}(A, C_i \setminus A), d_{avg}(A', C_j \setminus A'))$.

(4) When $\alpha > 3\frac{\max_i |C_i|}{\min_i |C_i|-1}$, for any point p, all its $\min_i |C_i|/2$ nearest neighbors are in the same optimal cluster.

Proof. (1) This follows by considering a specific perturbation and using the fact that the optimum does not change after the perturbation. We define a perturbation as follows: $d'(p, q) = \alpha d(p, q)$ if $p \in A, q \in C_i \setminus A$ or $q \in A, p \in C_i \setminus A$, and $d'(p, q) = d(p, q)$ otherwise. d' is a valid α-perturbation of d, so the optimal clustering after perturbation should remain the same. Specially, its cost should be smaller than that of the clustering obtained by replacing C_i, C_j with $C_i \setminus A, A \cup C_j$. After canceling the terms common in the two costs, we have $2d'_{sum}(A, C_i \setminus A) < 2d'_{sum}(A, C_j)$, which implies $\alpha d_{sum}(A, C_i \setminus A) < d_{sum}(A, C_j)$.

(2) The first claim shows that we can bound $d_{sum}(A, C_i \setminus A)$ by $d_{sum}(A, C_j)$. We can divide C_j into two parts: a subset A' and the rest points $C_j \setminus A'$. Then from the triangle inequality we can bound $d_{sum}(A, C_j)$ by $d_{sum}(A, A')$ and $d_{sum}(A', C_j \setminus A')$. Formally, we have from the first claim:

$$\begin{aligned}
\alpha d_{sum}(A, C_i \setminus A) &< d_{sum}(A, C_j) = d_{sum}(A, A') + d_{sum}(A, C_j \setminus A') \\
&= d_{sum}(A, A') + \sum_{p \in A} \sum_{q \in C_j \setminus A'} d(p, q) \\
&\leq d_{sum}(A, A') + \frac{1}{|A'|} \sum_{p \in A} \sum_{q \in C_j \setminus A'} \sum_{p' \in A'} [d(p, p') + d(p', q)] \\
&= d_{sum}(A, A') + \frac{|C_j \setminus A'|}{|A'|} d_{sum}(A, A') + \frac{|A|}{|A'|} d_{sum}(A', C_j \setminus A') \\
&= \frac{|C_j|}{|A'|} d_{sum}(A, A') + \frac{|A|}{|A'|} d_{sum}(A', C_j \setminus A').
\end{aligned}$$

where the last inequality comes from the triangle inequality.

(3) We have from the second claim:

$$\alpha d_{sum}(A, C_i \setminus A) < \frac{|C_j|}{|A'|} d_{sum}(A, A') + \frac{|A|}{|A'|} d_{sum}(A', C_j \setminus A'), \tag{5}$$

$$\alpha d_{sum}(A', C_j \setminus A') < \frac{|C_i|}{|A|} d_{sum}(A', A) + \frac{|A'|}{|A|} d_{sum}(A, C_i \setminus A). \tag{6}$$

Divide Inequality 5 by $|A|$, divide Inequality 6 by $|A'|$, and add them up:

$$(\alpha-1)\frac{d_{sum}(A',C_j\setminus A')}{|A'|}+(\alpha-1)\frac{d_{sum}(A,C_i\setminus A)}{|A|} \quad < \quad (|C_i|+|C_j|)\frac{d_{sum}(A,A')}{|A||A'|}.$$

Suppose $d_{avg}(A,A')\leq \min(d_{avg}(A,C_i\setminus A), d_{avg}(A',C_j\setminus A'))$, then

$$\begin{aligned}(\alpha-1)|C_j\setminus A|d_{avg}(A,A')+(\alpha-1)|C_i\setminus A|d_{avg}(A,A') &< (|C_i|+|C_j|)d_{avg}(A,A'),\\ (\alpha-1)|C_j\setminus A|+(\alpha-1)|C_i\setminus A| &< |C_i|+|C_j|,\end{aligned}$$

which is contradictory when $\alpha > 3\frac{\max_i |C_i|}{\min_i |C_i|-1}$, and $|C_j\setminus A'|$ and $|C_i\setminus A|$ are larger than $\min_i |C_i|/2$.

(4) Suppose p comes from the optimal cluster C_i. Let $p'=\arg\min_{q\notin C_i} d(p,q)$, and suppose p' comes from the optimal cluster C_j. There are two cases. The first case is that $d_{avg}(p,C_i\setminus\{p\})\geq d_{avg}(p',C_j\setminus\{p'\})$. From this assumption and Fact 4.(2) we have

$$\alpha d_{sum}(p,C_i\setminus\{p\}) \quad < \quad |C_j|d(p,p')+|C_j\setminus\{p'\}|d_{avg}(p,C_i\setminus\{p\})$$

which leads to

$$d_{avg}(p,C_i\setminus\{p\}) \quad < \quad \frac{|C_j|}{\alpha(|C_i|-1)-(|C_j|-1)}d(p,p').$$

The second case is that $d_{avg}(p,C_i\setminus\{p\}) < d_{avg}(p',C_j\setminus\{p'\})$, then a similar argument leads to

$$d_{avg}(p,C_i\setminus\{p\})\leq d_{avg}(p',C_j\setminus\{p'\}) < \frac{|C_i|}{\alpha(|C_j|-1)-(|C_i|-1)}d(p,p').$$

In both cases, when $\alpha\geq 3\frac{\max_i |C_i|}{\min_i |C_i|-1}$, $d_{avg}(p,C_i\setminus\{p\}) < d(p,p')/2$. Let $E=\{q\in C_i : d(p,q)\geq d(p,p')\}$. Then we have

$$|E|d(p,p')\leq d_{sum}(p,E)\leq d_{sum}(p,C_i\setminus\{p\}) < \frac{|C_i|-1}{2}d(p,p').$$

This means $|E| < \frac{|C_i|}{2}$, i.e. more than $|C_i|/2$ points in C_i are within distance less than $\min_{q\notin C_i} d(p,q)$. Therefore, for any point p, all its $\min_i |C_i|/2$ nearest neighbors are in the same optimal cluster. □

Theorem 6. *For $(3\frac{\max_i |C_i|}{\min_i |C_i|-1})$-perturbation resilient instances, Algorithm 3 finds the optimal min-sum k-clustering in polynomial time.*

Proof. It is sufficient to show that in Algorithm 3, $\mathcal{C}'$ is always laminar to the optimal clustering $\mathcal{C}$, i.e. for any $A\in\mathcal{C}'$ and $C\in\mathcal{C}$, we have either $A\subseteq C$, or $C\subseteq A$, or $A\cap C=\varnothing$. Then the minimum cost pruning of T will be the optimal clustering, which can be obtained by dynamic programming.

Intuitively, Fact 4.(4) implies that $\mathcal{C}'$ is laminar initially, and Fact 4.(3) can be used to show that the merge steps preserve the laminarity, so $\mathcal{C}'$ is always laminar to the optimal clustering.

Formally, we prove the laminarity by induction. By Fact 4.(4), $\mathcal{C}'$ is laminar initially. It is sufficient to prove that if the current clustering is laminar, then the merge step keeps the laminiarity. 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, i.e. $A\subset C_i, A'\subset C_j\neq C_i$; (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). By Fact 4.(3), the first case cannot happen. In the second case, for any E that is a subset of $C_i \setminus A$ in the current clustering, we have $d_{avg}(A, E) \geq d_{avg}(A, A')$. We know that $d_{avg}(A, C_i \setminus A)$ 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 $d_{avg}(A, C_i \setminus A) \geq d_{avg}(A, A')$. Also, $d_{avg}(A, A')$ 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 $d_{avg}(A, C_j) \leq d_{avg}(A, A') \leq d_{avg}(A, C_i \setminus A)$. This means

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

where the last inequality comes from $\alpha \geq 3\frac{\max_i |C_i|}{\min_i |C_i|-1}$ and $|C_i \setminus A| \geq \min_i |C_i|/2$. This contradicts Fact 4.(1). 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 in Algorithm 3 takes $O(n^2 \log n)$ time. To compute average distances between clusters, we can record the size of each cluster, and $d_{sum}(C'_i, C'_j)$ for any C'_i, C'_j in the current clustering, and update $d_{sum}(C'_i \cup C'_j, C'_l) = d_{sum}(C'_i, C'_l) + d_{sum}(C'_j, C'_l)$ 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. □

E Full Proof of Theorem 7

We sample a set S of size $n = \Theta(\frac{D^2}{\rho^2\eta^2} \ln \frac{Nk}{\delta})$ and run Algorithm 3 on S. We then output the implicit clustering of the whole space X that assigns each point $p \in X$ to $\tilde{C}_i \in \tilde{\mathcal{C}}$ such that $d_{sum}(p, \tilde{C}_i)$ is minimized. To prove the theorem, we first show that with high probability, $\mathcal{C}'$ in Algorithm 3 is always laminar to $\mathcal{C} \cap S$, and thus $\mathcal{C} \cap S$ is a pruning of the tree. Then we show that $\mathcal{C} \cap S$ is actually the minimum cost pruning $\tilde{\mathcal{C}}$. Using this fact, we prove that the implicit clustering obtained is the optimum clustering $\mathcal{C}$. We now formally present a detailed proof.

Fact 5. *Suppose the clustering instance (X, d) is α-perturbation resilient to the min-sum objective where $\alpha \geq 6\frac{\max_i |C_i|}{\min_i |C_i|-1}$. If the size of the sample $n = \Theta(\frac{D^2}{\rho^2\eta^2} \ln \frac{Nk}{\delta})$, then with probability at least $1 - \delta$, $\mathcal{C}'$ in Algorithm 3 is always laminar to $\mathcal{C} \cap S$.*

Proof. The intuition is that on X, for any $1 \leq i \neq j \leq k$, any $p \in C_i$, we have $\alpha d_{sum}(p, C_i) < d_{sum}(p, C_j)$. When n is sufficiently large, we can show $d_{sum}(p, C_i \cap S) \approx d_{sum}(p, C_i)|S|/|X|$, $d_{sum}(p, C_j \cap S) \approx d_{sum}(p, C_j)|S|/|X|$ and $\frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S|-1} \approx \frac{\max_i |C_i|}{\min_i |C_i|-1}$, and thus we have a similar claim on S. Then $\mathcal{C}'$ in Algorithm 3 is always laminar to $\mathcal{C} \cap S$.

Formally, it is sufficient to show that with probability at least $1 - \delta$, for any $1 \leq i \neq j \leq k, p \in C_i$, $\frac{|C_i|}{\sqrt[8]{2}}\frac{|S|}{|X|} \leq |C_i \cap S| \leq \sqrt[8]{2}|C_i|\frac{|S|}{|X|}$ and $3\frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S|-1} d_{sum}(p, C_i \cap S) < d_{sum}(p, C_j \cap S)$. Notice we have $\alpha d_{sum}(p, C_i) < d_{sum}(p, C_j)$ by Fact 4.(1). So we need to show with high probability:

(1) $|C_i \cap S| \leq \sqrt[8]{2}|C_i|\frac{|S|}{|X|}$ and $d_{sum}(p, C_i \cap S) \leq \sqrt[4]{2}\frac{|S|}{|X|} d_{sum}(p, C_i)$;

(2) $|C_j \cap S| \geq \frac{|C_j|}{\sqrt[8]{2}}\frac{|S|}{|X|}$ and $d_{sum}(p, C_j \cap S) \geq \frac{1}{\sqrt[4]{2}}\frac{|S|}{|X|} d_{sum}(p, C_j)$;

(3) $\frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S|-1} \leq \sqrt{2}\frac{\max_i |C_i|}{\min_i |C_i|-1}$.

To prove the first claim, we show that if n is sufficiently large, then with high probability, $\frac{|C_i \cap S|}{|S|} \leq \sqrt[8]{2}\frac{|C_i|}{|X|}$, and $d_{avg}(p, C_i \cap S) \leq \sqrt[8]{2} d_{avg}(p, C_i)$. On one hand, from Hoeffding's inequality, we have

$$Pr\left[\left|\frac{|C_i \cap S|}{|S|} - \frac{|C_i|}{|X|}\right| \geq (\sqrt[8]{2}-1)\frac{|C_i|}{|X|}\right] \leq 2\exp\{-2(\sqrt[8]{2}-1)^2\frac{|C_i|^2}{|X|^2}n\} \leq 2\exp\{-2(\sqrt[8]{2}-1)^2\rho^2 n\}.$$

On the other hand, under the condition $\left|\frac{|C_i \cap S|}{|S|} - \frac{|C_i|}{|X|}\right| < (\sqrt[8]{2}-1)\frac{|C_i|}{|X|}$, we have

$$\begin{aligned} Pr\left[d_{avg}(p, C_i \cap S) > \sqrt[8]{2} d_{avg}(p, C_i)\right] &\leq \exp\{-2(\sqrt[8]{2}-1)^2\left[\frac{d_{avg}(p, C_i)}{D}\right]^2 |C_i \cap S|\} \\ &\leq \exp\{-2(\sqrt[8]{2}-1)^2\frac{\eta^2}{D^2}(2-\sqrt[8]{2})\rho n\}. \end{aligned}$$

Therefore, if $n = \Theta(\frac{D^2}{\rho^2\eta^2}\ln\frac{Nk}{\delta})$, then the first claim is true with probability at least $1 - \frac{\delta}{4Nk}$. A similar argument holds for the second claim. By the union bound, with probability at least $1-\delta$, we have the two claims for any $p \in C_i$, any $C_j \neq C_i$. Under this condition, we have for any i, $\frac{|C_i|}{\sqrt[8]{2}|X|} \leq \frac{|C_i \cap S|}{|S|} \leq \frac{\sqrt[8]{2}|C_i|}{|X|}$, then the third claim follows:

$$\frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S| - 1} \leq \frac{\sqrt[8]{2}\frac{|S|}{|X|}\max_i |C_i|}{\frac{1}{\sqrt[8]{2}}\frac{|S|}{|X|}\min_i |C_i| - 1} \leq \frac{\sqrt[8]{2}\frac{|S|}{|X|}\max_i |C_i|}{\frac{|S|}{\sqrt[4]{2}|X|}(\min_i |C_i| - 1)} \leq \sqrt{2}\frac{\max_i |C_i|}{\min_i |C_i| - 1}.$$

□

Fact 6. *Suppose the clustering instance (X, d) is α-perturbation resilient to the min-sum objective where $\alpha \geq 6\frac{\max_i |C_i|}{\min_i |C_i| - 1}$. If the size of the sample $n = \Theta(\frac{D^2}{\rho^2\eta^2}\ln\frac{Nk}{\delta})$, then with probability at least $1-\delta$, the minimum min-sum cost pruning of the tree in Algorithm 3 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 that is not $\mathcal{C} \cap S$ 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 of the tree. We first prove a similar claim for $\mathcal{C}$ by the α-perturbation resilience, i.e. for any three different clusters $C_i, C_j, C_l \in \mathcal{C}$, any $A_X \subseteq C_i$, $\alpha d_{sum}(A_X, C_i \setminus A_X) < d_{sum}(C_j, C_l)$. Then we prove the claim for $\mathcal{C} \cap S$: for any $A \subseteq C_i \cap S$, $d_{sum}(A, C_i \cap S \setminus A) < d_{sum}(C_j \cap S, C_l \cap S)/2$. Finally we use it to prove $\mathcal{C} \cap S$ is the minimum cost pruning.

First, 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 $2d'_{sum}(A_X, C_i \setminus A_X) < 2d'_{sum}(C_j, C_l)$, which leads to

$$\alpha d_{sum}(A_X, C_i \setminus A_X) < d_{sum}(C_j, C_l). \tag{7}$$

Second, we prove for $\mathcal{C} \cap S$ the following claim: for any $A \subseteq C_i \cap S$, $d_{sum}(A, C_i \cap S \setminus A) < d_{sum}(C_j \cap S, C_l \cap S)/2$. On one hand, by summing Inequality 7 over all the subsets of C_i, we have

$$\begin{aligned} \alpha \sum_{A_X \subseteq C_i} d_{sum}(A_X, C_i \setminus A_X) &< 2^{|C_i|} d_{sum}(C_j, C_l), \\ \alpha 2^{|C_i|-1} \sum_{p,q \in C_i} d_{sum}(p, q) &< 2^{|C_i|} d_{sum}(C_j, C_l), \\ \frac{\alpha}{2} d_{sum}(C_i, C_i) &< d_{sum}(C_j, C_l). \end{aligned} \tag{8}$$

The second inequality follows from the fact that for any $p, q \in C_i$, in half of the choices of A_X, one of them is in A_X and the other is not. On the other hand, similar to the proof of Fact 5, we can show that with high probability, for any $p \in C_i$, $d_{sum}(p, C_i \cap S) \leq \sqrt[8]{2} \frac{|S|}{|X|} d_{sum}(p, C_i)$. So we have

$$\begin{aligned} d_{sum}(C_i \cap S, C_i \cap S) &= \sum_{p \in C_i \cap S} d_{sum}(p, C_i \cap S) \leq \sum_{p \in C_i \cap S} \sqrt[8]{2} \frac{|S|}{|X|} d_{sum}(p, C_i) \\ &= \sqrt[8]{2} \frac{|S|}{|X|} \sum_{q \in C_i} d_{sum}(C_i \cap S, q) \leq \sqrt[8]{2} \frac{|S|}{|X|} \sum_{q \in C_i} \sqrt[8]{2} \frac{|S|}{|X|} d_{sum}(C_i, q) \\ &= \sqrt[4]{2} \frac{|S|^2}{|X|^2} d_{sum}(C_i, C_i). \end{aligned} \tag{9}$$

We can also show that with high probability, for any $p \in C_i$ and any $C_j (j \neq i)$, $d_{sum}(p, C_j \cap S) \geq \frac{1}{\sqrt[8]{2}} \frac{|S|}{|X|} d_{sum}(p, C_j)$. So we have for any two different clusters C_j and C_l,

$$\begin{aligned} d_{sum}(C_j \cap S, C_l \cap S) &= \sum_{p \in C_j \cap S} d_{sum}(p, C_l \cap S) \geq \sum_{p \in C_j \cap S} \frac{1}{\sqrt[8]{2}} \frac{|S|}{|X|} d_{sum}(p, C_l) \\ &= \frac{1}{\sqrt[8]{2}} \frac{|S|}{|X|} \sum_{q \in C_l} d_{sum}(C_j \cap S, q) \geq \frac{1}{\sqrt[8]{2}} \frac{|S|}{|X|} \sum_{q \in C_l} \frac{1}{\sqrt[8]{2}} \frac{|S|}{|X|} d_{sum}(C_j, q) \\ &= \frac{1}{\sqrt[4]{2}} \frac{|S|^2}{|X|^2} d_{sum}(C_j, C_l). \end{aligned} \tag{10}$$

So we have for any $A \subseteq C_i \cap S$,

$$\begin{aligned} d_{sum}(A, C_i \cap S \setminus A) &\leq d_{sum}(C_i \cap S, C_i \cap S) \leq \sqrt[4]{2} \frac{|S|^2}{|X|^2} d_{sum}(C_i, C_i) \\ &< \frac{1}{2\sqrt[4]{2}} \frac{|S|^2}{|X|^2} d_{sum}(C_j, C_l) \leq \frac{1}{2} d_{sum}(C_j \cap S, C_l \cap S) \end{aligned}$$

where the second inequality comes from Inequality 9, the third comes from Inequality 8, and the last comes from Inequality 10.

We now use $d_{sum}(A, C_i \cap S \setminus A) < d_{sum}(C_j \cap S, C_l \cap S)/2$ 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 $S_{i,1}, \ldots, S_{i,m_i}$; after that, merge $C_{h+1} \cap S, \ldots, C_{h+l_g} \cap S$ into g unions, i.e. for $1 \leq j \leq g$, $l_0 = 0$, merge $l_j - l_{j-1} \geq 2$ clusters

$C_{h+l_{j-1}+1} \cap S, \ldots, C_{h+l_j} \cap S$ 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 h clusters is

$$\sum_{1 \le i \le h} \sum_{1 \le p \ne q \le m_i} d_{sum}(S_{i,p}, S_{i,q}) = \sum_{1 \le i \le h} \sum_{1 \le p \le m_i} d_{sum}(S_{i,p}, C_i \cap S \setminus S_{i,p}). \tag{11}$$

The cost increased by joining clusters is

$$\sum_{1 \le j \le g} \sum_{h+l_{j-1} < p \ne q \le h+l_j} d_{sum}(C_p \cap S, C_q \cap S). \tag{12}$$

To prove $\mathcal{C} \cap S$ is the minimum cost pruning, we need to show that the saved cost (11) is less than the increased cost (12). Since each term in (12) is twice larger than any term in (11), it is sufficient to show that the number of the terms in (12) is at least half the number of the terms in (11), i.e.

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

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

Theorem 7. *Suppose the clustering instance (X, d) is α-perturbation resilient to the min-sum objective where $\alpha \ge 6\frac{\max_i |C_i|}{\min_i |C_i| - 1}$. Then w.p. $\ge 1-\delta$, we can get an implicit optimum clustering in time $O((\frac{D^2}{\rho^2\eta^2} \ln \frac{Nk}{\delta})^3)$ where $\eta = \min_{p \in X, 1 \le i \le k} d_{avg}(p, C_i)$ is the minimum average distance between points and optimal clusters.*

Proof. As mentioned above, we sample a set S of size $n = \Theta(\frac{D^2}{\rho^2\eta^2} \ln \frac{Nk}{\delta})$ and run Algorithm 3 on S. We then output the implicit clustering of the whole space X that assigns each point $p \in X$ to $\tilde{C}_i \in \tilde{\mathcal{C}}$ such that $d_{sum}(p, \tilde{C}_i)$ is minimized. By Fact 5 and 6, the minimum cost pruning output by Algorithm 3 is $\mathcal{C} \cap S$. Notice in the proof of Fact 5, we showed for any $1 \le i \ne j \le k, p \in C_i$, $3\frac{\max_i |C_i \cap S|}{\min_i |C_i \cap S| - 1} d_{sum}(p, C_i \cap S) < d_{sum}(p, C_j \cap S)$. Therefore, by inserting p in $C_i \cap S$ such that $d_{sum}(p, C_i \cap S)$ is minimized, we can obtain the optimum clustering. $\square$

F Other Implementation Details

F.1 Dynamic Programming to Find 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 [2]. 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 $cost(p, m)$. 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 ($1 < m \le k$), 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 $1 \le m \le k$ for each tree node. The algorithm is presented in Algorithm 4. 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 $cost(p, 1)$ takes $O(t)$ time, computing $cost(p, m)(1 < m \le k)$ takes $O(k^2)$, in total the algorithm takes time $O(nt + nk^2)$.

Notice when T is a multi-branch tree and not suitable for dynamic programming, we need to turn it into a 2-branch tree 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, then merge $R_1 \cup R_2$ with R_3, repeat until we merge $R_1 \cup R_2 \cdots \cup R_{t-1}$ with R_t into R. In this way, we get a 2-branch tree T' and can run dynamic programming on it. Notice each pruning in T has a corresponding pruning in T', so the minimum cost pruning of T' has no greater cost than the minimum cost pruning of T.

Also notice when the cost function is center-based, such as k-median, the algorithm essentially computes a center for the node p when computing $cost(p, 1)$. So it can output the centers together with the pruning.

Algorithm 4 Dynamic Programming in Tree of Clusters

Input: tree of clusters T on data set S, distance function $d(\cdot,\cdot)$ on S, k.
• Traverse T bottom up, and denote the current tree node as p:
• Calculate $cost(p, 1)$.
• IF p is a leaf, $cost(p, m) = cost(p, 1)(1 < m \leq k)$,
• ELSE compute $cost(p, m)$ from $cost(p_1, m_1)$ and $cost(p_2, m_2)$ where p_1, p_2 are the children of p, and $m_1 + m_2 = m$.
• Traverse backwards to get the k-clustering that achieves $cost(r, k)$ where r is the root.
Output: the k-clustering.

F.2 An Efficient Implementation of Algorithm 1

Here we show an efficient implementation of Algorithm 1, namely Algorithm 5. The Phase 1 of this implementation takes time only $O(n^3)$.

Algorithm 5 Efficient Implementation of Algorithm 1

Input: Data set S, distance function $d(\cdot,\cdot)$ on S.
Phase 1: Sort all the pairwise distances in ascending order.
• For all $p \in S$ and $1 \leq i \leq n$, compute L^p, $\chi(p, i)$. Then compute $\chi^*(p, i)$ by Equation 13.
• Let the current clustering be n singleton clusters.
• For $d(p, q)$ in ascending order:
• Suppose $q = L^p_i$. Check if $d(p, q)$ satisfies the three claims in Fact 8,
• where the third claim can be checked by verifying if $\chi^*(p, i) = -1$.
• If so, merge all the clusters covered by $\mathbb{B}(p, d(p, q))$.
• Construct the tree T with points as leaves and internal nodes corresponding to the merges performed.
Phase 2: Apply dynamic programming on T to get the minimum cost pruning $\tilde{\mathcal{C}}$.
Output: Clustering $\tilde{\mathcal{C}}$.

Notice 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 7. *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. □

Fact 7 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 8. *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 $\mathcal{C}'$. We need to show that for any $C' \in \mathcal{C}'$ such that $C' \cap \mathbb{B}(c, d(c,p)) \neq \varnothing$, $C' \subseteq \mathbb{B}(c, d(c,p))$. 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 $c \notin \mathbb{B}(c_1, d(c_1, p_1))$, i.e. $d(c, c_1) > d(c_1, p_1)$. 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 $C' \subseteq \mathbb{B}(c, d(c,q))$. □

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 Lemma 8 takes $O(n^2)$, which is still not good enough. We can refine this step since intuitively, for every c, if $d(c,q)$ 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 $L^p = (L^p_1, \ldots, L^p_n)$ 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 $d(p, L^p_i)$ fail the third claim in Lemma 8. Formally, define $\chi^*(p,i)$ to be the maximum $j > i$ such that there exits $s \leq i$ satisfying $d(p, L^p_s) \geq d(L^p_s, L^p_j)$. If no such point L^p_j exists, let $\chi^*(p,i) = -1$. Then $d(p, L^p_i)$ 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 $d(p, L^p_{i-1})$, 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 $d(p, L^p_i) \geq d(L^p_i, L^p_j)$. Define $\chi(p,i)$ to be the maximum $j > i$ such that $d(p, L^p_i) \geq d(L^p_i, L^p_j)$; if no such j exists, let $\chi(p,i) = -1$. Then it is easy to verify that

$$\chi^*(p,i) = \max\{\chi^*(p,i-1), \chi(p,i)\}. \tag{13}$$

Notice 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 5.