Fully Scalable MPC Algorithms for Euclidean Clustering: Constant Approximation in Constant Rounds*

Artur Czumaj† Shaofeng H.-C. Jiang‡ Yaonan Jin§ Jianing Lou¶
Weicheng Wang ‖

November 3, 2026

Abstract

We study Euclidean k-MEANS and k-MEDIAN under the Massively Parallel Computation (MPC) model, focusing on the *fully scalable* setting. Previously, fully scalable algorithms for k-MEANS and/or k-MEDIAN either required $O(\log\log n \cdot \log\log\log n)$ rounds [Cohen-Addad, Kuhn, and Parsaeian; SODA'26] or guaranteed only an $O(\log n)$-approximation [Goranci, Jiang, Kiss, Kong, Qian, and Szilagyi; SODA'26]. In this paper, we close this gap by giving the first fully scalable algorithms for both problems, which work even when k is polynomially larger than the local memory, and *simultaneously* achieve an $O(1)$-approximation and $O(1)$ round complexity.

Technically, we devise a new variant of the MP algorithm [Mettu and Plaxton, SICOMP'03] that computes a *fractional solution satisfying the Lagrangian Multiplier Preserving (LMP) property*. Importantly, this LMP guarantee holds even under arbitrary distance distortions, a feature that is crucial for efficient MPC implementations. We then give a novel MPC rounding algorithm that first rounds to a bicriteria solution with $k+O(n^{1-\varepsilon})$ centers and then uses an *iterative kernelization* idea to convert it into one with exactly k centers.

*A preliminary version of the paper with a weaker result is available at https://arxiv.org/abs/2604.00954v1, version April 1, 2026. The current version subsumes the arxiv version from April 2026.

†University of Warwick. Email: A.Czumaj@warwick.ac.uk

‡Peking University. Email: shaofeng.jiang@pku.edu.cn

§Hong Kong University of Science and Technology. Email: jinyaonan@ust.hk

¶Peking University. Email: loujn@pku.edu.cn

‖Peking University. Email: 2300013053@stu.pku.edu.cn

Contents

1 Introduction **1**
- 1.1 Technical Overview . . . 2
  - 1.1.1 Challenges and Our Approach to LMP under Distance Distortion . . . 4
  - 1.1.2 Overview of Our MPC Rounding Algorithm . . . 7
- 1.2 Related Work . . . 9

2 Preliminaries **10**

3 Fractional MP Algorithm with Robust LMP Approximation **11**
- 3.1 Structural Properties of $\boldsymbol{r}^{(\beta)}$ . . . 14
- 3.2 Analysis of MP_β without Distance Distortions . . . 19
- 3.3 Analysis of MP_β under Distance Distortions (Proof of Lemma 3.1) . . . 24

4 $O(1)$-Approximation with $k + O(n^{1-\varepsilon})$ Centers **26**
- 4.1 Computing a Fractional Solution (Proof of Lemma 4.2) . . . 28
- 4.2 Rounding to $k + O(n^{1-\varepsilon})$ Centers (Proof of Lemma 4.3) . . . 31
  - 4.2.1 Algorithm Description and Analysis . . . 31
  - 4.2.2 MPC Implementation . . . 35

5 $O(1)$-Approximation with Exactly k Centers **40**

A Preprocessing **45**

B Proof of Lemma 4.4 **48**

C Missing Proofs in Section 4 **49**
- C.1 Proof of Claim 4.6 . . . 49
- C.2 Proof of Claim 4.12 . . . 51

D Proof of Claim 5.2 **53**
- D.1 Proof of Claim D.1. . . . 54
- D.2 Proof of Claim D.2 . . . 55

AI Methodology **58**

References **58**

1 Introduction

Euclidean k-Means is a central problem in combinatorial optimization and algorithm design. Given a dataset $P \subseteq \mathbb{R}^d$ of n points and an integer $k \geq 1$, the goal is to find a set $C \subseteq \mathbb{R}^d$ of at most k centers minimizing

$$\text{cost}(P, C) \;:=\; \sum_{p \in P} \min_{c \in C} \|p - c\|_2^2.$$

Euclidean k-Median instead minimizes the sum of (unsquared) distances. Both are special cases of (k, z)-Clustering, which minimizes the sum of z-th powers of distances ($z = 2$, $z = 1$, respectively).

We address the computational challenges of k-Means and k-Median on large-scale and high-dimensional datasets, focusing on the *Massively Parallel Computation (MPC)* model. The MPC model was introduced in [KSV10] and serves as a theoretical abstraction of several practical parallel computing frameworks, including MapReduce [DG08], Hadoop [Whi15], Spark [ZCF+10], and Dryad [IBY+07]. In the MPC model [KSV10, GSZ11, BKS17, ANOY14], an input of N words is distributed across machines with local memory $s = o(N)$ words each. Computation proceeds in *synchronous rounds*: in each round, every machine performs arbitrary local computation and then exchanges messages with other machines, sending and receiving at most s words. The output may be distributed across the machines. The primary complexity measure is the number of rounds, and total memory is a secondary one.

We aim to devise *fully scalable* MPC algorithms with only constant round complexity, i.e., algorithms that work with local memory $s = N^\varepsilon$ for every constant $\varepsilon \in (0, 1)$. Full scalability has become the gold standard and a widely pursued goal in MPC algorithm design [ANOY14, ASS+18, ASZ19, GU19, CDP21, EMMZ22, AAH+23, CC23, CGJ+24, CZ24, JMNZ24, ABJ+25, CGGJ25, CKP26, GJK+26], and achieving it has often proved to be particularly difficult. We also aim for low total memory, but this is a secondary concern: $N^{1+\varepsilon}$ total memory is typically considered acceptable. In Euclidean clustering, we count each coordinate of a point as one word, so the input size is $N = nd$, and a fully scalable algorithm must work with local memory $s = (nd)^\varepsilon$ for every constant $\varepsilon \in (0, 1)$.

The Challenge of $k \gg s$. For clustering, full scalability is demanding mainly because of the relation between the local memory s and the number k of clusters. When s is polynomially larger than k, k-Means and k-Median are well understood in MPC. Each machine summarizes its part of the data by a sample or a (composable) coreset of size roughly $k \cdot \text{polylog}(n)$, these summaries are merged in $O(1)$ rounds, and the resulting small instance is solved on a single machine (see, e.g., [HM04, EIM11, BEL13, BBLM14]). Full scalability rules out this approach. Since local memory $s = (nd)^\varepsilon$ must be supported for *every* constant $\varepsilon > 0$, any polynomial number of clusters, say $k = n^\delta$, eventually exceeds s. In the most interesting regime, k is polynomially larger than s; in this regime, no single machine can even store a solution, and summary-based approaches break down, since any coreset must contain at least k points [HV20, CLSS22, HLW24]. As observed in [BW18, CGJ+24], the techniques behind most sequential, streaming, and distributed clustering algorithms therefore do not yield fully scalable algorithms. Instead, the decision of which centers to open must be made by local computations on different machines, while still meeting the *global* constraint that at most k centers are opened in total. This global constraint is the crux of the problem.

This challenge has motivated a surge of interest in designing fully scalable algorithms for clustering problems [BW18, BEFM21, CLN+21, CMZ22, CCM23, CGJ+24, CGGJ25, CKP26, GJK+26].

Yet for k-Means and k-Median, previous fully scalable algorithms still had to compromise on other guarantees. In particular, no fully scalable algorithm was known to compute an $O(1)$-approximation with exactly k centers in $O(1)$ rounds. For algorithms that run in $O(1)$ rounds, existing results either (i) compute only polylog(n)-approximations [CLN+21, AAH+23, GJK+26]; (ii) achieve bi-criteria approximations that may use more than k centers [BW18, CGJ+24]; or (iii) work only on special inputs [CMZ22]. More recently, Cohen-Addad, Kuhn, and Parsaeian [CKP26] obtained an $O(1)$-approximation on general inputs with exactly k centers, but their algorithm requires $O(\log\log n \cdot \log\log\log n)$ rounds.

Our Results. We close this gap for k-Means and k-Median by giving the first fully scalable MPC algorithms that compute an $O(1)$-approximation with exactly k centers in $O(1)$ rounds. We state our result in Theorem 1.1, assuming $d = O(\log n)$ and that the aspect ratio of the dataset is bounded by poly(n) for simplicity; see Theorem 5.1 for the full statement. Both assumptions are standard in previous work on MPC clustering (e.g., [CKP26, GJK+26]) and are without loss of generality, as they can be ensured by an $O(1)$-round fully scalable preprocessing algorithm (see Section A).

Theorem 1.1 (Implied by Theorem 5.1)**.** *For any constant $\varepsilon \in (0,1)$, there exists a fully scalable MPC algorithm for k-Means and for k-Median that, given any n-point dataset in $\mathbb{R}^{O(\log n)}$ distributed across machines with local memory n^{ε}, computes an $\varepsilon^{-O(1/\varepsilon)}$-approximation with high probability.*[1] *The algorithm runs in $O(\varepsilon^{-2})$ rounds and uses $\widetilde{O}(n^{1+\varepsilon})$ total memory.*

Our full result (Theorem 5.1) applies more generally to (k,z)-Clustering, whose objective uses the power-z distances for any constant $z \geq 1$, while retaining an $O(1)$-approximation in $O(1)$ rounds in the fully scalable MPC setting. In addition, our algorithm works even in the setting of polylogarithmic local memory $s \geq \text{poly}(d\log n) = \text{polylog}(n)$, where it achieves a round complexity of $O(\varepsilon^{-1}\log_s n)$, matching the known lower bound [BW18, RVW18] up to a constant $O(\varepsilon^{-1})$ factor.

Our result in Theorem 1.1 significantly improves upon the previous $O(1)$-round $O(\log n)$-approximation for k-Median [GJK+26] and the $O(\log\log n \cdot \log\log\log n)$-round $O(1)$-approximation for k-Means [CKP26]. Both approaches appear to face inherent obstacles to further improvement. In particular, [GJK+26] relies on tree embeddings, and thus $O(\log n)$ is likely the best possible ratio due to a distance distortion lower bound [AKPW95, Bar98]. For [CKP26], the main bottleneck in round complexity is the need to compute ruling sets, which is a fundamental yet notoriously challenging problem in MPC. Ruling sets have been studied extensively on general graphs, but known MPC algorithms either require a superconstant number of rounds [Ona18, GU19, GGJ20, GLM+23, GP24, JKPS25] or are not fully scalable [CKPU23, HZ23, GP24]. Our new techniques bypass both of these barriers, as we discuss in more detail in Section 1.1 below.

1.1 Technical Overview

In this paper, we propose a new framework for clustering that, at a high level, consists of two steps: it first computes a *fractional* clustering solution via a new variant of the Mettu-Plaxton (MP) algorithm [MP03] with the *Lagrangian Multiplier Preserving (LMP) property*, and then applies a *rounding* procedure to convert it into an integral solution, i.e., a set of centers. Importantly, our framework works directly in Euclidean space and can be implemented mainly using approximate range queries/nearest-neighbor search, which are standard geometric primitives that admit

[1] By "with high probability", we mean with probability at least $1 - 1/\text{poly}(n)$, where the degree of poly(n) can be made arbitrarily large.

$O(1)$-round MPC implementations [CGJ+24]. This approach is fundamentally different from those of [CKP26, GJK+26] and bypasses their key technical barriers. On the one hand, working directly in Euclidean space allows us to bypass the *tree embeddings* used in [GJK+26] and their $\Omega(\log n)$ distance distortion. Moreover, their approach works only for k-Median and is unlikely to extend to k-Means, primarily because the expected distance-distortion guarantees of tree embeddings do not directly extend to squared distances. In contrast, our framework is more general: it works for (k, z)-Clustering with any $z \geq 1$. On the other hand, our framework avoids *ruling-set computation*, the bottleneck behind the $O(\log \log n \cdot \log \log \log n)$-round complexity of [CKP26]. In particular, this requires us to design a new rounding procedure, since previous sequential rounding algorithms for k-Means and/or k-Median typically rely on (variants of) ruling-set computation (as we discuss later).

Below, we highlight the key technical novelties of each of the two steps of our framework.

A New Variant of MP: Distortion-Robust and LMP. The key ingredient of our fractional algorithm, which also constitutes our main technical contribution, is a new variant of the well-known Mettu-Plaxton (MP) algorithm [MP03], which has been widely adapted to various sublinear computational settings for clustering [ARS03, BCIS05, GLS14, CJK+22, BCG+24, BGJ+24, CGJ+24]. As usual, our variant of MP aims to solve the *power-z facility location problem*: a (Lagrangian) relaxation of (k, z)-Clustering, but importantly, it additionally satisfies the *Lagrangian Multiplier Preserving (LMP) property* for all $z \geq 1$, even on arbitrarily distorted distances which need not form a metric or even be symmetric; see Algorithm 1 and Lemma 3.1. Indeed, LMP is a widely considered property that converts approximations for power-z facility location into approximations for (k, z)-Clustering via the *bi-point rounding procedure* [JV01, JMS02, JMM+03, LS16, BPR+17, CGLS23, GPST23, BCG+24]. Moreover, being robust to distance distortion is a natural and useful requirement, since approximating distances is a common way to gain efficiency. In our Euclidean MPC setting, this allows an $O(1)$-round MPC implementation via an approximate range query [CGJ+24] subroutine (which distorts the distance). We emphasize that our new distortion-robust LMP variant of the MP algorithm provides a general algorithmic framework that may find applications in computational settings beyond MPC.

While variants of MP with the LMP property exist [ARS03, BCG+24], their LMP guarantees may fail under even arbitrarily small distance distortions. In fact, handling distance distortion for LMP is a general challenge that was also identified in recent work [CKP26], which designs an MPC clustering algorithm based on the very different Jain-Vazirani framework [JV01]. Another limitation of existing LMP results for the MP algorithm is that they all work only for the $z = 1$ case, i.e., k-Median, and extending them to any $z \geq 1$ also poses a significant technical challenge. Our approach addresses both limitations. Technically, for $z = 1$, our algorithm is essentially the same as that of [BCG+24], but we develop a substantially different analysis that establishes the robustness of its LMP guarantee to distance distortions. This new analysis also allows us to further generalize the algorithm to arbitrary $z \geq 2$ while preserving the distortion-robust LMP property. We provide a more detailed discussion of the technical challenges and an overview of our approach in Section 1.1.1.

Bypassing Ruling-Set Computation: Iterative Kernelization. Our second technical contribution is a new rounding algorithm, which converts any feasible fractional solution, not necessarily one produced by our fractional MP algorithm, into a set of exactly k centers with only a constant-factor loss in the objective value. More importantly, this new algorithm can be implemented in $O(1)$ MPC rounds. This distinguishes our algorithm from previous sequential rounding

algorithms [CGTS99, CL12, BGH+26], which typically rely on a *filtering* operation that seems too global to implement efficiently in the fully scalable MPC model. Indeed, filtering is closely related to (and can often be reduced to) *ruling-set computation*, for which no $O(1)$-round fully scalable MPC algorithm is known, as discussed above.

We bypass this filtering step, or ruling-set computation more generally, through a *new rounding framework* that consists of two main parts. First, we give a rounding algorithm that only rounds the given fractional solution to a bicriteria solution with $k + O(n^{1-\varepsilon})$ centers. Then, building on this bicriteria algorithm, we design an *iterative kernelization* approach to obtain exactly k centers. Technically, our bicriteria rounding algorithm is based on a rounding scheme with an expected k-size guarantee from [ANS17], together with some new ideas of our own that eventually yield a high-probability bound of $k+O(n^{1-\varepsilon})$ centers. For the iterative kernelization, we use the technique from [BCF+26], which reduces the task of removing $O(n^{1-\varepsilon})$ extra centers to a clustering problem on a dataset of size only $O(n^{1-\varepsilon})$. Combining this technique with our bicriteria algorithm, we obtain a *kernelization* step that, roughly speaking, converts the clustering input into a new one with a significantly smaller number of centers k. Our iterative kernelization simply repeats this step until $k = O(1)$, at which point the resulting problem can be readily solved in $O(1)$ rounds. We provide a more detailed overview of the algorithm in Section 1.1.2.

We note that this overall plan is similar to that of [BGH+26] for sequential rounding, which starts by rounding to a bicriteria solution with $k+O(1)$ centers and then removes the extra centers. However, their computation of a bicriteria solution still relies on the global filtering operation, and their use of the pseudo-approximation reduction of [LS16] to remove the extra centers is also inefficient in our setting, since we may have a much larger number of $O(n^{1-\varepsilon})$ extra centers. We also note that similar ideas of iterative kernelization have appeared in a previous clustering algorithm [Tho04], but that work considers a very different setting of sequential graph clustering, and thus its concrete implementation differs substantially from ours. To the best of our knowledge, our algorithm is the first to employ iterative kernelization for rounding in the MPC model.

1.1.1 Challenges and Our Approach to LMP under Distance Distortion

We now discuss in more detail the technical challenges of achieving the LMP property in the presence of distance distortion and how our approach addresses them.

We begin with some more background. The LMP property is defined for the *power-z facility location problem*. In this problem, instead of imposing a strict bound of k on the number of centers, we are given a parameter $\lambda > 0$ as the opening cost per center, and the goal is to find a set $F \subseteq P$ minimizing $\lambda|F| + \mathrm{cost}_z(P, F)$, where $\lambda|F|$ is called the *opening cost* and $\mathrm{cost}_z(P, F)$ (identical to the (k, z)-Clustering cost) is called the *connection cost*. We say that a solution F is an LMP α-approximation for some $\alpha \geq 1$ if $\mathrm{cost}_z(P, F) + \alpha\lambda|F| \leq \alpha \cdot \mathrm{OPT}^{\mathrm{fl}}$, where $\mathrm{OPT}^{\mathrm{fl}}$ denotes the optimal value of the facility location problem. This is clearly a stronger guarantee than a standard α-approximation for facility location, which only requires that $\mathrm{cost}_z(P, F) + \lambda|F| \leq \alpha \cdot \mathrm{OPT}^{\mathrm{fl}}$. In this discussion, since we will consider only a single dataset P, we henceforth omit P from the cost notation, writing, for example, $\mathrm{cost}_z(F)$ instead of $\mathrm{cost}_z(P, F)$.

Challenges of Preserving LMP under Distance Distortion. While for non-LMP approximations to power-z facility location, a γ-factor distortion typically costs a factor of $O(\gamma^z)$ in the final approximation ratio, the stronger LMP property is more fragile: even a $1 + \varepsilon$ error can cause the resulting solution to no longer be LMP. Specifically, consider an LMP α-approximate solution F computed using the $(1 + \varepsilon)$-distorted distances, which means

$$\widetilde{\mathrm{cost}}_z(F) + \alpha\lambda|F| \leq \alpha\widetilde{\mathrm{OPT}}^{\mathrm{fl}}, \tag{1}$$

where both $\widetilde{\text{cost}_z}$ and $\widetilde{\text{OPT}}^{\text{fl}}$ are evaluated on the $(1+\varepsilon)$-distorted distances. However, it is possible that this solution F is not LMP with respect to the original distances. Indeed, while the opening cost $\lambda|F|$ does not change with respect to distance distortion, $\widetilde{\text{OPT}}^{\text{fl}}$ and $\widetilde{\text{cost}_z}(F)$ may change by a factor of, say, $1+\varepsilon$, and therefore the guarantee (1) only implies $(1+\varepsilon)\text{cost}_z(F)+\alpha\lambda|F| \le (1+\varepsilon)\alpha\,\text{OPT}^{\text{fl}}$. This still leaves a gap from the LMP guarantee we actually want, i.e., $\text{cost}_z(F)+\alpha'\lambda|F| \le \alpha'\,\text{OPT}^{\text{fl}}$, for (finite) $\alpha' \ge 1$; in fact, this gap cannot be bridged for a generic solution F, as shown by the following counterexample.

Example. Consider $z = 1$ and $\lambda = 1+\varepsilon$ (for any given $\varepsilon > 0$). Let the dataset consist of only two points x, y with $\text{dist}(x, y) = 1$, and distorted distance $\widetilde{\text{dist}}(x, y) = 1+\varepsilon$. On the original distances, the optimal value is $\text{OPT}^{\text{fl}} = 2+\varepsilon$. On the distorted distances, one optimal solution (which is clearly LMP 1-approximate) is to open both points as facilities, i.e., $F = \{x, y\}$. However, on the original distances, since $\text{cost}_z(F) = 0$, for any finite $\alpha' \ge 1$, we have

$$\text{cost}_z(F) + \alpha'\lambda|F| = \alpha' \cdot 2(1+\varepsilon) > \alpha' \cdot (2+\varepsilon) = \alpha' \cdot \text{OPT}^{\text{fl}}\,.$$

Therefore, F is not an LMP solution with any finite approximation ratio on the original distances.

Since LMP does not hold under distance distortions in general, one must utilize the properties of a specific algorithm, or a specialized way of distorting distances. Indeed, a recent work [CKP26] implements the seminal Jain-Vazirani algorithm (JV) [JV01] in MPC, but it only works through careful distance approximations (e.g., using Euclidean spanners) and highly sophisticated modifications to JV.

Our New Variant of MP Algorithm. We give a new variant of another seminal algorithm, the MP algorithm [MP03], and show that it achieves our goal of working under arbitrary distance distortion (without requiring any careful design of the distance approximation).

We first describe our new variant. In the standard MP algorithm, a key notion is the value r_p for each point $p \in P$, proposed by [MP03] (or its generalization to any $z \ge 1$ by [GLS14]), which is the unique r_p with $\sum_q [r_p - \text{dist}(p, q)]^+ = \lambda$. At a high level, our variant consists of two steps (see also Algorithm 1):

1. we first compute, for each $p \in P$, a value $0 < r_p^{(\beta)} \le \lambda$, which is defined in the same way as r_p but with the power-z distances scaled by a fixed parameter β;
2. then, we open each point p as a facility with opening amount $r_p^{(\beta)}/\lambda$.

We refer to this algorithm as MP_β. We remark that, for the special case $z = 1$, our variant is exactly the same as a previous algorithm proposed by [BCG+24] which also establishes the LMP property. However, as mentioned earlier, their analysis does not handle distance distortion. Hence, even in this special case $z = 1$, our significant contribution is a novel analysis that handles distortion. Moreover, our algorithm and analysis work for any $z \ge 1$, and establishing this generalization is also a nontrivial challenge that requires new ideas, which we discuss later.

Overview of the LMP Analysis for MP_β under Distance Distortion. We show that if we run MP_β on distances distorted by an $O(1)$ factor, then the resulting fractional solution, denoted by $\widetilde{\boldsymbol{r}} = (\widetilde{r}_p)_{p\in P}$, is an $O(\beta^{-1})$-approximate LMP solution with respect to the *original* distances (see also Lemma 3.1). Below, we give an overview of the analysis, assuming $\lambda = 1$ for simplicity.

Our key insight is that the β-scaling operation in MP_β already suffices for handling distance distortion. Specifically, we observe that the resulting solution $\widetilde{\boldsymbol{r}}$ directly translates to the following two-sided approximation guarantee:

$$r_p^{(\Omega(\beta))} \le \widetilde{r}_p \le r_p^{(\beta)}, \qquad \forall p \in P. \tag{2}$$

Hence, it suffices to prove that any such solution $\widetilde{\boldsymbol{r}}$ satisfying (2) is an LMP $O(\beta^{-1})$-approximation, which requires novel technical ideas. For example, one might be tempted to argue that there exists β' such that $\widetilde{r}_p = r_p^{(\beta')}$ for all p, but this is unfortunately not true, which makes it infeasible to directly apply the result for the non-distorted case (e.g., [BCG+24]). In our argument, we employ a metric reassignment argument to carefully relate the opening and connection costs of $\widetilde{\boldsymbol{r}}$ and $\boldsymbol{r}^{(\beta)}$, via re-distributing mass of assignments to facilities within a metric ball, and we also establish new crucial properties of the MP algorithm to justify the existence of such reassignment, making this argument valid. This new proof strategy also allows us to handle any $z \ge 1$, which bypasses a fundamental obstacle in generalizing the $z = 1$ proof in [BCG+24] to the case $z \ge 2$, even for non-distorted distances (as we discuss later).

We remark that the guarantee in (2) is stronger than a plain $O(1)$-approximation to $r_p^{(\beta)}$, which is what previous work achieves [BCIS05, CJK+22, CGJ+24]. Crucially, simply having an $O(1)$-approximation to the r_p's does not yield an LMP solution with a finite ratio, similar to the issue of LMP breaking under distorted distances.

Proof Overview: The Guarantee in (2) Implies the LMP Property. To start, we formally generalize the LMP property to the fractional solutions we consider. A fractional solution $\widetilde{\boldsymbol{r}}$ opens each point p with amount $\widetilde{r}_p$. The opening cost is $\|\widetilde{\boldsymbol{r}}\|_1 = \sum_{p\in P} \widetilde{r}_p$ (recall that we assume $\lambda = 1$). We define the connection cost $\mathrm{cost}_z(\widetilde{\boldsymbol{r}})$ by assigning each point to its nearest facilities, in non-decreasing order of distance according to the capacities specified by $\widetilde{\boldsymbol{r}}$, such that the total assigned amount for each point is 1. We say that $\widetilde{\boldsymbol{r}}$ is an LMP α-approximation for some $\alpha \ge 1$ if $\mathrm{cost}_z(\widetilde{\boldsymbol{r}}) + \alpha\|\widetilde{\boldsymbol{r}}\|_1 \le \alpha\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}$, where $\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}$ denotes the optimal value of the fractional power-z facility location problem.

We start with the case $z = 1$, for which [BCG+24] shows that the undistorted fractional solution $\boldsymbol{r}^{(\beta)}$ is LMP β^{-1}-approximate for $\beta \in (0, 1/4]$, i.e.,

$$\mathrm{cost}_z(\boldsymbol{r}^{(\beta)}) + \beta^{-1}\|\boldsymbol{r}^{(\beta)}\|_1 \le \beta^{-1}\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}. \tag{3}$$

A natural (and ideal) plan is to show (a) the opening cost $\|\widetilde{\boldsymbol{r}}\|_1 \le \|\boldsymbol{r}^{(\beta)}\|_1$, which follows immediately from $\widetilde{\boldsymbol{r}} \le \boldsymbol{r}^{(\beta)}$ in (2), and (b) the connection cost $\mathrm{cost}_z(\widetilde{\boldsymbol{r}}) \le \alpha \cdot \mathrm{cost}_z(\boldsymbol{r}^{(\beta)})$ for some constant $\alpha \ge 1$. This would readily imply that $\widetilde{\boldsymbol{r}}$ is an LMP solution with ratio $\alpha\beta^{-1} = O(\beta^{-1})$. However, for (b), we obtain only the following weaker guarantee, whose proof we outline later using the inequality $r_p^{(\Omega(\beta))} \le \widetilde{r}_p$ in (2):

$$\mathrm{cost}_z(\widetilde{\boldsymbol{r}}) \le \alpha \cdot (\mathrm{cost}_z(\boldsymbol{r}^{(\beta)}) + \|\boldsymbol{r}^{(\beta)}\|_1 - \|\widetilde{\boldsymbol{r}}\|_1), \tag{4}$$

i.e., with an additive term that roughly corresponds to the gap between $\|\boldsymbol{r}^{(\beta)}\|_1$ and $\|\widetilde{\boldsymbol{r}}\|_1$. Luckily, this still suffices to establish the LMP property for $\widetilde{\boldsymbol{r}}$:

$$\begin{aligned}
& \mathrm{cost}_z(\widetilde{\boldsymbol{r}}) + \alpha\beta^{-1} \cdot \|\widetilde{\boldsymbol{r}}\|_1 \\
\le\ & \alpha \cdot \mathrm{cost}_z(\boldsymbol{r}^{(\beta)}) + \alpha \cdot (\|\boldsymbol{r}^{(\beta)}\|_1 - \|\widetilde{\boldsymbol{r}}\|_1) + \alpha\beta^{-1} \cdot \|\widetilde{\boldsymbol{r}}\|_1 \\
\le\ & \alpha \cdot \mathrm{cost}_z(\boldsymbol{r}^{(\beta)}) + \alpha\beta^{-1} \cdot (\|\boldsymbol{r}^{(\beta)}\|_1 - \|\widetilde{\boldsymbol{r}}\|_1) + \alpha\beta^{-1} \cdot \|\widetilde{\boldsymbol{r}}\|_1 \\
=\ & \alpha \cdot \mathrm{cost}_z(\boldsymbol{r}^{(\beta)}) + \alpha\beta^{-1} \cdot \|\boldsymbol{r}^{(\beta)}\|_1
\end{aligned}$$

$$\leq \alpha\beta^{-1}\widehat{\mathrm{OPT}}_z^{\mathrm{fl}},$$

where the second inequality holds because $\|\boldsymbol{r}^{(\beta)}\|_1 - \|\widetilde{\boldsymbol{r}}\|_1 \geq 0$.

A Metric Reassignment Argument and New Properties of MP. To establish (4), we employ a metric reassignment argument. In particular, we consider the (fractional) assignment $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{P\times P}$ that realizes $\mathrm{cost}_z(\boldsymbol{r}^{(\beta)})$ (i.e., it assigns P to $\boldsymbol{r}^{(\beta)}$ according to the nearest neighbor rule), where $x_{p,q}$ denotes the amount assigned from p to q. This $\boldsymbol{x}$ may not be a feasible assignment with respect to $\widetilde{\boldsymbol{r}}$, since there may exist p, q such that $x_{p,q} > \widetilde{r}_q$, i.e., the assigned amount exceeds the opening at q. The plan is to use a reassignment argument to transform $\boldsymbol{x}$ into $\boldsymbol{x}'$ that is feasible with respect to $\widetilde{\boldsymbol{r}}$, with only a small overhead. Fix a point p. For each facility q such that the assigned amount $x_{p,q}$ exceeds the opening $\widetilde{r}_q$, we reassign the excess amount $x_{p,q} - \widetilde{r}_q \leq r_q^{(\beta)} - \widetilde{r}_q$ (note that $x_{p,q} \leq r_q^{(\beta)}$ since $\boldsymbol{x}$ is feasible with respect to $\boldsymbol{r}^{(\beta)}$) to other facilities located in the local area $B(p, O(\widetilde{r}_p))$. This is roughly where the additive term in (4) arises, although part of the cost also needs to be charged to $\mathrm{cost}_z(\boldsymbol{r}^{(\beta)})$.

To make this plan work, we establish a new property of the MP algorithm, which guarantees that the area $B(p, O(r_p^{(\beta)}))$ contains a sufficiently large total opening, i.e., $\sum_{x\in B(p,O(r_p^{(\beta)}))} r_x^{(\beta)} \geq 1$ for any $\beta \in (0, 1)$ and $p \in P$ (see Lemma 3.9). This further translates to a corresponding guarantee for $\widetilde{\boldsymbol{r}}$ via $\widetilde{\boldsymbol{r}} \geq \boldsymbol{r}^{(\Omega(\beta))}$ in (2).

Generalizing to $z \geq 2$: New Steps. Finally, we discuss how to generalize the algorithm and analysis to $z \geq 2$. The first step is to generalize the definition of the r_p's, and we make use of a natural generalization from [GLS14, CGJ+24]. While the above argument for $\widetilde{\boldsymbol{r}}$ generally works, it actually relies on one key premise: exact $\boldsymbol{r}^{(\beta)}$ is LMP (i.e., whether (3) holds), which has been proved in [BCG+24] for $z = 1$. Unfortunately, this premise is not known to hold for $z \geq 2$. Therefore, we need an additional step to prove that $\boldsymbol{r}^{(\beta)}$ is LMP. To this end, we generalize the analysis of [BCG+24] from the $z = 1$ case to $z \geq 2$. The key step in their approach is to define an assignment $\boldsymbol{x}'$ that is feasible with respect to $\boldsymbol{r}^{(\beta)}$, and to show that its connection cost, which upper-bounds $\mathrm{cost}_z(\boldsymbol{r}^{(\beta)})$, is good enough for LMP. However, we provide a counterexample showing that this assignment $\boldsymbol{x}'$ may not be feasible for $z \geq 2$ (see Example 3.19), which highlights a key difference from the $z = 1$ case. Nonetheless, we show that a slightly weaker inequality $\mathrm{cost}_z(\boldsymbol{r}^{(\beta)}) \leq O(1) \cdot \mathrm{cost}_z(\boldsymbol{x}')$ still holds (Lemma 3.16), which is sufficient to establish the LMP property of $\boldsymbol{r}^{(\beta)}$ for the $z \geq 2$ case. Interestingly, this is achieved using a similar idea as in the above reassignment argument: starting from $\boldsymbol{x}'$, we reassign, for each facility p, the excess amount to a local ball $B(p, O(r_p^{(\beta)}))$. The key difference is that we cannot afford to charge an additive term as in (4); instead, we exploit the definition of $\boldsymbol{x}'$ together with a new smoothness property of $r_p^{(\beta)}$ (Lemma 3.8) to remedy this.

1.1.2 Overview of Our MPC Rounding Algorithm

We now give a more detailed overview of our MPC rounding algorithm, which bypasses the filtering operation and ruling-set computation that have been major bottlenecks to efficient MPC implementations of previous rounding algorithms. Given a dataset P and a fractional solution $\boldsymbol{y} \in \mathbb{R}_{\geq 0}^P$ with $\|\boldsymbol{y}\|_1 = k$ (opening each p with amount y_p as a center), our algorithm returns an integral set $C \subseteq P$ with $|C| \leq k$ and $\mathrm{cost}_z(C) \leq O(1) \cdot \mathrm{cost}_z(\boldsymbol{y})$.

Starting Point: Expected Size Guarantee. We first observe that relaxing the requirement of opening *exactly* k centers to opening k centers *in expectation* requires only *local* computation. Indeed, there already exists a rounding algorithm with an expected k-size guarantee that determines whether to open each facility based only on "nearby" facilities and points [ANS17], and hence can be simulated using geometric range queries, which are standard $O(1)$-round primitives in MPC [CGJ+24]. However, it is still challenging to further strengthen the expected-size guarantee to a high-probability guarantee.

Converting to a Bicriteria Solution. Our idea is to instead aim for a bicriteria solution with $k + O(n^{1-\varepsilon})$ centers. For a random center set C with $\mathbb{E}[|C|] \le k$, we can use Markov's inequality to deduce that $\Pr[|C| > k + n^{1-\varepsilon}] \le \mathbb{E}[|C|]/(k + n^{1-\varepsilon}) \le 1 - \Omega(n^{-\varepsilon})$. Hence, by running the algorithm $\widetilde{O}(n^{\varepsilon})$ times independently in parallel, at least one of the resulting center sets satisfies the size guarantee $|C| \le k + n^{1-\varepsilon}$ with high probability. This parallel repetition increases the total memory usage by a factor of $\widetilde{O}(n^{\varepsilon})$, which we can afford.

However, there is another issue concerning the cost guarantee. What the algorithm of [ANS17] actually gives is

$$\mathbb{E}[|C|] \le k \quad \text{and} \quad \mathbb{E}[\mathrm{cost}_z(C)] \le O(1) \cdot \mathrm{cost}_z(\boldsymbol{y}), \tag{5}$$

where the cost is also bounded only in expectation. Unfortunately, there may be a correlation between size and cost, so that solutions with small size may have much larger costs. In particular, conditioning on the event $|C| \le k + n^{1-\varepsilon}$, which may have probability as small as $\Omega(n^{-\varepsilon})$, the conditional expected cost may be larger by a factor of up to $O(n^{\varepsilon})$. Consequently, simply repeating the algorithm does not necessarily guarantee that both $|C| \le k+n^{1-\varepsilon}$ and $\mathrm{cost}_z(C) \le O(1)\cdot\mathrm{cost}_z(\boldsymbol{y})$ hold *simultaneously* with high probability.

To tackle this issue, we introduce an additional algorithmic step based on *bi-point rounding* [JV01]. Specifically, despite the potential correlation between the size and cost of each solution, we show that, with high probability, there exists a *bi-point solution* (i.e., a convex combination of two center sets) among $t = n^{O(\varepsilon)}$ independently rounded solutions $C_1, \dots, C_t$. To see this, we examine the *average* size $\frac{1}{t}\sum_{i=1}^{t}|C_i|$ and the *average* cost $\frac{1}{t}\sum_{i=1}^{t}\mathrm{cost}_z(C_i)$. By standard concentration analysis, the following inequalities hold simultaneously with constant probability:

$$\frac{1}{t}\sum_{i=1}^{t}|C_i| \le k + O(n^{1-\varepsilon}) \quad \text{and} \quad \frac{1}{t}\sum_{i=1}^{t}\mathrm{cost}_z(C_i) \le O(1)\cdot\mathrm{cost}_z(\boldsymbol{y}) \tag{6}$$

The success probability can then be amplified to high probability by independent repetition. Now, given that (6) holds, a simple LP extreme-point argument implies that there exist two (possibly identical) solutions C_i, C_j among $C_1, \dots, C_t$ and a coefficient $\alpha \in [0,1]$ such that

$$\alpha|C_i| + (1-\alpha)|C_j| \le k + O(n^{1-\varepsilon}) \quad \text{and} \quad \alpha\mathrm{cost}_z(C_i) + (1-\alpha)\mathrm{cost}_z(C_j) \le O(1)\cdot\mathrm{cost}_z(\boldsymbol{y}). \tag{7}$$

That is, we obtain a bi-point solution (C_i, C_j, α). We then apply a bi-point rounding algorithm to obtain a center set C satisfying

$$|C| \le k + O(n^{1-\varepsilon}) \quad \text{and} \quad \mathrm{cost}_z(C) \le O(1)\cdot\mathrm{cost}_z(\boldsymbol{y}),$$

which is the desired bicriteria solution. Here, we note that bi-point rounding is already known to admit an $O(1)$-round implementation in the fully scalable MPC model; see [CKP26]. Therefore, the overall conversion from an algorithm with expected guarantees to one with high-probability guarantees only takes $O(1)$ MPC rounds.

Iterative Kernelization. So far, by combining our fractional algorithm with the rounding procedure above, we have obtained a bicriteria solution for (k, z)-Clustering with $k+O(n^{1-\varepsilon})$ centers (see also Theorem 4.1). The remaining task is to remove the extra $r = O(n^{1-\varepsilon})$ centers. This task has been extensively studied in the clustering literature (see, e.g., [Tho04, LS16, BCG+24, BCF25, CGL+25, BCF+26, BGH+26]), and we build on the technique of [BCF+26], which reduces this task to solving a clustering problem on a significantly smaller point set of size $O(r) = O(n^{1-\varepsilon})$. By combining our bicriteria algorithm with this technique, we obtain a *kernelization* step that, roughly speaking, converts the clustering input into a new one with the number of centers k reduced by a factor of n^{ε},[2] and a solution to the new input can be transformed into a solution to the original one with a constant $\mathrm{poly}(1/\varepsilon)$-factor loss in the approximation ratio. More importantly, this reduction mainly relies on approximate nearest neighbor search (as well as some basic operations such as sorting), and thus admits an $O(1)$-round MPC implementation; more concretely, $O(\log_s n) = O(\varepsilon^{-1})$ rounds in the fully scalable setting with $s = n^{\varepsilon}$ for any constant $\varepsilon \in (0, 1)$.

By iteratively applying this kernelization $O(\varepsilon^{-1})$ times, we eventually only need to solve clustering with a constant k, which can be handled readily in constant rounds. We lift this solution back through the sequence of $O(\varepsilon^{-1})$ kernelization steps for the original dataset. Since each kernelization step increases the approximation ratio by at most a $\mathrm{poly}(1/\varepsilon)$ factor, our final solution is an $\varepsilon^{-O(1/\varepsilon)}$-approximation. Moreover, each kernelization takes $O(\varepsilon^{-1})$ rounds, and we have only $O(\varepsilon^{-1})$ kernelizations. This gives an overall round complexity of $O(\varepsilon^{-2})$. Both the approximation ratio and the round complexity are thus constant for fixed ε.

1.2 Related Work

Fully scalable MPC algorithms have also been studied for fundamental problems in Euclidean spaces other than k-Means and k-Median. In the context of clustering, closely related problems include k-Center [BEFM21, CCM23, CGGJ25] and Facility Location [CGJ+24]. To date, the best known fully scalable algorithm for k-Center achieves an $O(\log n/\log\log n)$-approximation in a constant number of rounds [CGGJ25], and in low dimensions, this approximation ratio can be further improved to $2+\varepsilon$ (or $1+\varepsilon$ for a bicriteria solution). For Facility Location, the streaming algorithm of [CJK+22] can be directly translated into an $O(1)$-round MPC algorithm with an $O(1)$-approximation, but this only applies to estimating the optimal value. More recently, [CGJ+24] resolved the solution version of Facility Location by achieving the same $O(1)$-approximation in $O(1)$ rounds.

Beyond clustering, Minimum Spanning Tree (MST) has attracted considerable interest in Euclidean space. For this problem, there is a notable gap between the value version and the solution version. For the value version, using the streaming algorithm that achieves an $O(1/\varepsilon^2)$-approximation with space n^{ε} [CCJ+23], one can already obtain an $O(1)$-approximation in $O(1)$ rounds. In contrast, it remains open to find an $O(1)$-approximate solution for Euclidean MST in $O(1)$ rounds. Indeed, the currently best-known algorithms either find an $O(1)$-approximate MST in $\widetilde{O}(\log\log n)$ rounds [JMNZ24], or achieve a $\mathrm{polylog}(n)$-approximation in $O(1)$ rounds [AAH+23]. In low dimensions, e.g., $d = O(1)$, one can find an $O(1)$-approximation in $O(1)$ rounds [ANOY14]. Other fundamental problems studied in this context include the Earth Mover Distance [ANOY14] and Max-Cut [MW26].

Beyond the fully scalable regime, constant-round MPC algorithms for k-Median and k-Means have long been known when the local memory may grow with k, e.g., via sampling [EIM11], parallel k-means++ seeding [BMV+12], or composable coresets [BEL13, BBLM14]; all of them keep a

[2] More precisely, in our algorithm in Section 5, we reduce the size of a *candidate center set* (from which all centers are chosen), which serves as an upper bound on k.

summary of size $\Omega(k)$ on a single machine. Sequentially, LMP-based primal-dual algorithms [JV01, JMS02, JMM+03] underlie many of the best known ratios; notably, Ahmadian, Norouzi-Fard, Svensson, and Ward [ANFSW17] obtained a 6.357-approximation for Euclidean k-MEANS (and 2.633 for Euclidean k-MEDIAN) via a primal-dual algorithm that opens exactly k centers, but such algorithms rely on global pruning steps that are hard to parallelize. Very recently, Bhore, Filtser, Oh, Shin, and Singh [BFO+27] extended consistent hashing, the key primitive behind recent fully scalable geometric MPC algorithms, to doubling metrics, obtaining, among other results, a fully scalable bicriteria approximation for k-MEDIAN and k-MEANS that opens $(1+\varepsilon)k$ centers.

2 Preliminaries

Notation. For an integer $n \geq 1$, let $[n] := \{1, \ldots, n\}$, and for integers $m \leq n$, let $[m : n] := \{m, \ldots, n\}$. For $a \in \mathbb{R}$, let $[a]^+ := \max\{a, 0\}$. Throughout, $z \geq 1$ is a fixed integer, (V, dist) is a metric space (in the Euclidean setting, $V = \mathbb{R}^d$ and $\mathrm{dist}(p, q) = \|p - q\|_2$), and for a point p and a finite set Q we write $\mathrm{dist}(p, Q) := \min_{q \in Q} \mathrm{dist}(p, q)$. Vectors indexed by a finite set S (e.g., P, F, or $P \times F$) are in boldface, $\boldsymbol{a} \in \mathbb{R}^S$, with entries a_s; vector inequalities are entrywise, for example, $\boldsymbol{a} < \boldsymbol{b}$ if and only if $a_q < b_q$ for all $q \in S$.

For a point $p \in \mathbb{R}^d$ and a set $Q \subseteq \mathbb{R}^d$, we define $\mathrm{NN}(p, Q) := \mathrm{argmin}_{q \in Q} \mathrm{dist}(p, q)$ as a *nearest neighbor of p in Q*, with ties broken arbitrarily. We say that a point $q \in Q$ is an *α-approximate nearest neighbor* (*α-ANN* for short) of p in Q, for $\alpha \geq 1$, if $\mathrm{dist}(p, q) \leq \alpha \cdot \mathrm{dist}(p, Q)$. For a point set $Q \subseteq \mathbb{R}^d$, we define its *aspect ratio* as $\Delta(Q) := \frac{\max_{p,q \in Q} \mathrm{dist}(p,q)}{\min_{p \neq q \in Q} \mathrm{dist}(p,q)}$.

Fact 2.1 (Generalized Triangle Inequality)**.** *Let (P, dist) be a metric space. For any $p, q, t \in P$, it holds that* $\mathrm{dist}^z(p, q) \leq 2^{z-1} \cdot (\mathrm{dist}^z(p, t) + \mathrm{dist}^z(t, q))$.

Metric Clustering. We define a more general formulation of (k, z)-CLUSTERING as follows. For a general metric space (V, dist), the input consists of a dataset $P \subseteq V$, a candidate center set $F \subseteq V$, and an integer $1 \leq k \leq |F|$. The goal is to find a center set $C \subseteq F$ of size at most k that minimizes $\mathrm{cost}_z(P, C) := \sum_{p \in P} \mathrm{dist}^z(p, C)$, where $\mathrm{dist}(p, C) := \min_{c \in C} \mathrm{dist}(p, c)$. We define $\mathrm{OPT}^{\mathrm{cl}}_z(P, F, k) := \min_{C \subseteq F : |C| \leq k} \mathrm{cost}_z(P, C)$. This formulation includes the Euclidean (k, z)-CLUSTERING considered in Section 1 as the special case where $V = F = \mathbb{R}^d$ and dist is the Euclidean distance.

Fractional Solution to Clustering. An important notion in this paper is a *fractional* solution to (k, z)-CLUSTERING, defined as a vector $\boldsymbol{y} \in \mathbb{R}^F_{\geq 0}$ satisfying $1 \leq \|\boldsymbol{y}\|_1 \leq k$, where F is the given candidate center set. Each entry y_q represents the fractional opening of a candidate center $q \in F$, and the condition $1 \leq \|\boldsymbol{y}\|_1 \leq k$ means that at least one and at most k units of total fractional opening are placed on the candidate centers.

To define the *clustering cost* of $\boldsymbol{y}$, we start by defining the *fractional assignments* and the *assignment cost*. A fractional assignment is represented by a vector $\boldsymbol{x} = (x_{p,q})_{p \in P, q \in F} \in \mathbb{R}^{P \times F}_{\geq 0}$, where each entry $x_{p,q}$ represents the amount by which $p \in P$ is assigned to $q \in F$. The *assignment cost* of $\boldsymbol{x}$ is defined as

$$\sigma_z(P, \boldsymbol{x}) \;:=\; \sum_{p \in P} \sum_{q \in F} x_{p,q} \cdot \mathrm{dist}^z(p, q).$$

We say that $\boldsymbol{x}$ is *feasible* with respect to a fractional center set $\boldsymbol{y}$, denoted by $\boldsymbol{x} \sim \boldsymbol{y}$, if the following constraints are satisfied.

(C1) $\forall p \in P$, $\sum_{q \in F} x_{p,q} \geq 1$, i.e., each point p is assigned to a total of at least 1 unit of centers.

(C2) $\forall p \in P, q \in F$, $x_{p,q} \leq y_q$, i.e., the assignment amount of p to q is at most the opening of q.

We can then define the clustering cost of a fractional center set $\boldsymbol{y} \in \mathbb{R}^F_{\geq 0}$ with $\|\boldsymbol{y}\|_1 \geq 1$ as

$$\mathrm{cost}_z(P, \boldsymbol{y}) := \min_{\boldsymbol{x} \sim \boldsymbol{y}} \sigma_z(P, \boldsymbol{x}).$$

Let $\boldsymbol{x}^* \sim \boldsymbol{y}$ be the optimal feasible assignment with respect to $\boldsymbol{y}$, that is, $\mathrm{cost}_z(\boldsymbol{y}) = \sigma_z(\boldsymbol{x}^*)$. We define the individual cost of each point $p \in P$ with respect to $\boldsymbol{y}$ as

$$\mathrm{cost}_z(p, \boldsymbol{y}) \;:=\; \sum_{q \in F} x^*_{p,q} \cdot \mathrm{dist}^z(p, q) \;=\; \min_{\boldsymbol{x}_p \in \mathbb{R}^F_{\geq 0} : \|\boldsymbol{x}_p\|_1 \geq 1, \boldsymbol{x}_p \leq \boldsymbol{y}} \sum_{q \in F} x_{p,q} \cdot \mathrm{dist}^z(p, q),$$

where the last equality follows from the observation that the Constraints (C1) to (C2) are independent for each data point p; thus, $\boldsymbol{x}^*$ also minimizes the assignment cost of each individual point to the fractional center set $\boldsymbol{y}$.

We remark that the notion of a fractional solution is widely used in the clustering literature, especially in works that involve linear-programming relaxations (e.g. [CGTS99, CL12, LS16, CEMN22, BGH+26]). Moreover, it is well known that the gap between the optimal cost of a fractional solution and that of an integral solution (i.e., a center set), referred to as the *integrality gap* of (k, z)-Clustering, is $2^{O(z)}$ (see, e.g., [BGH+26]), as stated below.

Lemma 2.2 (Integrality Gap for Clustering [BGH+26])**.** $2^{-O(z)} \cdot \mathrm{OPT}^{\mathrm{cl}}_z(P, F, k) \leq \widehat{\mathrm{OPT}}^{\mathrm{cl}}_z(P, F, k) \leq \mathrm{OPT}^{\mathrm{cl}}_z(P, F, k)$*, where* $\widehat{\mathrm{OPT}}^{\mathrm{cl}}_z(P, F, k) := \min_{\boldsymbol{y} \in \mathbb{R}^F_{\geq 0} : 1 \leq \|\boldsymbol{y}\|_1 \leq k} \mathrm{cost}_z(P, \boldsymbol{y})$.

In this paper, when the context is clear, we may omit P from the notation and, for example, write $\mathrm{cost}_z(P, C)$, $\mathrm{cost}_z(P, \boldsymbol{y})$, and $\sigma_z(P, \boldsymbol{x})$ as $\mathrm{cost}_z(C)$, $\mathrm{cost}_z(\boldsymbol{y})$, and $\sigma_z(\boldsymbol{x})$, respectively.

3 Fractional MP Algorithm with Robust LMP Approximation

In this section, we present a new *fractional Mettu-Plaxton (MP) algorithm*, which admits a crucial *Lagrangian multiplier preserving (LMP)* property [JV01] and works even under *distance distortions*. This algorithm is a variant of the seminal MP algorithm [MP03]. A similar fractional MP algorithm is also considered in [BCG+24], but only for the case $z = 1$ and without allowing distance distortions, whereas our algorithm handles both general $z \geq 1$ and distance distortions. This algorithm is designed for the closely related Power-z Facility Location problem. Below, we start by defining Power-z Facility Location and the LMP property, then present our fractional MP algorithm, and finally discuss how to implement it efficiently in the MPC model.

Power-z Facility Location. In the Power-z Facility Location problem, the input additionally includes an opening cost $\lambda > 0$, and the hard size constraint $|C| \leq k$ in (k, z)-Clustering is replaced by a uniform opening cost λ per center in the objective. In this paper, we consider the special case where $P = F$, i.e., the centers/facilities are chosen from the dataset P, and focus on computing a fractional solution. Hence, the input consists of a dataset P and an opening cost λ, and the goal is to find a fractional solution $\boldsymbol{y} \in \mathbb{R}^P_{\geq 0}$ that minimizes $\lambda \cdot \|\boldsymbol{y}\|_1 + \mathrm{cost}_z(P, \boldsymbol{y})$. Let $\widehat{\mathrm{OPT}}^{\mathrm{fl}}_z(P, \lambda) := \min_{\boldsymbol{y} \in \mathbb{R}^P_{\geq 0} : \|\boldsymbol{y}\|_1 \geq 1} \lambda \cdot \|\boldsymbol{y}\|_1 + \mathrm{cost}_z(P, \boldsymbol{y})$ denote the optimal objective value of Power-z Facility Location over all fractional solutions.

Throughout this section, we consider only a single dataset P. Therefore, for simplicity, we omit P from the notation, writing $\mathrm{cost}_z(P, C)$, $\mathrm{cost}_z(P, \boldsymbol{y})$, $\sigma_z(P, \boldsymbol{x})$, and $\widehat{\mathrm{OPT}}^{\mathrm{fl}}_z(P, \lambda)$ as $\mathrm{cost}_z(C)$, $\mathrm{cost}_z(\boldsymbol{y})$, $\sigma_z(\boldsymbol{x})$, and $\widehat{\mathrm{OPT}}^{\mathrm{fl}}_z(\lambda)$, respectively.

Lagrangian Multiplier Preserving (LMP). We say that a fractional solution $\boldsymbol{y} \in \mathbb{R}^P_{\geq 0}$ (with $\|\boldsymbol{y}\|_1 \geq 1$) is an LMP α-approximation (for some $\alpha \geq 1$) if

$$\mathrm{cost}_z(\boldsymbol{y}) \;\leq\; \alpha \cdot \big(\widehat{\mathrm{OPT}}^{\mathrm{fl}}_z(\lambda) - \lambda \cdot \|\boldsymbol{y}\|_1\big).$$

An LMP approximation is a stronger notion than a standard approximation and is crucial for transforming a (fractional) solution for Power-z Facility Location into one for (k,z)-Clustering.

Robust Fractional MP Algorithm. We describe our new algorithm for Power-z Facility Location in Algorithm 1, which is parameterized by a scale factor $\beta \in (0,1)$, and we therefore refer to it as MP_β. This algorithm works for general metric spaces, where the distance function need not be Euclidean; for simplicity, we assume that the underlying set $V = P$ is the dataset itself.

At a high level, the algorithm MP_β computes a value $r_p^{(\beta)} \geq 0$ for each point $p \in P$, and then opens p with amount $r_p^{(\beta)}/\lambda$ as a facility; the value $r_p^{(\beta)}$ is computed in a manner similar to [MP03] (specifically, a generalized version as in [GLS14, CGJ+24] to handle power-z), but after scaling the power-z distances by β.

Algorithm 1: MP_β with $\beta \in (0,1)$ on input (P, dist).

1 For each point $p \in P$, compute a value $r_p^{(\beta)} > 0$ such that

$$\sum_{q \in P} \big[r_p^{(\beta)} - \beta \cdot \mathrm{dist}^z(p,q)\big]^+ = \lambda \tag{8}$$

2 **return** $\boldsymbol{y}^{(\beta)} := (r_p^{(\beta)}/\lambda)_{p \in P} \in \mathbb{R}^P_{\geq 0}$

Our main result of this section is the following Lemma 3.1, which establishes the correctness and robustness of MP_β. Specifically, we show that if MP_β is instead run on input $(P, \widetilde{\mathrm{dist}})$, where $\widetilde{\mathrm{dist}}$ is a distorted distance function, then by setting the scale factor β to be a sufficiently small value depending on the distortion factor, the algorithm MP_β still yields an LMP approximation with respect to the original distance dist. Notably, the distorted distance $\widetilde{\mathrm{dist}}$ is not required to be a metric (and need not even be symmetric).

Lemma 3.1. *Let $\varepsilon \in (0,1)$ be a constant and suppose that a distorted distance function $\widetilde{\mathrm{dist}}$ satisfies*

$$\mathrm{dist}(p,q) \;\leq\; \widetilde{\mathrm{dist}}(p,q) \;\leq\; \varepsilon^{-2} \cdot \mathrm{dist}(p,q), \qquad \forall p,q \in P. \tag{9}$$

Let $\widetilde{\boldsymbol{y}} \in \mathbb{R}^P_{\geq 0}$ denote the output of MP_β (Algorithm 1) with $\beta := 2^{-z-1} \cdot \varepsilon^{2z}$ on input $(P, \widetilde{\mathrm{dist}})$ (instead of (P, dist)). Then $\widetilde{\boldsymbol{y}}$ is a fractional solution to Power-z Facility Location with an LMP $2^{O(z^2)}\varepsilon^{-2z}$-approximation guarantee on the original metric space, i.e.,

$$\mathrm{cost}_z(\widetilde{\boldsymbol{y}}) \;\leq\; 2^{O(z^2)}\varepsilon^{-2z} \cdot \big(\widehat{\mathrm{OPT}}^{\mathrm{fl}}_z(\lambda) - \lambda \cdot \|\widetilde{\boldsymbol{y}}\|_1\big),$$

where both $\mathrm{cost}_z(\widetilde{\boldsymbol{y}})$ and $\widehat{\mathrm{OPT}}^{\mathrm{fl}}_z(\lambda)$ are evaluated with respect to the original distance dist.

We prove Lemma 3.1 using Sections 3.1 to 3.3. Specifically, in Section 3.1, we establish several structural properties of the vector $\boldsymbol{r}^{(\beta)} = (r_p^{(\beta)})_{p \in P}$ computed as in (8) on the original/undistorted metric (P, dist); in Section 3.2, we prove the correctness of MP_β without distance distortions; and finally, in Section 3.3, we prove the correctness of MP_β under distance distortions, i.e., Lemma 3.1.

MPC Implementation in Euclidean Space. Finally, we consider the Euclidean setting, where $P \subseteq \mathbb{R}^d$ and dist denotes the Euclidean distance, and design an efficient MPC algorithm based on the robust MP_β for computing a fractional LMP approximation for Power-z Facility Location.

Corollary 3.2. *For any constant $\varepsilon \in (0,1)$ and integer $z \geq 1$, there is a deterministic MPC algorithm for* Power-z Facility Location *in Euclidean space that,*

- *takes as input an n-point dataset $P \subseteq \mathbb{R}^d$, an opening cost $\lambda > 0$, and an upper bound $\Delta \geq \Delta(P)$ on the aspect ratio;*
- *computes an LMP $2^{O(z^2)}\varepsilon^{-2z}$-approximation for the dataset P.*

With local memory $s \geq \mathrm{poly}(d \log(n\Delta))$ per machine, the algorithm runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n \cdot \mathrm{poly}(d \log(n\Delta))$ total memory.

Proof. We present in Algorithm 2 a more involved (offline) algorithm than MP_β (Algorithm 1), which is more amenable to MPC implementation. The input to Algorithm 2 is $P \subseteq \mathbb{R}^d$, and it considers the Euclidean distance dist. We show that its output is equivalent to that of MP_β on input $(P, \widetilde{\mathrm{dist}})$, for some distorted distance $\widetilde{\mathrm{dist}}$, thereby establishing its correctness (i.e., an LMP approximation) via Lemma 3.1, and finally discuss its MPC implementation.

Throughout this proof, we assume that for every $p \neq q \in P$, $2 \leq \mathrm{dist}(p,q) \leq O(\Delta)$. This is without loss of generality, since we can apply a standard scaling preprocessing step to the dataset to ensure this property in $O(\log_s n)$ rounds using $O(n \cdot \mathrm{poly}(d \log(n\Delta)))$ total memory (see, e.g., [CGJ+24]).

Algorithm 2: MPC-implementable (offline) variant of MP_β

1 $L \leftarrow \Theta(\log \Delta)$ be sufficiently large
2 $\beta \leftarrow 2^{-z-1} \cdot \varepsilon^{2z}$ `// the same factor as specified in Lemma 3.1`
3 For each point $p \in P$, compute sets $A_p^{(0)} \subseteq \cdots \subseteq A_p^{(L)}$ such that
$\{q \in P : \mathrm{dist}(p,q) \leq \varepsilon^{-(\ell-1)}\} \subseteq A_p^{(\ell)} \subseteq \{q \in P : \mathrm{dist}(p,q) \leq \varepsilon^{-\ell}\}$, $\forall \ell \in [0:L]$
4 For each point $p \in P$, compute the solution $\widetilde{r}_p \geq 0$ to the equation
$\widetilde{r}_p + \sum_{\ell \in [L]} (|A_p^{(\ell)}| - |A_p^{(\ell-1)}|) \cdot [\widetilde{r}_p - \beta \cdot \varepsilon^{-\ell z}]^+ = \lambda$
`// It is easy to see from Fact 3.4 that there exists a unique solution` $\widetilde{r}_p$.
5 **return** $\widetilde{\boldsymbol{y}} := \widetilde{\boldsymbol{r}}/\lambda$

The correctness analysis of Algorithm 2 begins with the following Facts 3.3 and 3.4, concerning the sets $A_p^{(\ell)}$ computed in Line 3.

Fact 3.3. *$\varepsilon^{-(\ell-2)} \leq \mathrm{dist}(p,q) \leq \varepsilon^{-\ell}$, for any point $p \in P$, level $\ell \in [L]$, and point $q \in A_p^{(\ell)} \setminus A_p^{(\ell-1)}$.*

Proof. This is a direct consequence of Line 3. □

Fact 3.4. *$\{p\} = A_p^{(0)} \subseteq \cdots \subseteq A_p^{(L)} = P$ for every $p \in P$, when $L = \Theta(\log \Delta)$ is sufficiently large.*

Proof. The inclusion $A_p^{(0)} \subseteq \cdots \subseteq A_p^{(L)}$ is a direct consequence of Line 3. The first equality $A_p^{(0)} = \{p\}$ holds since $\varepsilon^{-0} = 1 < 2 \leq \min_{p \neq q \in P} \mathrm{dist}(p,q)$, and the second equality $A_p^{(L)} = P$ holds since a sufficiently large $L = \Theta(\log \Delta)$ leads to $\varepsilon^{-(L-1)} \geq \max_{p \neq q \in P} \mathrm{dist}(p,q) = O(\Delta)$. This finishes the proof of Fact 3.4. □

By Fact 3.4, the LHS of the equation in Line 4 is continuous and increasing in $\widetilde{r}_p$, which guarantees the existence and uniqueness of $\widetilde{r}_p$.

We define a distorted distance function $\widetilde{\mathrm{dist}}$ as follows:

- for every $p \in P$, let $\widetilde{\mathrm{dist}}(p,p) := 0 = \mathrm{dist}(p,p)$;
- for every $p \neq q \in P$, let $\ell \in [L]$ be such that $q \in A_p^{(\ell)} \setminus A_p^{(\ell-1)}$, and then let $\widetilde{\mathrm{dist}}(p,q) := \varepsilon^{-\ell}$; the existence and uniqueness of ℓ is guaranteed by Fact 3.4.

By Fact 3.3, we have $\mathrm{dist}(p,q) \le \widetilde{\mathrm{dist}}(p,q) \le \varepsilon^{-2} \cdot \mathrm{dist}(p,q)$ for every $p \neq q \in P$, and therefore the distorted distance function $\widetilde{\mathrm{dist}}$ satisfies (9). Moreover, by Fact 3.4, the definition of $\widetilde{\mathrm{dist}}$, and the computation of $\widetilde{r}_p$ (Line 4), we can deduce that

$$\sum_{q \in P} \left[\widetilde{r}_p - \beta \cdot \left(\widetilde{\mathrm{dist}}(p,q)\right)^z\right]^+ = \lambda.$$

Hence, the output $\widetilde{\boldsymbol{y}} = \widetilde{\boldsymbol{r}}/\lambda$ of Algorithm 2 is identical to the output of MP_β (Algorithm 1) with $\beta = 2^{-z-1} \cdot \varepsilon^{2z}$ on input $(P, \widetilde{\mathrm{dist}})$, and therefore Lemma 3.1 implies that $\widetilde{\boldsymbol{y}}$ is an LMP $2^{O(z^2)}\varepsilon^{-2z}$-approximation on the original Euclidean (P, dist).

Finally, we discuss the MPC implementation of Algorithm 2, and it suffices to focus on the nontrivial Lines 3 and 4.

- Line 3 aims to compute the sets $A_p^{(\ell)}$, whose faithful implementation in MPC is inefficient. Nonetheless, in fact, the latter Line 4 only requires the sizes $|A_p^{(\ell)}|$. For a single level $\ell \in [0:L]$, the sizes $\{|A_p^{(\ell)}|\}_{p \in P}$ can be computed deterministically using an MPC primitive from [CGJ+24, Theorem 3.1]. By running $L+1$ instances of this primitive—one per level $\ell \in [0:L]$—in parallel, we obtain all sizes $\{|A_p^{(\ell)}|\}_{p \in P, \ell \in [0:L]}$. This parallel invocation requires local memory $s \ge \mathrm{poly}(d \log n)$ and uses $O(\log_s n)$ rounds, exactly as in [CGJ+24, Theorem 3.1]; the total memory is $2^{O(\varepsilon d)} \cdot n \cdot \mathrm{poly}(d \log n) \cdot \log \Delta$, which is larger by a factor of $L = \Theta(\log \Delta)$ since we need to replicate the data L times for parallelization.
- To implement Line 4, we can redistribute the data so that for each point $p \in P$, the machine storing p also stores all sizes $\{|A_p^{(\ell)}|\}_{\ell \in [0:L]}$; then, the value $\widetilde{r}_p$ can be computed locally, with no further communication. Such redistribution can be MPC-implemented via, e.g., sorting [GSZ11], requiring local memory $s \ge L+1 = O(\log \Delta)$ (to store all sizes $\{|A_p^{(\ell)}|\}_{\ell \in [0:L]}$ in a single machine), using total memory $O(nL) = O(n \log \Delta)$, and in $O(\log_s n)$ rounds.

In total, the entire MPC implementation requires local memory $s \ge \mathrm{poly}(d \log(n\Delta))$, runs in $O(\log_s n)$ rounds, and uses total memory $2^{O(\varepsilon d)} \cdot n \cdot \mathrm{poly}(d \log(n\Delta))$. □

3.1 Structural Properties of $\boldsymbol{r}^{(\beta)}$

Throughout this and the following subsections, we fix a metric space (P, dist). For every $\beta > 0$, let $\boldsymbol{r}^{(\beta)} = (r_p^{(\beta)})_{p \in P}$ denote the vector computed by MP_β (in Line 1 of Algorithm 1) on (P, dist).

In this subsection, we present several structural properties of the vector $\boldsymbol{r}^{(\beta)} = (r_p^{(\beta)})_{p \in P}$:

- Lemma 3.5 bounds the number of points in every small-radius ball centered at each $p \in P$.
- Lemmas 3.6 to 3.8 derive a set of smoothness properties for the $r_p^{(\beta)}$ values.
- Lemma 3.9 demonstrates a local density property of the $r_p^{(\beta)}$ values, providing lower bounds on the sum of values over the neighborhood of each point (with a sufficiently large radius).

These properties play a crucial role in the analyses later in Sections 3.2 and 3.3.

Additional Notations: Balls and Rings. Before proceeding, we introduce some notations to better characterize the properties of $r_p^{(\beta)}$. For every point $p \in P$ and any radius $r \geq 0$, we define the (β-scaled power-z) ball $B_p^{(\beta)}(r)$ centered at p with radius r, and the (β-scaled power-z) ring $R_p^{(\beta)}(r)$ centered at p with inner radius r and outer radius $2r$, as follows.

$$B_p^{(\beta)}(r) \;:=\; \big\{q \in P :\; \beta \cdot \mathrm{dist}^z(p,q) \leq r\big\}. \tag{10}$$

$$R_p^{(\beta)}(r) \;:=\; B_p^{(\beta)}(2r) \setminus B_p^{(\beta)}(r) \;=\; \big\{q \in P :\; r < \beta \cdot \mathrm{dist}^z(p,q) \leq 2r\big\} \tag{11}$$

It is easy to verify that the defining equation for $r_p^{(\beta)}$ in (8) is equivalent to

$$\sum_{q \in B_p^{(\beta)}(r_p^{(\beta)})} \big(r_p^{(\beta)} - \beta \cdot \mathrm{dist}^z(p,q)\big) \;=\; \lambda. \tag{12}$$

Size Bounds for Balls. The following Lemma 3.5 provides an upper bound on the size of the small-radius ball $B_p^{(\beta)}(r)$. This lemma is adapted from [CGJ+24, Claim 4.4], and we include its proof for completeness.

Lemma 3.5. $|B_p^{(\beta)}(r)| \leq \frac{\lambda}{r_p^{(\beta)} - r}$, *for any scale factor* $\beta > 0$, *point* $p \in P$ *and radius* $0 < r < r_p^{(\beta)}$.

Proof. By definition, each point $q \in B_p^{(\beta)}(r)$ satisfies $r_p^{(\beta)} - \beta \cdot \mathrm{dist}^z(p,q) \geq r_p^{(\beta)} - r > 0$. This together with (8) gives $\lambda = \sum_{q \in P}[r_p^{(\beta)} - \beta \cdot \mathrm{dist}^z(p,q)]^+ \geq |B_p^{(\beta)}(r)| \cdot (r_p^{(\beta)} - r)$. The lemma then follows directly. □

Smoothness of $r_p^{(\beta)}$. We establish a set of *smoothness* properties for $r_p^{(\beta)}$ (Lemmas 3.6 to 3.8). First, Lemma 3.6 shows that $r_p^{(\beta)}$ is smooth with respect to β, in the sense that when β is scaled by a factor $c \geq 1$, the value of $r_p^{(\beta)}$ changes by a factor of at most c.

Lemma 3.6. $(\beta'/\beta) \cdot \boldsymbol{r}^{(\beta)} \leq \boldsymbol{r}^{(\beta')} \leq \boldsymbol{r}^{(\beta)}$, *for any scale factors* $\beta \geq \beta' > 0$.

Proof. It suffices to show $\frac{\beta'}{\beta} \cdot r_p^{(\beta)} \leq r_p^{(\beta')} \leq r_p^{(\beta)}$ for each point $p \in P$. Consider the function $\Phi_p^{(\beta')}(r) := \sum_{t \in P}[r - \beta' \cdot \mathrm{dist}^z(p,t)]^+$, which is increasing in r and satisfies $\Phi_p^{(\beta')}(r_p^{(\beta')}) = \lambda$ by (8); similarly for the other function $\Phi_p^{(\beta)}(r)$. (Note that $\lambda > 0$ is nonzero.) Then, we can verify the lemma as follows:

- $\Phi_p^{(\beta')}(\frac{\beta'}{\beta} \cdot r_p^{(\beta)}) = \frac{\beta'}{\beta} \cdot \Phi_p^{(\beta)}(r_p^{(\beta)}) = \frac{\beta'}{\beta} \cdot \lambda \leq \lambda = \Phi_p^{(\beta')}(r_p^{(\beta')}) \implies \frac{\beta'}{\beta} \cdot r_p^{(\beta)} \leq r_p^{(\beta')}$.
- $\Phi_p^{(\beta')}(r_p^{(\beta)}) \geq \Phi_p^{(\beta)}(r_p^{(\beta)}) = \lambda = \Phi_p^{(\beta')}(r_p^{(\beta')}) \implies r_p^{(\beta')} \leq r_p^{(\beta)}$.

This finishes the proof. □

Second, Lemma 3.7 shows that $r_p^{(\beta)}$ is smooth with respect to the location of $p \in P$, i.e., nearby points have comparable values of $r_p^{(\beta)}$. A similar result can be found in [CGJ+24, Claim 4.4].

Lemma 3.7. *For any scale factor* $\beta > 0$ *and any two points* $p, q \in P$,

$$r_p^{(\beta)} \;\leq\; r_q^{(2^{z-1}\beta)} + 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(p,q) \;\leq\; 2^{z-1} \cdot \big(r_q^{(\beta)} + \beta \cdot \mathrm{dist}^z(p,q)\big).$$

Proof. The second inequality, or equivalently $r_q^{(2^{z-1}\beta)} \le 2^{z-1} \cdot r_q^{(\beta)}$, follows directly from Lemma 3.6. Below, we prove the first inequality. By the generalized triangle inequality (Fact 2.1), we have

$$\begin{aligned}
\Phi_p^{(\beta)}\big(r_q^{(2^{z-1}\beta)} + 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(p,q)\big) &= \sum_{t \in P} \big[r_q^{(2^{z-1}\beta)} + 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(p,q) - \beta \cdot \mathrm{dist}^z(p,t)\big]^+ \\
&\triangleright \text{ Fact 2.1} \quad \ge \sum_{t \in P} \big[r_q^{(2^{z-1}\beta)} - 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(q,t)\big]^+ \\
&= \Phi_q^{(2^{z-1}\beta)}(r_q^{(2^{z-1}\beta)}) \;=\; \lambda \;=\; \Phi_p^{(\beta)}(r_p^{(\beta)}).
\end{aligned}$$

This implies $r_p^{(\beta)} \le r_q^{(2^{z-1}\beta)} + 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(p,q)$, since the function $\Phi_p^{(\beta)}$ is increasing (and $\lambda > 0$). □

Third, we establish Lemma 3.8, which is similar to Lemma 3.7 but, via a more careful argument, shows that $r_p^{(\beta)}$ and $r_q^{(\beta)}$ only differ by an additive term. This difference depends on the (power-z) distance $\mathrm{dist}^z(p,q)$ as well as the neighborhood structure of q (or p, by symmetry).

Lemma 3.8. *For any scale factor $\beta > 0$ and any two points $p, q \in P$,*

$$\begin{aligned}
r_p^{(\beta)} \;\le\; r_q^{(\beta)} + 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(p,q) + \frac{1}{|B_q^{(\beta)}(r_q^{(\beta)}/2)|} \cdot \Big(&\sum_{t \in B_q^{(\beta)}(r_q^{(\beta)}/2)} (2^{z-1} - 1) \cdot \beta \cdot \mathrm{dist}^z(t,q) \\
&+ \sum_{t \in R_q^{(\beta)}(r_q^{(\beta)}/2)} \big(r_q^{(\beta)} - \beta \cdot \mathrm{dist}^z(t,q)\big)\Big).
\end{aligned}$$

Proof. Let

$$\gamma := \frac{1}{|B_q^{(\beta)}(r_q^{(\beta)}/2)|} \cdot \Big(\sum_{t \in B_q^{(\beta)}(r_q^{(\beta)}/2)} (2^{z-1} - 1) \cdot \beta \cdot \mathrm{dist}^z(t,q) + \sum_{t \in R_q^{(\beta)}(r_q^{(\beta)}/2)} \big(r_q^{(\beta)} - \beta \cdot \mathrm{dist}^z(t,q)\big)\Big).$$

By the generalized triangle inequality (Fact 2.1), we have

$$\begin{aligned}
\Phi_p^{(\beta)}(r_q^{(\beta)} + 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(p,q) + \gamma) &= \sum_{t \in P} \big[r_q^{(\beta)} + 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(p,q) + \gamma - \beta \cdot \mathrm{dist}^z(t,p)\big]^+ \\
&\triangleright \text{ Fact 2.1} \quad \ge \sum_{t \in P} \big[r_q^{(\beta)} + \gamma - 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(t,q)\big]^+ \\
&\ge \sum_{t \in B_q^{(\beta)}(r_q^{(\beta)}/2)} \big(r_q^{(\beta)} + \gamma - 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(t,q)\big) \\
&= \sum_{t \in B_q^{(\beta)}(r_q^{(\beta)}/2)} \big(r_q^{(\beta)} - \beta \cdot \mathrm{dist}^z(t,q)\big) \\
&\quad + \sum_{t \in R_q^{(\beta)}(r_q^{(\beta)}/2)} \big(r_q^{(\beta)} - \beta \cdot \mathrm{dist}^z(t,q)\big) \\
\triangleright\, B_q^{(\beta)}(r_q^{(\beta)}) = B_q^{(\beta)}(r_q^{(\beta)}/2) \sqcup R_q^{(\beta)}(r_q^{(\beta)}/2) \quad &= \sum_{t \in B_q^{(\beta)}(r_q^{(\beta)})} \big(r_q^{(\beta)} - \beta \cdot \mathrm{dist}^z(t,q)\big) \\
&= \lambda \;=\; \Phi_p^{(\beta)}(r_p^{(\beta)}).
\end{aligned}$$

This implies $r_p^{(\beta)} \le r_q^{(\beta)} + 2^{z-1} \cdot \beta \cdot \mathrm{dist}^z(p,q) + \gamma$, since the function $\Phi_p^{(\beta)}$ is increasing (and $\lambda > 0$). □

Local Density. Finally, we establish Lemma 3.9, which shows that for each point $p \in P$, the sum of $r_q^{(\beta)}$ over all points q within distance $O(r_p^{(\beta)})$ (sufficiently large) is at least λ. This property is straightforward in the $z = 1$ case by combining the definition of $r_p^{(\beta)}$ (in (8)) with Lemma 3.7. However, this simple argument cannot extend to all $z \geq 1$ cases, because of the 2^{z-1} factor in the statement of Lemma 3.7, and a more involved argument is needed for the extension.

Lemma 3.9. *Let $\gamma^* := 2^{2z^2+z} = 2^{O(z^2)}$. For any scale factor $\beta > 0$ and every point $p \in P$,*

$$\sum_{q \in B_p^{(\beta)}(\gamma^* \cdot r_p^{(\beta)})} r_q^{(\beta)} \;\geq\; \lambda.$$

Proof. We show Lemma 3.9 by induction; specifically, Claim 3.10 verifies the base case, and Claim 3.11 verifies the induction step.

Claim 3.10 (Base Case for Lemma 3.9)**.** *For any scale factor $\beta > 0$ and every point $p \in P$,*

$$\sum_{q \in B_p^{(\beta)}(2^z \cdot r_p^{(\beta)})} r_q^{(\beta)} \;\geq\; (1/2^z) \cdot \lambda.$$

Proof. To ease the presentation, we reindex all points $P = \{p_1, \ldots, p_n\}$ such that $r_{p_1}^{(\beta)} \leq \cdots \leq r_{p_n}^{(\beta)}$; we will prove Claim 3.10 by induction on this ordering of points.

Base Case: The first point p_1 satisfies $r_{p_1}^{(\beta)} \leq r_q^{(\beta)}$, $\forall q \in P$, which together with (12) gives

$$\sum_{q \in B_{p_1}^{(\beta)}(2^z \cdot r_{p_1}^{(\beta)})} r_q^{(\beta)} \;\geq\; \sum_{q \in B_{p_1}^{(\beta)}(r_{p_1}^{(\beta)})} \left(r_{p_1}^{(\beta)} - \beta \cdot \mathrm{dist}^z(p_1, q)\right) \;=\; \lambda \;\geq\; (1/2^z) \cdot \lambda.$$

Induction Step: Fix an index $i \in [2 : n]$. We assume, as an induction hypothesis, that Claim 3.10 holds for points $p_1, \ldots, p_{i-1}$. We next establish Claim 3.10 for p_i. Below, we rewrite $p = p_i$ for notational brevity and conduct a case analysis.

Case 1: for every point $q \in B_p^{(\beta)}(r_p^{(\beta)})$, it holds that $r_q^{(\beta)} \geq \frac{1}{2^z} \cdot (r_p^{(\beta)} - \beta \cdot \mathrm{dist}^z(p, q))$. In this case, it follows directly that

$$\sum_{q \in B_p^{(\beta)}(r_p^{(\beta)})} r_q^{(\beta)} \;\geq\; \sum_{q \in B_p^{(\beta)}(r_p^{(\beta)})} (1/2^z) \cdot \left(r_p^{(\beta)} - \beta \cdot \mathrm{dist}^z(p, q)\right) \;=\; (1/2^z) \cdot \lambda,$$

which completes our analysis of Case 1.

Case 2: there exists some point $t \in B_p^{(\beta)}(r_p^{(\beta)})$ such that $r_t^{(\beta)} < \frac{1}{2^z} \cdot (r_p^{(\beta)} - \beta \cdot \mathrm{dist}^z(p, t))$. In this case, we make the following observations:

- $t = p_j$ for some index $j \in [i-1]$, since $r_t^{(\beta)} < \frac{1}{2^z} \cdot (r_p^{(\beta)} - \beta \cdot \mathrm{dist}^z(p, t)) < r_p^{(\beta)}$.
- $B_t^{(\beta)}(2^z \cdot r_t^{(\beta)}) \subseteq B_p^{(\beta)}(2^z \cdot r_p^{(\beta)})$. To see this, consider a specific point $q \in B_t^{(\beta)}(2^z \cdot r_t^{(\beta)})$; we have $\beta \cdot \mathrm{dist}^z(q, t) \leq 2^z \cdot r_t^{(\beta)}$. Moreover, by the premise of Case 2, the point t satisfies $\beta \cdot \mathrm{dist}^z(p, t) \leq r_p^{(\beta)}$ and $r_t^{(\beta)} < \frac{1}{2^z} \cdot (r_p^{(\beta)} - \beta \cdot \mathrm{dist}^z(p, t)) \leq \frac{1}{2^z} \cdot r_p^{(\beta)}$. By the generalized triangle inequality (Fact 2.1), we have

$$\begin{aligned} \beta \cdot \mathrm{dist}^z(q, p) \;&\leq\; \beta \cdot 2^{z-1} \cdot \left(\mathrm{dist}^z(q, t) + \mathrm{dist}^z(p, t)\right) \\ &\leq\; 2^{z-1} \cdot \left(2^z \cdot r_t^{(\beta)} + r_p^{(\beta)}\right) \end{aligned}$$

$$\begin{aligned} &\leq 2^{z-1} \cdot \left(2^z \cdot \tfrac{1}{2^z} + 1\right) \cdot r_p^{(\beta)} \\ &= 2^z \cdot r_p^{(\beta)}. \end{aligned}$$

This implies $q \in B_p^{(\beta)}(2^z \cdot r_p^{(\beta)})$ for every $q \in B_t^{(\beta)}(2^z \cdot r_t^{(\beta)})$, i.e., $B_t^{(\beta)}(2^z \cdot r_t^{(\beta)}) \subseteq B_p^{(\beta)}(2^z \cdot r_p^{(\beta)})$. Combining the above observations and the induction hypothesis completes our analysis in Case 2, as follows:

$$\sum_{q \in B_p^{(\beta)}(2^z \cdot r_p^{(\beta)})} r_q^{(\beta)} \;\geq\; \sum_{q \in B_t^{(\beta)}(2^z \cdot r_t^{(\beta)})} r_q^{(\beta)} \;\geq\; (1/2^z) \cdot \lambda.$$

Combining both cases completes the induction step, which immediately implies Claim 3.10. □

Claim 3.11 (Induction Step for Lemma 3.9)**.** *Given parameters $\alpha \in (0,1)$ and $\gamma \geq 1$, if it holds that*

$$\sum_{q \in B_p^{(\beta)}(\gamma \cdot r_p^{(\beta)})} r_q^{(\beta)} \;\geq\; \alpha \cdot \lambda, \qquad \forall \beta > 0,\ \forall p \in P, \tag{13}$$

then it also holds that

$$\sum_{q \in B_p^{(\beta)}(2^{2z} \cdot \gamma \cdot r_p^{(\beta)})} r_q^{(\beta)} \;\geq\; \min\{2\alpha, 1\} \cdot \lambda, \qquad \forall \beta > 0,\ \forall p \in P. \tag{14}$$

Proof. We fix a scale factor $\beta > 0$ and a point $p \in P$. Let $q^* := \operatorname{argmin}_{q \in B_p^{(\beta)}(2^z \cdot \gamma \cdot r_p^{(\beta)})} r_q^{(\beta)}$; we have $\beta \cdot \operatorname{dist}^z(q^*, p) \leq 2^z \cdot \gamma \cdot r_p^{(\beta)}$ and $r_{q^*}^{(\beta)} \leq r_p^{(\beta)}$. We conduct a case analysis.

Case 1: for every point $t \in B_{q^*}^{(\beta)}(r_{q^*}^{(\beta)})$, it holds that $r_t^{(\beta)} \geq r_{q^*}^{(\beta)}$. In this case, we observe that $B_{q^*}^{(\beta)}(r_{q^*}^{(\beta)}) \subseteq B_p^{(\beta)}(2^{2z} \cdot \gamma \cdot r_p^{(\beta)})$. To see this, consider a specific point $t \in B_{q^*}^{(\beta)}(r_{q^*}^{(\beta)})$; we have $\beta \cdot \operatorname{dist}^z(t, q^*) \leq r_{q^*}^{(\beta)}$. By the generalized triangle inequality (Fact 2.1), we have

$$\begin{aligned} \beta \cdot \operatorname{dist}^z(t,p) &\leq \beta \cdot 2^{z-1} \cdot \left(\operatorname{dist}^z(t,q^*) + \operatorname{dist}^z(p,q^*)\right) \\ &\leq 2^{z-1} \cdot \left(r_{q^*}^{(\beta)} + 2^z \cdot \gamma \cdot r_p^{(\beta)}\right) \\ \triangleright\ r_{q^*}^{(\beta)} \leq r_p^{(\beta)} \qquad &\leq 2^{2z} \cdot \gamma \cdot r_p^{(\beta)}. \end{aligned}$$

This implies $t \in B_p^{(\beta)}(2^{2z} \cdot \gamma \cdot r_p^{(\beta)})$ for every $t \in B_{q^*}^{(\beta)}(r_{q^*}^{(\beta)})$, i.e., $B_{q^*}^{(\beta)}(r_{q^*}^{(\beta)}) \subseteq B_p^{(\beta)}(2^{2z} \cdot \gamma \cdot r_p^{(\beta)})$. Combining this observation with the premise of Case 1 and (12) completes our analysis in Case 1, as follows:

$$\sum_{t \in B_p^{(\beta)}(2^{2z} \cdot \gamma \cdot r_p^{(\beta)})} r_t^{(\beta)} \;\geq\; \sum_{t \in B_{q^*}^{(\beta)}(r_{q^*}^{(\beta)})} \left(r_{q^*}^{(\beta)} - \beta \cdot \operatorname{dist}^z(q^*, t)\right) \;=\; \lambda.$$

Case 2: there exists some point $t \in B_{q^*}^{(\beta)}(r_{q^*}^{(\beta)})$ such that $r_t^{(\beta)} < r_{q^*}^{(\beta)} \leq r_p^{(\beta)}$. In this case, we make the following observations:

- $B_t^{(\beta)}(\gamma \cdot r_t^{(\beta)}) \cap B_p^{(\beta)}(\gamma \cdot r_p^{(\beta)}) = \emptyset$. To see this, assume for contradiction that there exists some point $q \in B_t^{(\beta)}(\gamma \cdot r_t^{(\beta)}) \cap B_p^{(\beta)}(\gamma \cdot r_p^{(\beta)})$. It follows that $\beta \cdot \operatorname{dist}^z(p,t) \leq \beta \cdot 2^{z-1} \cdot (\operatorname{dist}^z(p,q) + \operatorname{dist}^z(t,q)) \leq 2^{z-1} \cdot \gamma \cdot (r_t^{(\beta)} + r_p^{(\beta)}) \leq 2^z \cdot \gamma \cdot r_p^{(\beta)} \implies t \in B_p^{(\beta)}(2^z \cdot \gamma \cdot r_p^{(\beta)})$; given the premise of Case 2 that $r_t^{(\beta)} < r_{q^*}^{(\beta)}$, this contradicts the choice of $q^* = \operatorname{argmin}_{q \in B_p^{(\beta)}(2^z \cdot \gamma \cdot r_p^{(\beta)})} r_q^{(\beta)}$.

- $B_t^{(\beta)}(\gamma \cdot r_t^{(\beta)}) \subseteq B_p^{(\beta)}(2^{2z} \cdot \gamma \cdot r_p^{(\beta)})$. To see this, consider a specific point $q \in B_t^{(\beta)}(\gamma \cdot r_t^{(\beta)})$; we have $\beta \cdot \mathrm{dist}^z(t,q) \le \gamma \cdot r_t^{(\beta)}$. By applying the generalized triangle inequality (Fact 2.1) twice, we have

$$\begin{aligned} \beta \cdot \mathrm{dist}^z(p,q) &\le \beta \cdot 2^{z-1} \cdot \big(\mathrm{dist}^z(p,q^*) + 2^{z-1} \cdot \big(\mathrm{dist}^z(q^*,t) + \mathrm{dist}^z(t,q)\big)\big) \\ &\le 2^{z-1} \cdot \big(2^z \cdot \gamma \cdot r_p^{(\beta)} + 2^{z-1} \cdot \big(r_{q^*}^{(\beta)} + \gamma \cdot r_t^{(\beta)}\big)\big) \\ \triangleright\ r_t^{(\beta)} < r_{q^*}^{(\beta)} \le r_p^{(\beta)} \qquad &\le 2^{2z} \cdot \gamma \cdot r_p^{(\beta)}. \end{aligned}$$

  This implies $q \in B_p^{(\beta)}(2^{2z} \cdot \gamma \cdot r_p^{(\beta)})$ for all $q \in B_t^{(\beta)}(\gamma \cdot r_t^{(\beta)})$, i.e., $B_t^{(\beta)}(\gamma \cdot r_t^{(\beta)}) \subseteq B_p^{(\beta)}(2^{2z} \cdot \gamma \cdot r_p^{(\beta)})$.

Combining the above observations and the premise of Claim 3.11 completes our analysis in Case 2, as follows:

$$\sum_{q \in B_p^{(\beta)}(2^{2z} \cdot \gamma \cdot r_p^{(\beta)})} r_q^{(\beta)} \ \ge \sum_{q \in B_t^{(\beta)}(\gamma \cdot r_t^{(\beta)})} r_q^{(\beta)} + \sum_{q \in B_p^{(\beta)}(\gamma \cdot r_p^{(\beta)})} r_q^{(\beta)} \ \ge\ 2\alpha \cdot \lambda.$$

Combining both cases completes the proof of Claim 3.11. □

Let us return to the proof of Lemma 3.9; applying Claims 3.10 and 3.11 gives, for every integer $i \ge 0$,

$$\sum_{q \in B_p^{(\beta)}(2^{(2i+1)z} \cdot r_p^{(\beta)})} r_q^{(\beta)} \ \ge\ \min\{2^{i-z}, 1\} \cdot \lambda.$$

By setting $i = z$, and thus $\gamma^* = 2^{2z^2+z}$, Lemma 3.9 follows. □

3.2 Analysis of MP_β without Distance Distortions

In this subsection, we analyze MP_β (Algorithm 1) on input a metric space (P, dist) without any distance distortion, and show that it yields a fractional LMP approximation for Power-z Facility Location. The result is formally stated in the following lemma, which can be viewed as a generalization of [BCG+24, Theorem 13.1] from the $z = 1$ case to any $z \ge 1$.

Lemma 3.12. *Let $\beta^* := 2^{-z-1}$ and let (P, dist) be a metric space. For any scale factor $\beta \in (0, \beta^*]$, the algorithm MP_β (Algorithm 1) on input (P, dist) returns a fractional solution $\boldsymbol{y}^{(\beta)}$ to Power-z Facility Location with an LMP $2^{O(z^2)}\beta^{-1}$-approximation guarantee, i.e.,*

$$\mathrm{cost}_z(\boldsymbol{y}^{(\beta)}) \ \le\ 2^{O(z^2)} \cdot \beta^{-1} \cdot (\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda) - \lambda \cdot \|\boldsymbol{y}^{(\beta)}\|_1).$$

Before proving Lemma 3.12, we present the following technical claim, which relates the clustering cost $\mathrm{cost}_z(\boldsymbol{y})$ of $\boldsymbol{y}$ to the assignment cost $\sigma_z(\boldsymbol{x})$ for some assignment $\boldsymbol{x}$ that may be infeasible with respect to $\boldsymbol{y}$.

Claim 3.13. *Suppose that a fractional solution $\boldsymbol{y} \in \mathbb{R}_{\ge 0}^P$, a fractional assignment $\boldsymbol{x} \in \mathbb{R}_{\ge 0}^{P \times P}$, and a vector $\boldsymbol{\mu} \in \mathbb{R}_{\ge 0}^P$ satisfy the following conditions:*

1. *Each point $p \in P$ satisfies $\sum_{q \in P : \mathrm{dist}^z(p,q) \le \mu_p} y_q \ge 1$.*
2. *Each point $p \in P$ satisfies $\sum_{q \in P} x_{p,q} = 1$.*
3. *Each pair $p, q \in P$ with $x_{p,q} > 0$ satisfies $\mathrm{dist}^z(p,q) \le \mu_p$.*

Then, it holds that

$$\mathrm{cost}_z(\boldsymbol{y}) \;\le\; \sigma_z(\boldsymbol{x}) + \sum_{p,q\in P} [x_{p,q} - y_q]^+ \cdot \mu_p.$$

Proof. We introduce two auxiliary assignments $\underline{\boldsymbol{x}} = (\underline{x}_{p,q})_{p,q\in P}$ and $\overline{\boldsymbol{x}} = (\overline{x}_{p,q})_{p,q\in P}$:

- $\underline{\boldsymbol{x}}$ is defined as $\underline{x}_{p,q} := \min\{x_{p,q},\ y_q\}$, $\forall p, q \in P$.
- $\overline{\boldsymbol{x}}$ is defined as $\overline{x}_{p,q} := y_q \cdot \mathbb{I}[\mathrm{dist}^z(p,q) \le \mu_p]$, $\forall p, q \in P$, where $\mathbb{I}[\cdot]$ is the indicator function.

The premises of Claim 3.13 guarantee $\underline{\boldsymbol{x}} \le \overline{\boldsymbol{x}}$ and for every $p \in P$, $\sum_{q\in P} \underline{x}_{p,q} \le \sum_{q\in P} x_{p,q} = 1 \le \sum_{q\in P:\mathrm{dist}^z(p,q)\le\mu_p} y_q = \sum_{q\in P} \overline{x}_{p,q}$. This further implies the existence of an assignment $\widetilde{\boldsymbol{x}} = (\widetilde{x}_{p,q})_{p,q\in P}$ with $\underline{\boldsymbol{x}} \le \widetilde{\boldsymbol{x}} \le \overline{\boldsymbol{x}}$ and $\sum_{q\in P} \widetilde{x}_{p,q} = 1$, $\forall p \in P$. Note that $\widetilde{x}_{p,q} \le \overline{x}_{p,q} \le y_q$, $\forall p, q \in P$. Hence, this is a feasible assignment $\widetilde{\boldsymbol{x}} \sim \boldsymbol{y}$, and we can deduce that

$$\begin{aligned}
\mathrm{cost}_z(\boldsymbol{y}) \;\le\; \sigma_z(\widetilde{\boldsymbol{x}}) \;&=\; \sigma_z(\boldsymbol{x}) + \sum_{p,q\in P} (\widetilde{x}_{p,q} - x_{p,q}) \cdot \mathrm{dist}^z(p,q) \\
&\le\; \sigma_z(\boldsymbol{x}) + \sum_{p,q\in P} [\widetilde{x}_{p,q} - x_{p,q}]^+ \cdot \mathrm{dist}^z(p,q) \\
&\le\; \sigma_z(\boldsymbol{x}) + \sum_{p,q\in P} [\widetilde{x}_{p,q} - x_{p,q}]^+ \cdot \mu_p \\
&=\; \sigma_z(\boldsymbol{x}) + \sum_{p,q\in P} [x_{p,q} - \widetilde{x}_{p,q}]^+ \cdot \mu_p \\
\triangleright\ \widetilde{x}_{p,q} \ge \underline{x}_{p,q} = \min\{x_{p,q},\ y_q\} \qquad &\le\; \sigma_z(\boldsymbol{x}) + \sum_{p,q\in P} [x_{p,q} - \min\{x_{p,q},\ y_q\}]^+ \cdot \mu_p \\
&=\; \sigma_z(\boldsymbol{x}) + \sum_{p,q\in P} [x_{p,q} - y_q]^+ \cdot \mu_p.
\end{aligned}$$

Here, the third inequality holds since $[\widetilde{x}_{p,q} - x_{p,q}]^+ > 0$ only if $0 < \widetilde{x}_{p,q} \le \overline{x}_{p,q}$, and then only if $\mathrm{dist}^z(p,q) \le \mu_p$. The second equality holds since $\sum_{q\in P} \widetilde{x}_{p,q} = \sum_{q\in P} x_{p,q} = 1 \implies \sum_{q\in P} [\widetilde{x}_{p,q} - x_{p,q}]^+ = \sum_{q\in P} [x_{p,q} - \widetilde{x}_{p,q}]^+$, $\forall p \in P$. This completes the proof of Claim 3.13. □

We are now ready to prove Lemma 3.12.

Proof of Lemma 3.12. Let $\boldsymbol{r}^{(\beta)} = (r_p^{(\beta)})_{p\in P}$ denote the vector computed by MP_β (in Line 1 of Algorithm 1); then, the output of MP_β is $\boldsymbol{y}^{(\beta)} = \boldsymbol{r}^{(\beta)}/\lambda$. The proof follows a primal-dual approach. To begin, we present the primal and dual LPs for Power-z Facility Location, denoted by $\mathrm{LP}^{\mathrm{fl}}_\lambda$ and $\mathrm{DP}^{\mathrm{fl}}_\lambda$, respectively.

$$\begin{aligned}
\mathrm{LP}^{\mathrm{fl}}_\lambda: \quad \min \quad & \lambda \cdot \sum_{p\in P} y_p + \sum_{p,q\in P} x_{p,q} \cdot \mathrm{dist}^z(p,q) && && \text{(L)} \\
\text{s.t.} \quad & \sum_{q\in P} x_{p,q} \ge 1 && \forall p \in P && \text{(L1)} \\
& x_{p,q} \le y_q && \forall p, q \in P && \text{(L2)} \\
& x_{p,q}, y_p \ge 0 && \forall p, q \in P && \text{(L3)} \\
\mathrm{DP}^{\mathrm{fl}}_\lambda: \quad \max \quad & \sum_{p\in P} v_p && && \text{(D)}
\end{aligned}$$

$$\text{s.t.} \quad \sum_{p\in P} \left[v_p - \text{dist}^z(p,q)\right]^+ \le \lambda, \qquad \forall q \in P \tag{D1}$$

$$v_p \ge 0, \qquad \forall p \in P \tag{D2}$$

The optimal primal value is exactly $\widehat{\text{OPT}}_z^{\text{fl}}(\lambda) = \min_{\boldsymbol{y}\in\mathbb{R}^P_{\ge 0}:\|\boldsymbol{y}\|_1\ge 1} \lambda\cdot\|\boldsymbol{y}\|_1 + \text{cost}_z(\boldsymbol{y})$, since Constraints (L1) to (L2) are identical to Constraints (C1) to (C2) (see Section 2). Further, by weak duality, every feasible dual solution $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_1 = \sum_{p\in P} v_p \le \widehat{\text{OPT}}_z^{\text{fl}}(\lambda)$.

We define a concrete fractional assignment $\boldsymbol{x}^{(\beta)} = (x^{(\beta)}_{p,q})_{p,q\in P}$ as follows:

$$x^{(\beta)}_{p,q} \;:=\; (1/\lambda)\cdot\left[r_p^{(\beta)} - \beta\cdot\text{dist}^z(p,q)\right]^+, \qquad \forall p,q\in P. \tag{15}$$

To prove Lemma 3.12, we use the assignment cost $\sigma_z(\boldsymbol{x}^{(\beta)})$ as an intermediate bridge. Specifically, we show in Lemma 3.14 that the cost $\sigma_z(\boldsymbol{x}^{(\beta)})$ satisfies the desired LMP property, and then show in Lemma 3.16 that $\text{cost}_z(\boldsymbol{y}^{(\beta)})$ is not much larger than $\sigma_z(\boldsymbol{x}^{(\beta)})$. We note that the proof of Lemma 3.14 follows almost the same approach as in [BCG+24], except that we use our new smoothness property in Lemma 3.7, rather than the property they use which only holds for the $z = 1$ case.

Lemma 3.14. $\sigma_z(\boldsymbol{x}^{(\beta)}) \le \beta^{-1}\cdot(\widehat{\text{OPT}}_z^{\text{fl}}(\lambda) - \lambda\cdot\|\boldsymbol{y}^{(\beta)}\|_1)$.

Proof. We establish this lemma by analyzing a concrete *dual solution* $\boldsymbol{v}^{(\beta)} := \boldsymbol{r}^{(2\beta)}$. We first verify that this dual solution is feasible; by weak duality, this means $\|\boldsymbol{v}^{(\beta)}\|_1 \le \widehat{\text{OPT}}_z^{\text{fl}}(\lambda)$. Constraint (D2) trivially holds since $\boldsymbol{r}^{(2\beta)} \in \mathbb{R}^P_{\ge 0}$, and we can verify Constraint (D1) via Lemma 3.7:

$$\begin{aligned}
\sum_{p\in P}\left[r_p^{(2\beta)} - \text{dist}^z(p,q)\right]^+ &\le \sum_{p\in P}\left[r_q^{(2^z\cdot\beta)} + 2^z\cdot\beta\cdot\text{dist}^z(p,q) - \text{dist}^z(p,q)\right]^+ \\
\triangleright\ \beta\le\beta^* = \tfrac{1}{2^{z+1}} \quad &\le \sum_{p\in P}\left[r_q^{(2^z\cdot\beta)} - 2^z\cdot\beta\cdot\text{dist}^z(p,q)\right]^+ \\
\triangleright\ (8) \quad &= \lambda.
\end{aligned}$$

Moreover, the following Claim 3.15 essentially restates [BCG+24, Lemma 13.6] under our notation.

Claim 3.15 ([BCG+24, Lemma 13.6])**.** *For each point $p\in P$,*

$$r_p^{(\beta)} + \sum_{q\in P}(1/\lambda)\cdot\left[r_p^{(\beta)} - \beta\cdot\text{dist}^z(p,q)\right]^+\cdot\beta\cdot\text{dist}^z(p,q) \;\le\; r_p^{(2\beta)}.$$

Proof. The proof is almost identical to that of [BCG+24, Lemma 13.6], although they only consider the special case with $z = 1$ and $\beta = 1/4$. Specifically, rescaling all distances in their original proof directly gives the extension to $\beta\in(0,1)$. Also, replacing their distance function d with dist^z directly gives the extension to $z\ge 1$; in this regard, we emphasize that their original proof does not require the triangle inequality, so all the arguments carry over. □

By combining everything, we can deduce Lemma 3.14 as follows:

$$\begin{aligned}
\sigma_z(\boldsymbol{x}^{(\beta)}) &= \sum_{p\in P}\sum_{q\in P} x^{(\beta)}_{p,q}\cdot\text{dist}^z(p,q) \\
\triangleright\ (15) \quad &= \sum_{p\in P}\sum_{q\in P}(1/\lambda)\cdot\left[r_p^{(\beta)} - \beta\cdot\text{dist}^z(p,q)\right]^+\cdot\text{dist}^z(p,q)
\end{aligned}$$

$$
\begin{aligned}
&\triangleright \text{ Claim 3.15} && \le \beta^{-1} \cdot \sum_{p\in P} \big(r_p^{(2\beta)} - r_p^{(\beta)}\big) \\
&\triangleright\ \boldsymbol{y}^{(\beta)} = (1/\lambda)\cdot \boldsymbol{r}^{(\beta)} && = \beta^{-1} \cdot \big(\|\boldsymbol{r}^{(2\beta)}\|_1 - \lambda \cdot \|\boldsymbol{y}^{(\beta)}\|_1\big) \\
&\triangleright \text{ weak duality} && \le \beta^{-1} \cdot \big(\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda) - \lambda \cdot \|\boldsymbol{y}^{(\beta)}\|_1\big).
\end{aligned}
$$

□

In the $z = 1$ case considered in [BCG+24], the assignment $\boldsymbol{x}^{(\beta)}$ is feasible with respect to $\boldsymbol{y}^{(\beta)}$ (simply by Lemma 3.7). However, this feasibility does not extend to $z \ge 2$, and we provide a counterexample in Example 3.19 at the end of this subsection. To address this issue, we additionally establish the following Lemma 3.16.

Lemma 3.16. $\mathrm{cost}_z(\boldsymbol{y}^{(\beta)}) \le 2^{O(z^2)} \cdot \sigma_z(\boldsymbol{x}^{(\beta)})$.

Proof. We first upper-bound the difference between $\mathrm{cost}_z(\boldsymbol{y}^{(\beta)})$ and $\sigma_z(\boldsymbol{x}^{(\beta)})$ via Claim 3.13. To this end, we verify the three premises of Claim 3.13 for the choices $\boldsymbol{y} := \boldsymbol{y}^{(\beta)} = (1/\lambda)\cdot\boldsymbol{r}^{(\beta)}$, $\boldsymbol{x} := \boldsymbol{x}^{(\beta)}$, and $\boldsymbol{\mu} := \gamma^* \cdot \beta^{-1} \cdot \boldsymbol{r}^{(\beta)}$ (for the constant $\gamma^* = 2^{O(z^2)}$ specified in Lemma 3.9) as follows:

1. For every $p \in P$, by Lemma 3.9, we have $\sum_{q\in P:\mathrm{dist}^z(p,q)\le\mu_p} y_q = \sum_{q\in B_p^{(\beta)}(\gamma^*\cdot r_p^{(\beta)})} r_q^{(\beta)}/\lambda \ge 1$.
2. For every $p \in P$, $\sum_{q\in P} x_{p,q} = \sum_{q\in P} x_{p,q}^{(\beta)} = \sum_{q\in P}(1/\lambda)\cdot[r_p^{(\beta)} - \beta\cdot\mathrm{dist}^z(p,q)]^+ = 1$, where the second step applies (15), and the last step follows from (8).
3. For every $p, q \in P$ with $0 < x_{p,q} = x_{p,q}^{(\beta)} = (1/\lambda)\cdot[r_p^{(\beta)} - \beta\cdot\mathrm{dist}^z(p,q)]^+$, it follows directly that $\mathrm{dist}^z(p,q) \le \beta^{-1}\cdot r_p^{(\beta)} \le \gamma^*\cdot\beta^{-1}\cdot r_p^{(\beta)} = \mu_p$.

Then, by applying Claim 3.13, we can derive

$$
\mathrm{cost}_z(\boldsymbol{y}^{(\beta)}) \;\le\; \sigma_z(\boldsymbol{x}^{(\beta)}) + \gamma^*\cdot\beta^{-1}\cdot\sum_{p,q\in P}[x_{p,q}^{(\beta)} - y_q^{(\beta)}]^+\cdot r_p^{(\beta)}. \tag{16}
$$

We divide all pairs $P\times P = \mathcal{F}\sqcup\mathcal{C}$ into the *"far" pairs* $\mathcal{F}$ and the *"close" pairs* $\mathcal{C}$, as follows:

$$
\begin{aligned}
\mathcal{F} &:= \big\{(p,q)\in P\times P:\ \beta\cdot\mathrm{dist}^z(p,q) > \tfrac{2}{3}\cdot 2^{-z}\cdot r_p^{(\beta)}\big\}, \\
\mathcal{C} &:= \big\{(p,q)\in P\times P:\ \beta\cdot\mathrm{dist}^z(p,q) \le \tfrac{2}{3}\cdot 2^{-z}\cdot r_p^{(\beta)}\big\}.
\end{aligned}
$$

Below, Claims 3.17 and 3.18 investigate these two kinds of pairs separately.

Claim 3.17. $\sum_{(p,q)\in\mathcal{F}}[x_{p,q}^{(\beta)} - y_q^{(\beta)}]^+\cdot r_p^{(\beta)} \le 2^{O(z)}\cdot\beta\cdot\sigma_z(\boldsymbol{x}^{(\beta)})$.

Proof. For every far pair $(p,q)\in\mathcal{F}$, by definition it satisfies that $r_p^{(\beta)} \le \frac{3}{2}\cdot 2^z\cdot\beta\cdot\mathrm{dist}^z(p,q)$. Hence, we have $\sum_{(p,q)\in\mathcal{F}}[x_{p,q}^{(\beta)} - y_q^{(\beta)}]^+\cdot r_p^{(\beta)} \le \sum_{(p,q)\in\mathcal{F}} x_{p,q}^{(\beta)}\cdot(\frac{3}{2}\cdot 2^z\cdot\beta\cdot\mathrm{dist}^z(p,q)) \le 2^{O(z)}\cdot\beta\cdot\sigma_z(\boldsymbol{x}^{(\beta)})$. □

Claim 3.18. $\sum_{(p,q)\in\mathcal{C}}[x_{p,q}^{(\beta)} - y_q^{(\beta)}]^+\cdot r_p^{(\beta)} \le 2^{O(z)}\cdot\beta\cdot\sigma_z(\boldsymbol{x}^{(\beta)})$.

Proof. For every close pair $(p,q)\in\mathcal{C}$, by definition we have $\beta\cdot\mathrm{dist}^z(p,q) \le \frac{2}{3}\cdot 2^{-z}\cdot r_p^{(\beta)} \le \frac{1}{3}\cdot r_p^{(\beta)}$. This in combination with (15) implies $r_p^{(\beta)} \le \frac{3}{2}\cdot\lambda\cdot x_{p,q}^{(\beta)}$, and in combination with (Lemma 3.7) $r_p^{(\beta)} \le 2^{z-1}\cdot(r_q^{(\beta)} + \beta\cdot\mathrm{dist}^z(p,q))$ implies $r_p^{(\beta)} \le 3\cdot 2^{z-2}\cdot r_q^{(\beta)}$ and then $\beta\cdot\mathrm{dist}^z(p,q) \le r_q^{(\beta)}/2$. Therefore, we have $\mathcal{C}\subseteq\{(p,q)\in P\times P:\ p\in B_q^{(\beta)}(r_q^{(\beta)}/2)\}$.

Below, we simply write $B_q^{(\beta)} = B_q^{(\beta)}(r_q^{(\beta)}/2)$ and $R_q^{(\beta)} = R_q^{(\beta)}(r_q^{(\beta)}/2)$ for notational brevity. Following the definitions of $\boldsymbol{y}^{(\beta)}$ and $\boldsymbol{x}^{(\beta)}$ (see Lemma 3.12 and (15)), we can deduce that

$$\begin{aligned}
\sum_{(p,q)\in\mathcal{C}} \big[x_{p,q}^{(\beta)} - y_q^{(\beta)}\big]^+ \cdot r_p^{(\beta)} &= (1/\lambda)\cdot \sum_{(p,q)\in\mathcal{C}} r_p^{(\beta)} \cdot \big[r_p^{(\beta)} - \beta\cdot \mathrm{dist}^z(p,q) - r_q^{(\beta)}\big]^+ \\
&\triangleright \text{ Lemma 3.8} \quad \le (1/\lambda)\cdot \sum_{(p,q)\in\mathcal{C}} r_p^{(\beta)} \cdot (2^{z-1}-1)\cdot\beta\cdot\mathrm{dist}^z(p,q) \\
&\qquad + (1/\lambda)\cdot \sum_{(p,q)\in\mathcal{C}} r_p^{(\beta)} \cdot \frac{1}{|B_q^{(\beta)}|}\cdot \Big(\sum_{t\in B_q^{(\beta)}} (2^{z-1}-1)\cdot\beta\cdot\mathrm{dist}^z(t,q) \\
&\qquad\qquad + \sum_{t\in R_q^{(\beta)}} \big(r_q^{(\beta)} - \beta\cdot\mathrm{dist}^z(t,q)\big)\Big) \\
&\triangleright\ r_p^{(\beta)} \le \tfrac{3}{2}\cdot\lambda\cdot x_{p,q}^{(\beta)} \quad \le (1/\lambda)\cdot \sum_{(p,q)\in\mathcal{C}} \big(\tfrac{3}{2}\cdot\lambda\cdot x_{p,q}^{(\beta)}\big)\cdot(2^{z-1}-1)\cdot\beta\cdot\mathrm{dist}^z(p,q) \\
&\triangleright\ r_p^{(\beta)} \le 3\cdot 2^{z-2}\cdot r_q^{(\beta)} \qquad + (1/\lambda)\cdot \sum_{(p,q)\in\mathcal{C}} \frac{3\cdot 2^{z-2}\cdot r_q^{(\beta)}}{|B_q^{(\beta)}|}\cdot \Big(\sum_{t\in B_q^{(\beta)}} (2^{z-1}-1)\cdot\beta\cdot\mathrm{dist}^z(t,q) \\
&\qquad\qquad + \sum_{t\in R_q^{(\beta)}} \big(r_q^{(\beta)} - \beta\cdot\mathrm{dist}^z(t,q)\big)\Big) \\
&\triangleright\ \mathcal{C}\subseteq\{(p,q):\ p\in B_q^{(\beta)}\} \quad \le (1/\lambda)\cdot \sum_{q\in P}\sum_{p\in B_q^{(\beta)}} \big(\tfrac{3}{2}\cdot\lambda\cdot x_{p,q}^{(\beta)}\big)\cdot(2^{z-1}-1)\cdot\beta\cdot\mathrm{dist}^z(p,q) \\
&\qquad + (1/\lambda)\cdot \sum_{q\in P} \big(3\cdot 2^{z-2}\cdot r_q^{(\beta)}\big)\cdot \Big(\sum_{t\in B_q^{(\beta)}} (2^{z-1}-1)\cdot\beta\cdot\mathrm{dist}^z(t,q) \\
&\qquad\qquad + \sum_{t\in R_q^{(\beta)}} \big(r_q^{(\beta)} - \beta\cdot\mathrm{dist}^z(t,q)\big)\Big) \\
&\le 2^{O(z)}\cdot\beta\cdot \sum_{q\in P}\sum_{p\in B_q^{(\beta)}} x_{p,q}^{(\beta)}\cdot\mathrm{dist}^z(p,q) \\
&\qquad + 2^{O(z)}\cdot(1/\lambda)\cdot\sum_{q\in P}\Big(\beta\cdot\sum_{t\in B_q^{(\beta)}} r_q^{(\beta)}\cdot\mathrm{dist}^z(t,q) \\
&\qquad\qquad + \sum_{t\in R_q^{(\beta)}} r_q^{(\beta)}\cdot\big(r_q^{(\beta)} - \beta\cdot\mathrm{dist}^z(t,q)\big)\Big).
\end{aligned}$$

For every $t\in B_q^{(\beta)}$, $\beta\cdot\mathrm{dist}^z(q,t)\le r_q^{(\beta)}/2$ (see (10)) $\implies r_q^{(\beta)} \le 2\cdot[r_q^{(\beta)} - \beta\cdot\mathrm{dist}^z(q,t)]^+ = 2\lambda\cdot x_{q,t}^{(\beta)}$. For every $t\in R_q^{(\beta)}$, $\beta\cdot\mathrm{dist}^z(t,q)\le r_q^{(\beta)} \le 2\beta\cdot\mathrm{dist}^z(t,q)$ (see (11)) $\implies r_q^{(\beta)} - \beta\cdot\mathrm{dist}^z(q,t) = [r_q^{(\beta)} - \beta\cdot\mathrm{dist}^z(q,t)]^+ = \lambda\cdot x_{q,t}^{(\beta)}$. Plugging these into the above equation completes the proof:

$$\begin{aligned}
\sum_{(p,q)\in\mathcal{C}} \big[x_{p,q}^{(\beta)} - y_q^{(\beta)}\big]^+ \cdot r_p^{(\beta)} &\le 2^{O(z)}\cdot\beta\cdot \sum_{q\in P}\sum_{p\in B_q^{(\beta)}} x_{p,q}^{(\beta)}\cdot\mathrm{dist}^z(p,q) \\
&\qquad + 2^{O(z)}\cdot(1/\lambda)\cdot\sum_{q\in P}\Big(\beta\cdot\sum_{t\in B_q^{(\beta)}} \big(2\lambda\cdot x_{q,t}^{(\beta)}\big)\cdot\mathrm{dist}^z(t,q)
\end{aligned}$$

$$+ \sum_{t \in R_q^{(\beta)}} \left(2\beta \cdot \mathrm{dist}^z(t,q)\right) \cdot \left(\lambda \cdot x_{q,t}^{(\beta)}\right)\Big)$$

$$\leq\ 2^{O(z)} \cdot \beta \cdot \sum_{p,q \in P} x_{p,q}^{(\beta)} \cdot \mathrm{dist}^z(p,q)$$

$$=\ 2^{O(z)} \cdot \beta \cdot \sigma_z(\boldsymbol{x}^{(\beta)}).$$ □

Applying Claims 3.17 and 3.18 to (16) completes the proof of Lemma 3.16, as follows:

$$\mathrm{cost}_z(\boldsymbol{y}^{(\beta)}) \ \leq\ \sigma_z(\boldsymbol{x}^{(\beta)}) + \gamma^* \cdot \beta^{-1} \cdot \sum_{(p,q) \in \mathcal{F} \sqcup \mathcal{C}} \left[x_{p,q}^{(\beta)} - y_q^{(\beta)}\right]^+ \cdot r_p^{(\beta)}$$

$$\leq\ \sigma_z(\boldsymbol{x}^{(\beta)}) + \gamma^* \cdot 2^{O(z)} \cdot \sigma_z(\boldsymbol{x}^{(\beta)})$$

$$\triangleright\ \gamma^* = 2^{O(z^2)} \qquad \leq\ 2^{O(z^2)} \cdot \sigma_z(\boldsymbol{x}^{(\beta)}).$$ □

Lemma 3.12 follows directly by combining Lemmas 3.14 and 3.16. □

Example 3.19 (Counterexample to $\boldsymbol{x}^{(\beta)}$'s Feasibility)**.** Let $\lambda = 1$, $\beta = \beta^* = 2^{-z-1} = 1/8$, and $z = 2$. Consider points on the 1-dimensional Euclidean line: place p at 0, q at $\sqrt{2}$, and $n \geq 1$ points at $\sqrt{5}$. Then $r_p^{(\beta)} = 5/8$ and therefore $x_{p,q}^{(\beta)} = 3/8$. On the other hand, it is easy to see that $r_q^{(\beta)} \leq \frac{1}{8}(\sqrt{5}-\sqrt{2})^2 + O(1/n) \approx 0.0844 + O(1/n)$. For sufficiently large n, we have $y_q^{(\beta)} = r_q^{(\beta)} < x_{p,q}^{(\beta)}$, which violates Constraint (L2). As a result, for this instance, the assignment $\boldsymbol{x}^{(\beta)}$ is not feasible with respect to $\boldsymbol{y}^{(\beta)}$.

3.3 Analysis of MP_β under Distance Distortions (Proof of Lemma 3.1)

In this subsection, we prove Lemma 3.1, which analyzes MP_β (Algorithm 1) under distance distortions, and we restate it below.

Lemma 3.1. *Let $\varepsilon \in (0,1)$ be a constant and suppose that a distorted distance function $\widetilde{\mathrm{dist}}$ satisfies*

$$\mathrm{dist}(p,q) \ \leq\ \widetilde{\mathrm{dist}}(p,q) \ \leq\ \varepsilon^{-2} \cdot \mathrm{dist}(p,q), \qquad \forall p,q \in P. \tag{9}$$

Let $\widetilde{\boldsymbol{y}} \in \mathbb{R}_{\geq 0}^P$ denote the output of MP_β (Algorithm 1) with $\beta := 2^{-z-1} \cdot \varepsilon^{2z}$ on input $(P, \widetilde{\mathrm{dist}})$ (instead of (P, dist)). Then $\widetilde{\boldsymbol{y}}$ is a fractional solution to Power-z Facility Location *with an LMP $2^{O(z^2)}\varepsilon^{-2z}$-approximation guarantee on the original metric space, i.e.,*

$$\mathrm{cost}_z(\widetilde{\boldsymbol{y}}) \ \leq\ 2^{O(z^2)}\varepsilon^{-2z} \cdot \left(\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda) - \lambda \cdot \|\widetilde{\boldsymbol{y}}\|_1\right),$$

where both $\mathrm{cost}_z(\widetilde{\boldsymbol{y}})$ and $\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda)$ are evaluated with respect to the original distance dist*.*

Proof. Let $\beta^* := 2^{-z-1}$; then $\beta = \beta^* \varepsilon^{2z}$. Let $\boldsymbol{y}^{(\beta)} := (1/\lambda) \cdot \boldsymbol{r}^{(\beta)}$ and $\boldsymbol{y}^{(\beta^*)} := (1/\lambda) \cdot \boldsymbol{r}^{(\beta^*)}$, which are the outputs of MP_β and MP_{β^*} on the original metric space (P, dist), respectively. Let $\widetilde{\boldsymbol{r}} = (\widetilde{r}_p)_{p \in P}$ where $\widetilde{r}_p$ is computed by MP_β when run on the distorted $(P, \widetilde{\mathrm{dist}})$, i.e.,

$$\sum_{q \in P} \left[\widetilde{r}_p - \beta \cdot \widetilde{\mathrm{dist}}^z(p,q)\right]^+ \ =\ \lambda, \qquad \forall p \in P.$$

We first show that the vector $\widetilde{\boldsymbol{r}}$ satisfies $\boldsymbol{r}^{(\beta)} \leq \widetilde{\boldsymbol{r}} \leq \boldsymbol{r}^{(\beta^*)}$. To see this, observe from (9) that, for every $p \in P$,

$$\sum_{q \in P} \left[\widetilde{r}_p - \beta^* \cdot \mathrm{dist}^z(p,q)\right]^+ \ \leq\ \sum_{q \in P} \left[\widetilde{r}_p - \beta \cdot \widetilde{\mathrm{dist}}^z(p,q)\right]^+ \ =\ \lambda \ =\ \sum_{q \in P} \left[r_p^{(\beta^*)} - \beta^* \cdot \mathrm{dist}^z(p,q)\right]^+$$

$$\sum_{q\in P}\big[\widetilde{r}_p-\beta\cdot\mathrm{dist}^z(p,q)\big]^+ \;\geq\; \sum_{q\in P}\big[\widetilde{r}_p-\beta\cdot\widetilde{\mathrm{dist}}^z(p,q)\big]^+ \;=\; \lambda \;=\; \sum_{q\in P}\big[r_p^{(\beta)}-\beta\cdot\mathrm{dist}^z(p,q)\big]^+$$

Since the expression $\sum_{q\in P}[r-\beta\cdot\mathrm{dist}^z(p,q)]^+$ is increasing in r (and $\lambda>0$ is nonzero), we infer from the above two equations that $r_p^{(\beta)}\leq\widetilde{r}_p\leq r_p^{(\beta^*)}$; therefore, $\boldsymbol{r}^{(\beta)}\leq\widetilde{\boldsymbol{r}}\leq\boldsymbol{r}^{(\beta^*)}$ holds.

Since $\widetilde{\boldsymbol{y}}=\widetilde{\boldsymbol{r}}/\lambda$, we have $\boldsymbol{y}^{(\beta)}\leq\widetilde{\boldsymbol{y}}\leq\boldsymbol{y}^{(\beta^*)}$. Finally, we use this fact to prove the LMP property of $\widetilde{\boldsymbol{y}}$. Recall that $\beta^*=2^{-z-1}$ and $\boldsymbol{y}^{(\beta^*)}$ is the output of MP_{β^*} on the original space (P,dist); by Lemma 3.12, we have

$$\mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)}) \;\leq\; 2^{O(z^2)}\cdot(\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda)-\lambda\cdot\|\boldsymbol{y}^{(\beta^*)}\|_1). \tag{17}$$

The basic idea is to relate the clustering cost $\mathrm{cost}_z(\widetilde{\boldsymbol{y}})$ to the assignment cost $\sigma_z(\boldsymbol{x}^{(\beta^*)})=\mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)})$, where $\boldsymbol{x}^{(\beta^*)}\sim\boldsymbol{y}^{(\beta^*)}$ is the optimal feasible assignment to $\boldsymbol{y}^{(\beta^*)}$. To this end, we apply Claim 3.13. We verify the premises of Claim 3.13 for the choices $\boldsymbol{y}:=\widetilde{\boldsymbol{y}}=(1/\lambda)\cdot\widetilde{\boldsymbol{r}}$, $\boldsymbol{x}:=\boldsymbol{x}^{(\beta^*)}$, and $\boldsymbol{\mu}:=\gamma^*\cdot\beta^{-1}\cdot\boldsymbol{r}^{(\beta)}$ (for the constant $\gamma^*=2^{O(z^2)}$ specified in Lemma 3.9):

1. For every $p\in P$, by definition we have $\widetilde{r}_p\geq r_p^{(\beta)}$. It then follows that $\sum_{q\in P:\mathrm{dist}^z(p,q)\leq\mu_p}\widetilde{r}_q\geq\sum_{q\in P:\mathrm{dist}^z(p,q)\leq\gamma^*\cdot\beta^{-1}\cdot r_p^{(\beta)}}r_q^{(\beta)}=\sum_{q\in B_p^{(\beta)}(\gamma^*\cdot r_p^{(\beta)})}r_q^{(\beta)}\geq\lambda$, where the second step follows from the definition of $B_p^{(\beta)}(\gamma^*\cdot r_p^{(\beta)})$ in (10), and the third step applies Lemma 3.9. This further implies that $\sum_{q\in P:\mathrm{dist}^z(p,q)\leq\mu_p}y_q\geq 1$.

2. For every $p\in P$, $\sum_{q\in P}x_{p,q}=\sum_{q\in P}x_{p,q}^{(\beta^*)}=1$, given the optimality of $\boldsymbol{x}^{(\beta^*)}\sim\boldsymbol{y}^{(\beta^*)}$.

3. Since $\boldsymbol{y}=\widetilde{\boldsymbol{y}}\leq\boldsymbol{y}^{(\beta^*)}$, we have $\sum_{q\in P:\mathrm{dist}^z(p,q)\leq\mu_p}y_q^{(\beta^*)}\geq\sum_{q\in P:\mathrm{dist}^z(p,q)\leq\mu_p}y_q\geq 1$ for every $p\in P$, i.e., the nearby points $\{q\in P:\mathrm{dist}^z(p,q)\leq\mu_p\}$ already have sufficient opening for p. So, in the optimal fractional assignment $\boldsymbol{x}^{(\beta^*)}\sim\boldsymbol{y}^{(\beta^*)}$ under Constraints (C1) and (C2) (see Section 2), every point $q\in P$ that p is assigned to (i.e., $x_{p,q}=x_{p,q}^{(\beta^*)}>0$) satisfies $\mathrm{dist}^z(p,q)\leq\mu_p$.

By applying Claim 3.13 and $\sigma_z(\boldsymbol{x}^{(\beta^*)})=\mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)})$, we can derive

$$\mathrm{cost}_z(\widetilde{\boldsymbol{y}}) \;\leq\; \mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)})+\sum_{p,q\in P}[x_{p,q}^{(\beta^*)}-\widetilde{y}_q]^+\cdot\gamma^*\cdot\beta^{-1}\cdot r_p^{(\beta)} \tag{18}$$

$$\triangleright\,\boldsymbol{r}^{(\beta)}\leq\boldsymbol{r}^{(\beta^*)} \qquad \leq\; \mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)})+\sum_{p,q\in P}[x_{p,q}^{(\beta^*)}-\widetilde{y}_q]^+\cdot\gamma^*\cdot\beta^{-1}\cdot r_p^{(\beta^*)}. \tag{19}$$

We divide all pairs $P\times P=\mathcal{F}\sqcup\mathcal{C}$ into the "far" pairs $\mathcal{F}$ and the "close" pairs $\mathcal{C}$, as follows:

$$\mathcal{F} \;:=\; \big\{(p,q)\in P\times P:\;\beta^*\cdot\mathrm{dist}^z(p,q)>\tfrac{1}{3\cdot 2^{z-1}}\cdot r_p^{(\beta^*)}\big\},$$
$$\mathcal{C} \;:=\; \big\{(p,q)\in P\times P:\;\beta^*\cdot\mathrm{dist}^z(p,q)\leq\tfrac{1}{3\cdot 2^{z-1}}\cdot r_p^{(\beta^*)}\big\}.$$

Below, Claims 3.20 and 3.21 reason about these two kinds of pairs separately.

Claim 3.20. $\sum_{(p,q)\in\mathcal{F}}[x_{p,q}^{(\beta^*)}-\widetilde{y}_q]^+\cdot r_p^{(\beta^*)}\leq\frac{3}{4}\cdot\mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)})$.

Proof. Each far pair $(p,q)\in\mathcal{F}$ satisfies $r_p^{(\beta^*)}\leq 3\cdot 2^{z-1}\cdot\beta^*\cdot\mathrm{dist}^z(p,q)=\frac{3}{4}\cdot\mathrm{dist}^z(p,q)$. Hence, we have $\sum_{(p,q)\in\mathcal{F}}[x_{p,q}^{(\beta^*)}-\widetilde{y}_q]^+\cdot r_p^{(\beta^*)}\leq\sum_{(p,q)\in\mathcal{F}}x_{p,q}^{(\beta^*)}\cdot\frac{3}{4}\cdot\mathrm{dist}^z(p,q)\leq\frac{3}{4}\cdot\sigma_z(\boldsymbol{x}^{(\beta^*)})=\frac{3}{4}\cdot\mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)})$. □

Claim 3.21. $\sum_{(p,q)\in\mathcal{C}}[x_{p,q}^{(\beta^*)}-\widetilde{y}_q]^+\cdot r_p^{(\beta^*)}\leq 3\cdot 2^{z-1}\cdot\lambda\cdot(\|\boldsymbol{y}^{(\beta^*)}\|_1-\|\widetilde{\boldsymbol{y}}\|_1)$.

Proof. Each close pair $(p,q) \in \mathcal{C}$ satisfies $\beta^* \cdot \mathrm{dist}^z(p,q) \le \frac{1}{3\cdot 2^{z-1}} \cdot r_p^{(\beta^*)}$. In combination with (Lemma 3.7) $r_p^{(\beta^*)} \le 2^{z-1} \cdot (r_q^{(\beta^*)} + \beta^* \cdot \mathrm{dist}^z(p,q))$, we can deduce that $r_p^{(\beta^*)} \le 3 \cdot 2^{z-2} \cdot r_q^{(\beta^*)}$ and then $\beta^* \cdot \mathrm{dist}^z(p,q) \le \frac{1}{2} \cdot r_q^{(\beta^*)}$. Therefore, we have $\mathcal{C} \subseteq \{(p,q) \in P \times P : \ p \in B_q^{(\beta^*)}(r_q^{(\beta^*)}/2)\}$. It follows that

$$
\begin{aligned}
\sum_{(p,q)\in\mathcal{C}} [x_{p,q}^{(\beta^*)} - \widetilde{y}_q]^+ \cdot r_p^{(\beta^*)} &\le \sum_{(p,q)\in P\times P: p\in B_q^{(\beta^*)}(r_q^{(\beta^*)}/2)} [x_{p,q}^{(\beta^*)} - \widetilde{y}_q]^+ \cdot r_p^{(\beta^*)} \\
\triangleright\ x_{p,q}^{(\beta^*)} \le y_q^{(\beta^*)} \text{ (feasibility)} \quad &\le \sum_{(p,q)\in P\times P: p\in B_q^{(\beta^*)}(r_q^{(\beta^*)}/2)} [y_q^{(\beta^*)} - \widetilde{y}_q]^+ \cdot r_p^{(\beta^*)} \\
&\le \sum_{(p,q)\in P\times P: p\in B_q^{(\beta^*)}(r_q^{(\beta^*)}/2)} [y_q^{(\beta^*)} - \widetilde{y}_q]^+ \cdot 3\cdot 2^{z-2} \cdot r_q^{(\beta^*)} \\
&\le \sum_{q\in P} |B_q^{(\beta^*)}(r_q^{(\beta^*)}/2)| \cdot [y_q^{(\beta^*)} - \widetilde{y}_q]^+ \cdot 3\cdot 2^{z-2} \cdot r_q^{(\beta^*)} \\
\triangleright \text{ Lemma 3.5} \quad &\le \sum_{q\in P} (2\lambda / r_q^{(\beta^*)}) \cdot [y_q^{(\beta^*)} - \widetilde{y}_q]^+ \cdot 3\cdot 2^{z-2} \cdot r_q^{(\beta^*)} \\
\triangleright\ \widetilde{\boldsymbol{y}} \le \boldsymbol{y}^{(\beta^*)} \quad &= 3\cdot 2^{z-1} \cdot \lambda \cdot (\|\boldsymbol{y}^{(\beta^*)}\|_1 - \|\widetilde{\boldsymbol{y}}\|_1).
\end{aligned}
$$

This completes the proof of Claim 3.21. □

Applying Lemma 3.12 and Claims 3.20 and 3.21 to (19) finishes the proof of Lemma 3.1, as follows:

$$
\begin{aligned}
\mathrm{cost}_z(\widetilde{\boldsymbol{y}}) &\le \mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)}) + \gamma^* \cdot \beta^{-1} \cdot \sum_{(p,q)\in\mathcal{F}\sqcup\mathcal{C}} [x_{p,q}^{(\beta^*)} - \widetilde{y}_q]^+ \cdot r_p^{(\beta^*)} \\
\triangleright \text{ Claims 3.20 and 3.21} \quad &\le \mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)}) + \gamma^* \cdot \beta^{-1} \cdot \left(\tfrac{3}{4} \cdot \mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)}) + 3\cdot 2^{z-1} \cdot \lambda \cdot (\|\boldsymbol{y}^{(\beta^*)}\|_1 - \|\widetilde{\boldsymbol{y}}\|_1)\right) \\
\triangleright\ \gamma^* = 2^{O(z^2)} \quad &\le 2^{O(z^2)} \cdot \beta^{-1} \cdot \left(\mathrm{cost}_z(\boldsymbol{y}^{(\beta^*)}) + \lambda \cdot (\|\boldsymbol{y}^{(\beta^*)}\|_1 - \|\widetilde{\boldsymbol{y}}\|_1)\right) \\
\triangleright\ (17) \quad &\le 2^{O(z^2)} \cdot \beta^{-1} \cdot \left(2^{O(z^2)} \cdot (\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda) - \lambda \cdot \|\boldsymbol{y}^{(\beta^*)}\|_1) + \lambda \cdot (\|\boldsymbol{y}^{(\beta^*)}\|_1 - \|\widetilde{\boldsymbol{y}}\|_1)\right) \\
&\le 2^{O(z^2)} \cdot \beta^{-1} \cdot \left(\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda) - \lambda \cdot \|\widetilde{\boldsymbol{y}}\|_1\right),
\end{aligned}
$$

where the last step holds since $\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda) - \lambda \cdot \|\boldsymbol{y}^{(\beta^*)}\|_1 \ge 0$ (Lemma 3.12) and $\|\boldsymbol{y}^{(\beta^*)}\|_1 - \|\widetilde{\boldsymbol{y}}\|_1 \ge 0$. □

4 $O(1)$-Approximation with $k + O(n^{1-\varepsilon})$ Centers

In this section, we design an MPC algorithm that computes a *bicriteria* $O(1)$-approximation for Euclidean (k,z)-Clustering (Theorem 4.1) using $O(n^{1-\varepsilon})$ additional centers.

Theorem 4.1 (Bicriteria Approximation)**.** *For any constant $\varepsilon \in (0,1)$ and integer $z \ge 1$, there is a randomized MPC algorithm for* Euclidean (k,z)-Clustering *that*

- *takes as input a dataset $P \subseteq \mathbb{R}^d$, a candidate center set $F \subseteq \mathbb{R}^d$, an integer $1 \le k \le |F|$, and upper bounds $n \ge |P|+|F|$ and $\Delta \ge \Delta(P \cup F)$ on the total number of points and aspect ratio, respectively;*

- *computes, with probability* $1 - \frac{1}{\mathrm{poly}(n)}$*, a center set* $C \subseteq F$ *satisfying*

$$|C| \le k + 1 + |F|/n^{\varepsilon} \le k + O(n^{1-\varepsilon}), \quad \mathrm{cost}_z(P, C) \le 2^{O(z^2)} \cdot \varepsilon^{-O(z)} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P, F, k).$$

With local memory $s \ge \mathrm{poly}(d \log(n\Delta))$ *per machine, the algorithm runs in* $O(\log_s n)$ *rounds and uses* $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \mathrm{poly}(d \log(n\Delta))$ *total memory.*

Here, Theorem 4.1 is stated in a slightly more general form that allows n to be any upper bound on $|P| + |F|$, for technical reasons that arise in Section 5; for example, we may run Theorem 4.1 on a much smaller input (P', F'), where $P' \subseteq P$ and $F' \subseteq F$, while still requiring the size guarantee $|C| \le k + 1 + |F'|/n^{\varepsilon}$ with respect to the original size $n = |P| + |F|$.

Throughout this section, we only consider a single dataset P. Hence, for simplicity, we henceforth write $\mathrm{cost}_z(P, C)$ as $\mathrm{cost}_z(C)$ for any center set C, $\mathrm{cost}_z(P, \boldsymbol{y})$ as $\mathrm{cost}_z(\boldsymbol{y})$ for any fractional solution $\boldsymbol{y}$, and $\sigma_z(P, \boldsymbol{x})$ as $\sigma_z(\boldsymbol{x})$ for any fractional assignment $\boldsymbol{x}$.

Proof. At a high level, the algorithm in Theorem 4.1 consists of two steps. First, we compute a fractional solution that opens exactly k total units of centers, using the fractional LMP algorithm presented in Section 3. We then round this fractional solution to a center set while introducing at most $1 + |F|/n^{\varepsilon}$ additional centers. We present formal statements of the algorithms for these two steps below.

Lemma 4.2 (Computing a Fractional Solution)**.** *For any constant* $\varepsilon \in (0, 1)$ *and integer* $z \ge 1$*, there is a randomized MPC algorithm for* Euclidean (k, z)-Clustering *that*

- *takes as input a dataset* $P \subseteq \mathbb{R}^d$*, a candidate center set* $F \subseteq \mathbb{R}^d$*, an integer* $1 \le k \le |F|$*, and upper bounds* $n \ge |P| + |F|$ *and* $\Delta \ge \Delta(P \cup F)$ *on the total number of points and aspect ratio, respectively;*
- *computes, with probability* $1 - \frac{1}{\mathrm{poly}(n)}$*, a fractional solution* $\boldsymbol{y} \in \mathbb{R}^F_{\ge 0}$ *satisfying*

$$\|\boldsymbol{y}\|_1 \;=\; k, \quad \mathrm{cost}_z(\boldsymbol{y}) \le 2^{O(z^2)} \cdot \varepsilon^{-O(z)} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P, F, k)$$

With local memory $s \ge \mathrm{poly}(d \log(n\Delta))$ *per machine, the algorithm runs in* $O(\log_s n)$ *rounds and uses* $2^{O(\varepsilon d)} \cdot n \cdot \mathrm{poly}(d \log(n\Delta))$ *total memory.*

Proof. The proof can be found in Section 4.1. □

Lemma 4.3 (Rounding with Additional Centers)**.** *For any constant* $\varepsilon \in (0, 1)$ *and integer* $z \ge 1$*, there is a randomized MPC algorithm for* Euclidean (k, z)-Clustering *that*

- *takes as input a dataset* $P \subseteq \mathbb{R}^d$*, a candidate center set* $F \subseteq \mathbb{R}^d$*, an integer* $1 \le k \le |F|$*, a fractional solution* $\boldsymbol{y} \in \mathbb{R}^F_{\ge 0}$ *with* $\|\boldsymbol{y}\|_1 = k$*, and upper bounds* $n \ge |P| + |F|$ *and* $\Delta \ge \Delta(P \cup F)$ *on the total number of points and aspect ratio, respectively;*
- *computes, with probability* $1 - \frac{1}{\mathrm{poly}(n)}$*, a center set* $C \subseteq F$ *satisfying*

$$|C| \;\le\; k + 1 + |F|/n^{\varepsilon}, \quad \mathrm{cost}_z(C) \le (z/\varepsilon)^{O(z)} \cdot \mathrm{cost}_z(\boldsymbol{y}).$$

With local memory $s \ge \mathrm{poly}(d \log(n\Delta))$ *per machine, the algorithm runs in* $O(\log_s n)$ *rounds and uses* $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \mathrm{poly}(d \log(n\Delta))$ *total memory.*

Proof. The proof can be found in Section 4.2. □

The algorithm for Theorem 4.1 is then obtained straightforwardly by combining the above two algorithms: we first run the algorithm from Lemma 4.2 to compute a fractional solution $\boldsymbol{y} \in \mathbb{R}^F_{\geq 0}$, and then apply the algorithm from Lemma 4.3 to obtain a solution $C \subseteq F$. The size bound $|C| \leq k + 1 + |F|/n^{\varepsilon}$ follows directly from Lemma 4.3, and the cost satisfies

$$\mathrm{cost}_z(C) \;\leq\; (z/\varepsilon)^{O(z)} \cdot \mathrm{cost}_z(\boldsymbol{y}) \;\leq\; 2^{O(z^2)} \cdot \varepsilon^{-O(z)} \cdot \mathrm{OPT}^{\mathrm{cl}}_z(P, F, k).$$

The round complexity and total memory are at most the sums of the corresponding bounds in Lemma 4.2 and Lemma 4.3, and are thus as stated in Theorem 4.1. Finally, both the algorithm in Lemma 4.2 and that in Lemma 4.3 succeed with probability $1 - \frac{1}{\mathrm{poly}(n)}$. Therefore, by a union bound, the overall algorithm succeeds with probability at least $1 - \frac{1}{\mathrm{poly}(n)}$. This finishes the proof of Theorem 4.1. □

4.1 Computing a Fractional Solution (Proof of Lemma 4.2)

In this subsection, we prove Lemma 4.2, which is restated below for convenience.

Lemma 4.2 (Computing a Fractional Solution)**.** *For any constant $\varepsilon \in (0, 1)$ and integer $z \geq 1$, there is a randomized MPC algorithm for* Euclidean (k, z)-Clustering *that*

- *takes as input a dataset $P \subseteq \mathbb{R}^d$, a candidate center set $F \subseteq \mathbb{R}^d$, an integer $1 \leq k \leq |F|$, and upper bounds $n \geq |P| + |F|$ and $\Delta \geq \Delta(P \cup F)$ on the total number of points and aspect ratio, respectively;*
- *computes, with probability $1 - \frac{1}{\mathrm{poly}(n)}$, a fractional solution $\boldsymbol{y} \in \mathbb{R}^F_{\geq 0}$ satisfying*

$$\|\boldsymbol{y}\|_1 \;=\; k, \quad \mathrm{cost}_z(\boldsymbol{y}) \leq 2^{O(z^2)} \cdot \varepsilon^{-O(z)} \cdot \mathrm{OPT}^{\mathrm{cl}}_z(P, F, k)$$

With local memory $s \geq \mathrm{poly}(d \log(n\Delta))$ per machine, the algorithm runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n \cdot \mathrm{poly}(d \log(n\Delta))$ total memory.

Proof. We divide the proof into two parts. First, we consider the special case where $F = P$, which is exactly the setting considered by our algorithm for Power-z Facility Location in Corollary 3.2. Then, we apply a postprocessing step to transform the fractional solution $\boldsymbol{y} \in \mathbb{R}^P_{\geq 0}$ into a fractional solution in $\mathbb{R}^F_{\geq 0}$, as desired.

Algorithm for the Case $F = P$. Our MPC algorithm for the case $F = P$, shown in Algorithm 3, is adapted from the algorithm of [BCG+24, Section 14], which invokes an LMP algorithm for Power-z Facility Location as a subroutine. We replace that subroutine with our MPC algorithm from Corollary 3.2. Below, we describe and analyze Algorithm 3 assuming $2 \leq k \leq |P|$, as the algorithm cannot handle the case $k = 1$ (specifically, the index ℓ^* computed in Line 4 may not exist). For the case $k = 1$, we instead use a much simpler uniform sampling-based algorithm, which we discuss later.

At a high level, this algorithm invokes Corollary 3.2 with different values of λ to yield a series of fractional center sets $\{\boldsymbol{y}^{(\ell)}\}_{\ell \in [0:L]}$. It then identifies two particular fractional center sets $\boldsymbol{y}^{(\ell^*-1)}$ and $\boldsymbol{y}^{(\ell^*)}$, such that $\|\boldsymbol{y}^{(\ell^*-1)}\|_1 \geq k$ and $\|\boldsymbol{y}^{(\ell^*)}\|_1 \leq k$, and outputs their linear combination $\boldsymbol{y}$ satisfying $\|\boldsymbol{y}\|_1 = k$. The LMP approximation guarantee established in Corollary 3.2 is key to showing that

Algorithm 3: Computing a fractional solution to Euclidean (k, z)-Clustering

1 $L \leftarrow O(\log(n\Delta))$ (with a sufficiently large hidden constant), $\lambda_\ell \leftarrow 2^{\ell z}$ for every $\ell \in [0 : L]$.
2 $\boldsymbol{y}^{(0)} \leftarrow \mathbf{1}^P$ (i.e., $y_p^{(0)} \leftarrow 1$ for every point $p \in P$)
3 for each $\ell \in [L]$, $\boldsymbol{y}^{(\ell)} \leftarrow$ the fractional center set computed by Corollary 3.2 with $\lambda = \lambda_\ell$
4 find an arbitrary $\ell^* \in [L]$ such that $\|\boldsymbol{y}^{(\ell^*-1)}\|_1 \geq k$ and $\|\boldsymbol{y}^{(\ell^*)}\|_1 \leq k$
`// The existence of such an` ℓ^* `is guaranteed by Facts B.1 and B.2.`
5 compute $\alpha \in [0, 1]$ such that $\alpha \cdot \|\boldsymbol{y}^{(\ell^*-1)}\|_1 + (1-\alpha) \cdot \|\boldsymbol{y}^{(\ell^*)}\|_1 = k$
6 **return** $\boldsymbol{y} \leftarrow \alpha \cdot \boldsymbol{y}^{(\ell^*-1)} + (1-\alpha) \cdot \boldsymbol{y}^{(\ell^*)}$

$\boldsymbol{y}$ is a good fractional solution to (k, z)-Clustering. In what follows, we assume without loss of generality that for every $p \neq q \in (P \cup F)$, $2 \leq \text{dist}(p, q) \leq O(\Delta)$ (as in the proof of Corollary 3.2).

The correctness analysis of Algorithm 3 is (almost) identical to that in [BCG+24, Section 14], which is summarized in Lemma 4.4 in our notation.[3] For completeness, we provide a full proof in Appendix B.

Lemma 4.4 (Correctness of Algorithm 3)**.** *Assume that $2 \leq k \leq |P|$. Algorithm 3 returns a fractional center set $\boldsymbol{y} \in \mathbb{R}^P_{\geq 0}$ such that $\|\boldsymbol{y}\|_1 = k$ and $\text{cost}_z(\boldsymbol{y}) \leq 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \text{OPT}^{\text{cl}}_z(P, P, k)$.*

Proof. The proof can be found in Appendix B. □

MPC Implementation of Algorithm 3. Line 1 is trivial. Lines 2 and 3 for computing the fractional center sets $\{\boldsymbol{y}^{(\ell)}\}_{\ell \in [0:L]}$ can be easily parallelized; the round complexity and local memory requirement are the same as those of Corollary 3.2, while the total memory is that of Corollary 3.2 multiplied by $L = \Theta(\log(n\Delta))$ (due to parallelization). Line 4 requires the level ℓ^*, which can be found by first computing each norm $\|\boldsymbol{y}^{(\ell)}\|_1$ by aggregation and then gathering all pairs $(\ell, \|\boldsymbol{y}^{(\ell)}\|_1)$ at a single (leader) machine; this requires local memory $s \geq \Theta(L+1) = O(\log(n\Delta))$. Line 5 requires the value α, which can be found locally on the leader machine. After broadcasting α and ℓ^* and bringing the entries $y_p^{(\ell^*-1)}$ and $y_p^{(\ell^*)}$ to the same machine for every point $p \in P$, the computation of $\boldsymbol{y} = \alpha \cdot \boldsymbol{y}^{(\ell^*-1)} + (1-\alpha) \cdot \boldsymbol{y}^{(\ell^*)}$ is straightforward; this computation takes $O(\log_s n)$ rounds.

In sum, the above implementation of Algorithm 3 requires local memory $s \geq \text{poly}(d \log(n\Delta))$, runs in $O(\log_s n)$ rounds, and uses total memory $2^{O(\varepsilon d)} \cdot n \cdot \text{poly}(d \log(n\Delta))$. We also note that the above implementation of Algorithm 3 (for $k \geq 2$) is deterministic.

Special Case where $k = 1$. When $k = 1$, it is well known that selecting a center uniformly at random from P provides a $2^{O(z)}$-approximation in expectation (see, e.g., [AV07]). We can boost the success probability to $1 - \frac{1}{\text{poly}(n)}$ by independently sampling $O(\log n)$ such centers in parallel and selecting the best one as the final solution. Since both uniform sampling and computing the cost of a singleton center set can be efficiently implemented in the MPC model (via broadcast and converge-cast), this case can be addressed within the claimed number of rounds and space.

Extending to the Case $F \neq P$. We first handle the case $k \geq |P|$. For every $p \in P$, compute an $O(\varepsilon^{-1})$-ANN $\pi(p) \in F$. Let $S := \{\pi(p) : p \in P\}$. We place one unit of opening at every center in S and at any $k - |S|$ additional centers in $F \setminus S$. Such additional centers exist because $|S| \leq |P| \leq$

[3] While the analysis in [BCG+24, Section 14] only considers the case $z = 1$, it can be easily extended to general $z \geq 1$, as stated in Lemma 4.4.

$k \le |F|$. This gives total opening k and cost at most $O(\varepsilon^{-z}) \cdot \mathrm{cost}_z(F) \le O(\varepsilon^{-z}) \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P, F, k)$. Below, we focus on the case $1 \le k < |P|$.

We can extend the preceding algorithm to an arbitrary candidate center set $F \subseteq \mathbb{R}^d$, as follows. We first compute a fractional solution $\boldsymbol{y} \in \mathbb{R}^P_{\ge 0}$ using the preceding algorithm. Then, for each $p \in P$, we compute an $O(\varepsilon^{-1})$-ANN $\pi(p) \in F$ of p in F, and define a new fractional solution $\boldsymbol{y}' \in \mathbb{R}^F_{\ge 0}$ by

$$\forall c \in F, \qquad y'_c := \sum_{p \in P : \pi(p) = c} y_p.$$

We return the resulting fractional solution $\boldsymbol{y}'$ as the final output.

The MPC implementation is straightforward. To compute $\{\pi(p)\}_{p \in P}$, we use the deterministic approximate nearest-neighbor primitive from [CGJ+24, Section 3.1]. Given $\{\pi(p)\}_{p \in P}$, computing the new fractional solution $\boldsymbol{y}'$ reduces to an aggregation task, which can be implemented using standard MPC sorting and converge-cast (see, e.g., [CKP26, Lemma 5.10]). The bottleneck is therefore the approximate nearest-neighbor computation. Hence, the implementation requires local memory $s \ge \mathrm{poly}(d \log(n\Delta))$, runs in $O(\log_s n)$ rounds, and uses total memory $2^{O(\varepsilon d)} \cdot n \cdot \mathrm{poly}(d \log(n\Delta))$.

We now establish the correctness of this extension. Clearly, the construction of $\boldsymbol{y}'$ preserves the total fractional mass, and thus $\|\boldsymbol{y}'\|_1 = \|\boldsymbol{y}\|_1 = k$. We then bound the cost of $\boldsymbol{y}'$. Let $\boldsymbol{x} \in \mathbb{R}^{P \times P}_{\ge 0}$ be an optimal fractional assignment with respect to $\boldsymbol{y}$, i.e., $\sigma_z(\boldsymbol{x}) = \mathrm{cost}_z(\boldsymbol{y})$, and define

$$\forall p \in P,\ c \in F, \qquad x'_{p,c} := \sum_{q \in P : \pi(q) = c} x_{p,q}.$$

Clearly, this assignment is feasible with respect to $\boldsymbol{y}'$. For every $p, q \in P$, by the triangle inequality,

$$\begin{aligned} \mathrm{dist}(p, \pi(q)) &\le \mathrm{dist}(p, q) + \mathrm{dist}(q, \pi(q)) \le \mathrm{dist}(p, q) + O(\varepsilon^{-1}) \cdot \mathrm{dist}(q, F) \\ &\le O(\varepsilon^{-1}) \cdot (\mathrm{dist}(p, q) + \mathrm{dist}(p, F)). \end{aligned}$$

Therefore, by the generalized triangle inequality (Fact 2.1), we obtain

$$\begin{aligned} \mathrm{cost}_z(\boldsymbol{y}') \le \sigma_z(\boldsymbol{x}') &= \sum_{p \in P} \sum_{c \in F} x'_{p,c} \cdot \mathrm{dist}^z(p, c) \\ &= \sum_{p, q \in P} x_{p,q} \cdot \mathrm{dist}^z(p, \pi(q)) \\ &\le 2^{O(z)} \varepsilon^{-z} \cdot \Big(\sum_{p,q \in P} x_{p,q} \cdot \mathrm{dist}^z(p, q) + \sum_{p,q \in P} x_{p,q} \cdot \mathrm{dist}^z(p, F) \Big) \\ &= 2^{O(z)} \varepsilon^{-z} \cdot \Big(\mathrm{cost}_z(\boldsymbol{y}) + \mathrm{cost}_z(F) \Big) \\ &\le 2^{O(z^2)} \cdot \varepsilon^{-3z} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P, P, k) + 2^{O(z)} \varepsilon^{-z} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P, F, k), \end{aligned} \tag{20}$$

where the last step follows from $\mathrm{cost}_z(\boldsymbol{y}) \le 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P, P, k)$ by Lemma 4.4 (or the sampling guarantee above when $k = 1$), and from $\mathrm{cost}_z(F) \le \mathrm{OPT}_z^{\mathrm{cl}}(P, F, k)$ by the definition of $\mathrm{OPT}_z^{\mathrm{cl}}(P, F, k)$.

Finally, we compare $\mathrm{OPT}_z^{\mathrm{cl}}(P, P, k)$ with $\mathrm{OPT}_z^{\mathrm{cl}}(P, F, k)$, as follows. Let $C^\star \subseteq F$ be an optimal size-k center set, i.e., $\mathrm{cost}_z(C^\star) = \mathrm{OPT}_z^{\mathrm{cl}}(P, F, k)$. Let $C' := \{\mathrm{NN}(c, P) : c \in C^\star\} \subseteq P$. Then, $|C'| \le |C^\star| \le k$, and thus $\mathrm{OPT}_z^{\mathrm{cl}}(P, P, k) \le \mathrm{cost}_z(C')$. For every point $p \in P$, letting $c := \mathrm{NN}(p, C^\star)$, we have

$$\mathrm{dist}(p, C') \le \mathrm{dist}(p, \mathrm{NN}(c, P)) \le \mathrm{dist}(p, c) + \mathrm{dist}(c, P) \le 2\,\mathrm{dist}(p, c) = 2\,\mathrm{dist}(p, C^\star).$$

Therefore,

$$\mathrm{OPT}_z^{\mathrm{cl}}(P,P,k) \;\le\; \mathrm{cost}_z(C') \;\le\; 2^z \cdot \mathrm{cost}_z(C^\star) \;=\; 2^z \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P,F,k).$$

Plugging this into (20) gives

$$\mathrm{cost}_z(\boldsymbol{y}') \;\le\; 2^{O(z^2)}\varepsilon^{-3z} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P,F,k),$$

which verifies the correctness of $\boldsymbol{y}'$. □

4.2 Rounding to $k + O(n^{1-\varepsilon})$ Centers (Proof of Lemma 4.3)

In this subsection, we prove Lemma 4.3, which is restated below for convenience.

Lemma 4.3 (Rounding with Additional Centers)**.** *For any constant $\varepsilon \in (0,1)$ and integer $z \ge 1$, there is a randomized MPC algorithm for* Euclidean (k,z)-Clustering *that*

- *takes as input a dataset $P \subseteq \mathbb{R}^d$, a candidate center set $F \subseteq \mathbb{R}^d$, an integer $1 \le k \le |F|$, a fractional solution $\boldsymbol{y} \in \mathbb{R}^F_{\ge 0}$ with $\|\boldsymbol{y}\|_1 = k$, and upper bounds $n \ge |P|+|F|$ and $\Delta \ge \Delta(P \cup F)$ on the total number of points and aspect ratio, respectively;*
- *computes, with probability $1 - \frac{1}{\mathrm{poly}(n)}$, a center set $C \subseteq F$ satisfying*

$$|C| \;\le\; k + 1 + |F|/n^{\varepsilon}, \quad \mathrm{cost}_z(C) \le (z/\varepsilon)^{O(z)} \cdot \mathrm{cost}_z(\boldsymbol{y}).$$

With local memory $s \ge \mathrm{poly}(d\log(n\Delta))$ per machine, the algorithm runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \mathrm{poly}(d\log(n\Delta))$ total memory.

Throughout this subsection, we assume without loss of generality that $n^\varepsilon \ge 1$ is an integer. The proof of Lemma 4.3 spans Sections 4.2.1 and 4.2.2. Specifically, in Section 4.2.1, we present the algorithm in the offline setting (without considering its MPC implementation) and analyze the correctness of the returned center set, showing that it satisfies the desired size and cost guarantees in Lemma 4.3. Then, in Section 4.2.2, we show how to implement the algorithm in the MPC setting with the stated round complexity and total memory, thereby completing the proof of Lemma 4.3.

4.2.1 Algorithm Description and Analysis

We design the offline algorithm for Lemma 4.3 in three steps. In the first step, we design a weaker algorithm that returns a center set $C \subseteq F$ for which the size and cost guarantees hold only *in expectation*; this step applies the exponential-clock rounding scheme of [ANS17]. In the second step, we strengthen these expected guarantees to hold with constant probability via *bi-point rounding*. In the third step, we further boost the success probability to high probability via simple repetition.

Step 1: Expected Guarantees via Exponential Clocks. An offline version of the algorithm for the first step is presented in Algorithm 4. The algorithm starts by computing an auxiliary fractional solution $\boldsymbol{y}'$ whose cost and total opening are comparable to those of $\boldsymbol{y}$, together with a feasible assignment $\boldsymbol{x}' \sim \boldsymbol{y}'$, and then applies the exponential-clock rounding scheme of [ANS17] on $(\boldsymbol{x}', \boldsymbol{y}')$. Crucially, both $\boldsymbol{y}'$ and $\boldsymbol{x}'$ take values only in $\{0, 1/n^\varepsilon\}$, which is important for both (i) satisfying the requirement of the rounding scheme of [ANS17] (see the proof of Claim 4.5) and (ii) facilitating an efficient MPC implementation.

The correctness of Algorithm 4 is established in the following claim.

Algorithm 4: Rounding with expected size and cost guarantees

1 construct a multiset Q of points from F and a fractional solution $\boldsymbol{y}' := \frac{1}{n^\varepsilon} \cdot \mathbf{1}^Q$ such that $\|\boldsymbol{y}'\|_1 \le k + |F|/n^\varepsilon$ and $\mathrm{cost}_z(\boldsymbol{y}') \le \mathrm{cost}_z(\boldsymbol{y})$

2 compute an assignment $\boldsymbol{x}' \in \{0, 1/n^\varepsilon\}^{P \times Q}$ such that $\boldsymbol{x}' \sim \boldsymbol{y}'$, $\sum_{q \in Q} x'_{p,q} = 1$ for every $p \in P$, and $\sigma_z(\boldsymbol{x}') \le O(\varepsilon^{-z}) \cdot \mathrm{cost}_z(\boldsymbol{y}')$

`/* Lines 3 to 6: exponential-clock rounding [ANS17] on` $(\boldsymbol{x}', \boldsymbol{y}')$ `*/`

3 independently sample $\alpha_q \sim \mathrm{Exp}(1/n^\varepsilon)$ for every $q \in Q$

4 independently sample $\beta_p \sim \mathrm{Exp}(1)$ for every $p \in P$

5 for every $p \in P$, $f(p) \leftarrow \mathrm{argmin}_{q \in Q : x'_{p,q} > 0} \alpha_q$

6 **return** $C \leftarrow$ the set $\left\{ f(p) : p \in P, p = \mathrm{argmin}_{p' \in P : x'_{p', f(p)} > 0} \beta_{p'} \right\} \subseteq F$

Claim 4.5 (Correctness of Algorithm 4)**.** *The center set $C \subseteq F$ returned by Algorithm 4 satisfies $\mathbb{E}[|C|] \le k + |F|/n^\varepsilon$ and $\mathbb{E}[\mathrm{cost}_z(C)] \le (z/\varepsilon)^{O(z)} \cdot \mathrm{cost}_z(\boldsymbol{y})$.*

Proof. We first have the following Claim 4.6, which establishes the correctness of the exponential-clock rounding scheme on $(\boldsymbol{x}', \boldsymbol{y}')$ in Lines 3 to 6. This claim follows by essentially the same argument as in [ANS17, Section 4], with their analysis generalized from the case $z = 1$ to arbitrary $z \ge 1$; for completeness, we provide a proof in Appendix C.1. We also note that the analysis of [ANS17, Section 4] requires the input $(\boldsymbol{x}', \boldsymbol{y}')$ to the exponential-clock rounding scheme to satisfy (i) $\boldsymbol{x}' \sim \boldsymbol{y}'$ is feasible, and (ii) $x'_{p,q} \in \{0, y'_q\}$ for every $p \in P$ and $q \in Q$; both requirements are guaranteed by the constructions in Lines 1 and 2.[4]

Claim 4.6. $\mathbb{E}[|C|] \le \|\boldsymbol{y}'\|_1$ *and* $\mathbb{E}[\mathrm{cost}_z(C)] \le (z/\varepsilon)^{O(z)} \cdot \mathrm{cost}_z(\boldsymbol{y}')$.

Proof. The proof can be found in Appendix C.1. □

Combining Claim 4.6 with the guarantees that $\|\boldsymbol{y}'\|_1 \le k + |F|/n^\varepsilon$ and $\mathrm{cost}_z(\boldsymbol{y}') \le \mathrm{cost}_z(\boldsymbol{y})$ from Line 1 completes the proof of Claim 4.5. □

Step 2: Constant-probability Guarantee via Bi-point Rounding. The solution returned by Algorithm 4 satisfies both the size and cost bounds only in expectation, as stated in Claim 4.5. Using Algorithm 4 as a subroutine, we now give an algorithm that, with constant probability, returns a solution C satisfying $|C| \le k + O(n^{1-\varepsilon})$ and $\mathrm{cost}_z(C) \le (z/\varepsilon)^{O(z)} \cdot \mathrm{cost}_z(\boldsymbol{y})$.

The algorithm is presented in Algorithm 5. The key idea is that, by repeating Algorithm 4 sufficiently many times, we can obtain a *bi-point solution* with *worst-case* size and cost guarantees with constant probability. Here, a bi-point solution is represented by a tuple (C_1, C_2, a), where C_1, C_2 are two center sets and $a \in [0, 1]$; its size and cost are defined as $a|C_1| + (1-a)|C_2|$ and $a\mathrm{cost}_z(C_1) + (1-a)\mathrm{cost}_z(C_2)$, respectively. Provided that we obtain a bi-point solution, we then apply the classical bi-point rounding algorithm of [JV01] to obtain a center set whose size is at most (the ceiling of) the size of the bi-point solution, while its expected cost is within an $\varepsilon^{-O(z)}$ factor of the cost of the bi-point solution. In this algorithm description, we do not give the details of the bi-point rounding procedure, and we note that an efficient MPC implementation has already been given in [CKP26] (see also Claim 4.12).

[4] The algorithm in [ANS17, Section 3] is designed for a dynamic setting over T time steps and additionally involves a vector $\boldsymbol{z}$ that accounts for changes in $\boldsymbol{x}'$. In our setting, there is only a single time step, i.e., $T = 1$, and hence the vector $\boldsymbol{z}$ is not needed.

Algorithm 5: Computing a solution with constant-probability guarantees

1 independently run Algorithm 4 $h := n^{2\varepsilon}$ times to obtain solutions $C_1, C_2, \ldots, C_h \subseteq F$
2 for every $i \in [h]$, compute $\widehat{\text{cost}}_z(C_i)$ such that $\text{cost}_z(C_i) \leq \widehat{\text{cost}}_z(C_i) \leq O(\varepsilon^{-z}) \cdot \text{cost}_z(C_i)$
3 find a bi-point solution (C_i, C_j, a) for some $i, j \in [h]$ (possibly equal) and $a \in [0,1]$ minimizing $a\widehat{\text{cost}}_z(C_i) + (1-a)\widehat{\text{cost}}_z(C_j)$ subject to $a|C_i| + (1-a)|C_j| \leq k + 2 \cdot |F|/n^{\varepsilon}$
4 **if** *no such bi-point solution is found* **then**
5 | pick an arbitrary bi-point solution (C_i, C_j, a)
6 run the bi-point rounding [JV01] on (C_i, C_j, a) to obtain a solution $C \subseteq F$ such that
(i) $|C| \leq a|C_i| + (1-a)|C_j| + 1$, and
(ii) $\mathbb{E}[\text{cost}_z(C)] \leq \varepsilon^{-O(z)} \cdot \big(a\text{cost}_z(C_i) + (1-a)\text{cost}_z(C_j)\big)$
7 **return** C

Claim 4.7 (Correctness of Algorithm 5)**.** *The center set $C \subseteq F$ returned by Algorithm 5 satisfies the following guarantees with probability at least $1/4$:*

$$|C| \;\leq\; k + 1 + 2 \cdot |F|/n^{\varepsilon}, \quad \textit{and} \quad \text{cost}_z(C) \;\leq\; (z/\varepsilon)^{O(z)} \cdot \text{cost}_z(\boldsymbol{y}).$$

Proof. By Claim 4.5, for every $i \in [h]$, we have

$$\mathbb{E}[|C_i|] \;\leq\; k + |F|/n^{\varepsilon}, \quad \mathbb{E}[\text{cost}_z(C_i)] \;\leq\; (z/\varepsilon)^{O(z)} \cdot \text{cost}_z(\boldsymbol{y}).$$

We claim that the following event occurs with probability at least $1/2$; we prove this claim at the end of the proof:

$$\begin{aligned} \exists \text{ a bi-point solution } (C_i, C_j, a) \text{ s.t. } & a|C_i| + (1-a)|C_j| \;\leq\; k + 2 \cdot |F|/n^{\varepsilon}, \\ & a\widehat{\text{cost}}_z(C_i) + (1-a)\widehat{\text{cost}}_z(C_j) \;\leq\; (z/\varepsilon)^{O(z)} \cdot \text{cost}_z(\boldsymbol{y}). \end{aligned} \tag{21}$$

Conditioned on this event, Line 3 finds such a bi-point solution, after which Line 6 computes a solution C satisfying $|C| \leq k + 1 + 2 \cdot |F|/n^{\varepsilon}$ and

$$\begin{aligned} \mathbb{E}[\text{cost}_z(C)] \;&\leq\; \varepsilon^{-O(z)} \cdot \big(a\text{cost}_z(C_i) + (1-a)\text{cost}_z(C_j)\big) \\ \triangleright \text{ Line 2} \quad &\leq\; \varepsilon^{-O(z)} \cdot \big(a\widehat{\text{cost}}_z(C_i) + (1-a)\widehat{\text{cost}}_z(C_j)\big) \;\leq\; (z/\varepsilon)^{O(z)} \cdot \text{cost}_z(\boldsymbol{y}). \end{aligned}$$

Thus, by Markov's inequality, we conclude that C satisfies the size and cost guarantees stated in Claim 4.7 with conditional probability at least $1/2$ given Event (21). Overall, the algorithm returns a desired solution with probability at least $1/4$.

It remains to prove that Event (21) occurs with probability at least $1/2$. Define the random variables $v_s := \frac{1}{h}\sum_{i=1}^{h} |C_i|$ and $v_c := \frac{1}{h}\sum_{i=1}^{h} \text{cost}_z(C_i)$. By Claim 4.5, we have

$$\mathbb{E}[v_s] \;\leq\; k + |F|/n^{\varepsilon}, \quad \mathbb{E}[v_c] \;\leq\; (z/\varepsilon)^{O(z)} \cdot \text{cost}_z(\boldsymbol{y}).$$

We then examine the concentration of both random variables. For v_s, note that $0 \leq |C_i| \leq |F|$ for every $i \in [h]$. By Hoeffding's inequality, we have

$$\begin{aligned} \Pr\left[v_s \geq k + 2 \cdot |F|/n^{\varepsilon}\right] \;&\leq\; \Pr\left[v_s - \mathbb{E}[v_s] \geq |F|/n^{\varepsilon}\right] \\ &\leq\; \exp\left(\frac{-2h \cdot (|F|/n^{\varepsilon})^2}{|F|^2}\right) \;\leq\; e^{-2}. \end{aligned}$$

For v_c, we simply apply Markov's inequality to obtain $\Pr[v_c > 4\,\mathbb{E}[v_c]] \le 1/4$. By a union bound, the event

$$\left\{ v_s \;=\; \frac{1}{h}\sum_{i=1}^{h} |C_i| \;\le\; k + 2\cdot|F|/n^{\varepsilon}, \quad v_c \;=\; \frac{1}{h}\sum_{i=1}^{h} \mathrm{cost}_z(C_i) \;\le\; (z/\varepsilon)^{O(z)}\cdot \mathrm{cost}_z(\boldsymbol{y}) \right\}$$

holds with probability at least $1 - e^{-2} - 1/4 > 1/2$. Then, conditioning on this event, we show the existence of a bi-point solution (C_i, C_j, a) satisfying the bounds in (21). Let $q_i := (|C_i|, \mathrm{cost}_z(C_i)) \in \mathbb{R}^2$ for every $i \in [h]$. Since $(v_s, v_c) = \frac{1}{h}\sum_{i=1}^{h} q_i$, the point (v_s, v_c) lies in the convex hull of $\{q_i\}_{i\in[h]}$. Let (v_s', v_c') be the point obtained by moving from (v_s, v_c) as far as possible in the direction $(-1,-1)$ while remaining in the convex hull. Then (v_s', v_c') lies on its boundary and satisfies $v_s' \le v_s$ and $v_c' \le v_c$. Since this convex hull lies in $\mathbb{R}^2$ and (v_s', v_c') is on its boundary, there exist $i, j \in [h]$ and $a \in [0,1]$ such that $(v_s', v_c') = aq_i + (1-a)q_j$. In other words, $a|C_i| + (1-a)|C_j| \le k + 2\cdot|F|/n^{\varepsilon}$ and $a\mathrm{cost}_z(C_i) + (1-a)\mathrm{cost}_z(C_j) \le (z/\varepsilon)^{O(z)}\mathrm{cost}_z(\boldsymbol{y})$. Finally, by Line 2, we obtain $a\widehat{\mathrm{cost}}_z(C_i) + (1-a)\widehat{\mathrm{cost}}_z(C_j) \le (z/\varepsilon)^{O(z)}\mathrm{cost}_z(\boldsymbol{y})$, with a larger hidden constant in the factor $(z/\varepsilon)^{O(z)}$. Thus, (C_i, C_j, a) satisfies (21). This finishes the proof. □

Step 3: High-probability Guarantee via Repetition (The Final Algorithm). Finally, we give an algorithm for Lemma 4.3 that computes a solution C such that both the size and cost bounds hold *with high probability*. The algorithm simply repeats Algorithm 5 independently $O(\log n)$ times and returns a solution with the minimum estimated cost among those satisfying the size bound.

Algorithm 6: Computing a solution with high-probability guarantees

```
1 independently run Algorithm 5 O(log n) times (with a sufficiently large hidden constant)
2 for every resulting solution C, compute a value cost_z^(C) as in Line 2 of Algorithm 5
3 if there exists a resulting solution satisfying |C| ≤ k + 1 + 2·|F|/n^ε then
4 |   return the solution with the minimum estimated cost cost_z^(C) among all such
  |     solutions
5 else
6 |   return an arbitrary subset of F
```

Claim 4.8 (Correctness of Algorithm 6)**.** *With probability* $1 - \frac{1}{\mathrm{poly}(n)}$*, Algorithm 6 returns a solution* $C \subseteq F$ *such that* $|C| \le k + 1 + 2\cdot|F|/n^{\varepsilon}$ *and* $\mathrm{cost}_z(C) \le (z/\varepsilon)^{O(z)}\cdot\mathrm{cost}_z(\boldsymbol{y})$.

Proof. By Claim 4.7, with probability at least $1/4$, a run of Algorithm 5 returns a solution $C \subseteq F$ such that

$$|C| \le k + 1 + 2\cdot|F|/n^{\varepsilon} \quad \text{and} \quad \mathrm{cost}_z(C) \le (z/\varepsilon)^{O(z)}\cdot\mathrm{cost}_z(\boldsymbol{y}).$$

With a sufficiently large hidden constant in the $O(\log n)$ independent repetitions, at least one run returns such a solution with probability $1 - \frac{1}{\mathrm{poly}(n)}$; denote one such solution by $C^\star$. Let C be the solution returned by Algorithm 6. The size bound $|C| \le k + 1 + 2\cdot|F|/n^{\varepsilon}$ follows directly from Line 4. As for the cost bound, by the selection of C in Line 4 and the guarantee of $\widehat{\mathrm{cost}}_z(C)$ in Line 2, it follows that

$$\mathrm{cost}_z(C) \;\le\; \widehat{\mathrm{cost}}_z(C) \;\le\; \widehat{\mathrm{cost}}_z(C^\star) \;\le\; O(\varepsilon^{-z})\cdot\mathrm{cost}_z(C^\star) \;\le\; (z/\varepsilon)^{O(z)}\cdot\mathrm{cost}_z(\boldsymbol{y}).$$

This finishes the proof of Claim 4.8. □

Finally, we note that Claim 4.8 guarantees a size bound of $|C| \leq k+1+2\cdot|F|/n^{\varepsilon}$, which can be easily strengthened to $|C| \leq k+1+|F|/n^{\varepsilon}$, as desired in Lemma 4.3, by scaling n by an appropriate constant factor. This scaling does not affect the asymptotic approximation ratio, round complexity, or total memory bound.

4.2.2 MPC Implementation

In this subsection, we discuss in turn how to implement the algorithms from the three steps in Section 4.2.1 in the MPC model, and show that all these implementations achieve the asymptotic round complexity and total memory bounds stated in Lemma 4.3, thereby completing its proof.

Throughout, we say that an MPC algorithm successfully implements an offline algorithm with probability $1-\frac{1}{\mathrm{poly}(n)}$ if there exists a success event of probability at least $1-\frac{1}{\mathrm{poly}(n)}$ such that, conditioned on this event, all guarantees established for the offline algorithm remain valid for the MPC implementation; we make no guarantee on its output when the success event does not occur.

MPC Implementation of Algorithm 4. We first discuss the MPC implementation of Algorithm 4 from the first step in Section 4.2.1.

Claim 4.9 (MPC Implementation of Algorithm 4)**.** *There exists an MPC algorithm that successfully implements Algorithm 4 with probability at least $1-\frac{1}{\mathrm{poly}(n)}$; specifically, conditioned on successful implementation, the guarantees in Claim 4.5 hold.*

With local memory $s \geq \mathrm{poly}(d\log(n\Delta))$ per machine, the algorithm runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)}\cdot n^{1+O(\varepsilon)}\cdot\mathrm{poly}(d\log(n\Delta))$ total memory.

Proof. To implement Line 1, we define the multiset Q as follows: for every $q\in F$, Q contains exactly $\lceil n^{\varepsilon}y_q\rceil$ copies of q; thus, $n^{\varepsilon}\leq|Q|\leq n^{\varepsilon}k+|F|=O(n^{1+\varepsilon})$. Such a multiset Q can be constructed in the MPC model by replicating each $q\in F$ into $\lceil n^{\varepsilon}y_q\rceil$ copies. The fractional solution $\boldsymbol{y}'=\frac{1}{n^{\varepsilon}}\cdot\mathbf{1}^{Q}$ can then be constructed trivially, and its total opening satisfies

$$\|\boldsymbol{y}'\|_1 \;=\; \frac{1}{n^{\varepsilon}}\cdot\sum_{q\in F}\lceil n^{\varepsilon}\cdot y_q\rceil \;\leq\; \frac{1}{n^{\varepsilon}}\cdot\sum_{q\in F}(n^{\varepsilon}\cdot y_q+1) \;=\; \|\boldsymbol{y}\|_1+\frac{|F|}{n^{\varepsilon}}.$$

It is also easy to verify that $\mathrm{cost}_z(\boldsymbol{y}')\leq\mathrm{cost}_z(\boldsymbol{y})$, since, for every $q\in F$, the total opening at location q under $\boldsymbol{y}'$ (aggregated over all copies of q in Q) is at least the opening y_q of q under $\boldsymbol{y}$. The main bottleneck in implementing Line 1 is data replication, which is a standard procedure that can be implemented in $O(\log_s n)$ rounds using $O(dn^{1+\varepsilon})$ total memory (for storing the multiset Q of size $O(n^{1+\varepsilon})$); see, e.g., [NS22, Lemma 1].

The MPC implementation of Line 2 is somewhat involved, and we summarize its guarantee in the following claim.

Claim 4.10 (MPC Implementation of Line 2 of Algorithm 4)**.** *There exists an MPC algorithm that, given the fractional solution $\boldsymbol{y}'=\frac{1}{n^{\varepsilon}}\cdot\mathbf{1}^{Q}$ computed in Line 1 of Algorithm 4, with probability $1-\frac{1}{\mathrm{poly}(n)}$ computes for every $p\in P$ a size-n^{ε} submultiset $Q(p)$ of Q such that the assignment $\boldsymbol{x}'\in\{0,1/n^{\varepsilon}\}^{P\times Q}$ defined below satisfies $\boldsymbol{x}'\sim\boldsymbol{y}'$ and $\sigma_z(\boldsymbol{x}')\leq O(\varepsilon^{-z})\cdot\mathrm{cost}_z(\boldsymbol{y}')$.*

$$\forall p\in P, q\in Q,\quad x'_{p,q}=\begin{cases}1/n^{\varepsilon} & q\in Q(p)\\ 0 & q\notin Q(p)\end{cases}. \tag{22}$$

With local memory $s\geq\mathrm{poly}(d\log(n\Delta))$ per machine, the algorithm runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)}\cdot n^{1+O(\varepsilon)}\cdot\mathrm{poly}(d\log(n\Delta))$ total memory.

We defer the proof of Claim 4.10 to later and now proceed to implement the remaining lines of Algorithm 4 in the MPC model, whose implementation is relatively straightforward.

Lines 3 and 4 can be trivially implemented in one MPC round. We finally describe how to implement Lines 5 and 6 using the multisets $\{Q(p)\}_{p\in P}$ computed in Claim 4.10, as follows:

- Line 5 is an aggregation task that computes, for every $p \in P$, the minimum α_q over all $q \in Q(p)$. Hence, we can use standard sorting and converge-cast to aggregate these values for all $p \in P$ simultaneously (see, e.g., [CKP26, Lemma 5.10] for a similar aggregation task), in $O(\log_s n)$ rounds using $O(n^{1+\varepsilon} \cdot \text{poly}(d \log(n\Delta)))$ total memory.
- Line 6 can be implemented by an aggregation task followed by a simple lookup. First, for every $q \in \bigcup_{p\in P} Q(p)$, we compute the minimum β_p over all $p \in P$ with $q \in Q(p)$, which determines $g(q) := \text{argmin}_{p\in P: x'_{p,q}>0} \beta_p$. Then, for every $p \in P$, we check whether $g(f(p)) = p$ via sorting and, if so, add $f(p)$ to C. To ensure that C is a set, we perform a final sorting step to remove duplicate points. Hence, Line 6 can be implemented in $O(\log_s n)$ rounds using $O(n^{1+\varepsilon} \cdot \text{poly}(d \log(n\Delta)))$ total memory.

This completes our discussion of the MPC implementation of Algorithm 4. Overall, the implementation runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \text{poly}(d \log(n\Delta))$ total memory. The operations used above are either deterministic or satisfy the high-probability guarantees of their corresponding primitives. By a union bound, the entire implementation successfully implements Algorithm 4 with probability at least $1 - 1/\text{poly}(n)$. This finishes the proof of Claim 4.9. □

It remains only to prove Claim 4.10, which we do below.

Proof of Claim 4.10. For simplicity, we only consider the case where Q is a set rather than a multiset; the same argument can be easily generalized to the multiset case, for example, by distinguishing colocated points using their IDs.

Algorithm 7: A more detailed implementation of Line 2 of Algorithm 4

```
1 B ← 2n^{2ε} and L ← O(log n) (with a sufficiently large hidden constant)
2 for every ℓ ∈ [L] do
3 |   independently sample v_q^(ℓ) uniformly from [B] for every q ∈ Q
4 |   for every i ∈ [B], let Q_i^(ℓ) ← {q ∈ Q : v_q^(ℓ) = i}
5 for every (ℓ, i) ∈ [L] × [B] such that Q_i^(ℓ) ≠ ∅ do
6 |   compute, for every p ∈ P, an O(ε^{-1})-ANN q̂_i^(ℓ)(p) ∈ Q_i^(ℓ) in Q_i^(ℓ)
                                  // the algorithm fails if fewer than n^ε points exist
7 for every p ∈ P, let Q(p) ← the n^ε points in {q̂_i^(ℓ)(p) : ℓ ∈ [L], i ∈ [B], Q_i^(ℓ) ≠ ∅} closest to p
8 return {Q(p)}_{p∈P}
```

We present in Algorithm 7 a more detailed implementation of Line 2 of Algorithm 4. We start by describing the MPC implementation of Algorithm 7:

- We first implement the first **for** loop (Lines 2 to 4), which aims to compute $\{Q_i^{(\ell)}\}_{\ell\in[L],i\in[B]}$. To this end, for every $(\ell, i) \in [L] \times [B]$, we allocate a set $\mathcal{M}_i^{(\ell)}$ of machines responsible for storing $Q_i^{(\ell)}$. We replicate Q into L copies. For every point $q \in Q$ in the ℓ-th copy, we independently sample $v_q^{(\ell)} \in [B]$ locally and route q to $\mathcal{M}_i^{(\ell)}$, where $i = v_q^{(\ell)}$. After this step, each $\mathcal{M}_i^{(\ell)}$ stores

exactly $Q_i^{(\ell)}$. This step runs in $O(\log_s n)$ rounds, where the main bottleneck is replicating Q into $L = O(\log n)$ copies. Moreover, since each $\mathcal{M}_i^{(\ell)}$ stores at most $|Q| = O(n^{1+\varepsilon})$ points and there are $LB = O(n^{2\varepsilon} \log n)$ such sets of machines, this implementation of the first **for** loop uses $n^{1+3\varepsilon} \cdot \text{poly}(d \log(n\Delta))$ total memory.

- To implement the second **for** loop (Lines 5 to 6), we first replicate the query set P to every nonempty machine set $\mathcal{M}_i^{(\ell)}$. We then run the MPC approximate nearest-neighbor algorithm from [CGJ+24, Section 3.1] on each $\mathcal{M}_i^{(\ell)}$ in parallel, taking $Q_i^{(\ell)}$ as the terminal set and P as the query set. This runs in $O(\log_s n)$ rounds. Since there are at most $LB = O(n^{2\varepsilon} \log n)$ machine sets, the total memory is that required by [CGJ+24, Section 3.1] multiplied by $O(n^{2\varepsilon} \log n)$, giving a total memory of $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \text{poly}(d \log(n\Delta))$.
- Finally, we construct $Q(p)$ for every $p \in P$. For every $p \in P$, $\ell \in [L]$, and $i \in [B]$ such that $Q_i^{(\ell)} \neq \emptyset$, we create a tuple $(p, \text{dist}(p, \widehat{q}_i^{(\ell)}(p)), \widehat{q}_i^{(\ell)}(p))$; there are at most $nLB = O(n^{1+2\varepsilon} \log n)$ such tuples. We first remove duplicate tuples and then sort the remaining tuples lexicographically. For every $p \in P$, we retain the first n^{ε} tuples associated with p and let $Q(p)$ consist of the corresponding points $\widehat{q}_i^{(\ell)}(p)$. These operations can be implemented in $O(\log_s n)$ rounds using $n^{1+2\varepsilon} \cdot \text{poly}(d \log(n\Delta))$ total memory.

Combining the above three steps, Algorithm 7 can be implemented in $O(\log_s n)$ rounds using $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \text{poly}(d \log(n\Delta))$ total memory.

It remains to prove the correctness of Algorithm 7. To this end, for every $p \in P$, let $Q^\star(p) \subseteq Q$ be the n^{ε} closest points in Q to p. Whenever all $Q(p)$ are well-defined, the definition of $\boldsymbol{x}'$ in (22) and the definition of $\boldsymbol{y}' = \frac{1}{n^{\varepsilon}} \cdot \mathbf{1}^Q$ imply that $\boldsymbol{x}' \sim \boldsymbol{y}'$ is feasible, and

$$\sigma_z(\boldsymbol{x}') = \frac{1}{n^{\varepsilon}} \sum_{p \in P} \sum_{q \in Q(p)} \text{dist}^z(p, q), \qquad \text{cost}_z(\boldsymbol{y}') = \frac{1}{n^{\varepsilon}} \sum_{p \in P} \sum_{q \in Q^\star(p)} \text{dist}^z(p, q). \tag{23}$$

Hence, it suffices to show that, for every fixed point $p \in P$, with probability $1 - \frac{1}{\text{poly}(n)}$, Line 7 finds n^{ε} points, so that $Q(p)$ is well-defined, and

$$\sum_{q \in Q(p)} \text{dist}^z(p, q) \;\leq\; O(\varepsilon^{-z}) \cdot \sum_{q \in Q^\star(p)} \text{dist}^z(p, q). \tag{24}$$

By a union bound over all $p \in P$, with probability $1 - \frac{1}{\text{poly}(n)}$, $Q(p)$ is well-defined and (24) holds simultaneously for all $p \in P$. Combining this with (23) yields $\sigma_z(\boldsymbol{x}') \leq O(\varepsilon^{-z}) \cdot \text{cost}_z(\boldsymbol{y}')$. For the remainder of the proof, we fix an arbitrary point $p \in P$. We first establish the following claim.

Claim 4.11. *With probability $1 - \frac{1}{\text{poly}(n)}$, there exists $\ell \in [L]$ such that all points in $Q^\star(p)$ are separated by the partition $\{Q_i^{(\ell)}\}_{i \in [B]}$.*

Proof. Fix an $\ell \in [L]$. For any two distinct points $q, q' \in Q^\star(p)$, the values $v_q^{(\ell)}$ and $v_{q'}^{(\ell)}$ are sampled independently and uniformly from $[B]$. Hence, $\Pr[v_q^{(\ell)} = v_{q'}^{(\ell)}] = 1/B$. Since $|Q^\star(p)| = n^{\varepsilon}$ and $B = 2n^{2\varepsilon}$, a union bound over all pairs of distinct points gives

$$\Pr\left[\exists q \neq q' \in Q^\star(p), v_q^{(\ell)} = v_{q'}^{(\ell)}\right] \;\leq\; \binom{n^{\varepsilon}}{2} \cdot \frac{1}{B} \;\leq\; \frac{1}{4}.$$

Thus, with probability at least 3/4, all points in $Q^\star(p)$ lie in distinct parts of the partition $\{Q_i^{(\ell)}\}_{i\in[B]}$. Since the random partitions are independent across $\ell \in [L]$ and $L = O(\log n)$ is sufficiently large, with probability $1 - \frac{1}{\mathrm{poly}(n)}$, there exists $\ell \in [L]$ such that all points in $Q^\star(p)$ lie in distinct parts of the partition $\{Q_i^{(\ell)}\}_{i\in[B]}$. □

Condition on the high-probability event in Claim 4.11 and fix any such $\ell \in [L]$. For every $q \in Q^\star(p)$, let $i_q \in [B]$ be the unique index such that $q \in Q_{i_q}^{(\ell)}$. The corresponding sets $Q_{i_q}^{(\ell)}$ are pairwise disjoint, so the ANN points $\widehat{q}_{i_q}^{(\ell)}(p)$ computed in Line 6 are distinct, and hence $Q(p)$ is well-defined. For every $q \in Q^\star(p)$, by Line 6, we have

$$\mathrm{dist}^z\left(p, \widehat{q}_{i_q}^{(\ell)}(p)\right) \;\le\; O(\varepsilon^{-z}) \cdot \mathrm{dist}^z\left(p, Q_{i_q}^{(\ell)}\right) \;\le\; O(\varepsilon^{-z}) \cdot \mathrm{dist}^z(p,q),$$

where the second inequality follows from $q \in Q_{i_q}^{(\ell)}$. Summing over all $q \in Q^\star(p)$ gives

$$\sum_{q\in Q^\star(p)} \mathrm{dist}^z\left(p, \widehat{q}_{i_q}^{(\ell)}(p)\right) \;\le\; O(\varepsilon^{-z}) \cdot \sum_{q\in Q^\star(p)} \mathrm{dist}^z(p,q).$$

Finally, by the definition of $Q(p)$ in Line 7, it follows that

$$\sum_{q\in Q(p)} \mathrm{dist}^z(p,q) \;\le\; \sum_{q\in Q^\star(p)} \mathrm{dist}^z\left(p, \widehat{q}_{i_q}^{(\ell)}(p)\right).$$

Combining the above two inequalities proves (24), thereby completing the correctness proof of Algorithm 7. This finishes the proof of Claim 4.10. □

MPC Implementation of Algorithm 5. We then discuss the MPC implementation of Algorithm 5 from the second step in Section 4.2.1. For this, we need the following claim concerning the MPC implementation of the bi-point rounding procedure used in Line 6. The claim follows almost directly from [CKP26], with the only modification being a generalization of their $z = 2$ case to arbitrary $z \ge 1$; for completeness, we provide a proof in Appendix C.2.

Claim 4.12 (Bi-point Rounding [CKP26])**.** *There exists an MPC algorithm that, given a bi-point solution (C_1, C_2, a), computes a random solution $C \subseteq C_1 \cup C_2 \subseteq F$ such that*

$$|C| \le a|C_1| + (1-a)|C_2| + 1 \quad \textit{and} \quad \mathbb{E}[\mathrm{cost}_z(C)] \le \varepsilon^{-O(z)} \cdot \big(a\mathrm{cost}_z(C_1) + (1-a)\mathrm{cost}_z(C_2)\big).$$

If the local memory per machine is $s \ge \mathrm{poly}(d\log(n\Delta))$, the algorithm runs in $O(\log_s n)$ rounds and uses total memory $2^{O(\varepsilon d)} \cdot n \cdot \mathrm{poly}(d\log(n\Delta))$.

Proof. The proof can be found in Appendix C.2. □

Claim 4.13 (MPC Implementation of Algorithm 5)**.** *There exists an MPC algorithm that successfully implements Algorithm 5 with probability at least $1 - \frac{1}{\mathrm{poly}(n)}$; specifically, conditioned on successful implementation, the guarantee in Claim 4.7 holds.*

With local memory $s \ge \mathrm{poly}(d\log(n\Delta))$ per machine, the algorithm runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \mathrm{poly}(d\log(n\Delta))$ total memory.

Proof. Line 1 can be implemented straightforwardly by running $h = n^{2\varepsilon}$ independent instances of the MPC implementation of Algorithm 4 in Claim 4.9 in parallel. By Claim 4.9, this takes $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \text{poly}(d \log(n\Delta))$ total memory.

To implement Line 2, we note that, for a fixed $i \in [h]$, it suffices to compute, for every point $p \in P$, an approximate-nearest center $c_i(p) \in C_i$ satisfying $\text{dist}(p, c_i(p)) \leq O(\varepsilon^{-1}) \cdot \text{dist}(p, C_i)$, and then aggregate the power-z distances $\text{dist}^z(p, c_i(p))$. The computation of $c_i(p)$ for all $p \in P$ is the main bottleneck. This can be achieved using the MPC approximate nearest-neighbor algorithm from [CGJ+24, Section 3.1], which runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n \cdot \text{poly}(d \log(n\Delta))$ total memory. Since we can run this computation for all $i \in [h]$ in parallel, Line 2 can be implemented in $O(\log_s n)$ rounds using $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \text{poly}(d \log(n\Delta))$ total memory.

To implement Lines 3 to 5, we enumerate all pairs $(i, j) \in [h]^2$. For each pair, we compute an optimal value $a \in [0, 1]$ minimizing $a\widehat{\text{cost}}_z(C_i) + (1-a)\widehat{\text{cost}}_z(C_j)$ subject to $a|C_i| + (1-a)|C_j| \leq k + 2 \cdot |F|/n^{\varepsilon}$, whenever such a feasible value exists. We then select, among all resulting bi-point solutions, the one with the minimum estimated cost $a\widehat{\text{cost}}_z(C_i) + (1-a)\widehat{\text{cost}}_z(C_j)$; if no such solution exists, we simply pick an arbitrary bi-point solution. In the MPC model, this can be implemented by distributing the records $(|C_i|, \widehat{\text{cost}}_z(C_i), |C_j|, \widehat{\text{cost}}_z(C_j))$ so that the value a for each pair (i, j) can be computed locally, and then selecting the resulting bi-point solution (C_i, C_j, a) with the minimum value of $a\widehat{\text{cost}}_z(C_i) + (1-a)\widehat{\text{cost}}_z(C_j)$. This requires only standard MPC routing and aggregation, and takes $O(\log_s n)$ rounds using $O(h^2)$ additional total memory.

Finally, we run the bi-point rounding algorithm from Claim 4.12 on the selected (C_i, C_j, a).

The above implementation runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} n^{1+O(\varepsilon)} \text{poly}(d \log(n\Delta))$ total memory. Moreover, since all operations, including the MPC implementation of Algorithm 4, sorting, and aggregation, are either deterministic or succeed with probability at least $1 - 1/\text{poly}(n)$, a union bound implies that the entire implementation successfully implements Algorithm 5 with probability at least $1 - 1/\text{poly}(n)$. □

MPC Implementation of Algorithm 6. Finally, we discuss the MPC implementation of Algorithm 6 from the third step in Section 4.2.1. Since this is the final algorithm used to establish Lemma 4.3, its MPC implementation, together with the correctness guarantee in Claim 4.8, completes the proof of Lemma 4.3.

Claim 4.14 (MPC Implementation of Algorithm 6)**.** *There exists an MPC algorithm that successfully implements Algorithm 6 with probability at least* $1 - \frac{1}{\text{poly}(n)}$*; specifically, conditioned on successful implementation, the guarantee in Claim 4.8 holds.*

With local memory $s \geq \text{poly}(d \log(n\Delta))$ *per machine, the algorithm runs in* $O(\log_s n)$ *rounds and uses* $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \text{poly}(d \log(n\Delta))$ *total memory.*

Proof. The computation of $\widehat{\text{cost}}_z(C)$ for all solutions C returned by the instances of Algorithm 5 can be implemented in the same way as Line 2 of Algorithm 5. Then, we compute every $|C|$ by aggregation, discard all solutions with size larger than $k + 1 + 2 \cdot |F|/n^{\varepsilon}$, and select the remaining solution with minimum $\widehat{\text{cost}}_z(C)$ by aggregating over the $O(\log n)$ runs; if no solution remains, we return an arbitrary subset of F. Therefore, the main bottleneck is running $O(\log n)$ independent instances of Algorithm 5 in parallel, which, by Claim 4.13, takes $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \text{poly}(d \log(n\Delta))$ total memory. The subroutine calls above are either deterministic or succeed with probability at least $1 - \frac{1}{\text{poly}(n)}$. By a union bound, the entire implementation successfully implements Algorithm 6 with probability at least $1 - 1/\text{poly}(n)$. This finishes the proof. □

5 $O(1)$-Approximation with Exactly k Centers

In this section, we design an MPC algorithm for Euclidean (k,z)-Clustering that achieves a true $O(1)$-approximation using *exactly* k centers; this is the main result of this paper, and is presented in Theorem 5.1 below.

Theorem 5.1 (True Approximation)**.** *For any constant $\varepsilon \in (0,1)$ and integer $z \geq 1$, there is a randomized MPC algorithm for* Euclidean (k,z)-Clustering *that,*

- *takes as input a dataset $P \subseteq \mathbb{R}^d$, a candidate center set $F \subseteq \mathbb{R}^d$, with $n := |P| + |F|$ denoting the total number of points, an integer $1 \leq k \leq |F|$, and an upper bound $\Delta \geq \Delta(P \cup F)$ on the aspect ratio;*
- *computes, with high probability, a center set $C \subseteq F$ satisfying*

$$|C| \;=\; k, \quad \mathrm{cost}_z(P,C) \;\leq\; 2^{O(z^2/\varepsilon)}\varepsilon^{-O(z/\varepsilon)} \cdot \mathrm{OPT}^{\mathrm{cl}}_z(P,F,k).$$

With local memory $s \geq 2^{\Omega(1/\varepsilon)} \cdot \mathrm{poly}(d\log(n\Delta))$ per machine, the algorithm runs in $O(\varepsilon^{-1}\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \mathrm{poly}(\varepsilon^{-1} d\log(n\Delta))$ total memory.

Before we prove Theorem 5.1, we discuss how it implies Theorem 1.1: For an input n-point dataset $P \subseteq \mathbb{R}^d$ with $d = O(\log n)$, we set $F = P$ and take $z = 2$ for k-Means and $z = 1$ for k-Median. Here, we note that k-Median and k-Means allow centers to be chosen from the entire $\mathbb{R}^d$, but restricting the candidate center set to $F = P$ incurs only an additional 2^z factor in the approximation ratio, which we can afford. As for the aspect ratio, we provide a preprocessing algorithm that reduces it to $\mathrm{poly}(n)$ (without knowing the original aspect ratio) while incurring only an additional $2^{O(z)}$ factor in the approximation ratio; see Appendix A. With these settings, for both k-Means and k-Median, the approximation ratio is $\varepsilon^{-O(1/\varepsilon)}$, and the total memory is $\widetilde{O}(n^{1+O(\varepsilon)})$, which becomes $\widetilde{O}(n^{1+\varepsilon})$ after appropriately rescaling ε in Theorem 5.1. Moreover, since $d = O(\log n)$ and the reduced aspect ratio is $\mathrm{poly}(n)$, the local memory $s = n^{\varepsilon}$ satisfies $s \geq 2^{\Omega(1/\varepsilon)} \cdot \mathrm{poly}(d\log(n\Delta))$, as required by Theorem 5.1. The round complexity is thus $O(\varepsilon^{-1}\log_s n) = O(\varepsilon^{-2})$. Theorem 1.1 then follows.

The remainder of this section is devoted to proving Theorem 5.1. We first describe a kernelization subroutine from [BCF+26] designed for a special *restricted clustering* problem, and then present our algorithm, which follows a novel *iterative kernelization* scheme.

One Kernelization Step. We use the technique from [BCF+26], which intuitively yields a kernelization for the (r,z)-Restricted Clustering problem. In this problem, the input consists of a dataset $P \subseteq \mathbb{R}^d$, a candidate center set $F \subseteq \mathbb{R}^d$, and an integer $1 \leq r \leq |F| - 1$ specifying the number of centers to remove from F (rather than the number k of centers to select as in (k,z)-Clustering). The goal is to find a set $R \subseteq F$ with $|R| = r$ that minimizes $\mathrm{cost}_z(P, F \setminus R)$. Observe that this problem is equivalent to (k,z)-Clustering when $k + r = |F|$. Therefore, the optimal objective value for (r,z)-Restricted Clustering is exactly $\mathrm{OPT}^{\mathrm{cl}}_z(P,F,|F|-r)$.

For (r,z)-Restricted Clustering, [BCF+26] identifies a size-$O(r)$ subset of F and shows that it suffices to find the set R within this subset, thereby yielding a kernelization step that reduces the size of the candidate center set. This also yields a kernelization step for (k,z)-Clustering that reduces the size of the candidate center set to $O(|F| - k)$ and forms a core step of our algorithm for Theorem 5.1, which we discuss later.

We state the kernelization subroutine of [BCF+26] in Algorithm 8 using our notation, where $(P^{\mathrm{in}}, F^{\mathrm{in}})$ and $(P^{\mathrm{out}}, F^{\mathrm{out}})$ denote the pairs of dataset and candidate center set before and after the

kernelization step, respectively, and summarize its guarantee in Claim 5.2. The proof of Claim 5.2 follows almost directly from the analysis in [BCF+26, Section 11], but generalizes their $z = 2$ case to arbitrary $z \geq 1$; for completeness, we provide the proof in Appendix D. Moreover, we show that their algorithm can be efficiently implemented in the MPC model, as stated in Claim 5.3.

Algorithm 8: Kernelization for (r, z)-Restricted Clustering on $(P^{\text{in}}, F^{\text{in}})$

1 compute, for every $c \in F^{\text{in}}$, a value $\tau(c)$ such that
$\text{dist}(c, F^{\text{in}} \setminus \{c\}) \leq \tau(c) \leq O(\varepsilon^{-1}) \cdot \text{dist}(c, F^{\text{in}} \setminus \{c\})$
2 compute, for every $p \in P^{\text{in}}$, an $O(\varepsilon^{-1})$-ANN $\pi(p) \in F^{\text{in}}$
3 for every $c \in F^{\text{in}}$, $w(c) \leftarrow |\{p \in P^{\text{in}} : \pi(p) = c\}|$
4 sort the centers in F^{in} as $c_1, \ldots, c_{|F^{\text{in}}|}$ such that
$w(c_1) \cdot \tau^z(c_1) \leq w(c_2) \cdot \tau^z(c_2) \leq \cdots \leq w(c_{|F^{\text{in}}|}) \cdot \tau^z(c_{|F^{\text{in}}|})$
5 $T_1 \leftarrow$ the set $\{c_i\}_{i \in [\min\{6r, |F^{\text{in}}|\}]}$
6 $T_2 \leftarrow$ the set $\{s_i\}_{i \in [\min\{6r, |F^{\text{in}}|\}]}$, where each s_i is an $O(\varepsilon^{-1})$-ANN of c_i in $F^{\text{in}} \setminus T_1$
// $T_2 \leftarrow \emptyset$ `if` $T_1 = F^{\text{in}}$
7 **return** $(P^{\text{out}}, F^{\text{out}})$, where $F^{\text{out}} \leftarrow T_1 \cup T_2$ and $P^{\text{out}} \leftarrow \{p \in P^{\text{in}} : \pi(p) \in F^{\text{out}}\}$

Claim 5.2 (Guarantees of Algorithm 8)**.** *The output $(P^{\text{out}}, F^{\text{out}})$ of Algorithm 8 satisfies the following properties.*

- $F^{\text{out}} \subseteq F^{\text{in}}$ *and* $r + 1 \leq |F^{\text{out}}| \leq 12r$.
- *for every $R \subseteq F^{\text{out}} \subseteq F^{\text{in}}$ with $|R| = r$ that is an α-approximate solution to (r, z)-Restricted Clustering on $(P^{\text{out}}, F^{\text{out}})$ for some $\alpha \geq 1$, i.e.,*

$$\text{cost}_z(P^{\text{out}}, F^{\text{out}} \setminus R) \leq \alpha \cdot \text{OPT}^{\text{cl}}_z(P^{\text{out}}, F^{\text{out}}, |F^{\text{out}}| - r),$$

R is an $(\varepsilon^{-O(z)} \cdot \alpha)$-approximate solution to (r, z)-Restricted Clustering on $(P^{\text{in}}, F^{\text{in}})$, i.e.,

$$\text{cost}_z(P^{\text{in}}, F^{\text{in}} \setminus R) \leq \varepsilon^{-O(z)} \cdot \alpha \cdot \text{OPT}^{\text{cl}}_z(P^{\text{in}}, F^{\text{in}}, |F^{\text{in}}| - r).$$

Proof. The proof can be found in Appendix D. □

Claim 5.3 (MPC Implementation of Algorithm 8)**.** *There is an MPC implementation of Algorithm 8 that, with local memory $s \geq \text{poly}(d \log(n\Delta))$ per machine, runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n \cdot \text{poly}(d \log(n\Delta))$ total memory. Here, $n := |P^{\text{in}}| + |F^{\text{in}}|$ denotes the total number of points and $\Delta \geq \Delta(P^{\text{in}} \cup F^{\text{in}})$ is an upper bound on the aspect ratio.*

Proof. We describe how to implement each step of Algorithm 8 in the MPC model: Line 1 can be implemented using the MPC approximate nearest-neighbor algorithm from [CGJ+24, Section 3.1]. Here, since we need an $O(\varepsilon^{-1})$-ANN of c in $F^{\text{in}} \setminus \{c\}$, excluding c itself, we slightly modify the composable function $f(Y)$ used in [CGJ+24, Section 3.1] to return the $\min\{2, |Y|\}$ points in Y with the smallest IDs, or $\emptyset$ if $Y = \emptyset$; then the same algorithm and argument in [CGJ+24, Section 3.1] can be applied to implement Line 1. Similarly, Line 2 can be implemented by applying the same MPC approximate nearest-neighbor algorithm with F^{in} as the terminal set and P^{in} as the query set. Given $\{\pi(p)\}_{p \in P^{\text{in}}}$, Line 3 can be implemented by sorting the pairs $\{(\pi(p), p)\}_{p \in P^{\text{in}}}$ according to their first coordinates and, in parallel, aggregating the number of points assigned to each center using a standard MPC converge-cast operation (see, e.g., [CKP26, Lemma 5.10]). Line 4 can then

be implemented via standard MPC sorting. The set T_1 in Line 5 is obtained by retaining the first $\min\{6r, |F^{\text{in}}|\}$ centers in this ordering. To implement Line 6, if $T_1 \neq F^{\text{in}}$, we again invoke the MPC approximate nearest-neighbor algorithm from [CGJ+24, Section 3.1], this time taking $F^{\text{in}} \setminus T_1$ as the terminal set and T_1 as the query set. This computes all centers s_i simultaneously in $O(\log_s n)$ rounds. Finally, $F^{\text{out}} = T_1 \cup T_2$ can be constructed via sorting and duplicate removal. Given F^{out}, the new dataset $P^{\text{out}} = \{p \in P^{\text{in}} : \pi(p) \in F^{\text{out}}\}$ can be computed straightforwardly.

The bottleneck consists of a constant number of calls to the MPC approximate nearest-neighbor algorithm from [CGJ+24, Section 3.1]; all other steps use standard MPC sorting, routing, and converge-cast primitives. Therefore, the entire implementation runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n \cdot \text{poly}(d \log(n\Delta))$ total memory. □

Algorithm for Theorem 5.1: Iterative Kernelization. We now present our iterative kernelization for Theorem 5.1 in Algorithm 9. Specifically, starting from the tuple $(P_0, F_0, k_0) = (P, F, k)$ given as input to Theorem 5.1, we iteratively construct a sequence of tuples $(P_\ell, F_\ell, k_\ell)_{\ell \in [0:L]}$, where the size of the candidate center set decreases rapidly with ℓ. In each iteration $\ell \in [0 : L-1]$, we construct $(P_{\ell+1}, F_{\ell+1}, k_{\ell+1})$ from the current tuple (P_ℓ, F_ℓ, k_ℓ) via the following two steps:

1. We first compute a preliminary bicriteria solution $C_\ell^{\text{pre}} \subseteq F_\ell$ of size at most $k_\ell + 1 + |F_\ell|/n^\varepsilon$; in the MPC model, this is provided by Theorem 4.1.

2. Then, we apply Algorithm 8 with $r = |C_\ell^{\text{pre}}| - k_\ell$ and $(P^{\text{in}}, F^{\text{in}}) = (P_\ell, C_\ell^{\text{pre}})$ to obtain $(P^{\text{out}}, F^{\text{out}})$, and set $(P_{\ell+1}, F_{\ell+1}, k_{\ell+1}) = (P^{\text{out}}, F^{\text{out}}, |F^{\text{out}}| - r)$. By Claim 5.2, the new candidate center set satisfies

$$|F_{\ell+1}| \;=\; |F^{\text{out}}| \;\le\; 12(|C_\ell^{\text{pre}}| - k_\ell) \;\le\; 12(1 + |F_\ell|/n^\varepsilon).$$

We note that we do not directly apply Algorithm 8 to reduce (P_ℓ, F_ℓ, k_ℓ) (i.e., with $r = |F_\ell| - k_\ell$ and $(P^{\text{in}}, F^{\text{in}}) = (P_\ell, F_\ell)$), since this would only guarantee a candidate center set of size at most $12(|F_\ell| - k_\ell)$, which may not yield any meaningful reduction when k_ℓ is small.

Finally, we compute an optimal solution C_L to (k, z)-Clustering on the last tuple (P_L, F_L, k_L), and then iteratively lift the solution back through the sequence of tuples in the way described in Claim 5.2, eventually obtaining a solution C_0 to the original input $(P_0, F_0, k_0) = (P, F, k)$.

We are now ready to prove Theorem 5.1 by analyzing the algorithm in Algorithm 9.

Proof of Theorem 5.1. We analyze the correctness of Algorithm 9 and its MPC implementation separately below.

Correctness of Algorithm 9. The solution returned in Line 1 is optimal, so assume this case does not apply. We prove the following stronger guarantee by backward induction on ℓ:

$$\forall \ell \in [0:L], \qquad |C_\ell| \;=\; k_\ell, \quad \text{cost}_z(P_\ell, C_\ell) \;\le\; \left(2^{O(z^2)}\varepsilon^{-O(z)}\right)^{L-\ell+1} \cdot \text{OPT}_z^{\text{cl}}(P_\ell, F_\ell, k_\ell). \tag{25}$$

The base case is $\ell = L$. By Line 8, we directly have

$$|C_\ell| \;=\; k_\ell, \quad \text{cost}_z(P_\ell, C_\ell) \;=\; \text{OPT}_z^{\text{cl}}(P_\ell, F_\ell, k_\ell) \;\le\; \left(2^{O(z^2)}\varepsilon^{-O(z)}\right)^{L-\ell+1} \cdot \text{OPT}_z^{\text{cl}}(P_\ell, F_\ell, k_\ell).$$

This proves (25) for the base case $\ell = L$.

For the inductive step, suppose that (25) holds for $\ell + 1 \in [L]$. We show below that (25) also holds for ℓ. We first have the following fact.

Algorithm 9: Clustering via Iterative Kernelization on Instance (P, F, k)

1 **if** $|F| = k$ **then return** F
2 let $(P_0, F_0, k_0) \leftarrow (P, F, k)$, and $L \leftarrow \lceil 4/\varepsilon \rceil$
3 **for** $\ell \leftarrow 0$ **to** $L - 1$ **do**
4 compute a preliminary bicriteria solution $C_\ell^{\mathrm{pre}} \subseteq F_\ell$ such that
(i) $k_\ell + 1 \le |C_\ell^{\mathrm{pre}}| \le k_\ell + 1 + |F_\ell|/n^\varepsilon$ and
(ii) $\mathrm{cost}_z(P_\ell, C_\ell^{\mathrm{pre}}) \le 2^{O(z^2)} \cdot \varepsilon^{-O(z)} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, F_\ell, k_\ell)$
5 $r_\ell \leftarrow |C_\ell^{\mathrm{pre}}| - k_\ell$ // $1 \le r_\ell \le |C_\ell^{\mathrm{pre}}| - 1$
6 run Algorithm 8 with $r = r_\ell$ and $(P^{\mathrm{in}}, F^{\mathrm{in}}) = (P_\ell, C_\ell^{\mathrm{pre}})$ to obtain $(P^{\mathrm{out}}, F^{\mathrm{out}})$
7 $(P_{\ell+1}, F_{\ell+1}, k_{\ell+1}) \leftarrow (P^{\mathrm{out}}, F^{\mathrm{out}}, |F^{\mathrm{out}}| - r_\ell)$
// `Algorithm 8 ensures that` $F_{\ell+1} \subseteq C_\ell^{\mathrm{pre}} \subseteq F_\ell$
8 compute a solution $C_L \subseteq F_L$ of size k_L such that $\mathrm{cost}_z(P_L, C_L) = \mathrm{OPT}_z^{\mathrm{cl}}(P_L, F_L, k_L)$
9 **for** $\ell \leftarrow L - 1$ **downto** 0 **do**
10 $C_\ell \leftarrow C_\ell^{\mathrm{pre}} \setminus (F_{\ell+1} \setminus C_{\ell+1}) \subseteq F_\ell$
11 **return** $C \leftarrow C_0$

Fact 5.4. *$C_{\ell+1} \subseteq F_{\ell+1} \subseteq C_\ell^{\mathrm{pre}} \subseteq F_\ell$, and $r_\ell = |C_\ell^{\mathrm{pre}}| - k_\ell = |F_{\ell+1}| - k_{\ell+1}$.*

Proof. The inclusion $C_{\ell+1} \subseteq F_{\ell+1}$ follows by considering two cases. If $\ell + 1 = L$, it follows directly from Line 8. Otherwise, it follows from $C_{\ell+1} \subseteq C_{\ell+1}^{\mathrm{pre}}$ (Line 10) and $C_{\ell+1}^{\mathrm{pre}} \subseteq F_{\ell+1}$ (Line 4). The inclusion $F_{\ell+1} \subseteq C_\ell^{\mathrm{pre}}$ follows from the fact that $(P_{\ell+1}, F_{\ell+1}) = (P^{\mathrm{out}}, F^{\mathrm{out}})$ is the output of Algorithm 8 with $r = r_\ell$ and $(P^{\mathrm{in}}, F^{\mathrm{in}}) = (P_\ell, C_\ell^{\mathrm{pre}})$ (Line 6), together with the guarantee of Algorithm 8 stated in Claim 5.2. The inclusion $C_\ell^{\mathrm{pre}} \subseteq F_\ell$ follows directly from Line 4. Finally, the equation $r_\ell = |C_\ell^{\mathrm{pre}}| - k_\ell = |F_{\ell+1}| - k_{\ell+1}$ follows directly from Lines 5 and 7. □

The size guarantee $|C_\ell| = k_\ell$ follows directly from Fact 5.4, the definition $C_\ell = C_\ell^{\mathrm{pre}} \setminus (F_{\ell+1} \setminus C_{\ell+1})$ in Line 10, and (25) for $\ell + 1$, as follows:

$$|C_\ell| \;=\; |C_\ell^{\mathrm{pre}}| - (|F_{\ell+1}| - |C_{\ell+1}|) \;=\; |C_\ell^{\mathrm{pre}}| - (|F_{\ell+1}| - k_{\ell+1}) \;=\; k_\ell.$$

We then prove the cost guarantee of C_ℓ, i.e., $\mathrm{cost}_z(P_\ell, C_\ell) \le \left(2^{O(z^2)}\varepsilon^{-O(z)}\right)^{L-\ell+1} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, F_\ell, k_\ell)$. Let $R := F_{\ell+1} \setminus C_{\ell+1}$. We have the following facts:

- $|R| = |F_{\ell+1}| - |C_{\ell+1}| = |F_{\ell+1}| - k_{\ell+1} = |C_\ell^{\mathrm{pre}}| - k_\ell = r_\ell$ by Fact 5.4 and (25) for $\ell + 1$.
- $\mathrm{cost}_z(P_{\ell+1}, F_{\ell+1} \setminus R) \le \left(2^{O(z^2)}\varepsilon^{-O(z)}\right)^{L-\ell} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P_{\ell+1}, F_{\ell+1}, k_{\ell+1})$ by (25) for $\ell + 1$.
- $(P_{\ell+1}, F_{\ell+1})$ is obtained by applying Algorithm 8 with $r = r_\ell$ and $(P^{\mathrm{in}}, F^{\mathrm{in}}) = (P_\ell, C_\ell^{\mathrm{pre}})$, as specified in Lines 6 and 7.

The first two facts together imply that R is a $\left(2^{O(z^2)}\varepsilon^{-O(z)}\right)^{L-\ell}$-approximate solution to (r_ℓ, z)-Restricted Clustering on $(P_{\ell+1}, F_{\ell+1})$, and the third fact allows us to apply Claim 5.2 and deduce that R is a $\left(\varepsilon^{-O(z)} \cdot \left(2^{O(z^2)}\varepsilon^{-O(z)}\right)^{L-\ell}\right)$-approximate solution to (r_ℓ, z)-Restricted Clustering on $(P_\ell, C_\ell^{\mathrm{pre}})$, i.e.,

$$\begin{aligned} \mathrm{cost}_z(P_\ell, C_\ell) \;&\le\; \varepsilon^{-O(z)} \cdot \left(2^{O(z^2)}\varepsilon^{-O(z)}\right)^{L-\ell} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, C_\ell^{\mathrm{pre}}, |C_\ell^{\mathrm{pre}}| - r_\ell) \\ &=\; \varepsilon^{-O(z)} \cdot \left(2^{O(z^2)}\varepsilon^{-O(z)}\right)^{L-\ell} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, C_\ell^{\mathrm{pre}}, k_\ell) \end{aligned} \tag{26}$$

Finally, we upper-bound $\mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, C_\ell^{\mathrm{pre}}, k_\ell)$ in terms of $\mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, F_\ell, k_\ell)$ as follows: Let $C_\ell^\star \subseteq F_\ell$ with size k_ℓ be such that $\mathrm{cost}_z(P_\ell, C_\ell^\star) = \mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, F_\ell, k_\ell)$. Let $C'_\ell := \{\mathrm{NN}(c, C_\ell^{\mathrm{pre}}) : c \in C_\ell^\star\} \subseteq C_\ell^{\mathrm{pre}}$. Then, $|C'_\ell| \le |C_\ell^\star| = k_\ell$ and therefore $\mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, C_\ell^{\mathrm{pre}}, k_\ell) \le \mathrm{cost}_z(P_\ell, C'_\ell)$. For every $p \in P_\ell$, let $c \in C_\ell^\star$ be a closest center to p. By the definition of C'_ℓ,

$$\begin{aligned}\mathrm{dist}(p, C'_\ell) \;&\le\; \mathrm{dist}\left(p, \mathrm{NN}(c, C_\ell^{\mathrm{pre}})\right) \;\le\; \mathrm{dist}(p,c) + \mathrm{dist}(c, C_\ell^{\mathrm{pre}}) \;\le\; 2\,\mathrm{dist}(p,c) + \mathrm{dist}(p, C_\ell^{\mathrm{pre}}) \\ &=\; 2\,\mathrm{dist}(p, C_\ell^\star) + \mathrm{dist}(p, C_\ell^{\mathrm{pre}})\end{aligned}$$

Hence, by the generalized triangle inequality (Fact 2.1),

$$\begin{aligned}\mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, C_\ell^{\mathrm{pre}}, k_\ell) \;&\le\; \mathrm{cost}_z(P_\ell, C'_\ell) \;\le\; 2^{O(z)} \cdot \left(\mathrm{cost}_z(P_\ell, C_\ell^\star) + \mathrm{cost}_z(P_\ell, C_\ell^{\mathrm{pre}})\right) \\ &\le\; 2^{O(z^2)} \cdot \varepsilon^{-O(z)} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, F_\ell, k_\ell),\end{aligned}$$

where the last inequality follows from $\mathrm{cost}_z(P_\ell, C_\ell^\star) = \mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, F_\ell, k_\ell)$ and $\mathrm{cost}_z(P_\ell, C_\ell^{\mathrm{pre}}) \le 2^{O(z^2)} \cdot \varepsilon^{-O(z)} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, F_\ell, k_\ell)$ by Line 4. Plugging this bound into (26), we obtain

$$\mathrm{cost}_z(P_\ell, C_\ell) \;\le\; \left(2^{O(z^2)} \varepsilon^{-O(z)}\right)^{L-\ell+1} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P_\ell, F_\ell, k_\ell),$$

which proves (25) for ℓ. This completes the proof of (25) by the principle of mathematical induction.

Recall that $L = O(\varepsilon^{-1})$ by Line 2. Applying (25) with $\ell = 0$, the returned solution $C \leftarrow C_0$ satisfies

$$\mathrm{cost}_z(P, C) \;\le\; 2^{O(z^2/\varepsilon)} \cdot \varepsilon^{-O(z/\varepsilon)} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P, F, k).$$

This completes the analysis of the approximation ratio of Algorithm 9.

MPC Implementation of Algorithm 9. Throughout the discussion of the MPC implementation of Algorithm 9, we assume that $n^\varepsilon \ge 100$. Otherwise, both the dataset P and the candidate center set F have size at most $100^{1/\varepsilon} = 2^{O(1/\varepsilon)}$. Hence, the entire instance fits on a single machine with local memory $s \ge 2^{\Omega(1/\varepsilon)} \cdot \mathrm{poly}(d \log(n\Delta))$ and can therefore be solved directly.

We first establish the following claim, which upper-bounds the size of the candidate center set F_L in the final instance (P_L, F_L, k_L).

Claim 5.5. *For $L = \lceil 4/\varepsilon \rceil$ and $n^\varepsilon \ge 100$, it holds that $|F_L| \le O(1)$.*[5]

Proof. Consider a fixed $\ell \in [0 : L-1]$. By Line 4, we have $|C_\ell^{\mathrm{pre}}| \le k_\ell + 1 + |F_\ell|/n^\varepsilon$. Since the tuple $(P_{\ell+1}, F_{\ell+1}, k_{\ell+1})$ is constructed by running Algorithm 8 with $r = r_\ell = |C_\ell^{\mathrm{pre}}| - k_\ell$ and $(P^{\mathrm{in}}, F^{\mathrm{in}}) = (P_\ell, C_\ell^{\mathrm{pre}})$, it follows from Claim 5.2 that

$$|F_{\ell+1}| \;\le\; 12(|C_\ell^{\mathrm{pre}}| - k_\ell) \;\le\; 12\,(1 + |F_\ell|/n^\varepsilon)\,.$$

Define $\Phi(F) := |F| - \frac{12}{1-12/n^\varepsilon}$. The above inequality implies that

$$\begin{aligned}\Phi(F_{\ell+1}) \;&=\; |F_{\ell+1}| - \frac{12}{1-12/n^\varepsilon} \;\le\; 12 + \frac{12}{n^\varepsilon}|F_\ell| - \frac{12}{1-12/n^\varepsilon} \;=\; \frac{12}{n^\varepsilon}\left(|F_\ell| - \frac{12}{1-12/n^\varepsilon}\right) \\ &=\; \frac{12}{n^\varepsilon} \cdot \Phi(F_\ell).\end{aligned}$$

[5] We note that here only the final candidate center set F_L is guaranteed to have size $O(1)$, while the dataset P_L may still be large. However, this case can already be handled efficiently in MPC. Moreover, if desired, we can also further reduce the size of P_L to $O(1)$ using a coreset [HM04, HLW24].

Therefore,

$$\Phi(F_L) \;\le\; \left(\frac{12}{n^{\varepsilon}}\right)^{L} \Phi(F_0) \;\le\; \left(\frac{12}{n^{\varepsilon}}\right)^{L} n \;\le\; 1,$$

where the second inequality follows from $\Phi(F_0) \le |F_0| \le n$, and the third follows from $12 \le 100^{3/4} \le n^{3\varepsilon/4}$ and $L = \lceil 4/\varepsilon \rceil$, which give $(12/n^{\varepsilon})^{L} n \le n^{-\varepsilon L/4} n \le 1$. Finally, $|F_L| = \Phi(F_L) + \frac{12}{1-12/n^{\varepsilon}} \le O(1)$, which completes the proof of Claim 5.5. □

We now discuss the MPC implementation of Algorithm 9. Lines 1 and 2 are straightforward to implement. For the first **for** loop (Lines 3 to 7), Line 4 is implemented by applying Theorem 4.1, using the original input size n as the size upper bound. If necessary, we add arbitrary centers from F_ℓ to ensure $|C_\ell^{\mathrm{pre}}| \ge k_\ell + 1$ without increasing the cost. Line 6 is implemented using Claim 5.3. Hence, each iteration runs in $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \mathrm{poly}(d \log(n\Delta))$ total memory. Executing the $L = O(\varepsilon^{-1})$ iterations sequentially therefore takes $O(\varepsilon^{-1} \log_s n)$ rounds and increases the total memory by at most a factor of $O(\varepsilon^{-1})$, yielding a total memory bound of $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \mathrm{poly}(\varepsilon^{-1} d \log(n\Delta))$. To implement Line 8, we use the fact $|F_L| = O(1)$ from Claim 5.5 and enumerate all $O(1)$ subsets $C \subseteq F_L$ of size k_L. For each such C, we broadcast C to the machines storing P_L and compute $\mathrm{cost}_z(P_L, C)$ by locally evaluating $\mathrm{dist}^z(p, C)$ and aggregating these values using a standard MPC converge-cast. We then select a candidate of minimum cost. All candidate costs can be computed in parallel, so this step takes $O(\log_s n)$ rounds and uses $n \cdot \mathrm{poly}(d \log(n\Delta))$ total memory. Finally, for the second **for** loop (Lines 9 to 10), each update $C_\ell \leftarrow C_\ell^{\mathrm{pre}} \setminus (F_{\ell+1} \setminus C_{\ell+1})$ can be implemented using standard MPC sorting and set operations (see, e.g., [ASS+18]). Each iteration takes $O(\log_s n)$ rounds and uses $n \cdot \mathrm{poly}(d \log(n\Delta))$ total memory. Thus, executing the $L = O(\varepsilon^{-1})$ iterations sequentially takes $O(\varepsilon^{-1} \log_s n)$ rounds and uses $n \cdot \mathrm{poly}(\varepsilon^{-1} d \log(n\Delta))$ total memory.

Overall, the above MPC implementation for Algorithm 9 runs in $O(\varepsilon^{-1} \log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n^{1+O(\varepsilon)} \cdot \mathrm{poly}(\varepsilon^{-1} d \log(n\Delta))$ total memory. It remains to analyze the success probability, for which it suffices to consider the $L = O(\varepsilon^{-1})$ invocations of the algorithm in Theorem 4.1 in Line 4. Each invocation succeeds with high probability. Therefore, by a union bound over the $L = O(\varepsilon^{-1})$ invocations, which is a constant, all of them succeed with high probability as well. This finishes the proof of Theorem 5.1. □

Appendices

A Preprocessing

In this section, we discuss the MPC preprocessing for dimension reduction and aspect ratio reduction separately; both are fully scalable and round-efficient.

Dimension Reduction. To reduce the dimension to $O(\log n)$, we apply an MPC implementation of the JL transform due to [AAH+23], which is fully scalable and runs in $O(1)$ rounds.

Aspect Ratio Reduction. In the rest of this section, we present an MPC algorithm for reducing the aspect ratio of the input dataset P for (k, z)-Clustering when the candidate center set is $F = P$, i.e., when centers are chosen from P. For brevity, we write $\mathrm{OPT}_z^{\mathrm{cl}} = \mathrm{OPT}_z^{\mathrm{cl}}(P, P, k)$ and $\mathrm{cost}_z(C) = \mathrm{cost}_z(P, C)$. Specifically, given a parameter $1 \le \alpha \le \mathrm{poly}(n)$ and an n-point dataset

Algorithm 10: Reducing the aspect ratio

1 Compute a value η such that $\mathrm{OPT}_z^{\mathrm{cl}} \le \eta \le \beta \cdot \mathrm{OPT}_z^{\mathrm{cl}}$, where $\beta = n^{O(z)}$
2 Let $\alpha' = 2^{O(z)} \cdot \alpha$ be sufficiently large
3 Let $p_1, \dots, p_n$ be an arbitrary ordering of the points in P
4 Let $T \leftarrow \{p_{i,j} \in \mathbb{R} : p_i = (p_{i,1}, \dots, p_{i,d}) \in P, j \in [d]\}$, i.e. the set of all coordinate values
`// remove duplicates`
5 Sort T in non-decreasing order, denoting the resulting sequence by $t_1, \dots, t_m$ ($m = |T| \le nd$)
6 For every $i \in [m]$, compute $h(t_i) \leftarrow t_1 + \sum_{j=2}^{i} \min\{t_j - t_{j-1}, (\alpha' \cdot \eta)^{1/z}\}$
7 For every $p_i = (p_{i,1}, \dots, p_{i,d}) \in P$, compute $g(p_i) \leftarrow (h(p_{i,1}), \dots, h(p_{i,d}))$
8 For every $i \in [n]$, let $f(p_i) \leftarrow$ a nearest point to $g(p_i)$ in
$\mathrm{Net}_i := \{u \cdot \frac{\eta^{1/z}}{n\sqrt{d}\cdot\beta^{1/z}} + i \cdot \frac{\eta^{1/z}}{n^{100}\sqrt{d}\cdot\beta^{1/z}} : u \in \mathbb{Z}\}^d$
9 **return** $f(P)$

$P \subseteq \mathbb{R}^d$, the algorithm computes a new dataset of the form $f(P) = \{f(p) : p \in P\}$ for some injective mapping $f : P \to \mathbb{R}^d$, such that

(P1) the aspect ratio of $f(P)$ is at most $\mathrm{poly}(n)$, and

(P2) for any α-approximate solution $f(C) \subseteq f(P)$ to (k,z)-Clustering on $f(P)$, the set $C \subseteq P$ is a $2^{O(z)}\alpha$-approximate solution to (k,z)-Clustering on P.

The algorithm requires local memory $s \ge \mathrm{poly}(d \log n)$, runs in $O(\log_s n)$ rounds, and uses total memory $O(n \cdot \mathrm{poly}(d \log n))$. Moreover, the algorithm stores the mapping f as a set of pairs $\{(p, f(p)) : p \in P\}$, so that recovering C from a given $f(C) \subseteq f(P)$ can be done in $O(\log_s n)$ rounds (see e.g. [ASS+18]).

An offline version of our algorithm is presented in Algorithm 10. In the following, we first analyze the correctness of Algorithm 10, and then discuss its MPC implementation. Throughout, we consider the case $k < n$, for which $\mathrm{OPT}_z^{\mathrm{cl}} > 0$; the case $k = n$ is trivial, since we can simply return P as the solution. Moreover, by the above dimension reduction, we assume that $d = O(\log n)$.

Proof of Property (P1). Fix $p_i, p_j \in P$ with $i \ne j$. We first show that $\mathrm{dist}(f(p_i), f(p_j)) \ge \frac{(\mathrm{OPT}_z^{\mathrm{cl}})^{1/z}}{\mathrm{poly}(n)}$. According to Line 8, it is easy to verify that $\min_{p \in \mathrm{Net}_i, q \in \mathrm{Net}_j} \mathrm{dist}(p,q) \ge \frac{\eta^{1/z}}{n^{100}\sqrt{d}\cdot\beta^{1/z}} \ge \frac{(\mathrm{OPT}_z^{\mathrm{cl}})^{1/z}}{\mathrm{poly}(n)}$. Therefore, $\mathrm{dist}(f(p_i), f(p_j)) \ge \frac{(\mathrm{OPT}_z^{\mathrm{cl}})^{1/z}}{\mathrm{poly}(n)}$ because $f(p_i) \in \mathrm{Net}_i$ and $f(p_j) \in \mathrm{Net}_j$.

We then show that $\mathrm{dist}(f(p_i), f(p_j)) \le \mathrm{poly}(n) \cdot (\alpha \cdot \mathrm{OPT}_z^{\mathrm{cl}})^{1/z}$. We need the following fact.

Fact A.1. *For every $p, q \in P$, $|\mathrm{dist}(g(p), g(q)) - \mathrm{dist}(f(p), f(q))| \le \frac{2(\mathrm{OPT}_z^{\mathrm{cl}})^{1/z}}{n}$.*

Proof. By Line 8, rounding moves each point by at most $\frac{\eta^{1/z}}{n\cdot\beta^{1/z}} \le \frac{(\mathrm{OPT}_z^{\mathrm{cl}})^{1/z}}{n}$. The claimed bound follows from the triangle inequality. □

By Fact A.1, it suffices to show that $\mathrm{dist}(g(p_i), g(p_j)) \le \mathrm{poly}(n) \cdot (\alpha \cdot \mathrm{OPT}_z^{\mathrm{cl}})^{1/z}$. Notice that the construction of h in Line 6 ensures that for every two points, their corresponding coordinates after applying the mapping h differ by at most $m \cdot (\alpha' \cdot \eta)^{1/z} \le \mathrm{poly}(n) \cdot (\alpha \cdot \mathrm{OPT}_z^{\mathrm{cl}})^{1/z}$. Therefore, $\mathrm{dist}(g(p_i), g(p_j)) \le \mathrm{poly}(n) \cdot (\alpha \cdot \mathrm{OPT}_z^{\mathrm{cl}})^{1/z}$ follows directly, which, as stated, implies that $\mathrm{dist}(f(p_i), f(p_j)) \le \mathrm{poly}(n) \cdot (\alpha \cdot \mathrm{OPT}_z^{\mathrm{cl}})^{1/z}$.

Combining the above results, we confirm that the aspect ratio of $f(P)$ is at most $\mathrm{poly}(n)$.

Proof of Property (P2). For every solution $C \subseteq P$, define $\mathrm{cost}_z^g(C) := \sum_{p\in P} \mathrm{dist}^z(g(p), g(C))$ as the clustering cost after applying the map g, and define $\mathrm{cost}_z^f(C) := \sum_{p\in P} \mathrm{dist}^z(f(p), f(C))$ as the clustering cost after applying the map f.

We first have the following fact.

Fact A.2. *For every $p_i, p_j \in P$, the following guarantees hold.*

- *If* $\mathrm{dist}(p_i, p_j) \le (\alpha' \cdot \mathrm{OPT}_z^{\mathrm{cl}})^{1/z}$, *then* $\mathrm{dist}(g(p_i), g(p_j)) = \mathrm{dist}(p_i, p_j)$.
- *If* $\mathrm{dist}(g(p_i), g(p_j)) < (\alpha' \cdot \mathrm{OPT}_z^{\mathrm{cl}})^{1/z}$, *then* $\mathrm{dist}(g(p_i), g(p_j)) = \mathrm{dist}(p_i, p_j)$.

Proof. We first prove the first guarantee. By Line 7, $g(p_i) = (h(p_{i,1}), \ldots, h(p_{i,d}))$ and $g(p_j) = (h(p_{j,1}), \ldots, h(p_{j,d}))$. For every $u \in [d]$, since $|p_{i,u} - p_{j,u}| \le \mathrm{dist}(p_i, p_j) \le (\alpha' \cdot \mathrm{OPT}_z^{\mathrm{cl}})^{1/z} \le (\alpha' \cdot \eta)^{1/z}$, we have $h(p_{i,u}) - h(p_{j,u}) = p_{i,u} - p_{j,u}$ by Line 6. Therefore, we have $\mathrm{dist}(g(p_i), g(p_j)) = \mathrm{dist}(p_i, p_j)$.

We then prove the second guarantee by considering its contrapositive. If $\mathrm{dist}(g(p_i), g(p_j)) \neq \mathrm{dist}(p_i, p_j)$, then there exists $u \in [d]$ such that $h(p_{i,u}) - h(p_{j,u}) \neq p_{i,u} - p_{j,u}$. By Line 6, this implies $|h(p_{i,u}) - h(p_{j,u})| \ge (\alpha' \cdot \eta)^{1/z} \ge (\alpha' \cdot \mathrm{OPT}_z^{\mathrm{cl}})^{1/z}$, and therefore $\mathrm{dist}(g(p_i), g(p_j)) \ge (\alpha' \cdot \mathrm{OPT}_z^{\mathrm{cl}})^{1/z}$. □

Let $C^* \subseteq P$ be an optimal solution to (k, z)-Clustering on P. For every $p \in P$, let $c^* \in C^*$ denote the closest center in C^* to p (with ties broken arbitrarily); we have $\mathrm{dist}(p, c^*) \le (\mathrm{OPT}_z^{\mathrm{cl}})^{1/z}$, which implies that $\mathrm{dist}(g(p), g(c^*)) = \mathrm{dist}(p, c^*)$ by Fact A.2. Therefore, $\mathrm{cost}_z^g(C^*) \le \mathrm{cost}_z(C^*)$. Then, by Fact A.1 and the generalized triangle inequality (Fact 2.1), we have $\mathrm{cost}_z^f(C^*) \le 2^{z-1} \cdot \mathrm{cost}_z^g(C^*) + 2^{z-1} \cdot n \cdot (\frac{2(\mathrm{OPT}_z^{\mathrm{cl}})^{1/z}}{n})^z \le 2^{O(z)} \cdot \mathrm{cost}_z(C^*)$. This means that the optimal (k, z)-Clustering objective on the final dataset $f(P)$ is at most $2^{O(z)}$ times that of the input dataset P.

Then, consider an arbitrary solution $f(C) \subseteq f(P)$ with $|f(C)| = k$ (and thus $|C| = k$), and suppose that $f(C)$ is an α-approximate solution to (k, z)-Clustering on $f(P)$. Then we have $\mathrm{cost}_z^f(C) \le 2^{O(z)} \cdot \alpha \cdot \mathrm{OPT}_z^{\mathrm{cl}}$. By Fact A.1 and the generalized triangle inequality (Fact 2.1), it follows that $\mathrm{cost}_z^g(C) \le 2^{O(z)}\alpha \cdot \mathrm{OPT}_z^{\mathrm{cl}}$. Finally, recall that $\alpha' = 2^{O(z)}\alpha$ is sufficiently large; for every $p \in P$, since $\mathrm{dist}^z(g(p), g(C)) \le \mathrm{cost}_z^g(C) < \alpha' \cdot \mathrm{OPT}_z^{\mathrm{cl}}$, by Fact A.2 we have that $\mathrm{dist}^z(p, C) \le \mathrm{dist}^z(g(p), g(C))$. Therefore, $\mathrm{cost}_z(C) \le \mathrm{cost}_z^g(C) \le 2^{O(z)}\alpha \cdot \mathrm{OPT}_z^{\mathrm{cl}}$. This confirms that C is a $2^{O(z)}\alpha$-approximate solution to (k, z)-Clustering on the input dataset P.

MPC Implementation of Algorithm 10. Line 1 requires computing a coarse estimate of the optimal (k, z)-Clustering objective on P. To this end, we invoke the MPC k-Center algorithm from [CGGJ25, Lemma 7.2] to compute an upper estimate $E > 0$ of the optimal k-Center objective within a factor of $\mathrm{poly}(n)$. It is easy to verify that the value $\eta := n \cdot E^z$ approximates the optimal (k, z)-Clustering objective within a factor of $\beta = n^{O(z)}$, i.e., $\mathrm{OPT}_z^{\mathrm{cl}} \le \eta \le \beta \cdot \mathrm{OPT}_z^{\mathrm{cl}}$.

Then, Line 4 collects the distinct coordinate values, Line 5 uses MPC sorting, and Line 6 can be achieved via an MPC prefix-sum computation (see [GSZ11] for both MPC sorting and MPC prefix-sum computation). To implement Line 7, we use sorting to look up the value $h(x)$ for each coordinate x of every point, and then gather the mapped coordinates to form the resulting point. This takes $O(\log_s n)$ rounds. Finally, the rounding step in Line 8 can be easily implemented in a single MPC round.

Overall, the complexity of the above implementation is dominated by the MPC k-Center algorithm from [CGGJ25, Lemma 7.2], which requires local memory $s \ge \mathrm{poly}(d \log n)$, runs in $O(\log_s n)$ rounds, and uses total memory $O(n \cdot \mathrm{poly}(d \log n))$.

B Proof of Lemma 4.4

Fact B.1. $\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_0) = \mathrm{cost}_z(\boldsymbol{y}^{(0)}) + \lambda_0 \cdot \|\boldsymbol{y}^{(0)}\|_1$.

Proof. Denote by $\boldsymbol{y}^* \in \mathbb{R}^P_{\geq 0}$ an optimal fractional solution to Power-z Facility Location with $\lambda = \lambda_0$; we have $\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_0) = \mathrm{cost}_z(\boldsymbol{y}^*) + \lambda_0 \cdot \|\boldsymbol{y}^*\|_1 \geq (|P| - \|\boldsymbol{y}^*\|_1) + \lambda_0 \cdot \|\boldsymbol{y}^*\|_1 = |P| = \mathrm{cost}_z(\boldsymbol{y}^{(0)}) + \lambda_0 \cdot \|\boldsymbol{y}^{(0)}\|_1$, where the second step holds since $\min_{p \neq q \in P} \mathrm{dist}(p,q) \geq 1$, and the last step holds since $\boldsymbol{y}^{(0)}$ is the all-ones vector (see Line 2). We also have $\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_0) \leq \mathrm{cost}_z(\boldsymbol{y}^{(0)}) + \lambda_0 \cdot \|\boldsymbol{y}^{(0)}\|_1$ given its optimality. Combining both equations finishes the proof. □

Fact B.2. $\|\boldsymbol{y}^{(L)}\|_1 \leq 2$.

Proof. First, observe that $\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_L) \leq \lambda_L + n \cdot \Delta^{O(z)}$ by considering any solution $\boldsymbol{y}'$ with $\|\boldsymbol{y}'\|_1 = 1$. Then, by Corollary 3.2, the solution $\boldsymbol{y}^{(L)}$ satisfies that

$$\lambda_L \cdot \|\boldsymbol{y}^{(L)}\|_1 \;\leq\; \widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_L) \;\leq\; \lambda_L + n\Delta^{O(z)} \;\leq\; 2\lambda_L,$$

where the last step holds since $\lambda_L = 2^{Lz}$ and $L = \Theta(\log(n\Delta))$ is sufficiently large. Hence, $\|\boldsymbol{y}^{(L)}\|_1 \leq 2$, which finishes the proof. □

We are now ready to prove Lemma 4.4.

Proof of Lemma 4.4. By Facts B.1 and B.2, there exists some $\ell \in [L]$ such that $\|\boldsymbol{y}^{(\ell-1)}\|_1 \geq k$ and $\|\boldsymbol{y}^{(\ell)}\|_1 \leq k$. Consequently, Line 4 selects an arbitrary such index $\ell^* \in [L]$, and the output of Algorithm 3 is $\boldsymbol{y} := \alpha \cdot \boldsymbol{y}^{(\ell^*-1)} + (1-\alpha) \cdot \boldsymbol{y}^{(\ell^*)}$, where $\alpha \in [0,1]$ is chosen (in Line 5) such that $\|\boldsymbol{y}\|_1 = k$. Let $\boldsymbol{x}^{(\ell^*-1)} \sim \boldsymbol{y}^{(\ell^*-1)}$ and $\boldsymbol{x}^{(\ell^*)} \sim \boldsymbol{y}^{(\ell^*)}$ be the optimal feasible assignments with respect to $\boldsymbol{y}^{(\ell^*-1)}$ and $\boldsymbol{y}^{(\ell^*)}$, respectively. It is easy to verify that $\boldsymbol{x} := \alpha \cdot \boldsymbol{x}^{(\ell^*-1)} + (1-\alpha) \cdot \boldsymbol{x}^{(\ell^*)}$ is feasible with respect to $\boldsymbol{y} = \alpha \cdot \boldsymbol{y}^{(\ell^*-1)} + (1-\alpha) \cdot \boldsymbol{y}^{(\ell^*)}$. As a result,

$$\begin{aligned}
\mathrm{cost}_z(\boldsymbol{y}) \;\leq\; \sigma_z(\boldsymbol{x}) \;&=\; \sum_{p \in P} \sum_{q \in P} \left(\alpha \cdot x_{p,q}^{(\ell^*-1)} + (1-\alpha) \cdot x_{p,q}^{(\ell^*)} \right) \cdot \mathrm{dist}^z(p,q) \\
&=\; \alpha \cdot \sigma_z(\boldsymbol{x}^{(\ell^*-1)}) + (1-\alpha) \cdot \sigma_z(\boldsymbol{x}^{(\ell^*)}) \;=\; \alpha \cdot \mathrm{cost}_z(\boldsymbol{y}^{(\ell^*-1)}) + (1-\alpha) \cdot \mathrm{cost}_z(\boldsymbol{y}^{(\ell^*)}).
\end{aligned}$$

By Corollary 3.2 and Fact B.1, we have

$$\begin{aligned}
\mathrm{cost}_z(\boldsymbol{y}^{(\ell^*-1)}) \;&\leq\; 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \left(\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_{\ell^*-1}) - \lambda_{\ell^*-1} \cdot \|\boldsymbol{y}^{(\ell^*-1)}\|_1 \right) \\
&\leq\; 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \left(2^z \cdot \widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_{\ell^*-1}) - 2^z \cdot \lambda_{\ell^*-1} \cdot \|\boldsymbol{y}^{(\ell^*-1)}\|_1 \right) \\
&=\; 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \left(2^z \cdot \widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_{\ell^*-1}) - \lambda_{\ell^*} \cdot \|\boldsymbol{y}^{(\ell^*-1)}\|_1 \right),
\end{aligned}$$

where the last step follows from the fact that $\lambda_{\ell^*} = 2^z \lambda_{\ell^*-1}$; and

$$\mathrm{cost}_z(\boldsymbol{y}^{(\ell^*)}) \;\leq\; 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \left(\widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_{\ell^*}) - \lambda_{\ell^*} \cdot \|\boldsymbol{y}^{(\ell^*)}\|_1 \right).$$

Therefore,

$$\mathrm{cost}_z(\boldsymbol{y}) \;\leq\; 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \left(2^z \alpha \cdot \widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_{\ell^*-1}) + (1-\alpha) \cdot \widehat{\mathrm{OPT}}_z^{\mathrm{fl}}(\lambda_{\ell^*}) - \lambda_{\ell^*} \cdot \|\boldsymbol{y}\|_1 \right).$$

Then, let $\boldsymbol{y}^* \in \mathbb{R}^P_{\geq 0}$ with $\|\boldsymbol{y}^*\|_1 = k$ denote an optimal fractional solution to the EUCLIDEAN (k,z)-CLUSTERING problem on input (P,P,k). We have $2^z\alpha \cdot \widehat{\mathrm{OPT}}^{\mathrm{fl}}_z(\lambda_{\ell^*-1}) \leq 2^z\alpha \cdot (\lambda_{\ell^*-1} \cdot k + \mathrm{cost}_z(\boldsymbol{y}^*)) = \alpha \cdot \lambda_{\ell^*} \cdot k + 2^z\alpha \cdot \mathrm{cost}_z(\boldsymbol{y}^*)$ and $(1-\alpha) \cdot \widehat{\mathrm{OPT}}^{\mathrm{fl}}_z(\lambda_{\ell^*}) \leq (1-\alpha) \cdot \lambda_{\ell^*} \cdot k + (1-\alpha) \cdot \mathrm{cost}_z(\boldsymbol{y}^*)$. Hence,

$$\begin{aligned}
\mathrm{cost}_z(\boldsymbol{y}) &\leq 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \left(2^z\alpha \cdot \widehat{\mathrm{OPT}}^{\mathrm{fl}}_z(\lambda_{\ell^*-1}) + (1-\alpha) \cdot \widehat{\mathrm{OPT}}^{\mathrm{fl}}_z(\lambda_{\ell^*}) - \lambda_{\ell^*} \cdot \|\boldsymbol{y}\|_1\right) \\
&\leq 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \left(\lambda_{\ell^*} \cdot k + (2^z\alpha + 1 - \alpha) \cdot \mathrm{cost}_z(\boldsymbol{y}^*) - \lambda_{\ell^*} \cdot \|\boldsymbol{y}\|_1\right) \\
\triangleright\ \|\boldsymbol{y}\|_1 = k \quad &\leq 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \mathrm{cost}_z(\boldsymbol{y}^*) \ \leq\ 2^{O(z^2)} \cdot \varepsilon^{-2z} \cdot \mathrm{OPT}^{\mathrm{cl}}_z(P,P,k).
\end{aligned}$$

This finishes the proof. □

C Missing Proofs in Section 4

C.1 Proof of Claim 4.6

Proof. Throughout this proof, we consider the case where Q is a set rather than a multiset for simplicity; the same argument applies directly to the multiset case, since colocated points can be distinguished by their IDs. Recall that $\boldsymbol{x}' \sim \boldsymbol{y}'$ is feasible, $\sum_{q\in Q} x'_{p,q} = 1$ for every $p \in P$ and $x'_{p,q} \in \{0, y'_q\}$ for every $p \in P$ and $q \in Q$ by Line 2. In particular,

$$\forall p \in P, \qquad \sum_{q \in Q : x'_{p,q} > 0} y'_q \ =\ \sum_{q \in Q} x'_{p,q} \ =\ 1.$$

Let $Q_+ := \{q \in Q : \exists p \in P,\ x'_{p,q} > 0\}$ and, for every $q \in Q_+$, define

$$g(q) := \operatorname*{argmin}_{p \in P : x'_{p,q} > 0} \beta_p.$$

We note that points in $Q \setminus Q_+$ cannot be selected by any $f(p)$ and therefore are never included in C. By Line 6, a point $q \in Q_+$ is selected if and only if $f(g(q)) = q$.

Expected Size. For every $p \in P$ and $q \in Q$ with $x'_{p,q} > 0$, the exponential clocks have rates $y'_q = 1/n^\varepsilon$, and hence

$$\Pr[f(p) = q] \ =\ \frac{y'_q}{\sum_{q' \in Q : x'_{p,q'} > 0} y'_{q'}} \ =\ y'_q.$$

For every $q \in Q_+$, the value $g(q)$ depends only on the clocks $\{\beta_p\}_{p\in P}$ and is therefore independent of $\{\alpha_{q'}\}_{q'\in Q}$. Consequently,

$$\Pr[f(g(q)) = q] \ =\ \sum_{p \in P : x'_{p,q} > 0} \Pr[g(q) = p] \cdot \Pr[f(p) = q] \ =\ y'_q.$$

As a result,

$$\mathbb{E}[|C|] \ \leq\ \sum_{q \in Q_+} \Pr[f(g(q)) = q] \ =\ \sum_{q \in Q_+} y'_q \ \leq\ \|\boldsymbol{y}'\|_1.$$

Expected Cost. We follow the analysis of connection paths in [ANS17, Section 4.2] and extend it to the power-z objective. Let

$$E := \big\{\{p,q\} : p \in P,\ q \in Q_+,\ x'_{p,q} > 0\big\}$$

be the edge set of the *undirected* support graph on $P \sqcup Q_+$, where the two vertex classes are kept distinct even when their locations coincide. Construct a random *directed* graph G on the same vertex set by adding an arc $p \to f(p)$ for every $p \in P$ and an arc $q \to g(q)$ for every $q \in Q_+$. Every vertex in G has exactly one outgoing arc, and the selected points are exactly those belonging to directed cycles of length two. Moreover, with probability one, all clocks within each vertex class are distinct. On this event, G has no directed cycle of length greater than two (see [ANS17, Observation 3.1]).

For every $p \in P$, define its *connection path* $\Pi(p) = (e_p^{(1)}, \dots, e_p^{(t_p)})$ by starting at p and following the unique outgoing arc until immediately before a vertex would be visited for the second time. The last vertex from Q_+ on this path belongs to a directed 2-cycle and hence is included in C. Writing $\mathrm{dist}(e)$ for the distance between the endpoints of an edge or arc e, the triangle inequality gives

$$\mathrm{dist}(p, C) \ \le \ \sum_{i=1}^{t_p} \mathrm{dist}(e_p^{(i)}). \tag{27}$$

Let $\gamma := 5/4$. For $z > 1$, Hölder's inequality implies

$$\begin{aligned}
\mathrm{dist}^z(p, C) \ &\le \ \Big(\sum_{i=1}^{t_p} \gamma^{-i/z} \cdot \gamma^{i/z} \,\mathrm{dist}(e_p^{(i)})\Big)^z \\
&\le \ \Big(\sum_{i=1}^{t_p} \gamma^{-i/(z-1)}\Big)^{z-1} \cdot \sum_{i=1}^{t_p} \gamma^i \,\mathrm{dist}^z(e_p^{(i)}) \\
&\le \ \Big(\frac{1}{\gamma^{1/(z-1)} - 1}\Big)^{z-1} \cdot \sum_{i=1}^{t_p} \gamma^i \,\mathrm{dist}^z(e_p^{(i)}) \\
&\le \ \Big(\frac{z-1}{\ln \gamma}\Big)^{z-1} \cdot \sum_{i=1}^{t_p} \gamma^i \,\mathrm{dist}^z(e_p^{(i)}).
\end{aligned}$$

Thus, for every integer $z \ge 1$,

$$\mathrm{dist}^z(p, C) \ \le \ z^{O(z)} \cdot \sum_{i=1}^{t_p} \gamma^i \,\mathrm{dist}^z(e_p^{(i)}), \tag{28}$$

where the $z = 1$ case follows directly from (27).

For every $e \in E$, $p \in P$, and integer $i \ge 1$, let $\mu_{i,p}(e)$ be the indicator of the event that the i-th arc of $\Pi(p)$ traverses e in either direction. Let $w_i(e) := \sum_{p \in P} \mu_{i,p}(e)$. The connection-path bound in [ANS17, Section 4.2], applied to the feasible pair $(\boldsymbol{x}', \boldsymbol{y}')$ with $x'_{p,q} \in \{0, y'_q\}$, gives, for every $e = \{p, q\} \in E$,

$$\mathbb{E}[w_{2t+1}(e)] \ \le \ \frac{x'_{p,q}}{2^{\max\{0,t-1\}}}, \qquad \forall t \ge 0, \tag{29}$$

$$\mathbb{E}[w_{2t}(e)] \ \le \ \frac{x'_{p,q}}{2^{\max\{0,t-2\}}}, \qquad \forall t \ge 1. \tag{30}$$

Since $\gamma^2/2 < 1$, it follows that, for every $e = \{p, q\} \in E$,

$$\begin{aligned}\sum_{i\geq 1} \gamma^i \, \mathbb{E}[w_i(e)] &\leq x'_{p,q} \cdot \Big(\sum_{t\geq 0} \frac{\gamma^{2t+1}}{2^{\max\{0,t-1\}}} + \sum_{t\geq 1} \frac{\gamma^{2t}}{2^{\max\{0,t-2\}}} \Big) \\ &= O(x'_{p,q}).\end{aligned}$$

Combining this with (28) and linearity of expectation, we obtain

$$\begin{aligned}\mathbb{E}[\mathrm{cost}_z(C)] &\leq z^{O(z)} \cdot \sum_{p\in P} \sum_{i\geq 1} \sum_{e\in E} \gamma^i \, \mathrm{dist}^z(e) \cdot \mathbb{E}[\mu_{i,p}(e)] \\ &= z^{O(z)} \cdot \sum_{e\in E} \mathrm{dist}^z(e) \cdot \sum_{i\geq 1} \gamma^i \, \mathbb{E}[w_i(e)] \\ &\leq O(z^{O(z)}) \cdot \sum_{p\in P} \sum_{q\in Q} x'_{p,q} \, \mathrm{dist}^z(p, q) \\ &= O(z^{O(z)}) \cdot \sigma_z(\boldsymbol{x}') \\ &\leq (z/\varepsilon)^{O(z)} \cdot \mathrm{cost}_z(\boldsymbol{y}'),\end{aligned}$$

where the last step follows from Line 2 of Algorithm 4. Together with the expected size bound, this proves Claim 4.6. □

C.2 Proof of Claim 4.12

We describe the bi-point rounding algorithm in Algorithm 11.

Algorithm 11: Bi-point Rounding on (C_1, C_2, a)

```
/* Without loss of generality, assume that |C_1| ≤ |C_2|                    */
```
1 $\kappa \leftarrow \lceil a|C_1| + (1-a)|C_2| \rceil$
2 $C'_1 \leftarrow \{C_2(c) : c \in C_1\}$, where $C_2(c) \in C_2$ satisfies $\mathrm{dist}(c, C_2(c)) \leq O(\varepsilon^{-1}) \cdot \mathrm{dist}(c, C_2)$
3 add $|C_1| - |C'_1|$ arbitrary distinct points from $C_2 \setminus C'_1$ to C'_1 `// this ensures` $|C'_1| = |C_1|$
4 $C'_2 \leftarrow$ a uniformly random subset of $C_2 \setminus C'_1$ of size $\kappa - |C_1|$
5 **return** $C \leftarrow C_1 \cup C'_2$ with probability a, and $C \leftarrow C'_1 \cup C'_2$ with probability $1 - a$

Correctness of Algorithm 11. Clearly, the returned set $C \subseteq C_1 \cup C_2$ has size at most $\kappa = \lceil a|C_1| + (1-a)|C_2| \rceil \leq a|C_1| + (1-a)|C_2| + 1$. It then remains to analyze the expected cost of C.

If $|C_1| = |C_2|$, then Line 3 ensures that $C'_1 = C_2$, and therefore $C'_2 = \emptyset$. The algorithm returns C_1 with probability a and C_2 with probability $1 - a$. Consequently, $\mathbb{E}[\mathrm{cost}_z(C)] = a\mathrm{cost}_z(C_1) + (1-a)\mathrm{cost}_z(C_2)$, as desired.

We henceforth assume that $|C_1| < |C_2|$. Fix an arbitrary point $p \in P$. For $j \in \{1, 2\}$, choose $c_j \in C_j$ such that $\mathrm{dist}(p, c_j) = \mathrm{dist}(p, C_j)$. We distinguish two cases according to whether c_2 belongs to C'_1.

Case 1: $c_2 \in C'_1$. In this case, if the algorithm returns $C \leftarrow C_1 \cup C'_2$, which happens with probability a, then $\mathrm{dist}(p, C) \leq \mathrm{dist}(p, C_1)$. Otherwise, the algorithm returns $C \leftarrow C'_1 \cup C'_2$, which happens with probability $1-a$, and since $c_2 \in C'_1 \subseteq C$, we have $\mathrm{dist}(p, C) \leq \mathrm{dist}(p, c_2) = \mathrm{dist}(p, C_2)$. Therefore, $\mathbb{E}[\mathrm{dist}^z(p, C)] \leq a \cdot \mathrm{dist}^z(p, C_1) + (1-a) \cdot \mathrm{dist}^z(p, C_2)$.

Case 2: $c_2 \in C_2 \setminus C_1'$. In this case, we first prove the following inequality

$$\operatorname{dist}(p, C_1') \;\le\; O(\varepsilon^{-1}) \cdot \big(\operatorname{dist}(p, C_1) + \operatorname{dist}(p, C_2)\big). \tag{31}$$

To see this, recall that $C_2(c_1) \in C_1'$ (Line 2); we then have

$$\begin{aligned}
\operatorname{dist}(p, C_1') \;\le\; \operatorname{dist}(p, C_2(c_1)) \;&\le\; \operatorname{dist}(p, c_1) + \operatorname{dist}(c_1, C_2(c_1)) \\
\triangleright \text{ Line 2} \quad &\le\; \operatorname{dist}(p, c_1) + O(\varepsilon^{-1}) \cdot \operatorname{dist}(c_1, C_2) \\
&\le\; (1 + O(\varepsilon^{-1})) \cdot \operatorname{dist}(p, c_1) + O(\varepsilon^{-1}) \cdot \operatorname{dist}(p, C_2) \\
&=\; O(\varepsilon^{-1}) \cdot \big(\operatorname{dist}(p, C_1) + \operatorname{dist}(p, C_2)\big),
\end{aligned}$$

which proves (31). Then, we further distinguish the following two subcases:

(i) If $\operatorname{dist}(p, C_1) \le \operatorname{dist}(p, C_2)$, then

$$\begin{aligned}
\mathbb{E}[\operatorname{dist}^z(p, C)] \;&=\; a \cdot \mathbb{E}[\operatorname{dist}^z(p, C_1 \cup C_2')] + (1-a) \cdot \mathbb{E}[\operatorname{dist}^z(p, C_1' \cup C_2')] \\
&\le\; a \cdot \operatorname{dist}^z(p, C_1) + (1-a) \cdot \operatorname{dist}^z(p, C_1') \\
\triangleright (31) \quad &\le\; a \cdot \operatorname{dist}^z(p, C_1) + (1-a) \cdot \varepsilon^{-O(z)} \cdot \big(\operatorname{dist}(p, C_1) + \operatorname{dist}(p, C_2)\big)^z \\
&\le\; \varepsilon^{-O(z)} \cdot \big(a \cdot \operatorname{dist}^z(p, C_1) + (1-a) \cdot \operatorname{dist}^z(p, C_2)\big).
\end{aligned}$$

(ii) If $\operatorname{dist}(p, C_1) > \operatorname{dist}(p, C_2)$, assume for now that $\Pr[c_2 \in C_2'] \in (0, 1)$; the boundary cases are discussed later. Then, we have

$$\mathbb{E}[\operatorname{dist}^z(p, C)] \;=\; \mathbb{E}[\operatorname{dist}^z(p, C) \mid c_2 \in C_2'] \Pr[c_2 \in C_2'] + \mathbb{E}[\operatorname{dist}^z(p, C) \mid c_2 \notin C_2'] \Pr[c_2 \notin C_2'].$$

We analyze the two terms separately.

For the first term, since it must hold that $C_2' \subseteq C$ (Line 5), we have

$$\mathbb{E}[\operatorname{dist}^z(p, C) \mid c_2 \in C_2'] \;\le\; \operatorname{dist}^z(p, c_2) = \operatorname{dist}^z(p, C_2).$$

For the second term, since the choice between $C = C_1 \cup C_2'$ and $C = C_1' \cup C_2'$ (Line 5) is independent of the event $c_2 \notin C_2'$, we have

$$\begin{aligned}
\mathbb{E}[\operatorname{dist}^z(p, C) \mid c_2 \notin C_2'] \;&=\; \mathbb{E}[a \cdot \operatorname{dist}^z(p, C_1 \cup C_2') + (1-a) \cdot \operatorname{dist}^z(p, C_1' \cup C_2') \mid c_2 \notin C_2'] \\
&\le\; a \cdot \operatorname{dist}^z(p, C_1) + (1-a) \cdot \operatorname{dist}^z(p, C_1') \\
\triangleright (31) \quad &\le\; a \cdot \operatorname{dist}^z(p, C_1) + (1-a) \cdot \varepsilon^{-O(z)} \cdot \operatorname{dist}^z(p, C_1) \\
&\le\; \varepsilon^{-O(z)} \cdot \operatorname{dist}^z(p, C_1).
\end{aligned}$$

As a result,

$$\mathbb{E}[\operatorname{dist}^z(p, C)] \;\le\; \varepsilon^{-O(z)} \cdot \big(\operatorname{dist}^z(p, C_2) \cdot \Pr[c_2 \in C_2'] + \operatorname{dist}^z(p, C_1) \cdot \Pr[c_2 \notin C_2']\big).$$

Since C_2' is a uniformly random subset of $C_2 \setminus C_1'$ of size $\kappa - |C_1|$, we have

$$t \;:=\; \Pr[c_2 \in C_2'] \;=\; \frac{\kappa - |C_1|}{|C_2 \setminus C_1'|} \;=\; \frac{\kappa - |C_1|}{|C_2| - |C_1'|} \;\ge\; 1 - a,$$

where we are using the fact that $C_1' \subseteq C_2$, $|C_1| = |C_1'|$ and $\kappa = \lceil a|C_1| + (1-a)|C_2| \rceil \ge a|C_1| + (1-a)|C_2|$. Finally, since $\operatorname{dist}(p, C_1) > \operatorname{dist}(p, C_2)$, we have

$$\mathbb{E}[\operatorname{dist}^z(p, C)] \;\le\; \varepsilon^{-O(z)} \cdot \big(\operatorname{dist}^z(p, C_2) \cdot t + \operatorname{dist}^z(p, C_1) \cdot (1-t)\big)$$

$$\leq \varepsilon^{-O(z)} \cdot \big(\operatorname{dist}^z(p, C_2) \cdot (1-a) + \operatorname{dist}^z(p, C_1) \cdot a\big).$$

In all above cases, we have $\mathbb{E}[\operatorname{dist}^z(p, C)] \leq \varepsilon^{-O(z)} \cdot (a \cdot \operatorname{dist}^z(p, C_1) + (1-a) \cdot \operatorname{dist}^z(p, C_2))$. Summing over all $p \in P$ gives

$$\mathbb{E}[\operatorname{cost}_z(C)] \leq \varepsilon^{-O(z)} \cdot (a \cdot \operatorname{cost}_z(C_1) + (1-a) \cdot \operatorname{cost}_z(C_2)).$$

Finally, we discuss the boundary cases where $\Pr[c_2 \in C_2'] = 0$ or $\Pr[c_2 \in C_2'] = 1$. If $\Pr[c_2 \in C_2'] = 0$, then $a = 1$ since $\Pr[c_2 \in C_2'] \geq 1 - a$, and the desired bound follows from the analysis of the second term above. If $\Pr[c_2 \in C_2'] = 1$, the desired bound follows directly from the analysis of the first term above and $\operatorname{dist}(p, C_2) < \operatorname{dist}(p, C_1)$. Therefore, we complete the proof of the correctness of Algorithm 11.

MPC Implementation. Line 1 can be implemented in $O(\log_s n)$ rounds by aggregating $|C_1|$ and $|C_2|$ and broadcasting κ. Line 2 amounts to an approximate nearest-neighbor search, for which we can directly use [CGJ+24, Section 3.1]; this takes $O(\log_s n)$ rounds and uses $2^{O(\varepsilon d)} \cdot n \cdot \operatorname{poly}(d \log(n\Delta))$ total memory. Line 3 can be implemented straightforwardly. Line 4 can be implemented by generating a random permutation of $C_2 \setminus C_1'$ and selecting the first $\kappa - |C_1|$ points as C_2'. Finally, Line 5 can be implemented straightforwardly. Overall, Algorithm 11 can be implemented in $O(\log_s n)$ rounds using $2^{O(\varepsilon d)} \cdot n \cdot \operatorname{poly}(d \log(n\Delta))$ total memory.

D Proof of Claim 5.2

The first property in Claim 5.2 follows directly from the construction in Algorithm 8; specifically,

$$F^{\text{out}} = T_1 \cup T_2 \subseteq F^{\text{in}}, \quad r + 1 \leq |F^{\text{out}}| = |T_1| + |T_2| \leq 12r.$$

Here, the lower bound follows from $|T_1| = \min\{6r, |F^{\text{in}}|\} \geq r + 1$, since $1 \leq r < |F^{\text{in}}|$. If $T_1 = F^{\text{in}}$, then $F^{\text{out}} = F^{\text{in}}$ and $P^{\text{out}} = P^{\text{in}}$, so the second property holds immediately. Hence, assume $T_1 \neq F^{\text{in}}$ for the remainder of the proof. It remains to prove the second property, for which we need the following claim.

Claim D.1. *It holds that*

$$\operatorname{OPT}_z^{\text{cl}}(P^{\text{out}}, F^{\text{out}}, |F^{\text{out}}| - r) \leq \varepsilon^{-O(z)} \cdot \Big(\sum_{p \in P^{\text{in}}} \operatorname{dist}^z(p, \pi(p)) + \operatorname{OPT}_z^{\text{cl}}(P^{\text{in}}, F^{\text{in}}, |F^{\text{in}}| - r) \Big). \quad (32)$$

Proof. The proof can be found in Section D.1. □

Fix an arbitrary set $R \subseteq F^{\text{out}}$ of size r, and assume that, for some $\alpha \geq 1$,

$$\operatorname{cost}_z(P^{\text{out}}, F^{\text{out}} \setminus R) \leq \alpha \cdot \operatorname{OPT}_z^{\text{cl}}(P^{\text{out}}, F^{\text{out}}, |F^{\text{out}}| - r).$$

The goal is to prove

$$\operatorname{cost}_z(P^{\text{in}}, F^{\text{in}} \setminus R) \leq \varepsilon^{-O(z)} \cdot \alpha \cdot \operatorname{OPT}_z^{\text{cl}}(P^{\text{in}}, F^{\text{in}}, |F^{\text{in}}| - r). \quad (33)$$

For every $p \in P^{\text{out}}$, since $F^{\text{out}} \subseteq F^{\text{in}}$ by Algorithm 8, we have $F^{\text{out}} \setminus R \subseteq F^{\text{in}} \setminus R$, and hence

$$\operatorname{dist}(p, F^{\text{in}} \setminus R) \leq \operatorname{dist}(p, F^{\text{out}} \setminus R).$$

On the other hand, for every $p \in P^{\text{in}} \setminus P^{\text{out}}$, Line 7 implies that $\pi(p) \notin F^{\text{out}}$. Since $\pi(p) \in F^{\text{in}}$ and $R \subseteq F^{\text{out}}$, we have $\pi(p) \in F^{\text{in}} \setminus R$, and therefore

$$\text{dist}(p, F^{\text{in}} \setminus R) \;\leq\; \text{dist}(p, \pi(p)).$$

As a result, we have

$$\begin{aligned}
\text{cost}_z(P^{\text{in}}, F^{\text{in}} \setminus R) \;&\leq\; \text{cost}_z(P^{\text{out}}, F^{\text{out}} \setminus R) + \sum_{p \in P^{\text{in}} \setminus P^{\text{out}}} \text{dist}^z(p, \pi(p)) \\
&\leq\; \alpha \cdot \text{OPT}^{\text{cl}}_z(P^{\text{out}}, F^{\text{out}}, |F^{\text{out}}| - r) + \sum_{p \in P^{\text{in}} \setminus P^{\text{out}}} \text{dist}^z(p, \pi(p)) \\
\triangleright \text{ Claim D.1} \quad &\leq\; \varepsilon^{-O(z)} \cdot \alpha \cdot \Big(\sum_{p \in P^{\text{in}}} \text{dist}^z(p, \pi(p)) + \text{OPT}^{\text{cl}}_z(P^{\text{in}}, F^{\text{in}}, |F^{\text{in}}| - r) \Big).
\end{aligned}$$

Finally, by the construction of π in Line 2 and the definition of $\text{OPT}^{\text{cl}}_z(P^{\text{in}}, F^{\text{in}}, |F^{\text{in}}| - r)$, we have

$$\sum_{p \in P^{\text{in}}} \text{dist}^z(p, \pi(p)) \;\leq\; \varepsilon^{-O(z)} \cdot \text{cost}_z(P^{\text{in}}, F^{\text{in}}) \;\leq\; \varepsilon^{-O(z)} \cdot \text{OPT}^{\text{cl}}_z(P^{\text{in}}, F^{\text{in}}, |F^{\text{in}}| - r).$$

Combining everything together, we obtain (33), which finishes the proof of the second property in Claim 5.2.

D.1 Proof of Claim D.1.

For a point p and a nonempty set Q, we define $\text{NN}(p, Q) := \text{argmin}_{q \in Q} \text{dist}(p, q)$, with ties broken consistently. To prove Claim D.1, we need the following claim.

Claim D.2. *There exists a subset $\widehat{R} \subseteq T_1$ of size r, such that*

$$\sum_{c \in F^{\text{in}}} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus \widehat{R}) \;\leq\; \varepsilon^{-O(z)} \cdot \min_{\substack{R \subseteq F^{\text{in}} \\ |R| = r}} \sum_{c \in F^{\text{in}}} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus R). \tag{34}$$

Proof. The proof can be found in Section D.2. □

Let $\widehat{R} \subseteq T_1 \subseteq F^{\text{out}} \subseteq F^{\text{in}}$ be a subset of size r as guaranteed by Claim D.2, satisfying (34). We first have

$$\forall c \in F^{\text{out}}, \quad \text{dist}(c, F^{\text{out}} \setminus \widehat{R}) \;\leq\; O(\varepsilon^{-1}) \cdot \text{dist}(c, F^{\text{in}} \setminus \widehat{R}). \tag{35}$$

To see this, consider a fixed $c \in F^{\text{out}}$. If $c \notin \widehat{R}$, then (35) holds immediately since $\text{dist}(c, F^{\text{out}} \setminus \widehat{R}) = 0$. Otherwise, $c \in \widehat{R} \subseteq T_1$. Suppose that $c = c_i$, for some $i \in [6r]$, in the ordering of F^{in} computed in Line 4. Recall that, in Line 6, we compute a point $s_i \in F^{\text{in}} \setminus T_1$ satisfying $\text{dist}(c_i, s_i) \leq O(\varepsilon^{-1}) \cdot \text{dist}(c_i, F^{\text{in}} \setminus T_1)$, and include s_i in T_2. Combining the facts that $F^{\text{out}} = T_1 \cup T_2$ (Line 7), $T_1 \cap T_2 = \emptyset$ (Line 6) and $\widehat{R} \subseteq T_1$ (Claim D.2), we have $s_i \in F^{\text{out}} \setminus \widehat{R}$. Therefore,

$$\text{dist}(c_i, F^{\text{out}} \setminus \widehat{R}) \;\leq\; \text{dist}(c_i, s_i) \;\leq\; O(\varepsilon^{-1}) \cdot \text{dist}(c_i, F^{\text{in}} \setminus T_1).$$

Observe that $\text{NN}(c_i, F^{\text{in}} \setminus \widehat{R}) \in F^{\text{in}} \setminus \widehat{R} = (F^{\text{in}} \setminus T_1) \cup (T_1 \setminus \widehat{R})$, where $F^{\text{in}} \setminus T_1$ and $T_1 \setminus \widehat{R}$ are disjoint. We distinguish the following two cases.

- If $\text{NN}(c_i, F^{\text{in}} \setminus \widehat{R}) \in F^{\text{in}} \setminus T_1$, then $\text{dist}(c_i, F^{\text{in}} \setminus T_1) \leq \text{dist}(c_i, \text{NN}(c_i, F^{\text{in}} \setminus \widehat{R})) = \text{dist}(c_i, F^{\text{in}} \setminus \widehat{R})$. Therefore
$$\text{dist}(c_i, F^{\text{out}} \setminus \widehat{R}) \;\leq\; O(\varepsilon^{-1}) \cdot \text{dist}(c_i, F^{\text{in}} \setminus \widehat{R}).$$

- If $\mathrm{NN}(c_i, F^{\mathrm{in}} \setminus \widehat{R}) \in T_1 \setminus \widehat{R}$, since $(T_1 \setminus \widehat{R}) \subseteq (F^{\mathrm{out}} \setminus \widehat{R})$, we directly have

$$\mathrm{dist}(c_i, F^{\mathrm{out}} \setminus \widehat{R}) \;\le\; \mathrm{dist}(c_i, \mathrm{NN}(c_i, F^{\mathrm{in}} \setminus \widehat{R})) \;=\; \mathrm{dist}(c_i, F^{\mathrm{in}} \setminus \widehat{R}).$$

Overall, (35) holds. We use (35) to prove (32), as follows: Since $|F^{\mathrm{out}} \setminus \widehat{R}| = |F^{\mathrm{out}}| - |\widehat{R}| = |F^{\mathrm{out}}| - r$, we have

$$\begin{aligned}
&\mathrm{OPT}_z^{\mathrm{cl}}(P^{\mathrm{out}}, F^{\mathrm{out}}, |F^{\mathrm{out}}| - r) \\
&\le \mathrm{cost}_z(P^{\mathrm{out}}, F^{\mathrm{out}} \setminus \widehat{R}) \\
&\le \sum_{p \in P^{\mathrm{out}}} \big(\mathrm{dist}(p, \pi(p)) + \mathrm{dist}(\pi(p), F^{\mathrm{out}} \setminus \widehat{R})\big)^z \\
&\le 2^{z-1} \cdot \sum_{p \in P^{\mathrm{in}}} \mathrm{dist}^z(p, \pi(p)) + 2^{z-1} \cdot \sum_{c \in F^{\mathrm{out}}} w(c) \cdot \mathrm{dist}^z(c, F^{\mathrm{out}} \setminus \widehat{R}) \\
&\le 2^{z-1} \cdot \sum_{p \in P^{\mathrm{in}}} \mathrm{dist}^z(p, \pi(p)) + \varepsilon^{-O(z)} \cdot \sum_{c \in F^{\mathrm{out}}} w(c) \cdot \mathrm{dist}^z(c, F^{\mathrm{in}} \setminus \widehat{R}) \\
&\le 2^{z-1} \cdot \sum_{p \in P^{\mathrm{in}}} \mathrm{dist}^z(p, \pi(p)) + \varepsilon^{-O(z)} \cdot \min_{\substack{R \subseteq F^{\mathrm{in}} \\ |R| = r}} \sum_{c \in F^{\mathrm{in}}} w(c) \cdot \mathrm{dist}^z(c, F^{\mathrm{in}} \setminus R),
\end{aligned}$$

where the third step follows from the construction $P^{\mathrm{out}} = \{p \in P^{\mathrm{in}} : \pi(p) \in F^{\mathrm{out}}\}$ in Line 7 and the definition of w in Line 3; the fourth step follows from (35); and the last step follows from Claim D.2.

It remains to upper-bound $\min_{\substack{R \subseteq F^{\mathrm{in}} \\ |R| = r}} \sum_{c \in F^{\mathrm{in}}} w(c) \cdot \mathrm{dist}^z(c, F^{\mathrm{in}} \setminus R)$ as follows:

$$\begin{aligned}
&\min_{\substack{R \subseteq F^{\mathrm{in}} \\ |R| = r}} \sum_{c \in F^{\mathrm{in}}} w(c) \cdot \mathrm{dist}^z(c, F^{\mathrm{in}} \setminus R) \\
&\le \min_{\substack{R \subseteq F^{\mathrm{in}} \\ |R| = r}} \sum_{p \in P^{\mathrm{in}}} \mathrm{dist}^z(\pi(p), F^{\mathrm{in}} \setminus R) \\
&\le \min_{\substack{R \subseteq F^{\mathrm{in}} \\ |R| = r}} \sum_{p \in P^{\mathrm{in}}} 2^{z-1} \cdot \Big(\mathrm{dist}^z(p, \pi(p)) + \mathrm{dist}^z(p, F^{\mathrm{in}} \setminus R)\Big) \\
&= 2^{z-1} \cdot \sum_{p \in P^{\mathrm{in}}} \mathrm{dist}^z(p, \pi(p)) + 2^{z-1} \cdot \min_{\substack{R \subseteq F^{\mathrm{in}} \\ |R| = r}} \mathrm{cost}_z(P^{\mathrm{in}}, F^{\mathrm{in}} \setminus R) \\
&= 2^{z-1} \cdot \sum_{p \in P^{\mathrm{in}}} \mathrm{dist}^z(p, \pi(p)) + 2^{z-1} \cdot \mathrm{OPT}_z^{\mathrm{cl}}(P^{\mathrm{in}}, F^{\mathrm{in}}, |F^{\mathrm{in}}| - r),
\end{aligned}$$

where the first step follows from the definition of w (Line 3), and the last step follows from the definition of $\mathrm{OPT}_z^{\mathrm{cl}}(P^{\mathrm{in}}, F^{\mathrm{in}}, |F^{\mathrm{in}}| - r)$. Combining everything together, we obtain (32).

D.2 Proof of Claim D.2

Let $R^\star \subseteq F^{\mathrm{in}}$ be a set of size r that minimizes $\sum_{c \in F^{\mathrm{in}}} w(c) \cdot \mathrm{dist}^z(c, F^{\mathrm{in}} \setminus R^\star)$.

Since $T_1 \neq F^{\mathrm{in}}$ as we assume in Section D, we have $|T_1| = 6r$. Our proof strategy is to construct an auxiliary set $U \subseteq T_1$ of size at least r such that the centers in $R^\star \setminus T_1$ can be replaced, one-for-one, by centers in U while increasing the cost by only a bounded factor.

Below, we construct the set U, together with a mapping $h : U \to F^{\mathrm{in}}$, and analyze their properties; we then use these properties to complete the proof of Equation (34).

Construction of U and h. We construct the auxiliary set $U \subseteq T_1$ and a mapping h as follows:

1. $U_1 \leftarrow T_1 \setminus R^\star$.
2. $U_2 \leftarrow U_1 \setminus \{\mathrm{NN}(c, F^{\mathrm{in}} \setminus R^\star) : c \in R^\star\}$.
3. $U_3 \leftarrow U_2 \setminus \{\mathrm{NN}(c, U_2(c)) : c \in R^\star,\, U_2(c) \neq \emptyset\}$, where $U_2(c) := \{u \in U_2 : \mathrm{NN}(u, F^{\mathrm{in}} \setminus \{u\}) = c\}$.
4. For every $u \in U_3$, letting $c := \mathrm{NN}(u, F^{\mathrm{in}} \setminus \{u\})$, define $h(u) := c$ if $c \notin R^\star$, and $h(u) := \mathrm{NN}(c, U_2(c))$ otherwise. In the latter case, $U_2(c) \neq \emptyset$ since $u \in U_2(c)$.

   Observe that $h(u) \neq u$. Indeed, if $h(u) = c = \mathrm{NN}(u, F^{\mathrm{in}} \setminus \{u\})$, then $h(u) \neq u$ follows directly from the definition of $\mathrm{NN}(u, F^{\mathrm{in}} \setminus \{u\})$. Otherwise, $h(u) = \mathrm{NN}(c, U_2(c))$. By the third step, $\mathrm{NN}(c, U_2(c)) \notin U_3$, while $u \in U_3$. It therefore follows that $h(u) \neq u$.
5. Consider the undirected graph G on U_3 containing an edge $\{u, h(u)\}$ for every $u \in U_3$ with $h(u) \in U_3$. Since every vertex contributes at most one edge, every induced subgraph has at most as many edges as vertices and hence contains a vertex of degree at most 2. Therefore, the graph G is 2-degenerate and thus admits a proper 3-coloring.

   Finally, take an arbitrary 3-coloring and let U be the largest color class of G.

Clearly, we have $U \subseteq U_3 \subseteq U_2 \subseteq U_1 \subseteq T_1$.

Properties of U and h. We prove the following properties of U and the mapping h.

(a) $|U| \geq r$. To see this, observe that at most $3|R^\star| = 3r$ points are removed from T_1 in the first three steps. Since $|T_1| = 6r$, the set U_3 contains at least $3r$ points before the last step. By the last step, U contains at least one third of the points in U_3, so the final set satisfies $|U| \geq r$.

(b) $U \cap R^\star = \emptyset$. This property holds for the set U_1 constructed in the first step. Since $U \subseteq U_1$, it also holds for U.

(c) $\mathrm{NN}(c, F^{\mathrm{in}} \setminus R^\star) \notin U$, $\forall c \in R^\star$. This property holds for the set U_2 constructed in the second step. Since $U \subseteq U_2$, it also holds for U.

(d) $h(u) \in F^{\mathrm{in}} \setminus (R^\star \cup U)$, $\forall u \in U$. To see this, let $v := \mathrm{NN}(u, F^{\mathrm{in}} \setminus \{u\})$. By construction, if $v \notin R^\star$, then $h(u) = v \notin R^\star$; otherwise, $h(u) = \mathrm{NN}(v, U_2(v)) \in U_2 \subseteq T_1 \setminus R^\star$. Thus, in both cases, $h(u) \notin R^\star$. Moreover, if $h(u) \in U_3$, then $\{u, h(u)\}$ is an edge of the graph G. Since U is a color class of a proper coloring of G, u and $h(u)$ cannot both belong to U. Hence, $h(u) \notin U$, as required.

(e) $\mathrm{dist}(u, h(u)) \leq 2 \cdot \mathrm{dist}(u, F^{\mathrm{in}} \setminus \{u\})$, $\forall u \in U$. Again, let $v := \mathrm{NN}(u, F^{\mathrm{in}} \setminus \{u\})$. If $v \notin R^\star$, then $h(u) = v$, and hence $\mathrm{dist}(u, h(u)) = \mathrm{dist}(u, F^{\mathrm{in}} \setminus \{u\})$. Otherwise, since $u \in U_2(v)$ and $h(u) = \mathrm{NN}(v, U_2(v))$, we have $\mathrm{dist}(v, h(u)) \leq \mathrm{dist}(v, u)$. By the triangle inequality, $\mathrm{dist}(u, h(u)) \leq \mathrm{dist}(u, v) + \mathrm{dist}(v, h(u)) \leq 2 \cdot \mathrm{dist}(u, v) = 2 \cdot \mathrm{dist}(u, F^{\mathrm{in}} \setminus \{u\})$.

Completing the Proof of Claim D.2. We now return to the proof of Claim D.2, using the auxiliary set U and the mapping h constructed above.

Let $R_1 := R^\star \cap T_1$ and $R_2 := R^\star \setminus T_1$. Choose an arbitrary subset $R_2' \subseteq U \subseteq T_1$ of size $|R_2|$, and let $\widehat{R} := R_1 \cup R_2'$. Such a subset R_2' must exist since $|R_2| \leq r \leq |U|$ by Property (a). Clearly,

$\widehat{R} \subseteq T_1$. Moreover, since $U \cap R^\star = \emptyset$ by Property (b), $R_1 \subseteq R^\star$ and $R'_2 \subseteq U$, we have $R_1 \cap R'_2 = \emptyset$, and therefore $|\widehat{R}| = |R_1| + |R'_2| = |R_1| + |R_2| = r$. Observe that

$$\begin{aligned}
\sum_{c \in F^{\text{in}}} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus \widehat{R}) &= \sum_{c \in R_1} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus \widehat{R}) + \sum_{c \in R'_2} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus \widehat{R}) \\
\sum_{c \in F^{\text{in}}} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus R^\star) &= \sum_{c \in R_1} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus R^\star) + \sum_{c \in R_2} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus R^\star).
\end{aligned}$$

To prove Equation (34), it suffices to prove

$$\sum_{c \in R_1} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus \widehat{R}) \leq \sum_{c \in R_1} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus R^\star), \tag{36}$$

$$\sum_{c \in R'_2} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus \widehat{R}) \leq \varepsilon^{-O(z)} \cdot \sum_{c \in R_2} w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus R^\star). \tag{37}$$

We prove these two inequalities separately below.

Proof of (36). For every $c \in R_1$, by definition, $\text{NN}(c, F^{\text{in}} \setminus R^\star) \notin R^\star$. Moreover, by Property (c), $\text{NN}(c, F^{\text{in}} \setminus R^\star) \notin U$. Since $\widehat{R} \subseteq R^\star \cup U$, it follows that $\text{NN}(c, F^{\text{in}} \setminus R^\star) \notin \widehat{R}$, and hence $\text{NN}(c, F^{\text{in}} \setminus R^\star) \in F^{\text{in}} \setminus \widehat{R}$. Therefore,

$$\text{dist}(c, F^{\text{in}} \setminus \widehat{R}) \leq \text{dist}(c, \text{NN}(c, F^{\text{in}} \setminus R^\star)) = \text{dist}(c, F^{\text{in}} \setminus R^\star).$$

Taking the z-th power, multiplying by $w(c)$, and summing over all $c \in R_1$ proves (36).

Proof of (37). Let $\psi : R'_2 \to R_2$ be an arbitrary bijection (noting that $|R'_2| = |R_2|$). For every $c \in R'_2 \subseteq U$, we have $h(c) \in F^{\text{in}} \setminus (R^\star \cup U) \subseteq F^{\text{in}} \setminus \widehat{R}$ (since $\widehat{R} \subseteq R^\star \cup U$) by Property (d), and $\text{dist}(c, h(c)) \leq 2 \cdot \text{dist}(c, F^{\text{in}} \setminus \{c\})$ by Property (e). Therefore,

$$\text{dist}(c, F^{\text{in}} \setminus \widehat{R}) \leq \text{dist}(c, h(c)) \leq 2 \cdot \text{dist}(c, F^{\text{in}} \setminus \{c\}).$$

Moreover, since $c \in R'_2 \subseteq U \subseteq T_1$ and $\psi(c) \in R_2 = R^\star \setminus T_1$, by the construction of T_1 in Line 5, we have

$$w(c) \cdot \tau^z(c) \leq w(\psi(c)) \cdot \tau^z(\psi(c)).$$

By the definition of τ in Line 1, we then have

$$\begin{aligned}
w(c) \cdot \text{dist}^z(c, F^{\text{in}} \setminus \widehat{R}) &\leq w(c) \cdot 2^z \cdot \text{dist}^z(c, F^{\text{in}} \setminus \{c\}) \\
&\leq w(c) \cdot 2^z \cdot \tau^z(c) \\
&\leq w(\psi(c)) \cdot 2^z \cdot \tau^z(\psi(c)) \\
&\leq w(\psi(c)) \cdot \varepsilon^{-O(z)} \cdot \text{dist}^z(\psi(c), F^{\text{in}} \setminus \{\psi(c)\}) \\
&\leq w(\psi(c)) \cdot \varepsilon^{-O(z)} \cdot \text{dist}^z(\psi(c), F^{\text{in}} \setminus R^\star),
\end{aligned}$$

where the last step follows from $\psi(c) \in R_2 \subseteq R^\star$. Summing the above inequality over all $c \in R'_2$ and using the fact that ψ is a bijection proves (37).

AI Methodology

We used ChatGPT to assist with algorithm development. In particular, ChatGPT suggested some concrete steps that helped implement the iterative kernelization framework: considering a rounding algorithm of [ANS17] and converting its expected-size guarantee into a high-probability bicriteria guarantee (see Algorithms 4 and 5). However, all other major ideas, especially the entire algorithmic framework, the variant of the MP algorithm, and the overall iterative kernelization, were developed by the authors. The authors verified the correctness and originality of all content, including references.

References

[AAH+23] AmirMohsen Ahanchi, Alexandr Andoni, MohammadTaghi Hajiaghayi, Marina Knittel, and Peilin Zhong. Massively parallel tree embeddings for high dimensional spaces. In *SPAA*, pages 77–88. ACM, 2023.

[ABJ+25] Amir Azarmehr, Soheil Behnezhad, Rajesh Jayaram, Jakub Lacki, Vahab Mirrokni, and Peilin Zhong. Massively parallel minimum spanning tree in general metric spaces. In *SODA*, pages 143–174. SIAM, 2025.

[AKPW95] Noga Alon, Richard M. Karp, David Peleg, and Douglas B. West. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.*, 24(1):78–100, 1995.

[ANFSW17] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and Euclidean k-median by primal-dual algorithms. In *FOCS*, pages 61–72. IEEE Computer Society, 2017.

[ANOY14] Alexandr Andoni, Aleksandar Nikolov, Krzysztof Onak, and Grigory Yaroslavtsev. Parallel algorithms for geometric graph problems. In *STOC*, pages 574–583. ACM, 2014.

[ANS17] Hyung-Chan An, Ashkan Norouzi-Fard, and Ola Svensson. Dynamic facility location via exponential clocks. *ACM Trans. Algorithms*, 13(2):21:1–21:20, 2017.

[ARS03] Aaron Archer, Ranjithkumar Rajagopalan, and David B. Shmoys. Lagrangian relaxation for the k-median problem: New insights and continuity properties. In *ESA*, volume 2832 of *Lecture Notes in Computer Science*, pages 31–42. Springer, 2003.

[ASS+18] Alexandr Andoni, Zhao Song, Clifford Stein, Zhengyu Wang, and Peilin Zhong. Parallel graph connectivity in log diameter rounds. In *FOCS*, pages 674–685. IEEE Computer Society, 2018.

[ASZ19] Alexandr Andoni, Clifford Stein, and Peilin Zhong. Log diameter rounds algorithms for 2-vertex and 2-edge connectivity. In *ICALP*, volume 132 of *LIPIcs*, pages 14:1–14:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[AV07] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA*, pages 1027–1035. SIAM, 2007.

[Bar98] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *STOC*, pages 161–168. ACM, 1998.

[BBLM14] MohammadHossein Bateni, Aditya Bhaskara, Silvio Lattanzi, and Vahab Mirrokni. Distributed balanced clustering via mapping coresets. In *NIPS*, pages 2591–2599, 2014.

[BCF25] Sayan Bhattacharya, Martín Costa, and Ermiya Farokhnejad. Fully dynamic k-median with near-optimal update time and recourse. In *STOC*, pages 1166–1177. ACM, 2025.

[BCF+26] Sayan Bhattacharya, Martín Costa, Ermiya Farokhnejad, Shaofeng H.-C. Jiang, Yaonan Jin, and Jianing Lou. Fully dynamic Euclidean k-means. In *FOCS*. IEEE, 2026. To appear. A preliminary version is available at https://arxiv.org/abs/2507.11256.

[BCG+24] Sayan Bhattacharya, Martín Costa, Naveen Garg, Silvio Lattanzi, and Nikos Parotsidis. Fully dynamic k-clustering with fast update time and small recourse. In *FOCS*, pages 216–227. IEEE, 2024.

[BCIS05] Mihai Badoiu, Artur Czumaj, Piotr Indyk, and Christian Sohler. Facility location in sublinear time. In *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 866–877. Springer, 2005.

[BEFM21] MohammadHossein Bateni, Hossein Esfandiari, Manuela Fischer, and Vahab S. Mirrokni. Extreme k-center clustering. In *AAAI*, pages 3941–3949. AAAI Press, 2021.

[BEL13] Maria-Florina Balcan, Steven Ehrlich, and Yingyu Liang. Distributed k-means and k-median clustering on general topologies. In *NIPS*, pages 1995–2003, 2013.

[BFO+27] Sujoy Bhore, Arnold Filtser, Eunjin Oh, Hyeonjun Shin, and Himanshi Singh. Geometric hashing in MPC with applications to clustering and stochastic embeddings. In *SODA*. SIAM, 2027. To appear. A preliminary version is available at https://arxiv.org/abs/2610.21371.

[BGH+26] Jaroslaw Byrka, Yuhao Guo, Yang Hu, Shi Li, Chengzhang Wan, and Zaixuan Wang. k-clustering via iterative randomized rounding. In *FOCS*. IEEE, 2026. To appear. A preliminary version is available at https://arxiv.org/abs/2604.06046.

[BGJ+24] Sayan Bhattacharya, Gramoz Goranci, Shaofeng H.-C. Jiang, Yi Qian, and Yubo Zhang. Dynamic facility location in high dimensional Euclidean spaces. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pages 3763–3775. PMLR / OpenReview.net, 2024.

[BKS17] Paul Beame, Paraschos Koutris, and Dan Suciu. Communication steps for parallel query processing. *J. ACM*, 64(6):40:1–40:58, 2017.

[BMV+12] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable K-Means++. *Proceedings of the VLDB Endowment*, 5(7):622–633, 2012.

[BPR+17] Jaroslaw Byrka, Thomas W. Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Trans. Algorithms*, 13(2):23:1–23:31, 2017.

[BW18] Aditya Bhaskara and Maheshakya Wijewardena. Distributed clustering via LSH based data partitioning. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 569–578. PMLR, 2018.

[CC23] Sam Coy and Artur Czumaj. Deterministic massively parallel connectivity. *SIAM J. Comput.*, 52(5):1269–1318, 2023.

[CCJ+23] Xi Chen, Vincent Cohen-Addad, Rajesh Jayaram, Amit Levi, and Erik Waingarten. Streaming Euclidean MST to a constant factor. In *STOC*, pages 156–169. ACM, 2023.

[CCM23] Sam Coy, Artur Czumaj, and Gopinath Mishra. On parallel k-center clustering. In *SPAA*, pages 65–75. ACM, 2023.

[CDP21] Artur Czumaj, Peter Davies, and Merav Parter. Simple, deterministic, constant-round coloring in congested clique and MPC. *SIAM J. Comput.*, 50(5):1603–1626, 2021.

[CEMN22] Vincent Cohen-Addad, Hossein Esfandiari, Vahab S. Mirrokni, and Shyam Narayanan. Improved approximations for Euclidean k-means and k-median, via nested quasi-independent sets. In *STOC*, pages 1621–1628. ACM, 2022.

[CGGJ25] Artur Czumaj, Guichen Gao, Mohsen Ghaffari, and Shaofeng H.-C. Jiang. Fully scalable MPC algorithms for Euclidean k-center. In *ICALP*, volume 334 of *LIPIcs*, pages 64:1–64:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

[CGJ+24] Artur Czumaj, Guichen Gao, Shaofeng H.-C. Jiang, Robert Krauthgamer, and Pavel Veselý. Fully-scalable MPC algorithms for clustering in high dimension. In *ICALP*, volume 297 of *LIPIcs*, pages 50:1–50:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[CGL+25] Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, Chris Schwiegelshohn, and Ola Svensson. A $(2+\varepsilon)$-approximation algorithm for metric k-median. In *STOC*, pages 615–624. ACM, 2025.

[CGLS23] Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, and Chris Schwiegelshohn. Breaching the 2 LMP approximation barrier for facility location with applications to k-median. In *SODA*, pages 940–986. SIAM, 2023.

[CGTS99] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem. In *STOC*, pages 1–10. ACM, 1999.

[CJK+22] Artur Czumaj, Shaofeng H.-C. Jiang, Robert Krauthgamer, Pavel Veselý, and Mingwei Yang. Streaming facility location in high dimension via geometric hashing. In *FOCS*, pages 450–461. IEEE, 2022.

[CKP26] Vincent Cohen-Addad, Fabian Kuhn, and Zahra Parsaeian. An efficient Massively Parallel constant-factor approximation algorithm for the k-means problem. In *SODA*, pages 5489–5533. SIAM, 2026.

[CKPU23] Mélanie Cambus, Fabian Kuhn, Shreyas Pai, and Jara Uitto. Time and space optimal massively parallel algorithm for the 2-ruling set problem. In *DISC*, volume 281 of *LIPIcs*, pages 11:1–11:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[CL12] Moses Charikar and Shi Li. A dependent LP-rounding approach for the k-median problem. In *ICALP (1)*, volume 7391 of *Lecture Notes in Computer Science*, pages 194–205. Springer, 2012.

[CLN+21] Vincent Cohen-Addad, Silvio Lattanzi, Ashkan Norouzi-Fard, Christian Sohler, and Ola Svensson. Parallel and efficient hierarchical k-median clustering. In *NeurIPS*, pages 20333–20345, 2021.

[CLSS22] Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, and Chris Schwiegelshohn. Towards optimal lower bounds for k-median and k-means coresets. In *STOC*, pages 1038–1051. ACM, 2022.

[CMZ22] Vincent Cohen-Addad, Vahab S. Mirrokni, and Peilin Zhong. Massively parallel k-means clustering for perturbation resilient instances. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 4180–4201. PMLR, 2022.

[CZ24] Yi-Jun Chang and Da Wei Zheng. Fully scalable massively parallel algorithms for embedded planar graphs. In *SODA*, pages 4410–4450. SIAM, 2024.

[DG08] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.

[EIM11] Alina Ene, Sungjin Im, and Benjamin Moseley. Fast clustering using MapReduce. In *KDD*, pages 681–689. ACM, 2011.

[EMMZ22] Alessandro Epasto, Mohammad Mahdian, Vahab S. Mirrokni, and Peilin Zhong. Massively parallel and dynamic algorithms for minimum size clustering. In *SODA*, pages 1613–1660. SIAM, 2022.

[GGJ20] Mohsen Ghaffari, Christoph Grunau, and Ce Jin. Improved MPC algorithms for MIS, matching, and coloring on trees and beyond. In *DISC*, volume 179 of *LIPIcs*, pages 34:1–34:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[GJK+26] Gramoz Goranci, Shaofeng H.-C. Jiang, Peter Kiss, Qihao Kong, Yi Qian, and Eva Szilagyi. Tree embedding in high dimensions: Dynamic and massively parallel. In *SODA*, pages 1181–1213. SIAM, 2026.

[GLM+23] Chetan Gupta, Rustam Latypov, Yannic Maus, Shreyas Pai, Simo Särkkä, Jan Studený, Jukka Suomela, Jara Uitto, and Hossein Vahidi. Fast dynamic programming in trees in the MPC model. In *SPAA*, pages 443–453. ACM, 2023.

[GLS14] Joachim Gehweiler, Christiane Lammersen, and Christian Sohler. A distributed $O(1)$-approximation algorithm for the uniform facility location problem. *Algorithmica*, 68(3):643–670, 2014.

[GP24] Jeff Giliberti and Zahra Parsaeian. Massively parallel ruling set made deterministic. In *DISC*, volume 319 of *LIPIcs*, pages 29:1–29:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[GPST23] Kishen N. Gowda, Thomas W. Pensyl, Aravind Srinivasan, and Khoa Trinh. Improved bi-point rounding algorithms and a golden barrier for k-median. In *SODA*, pages 987–1011. SIAM, 2023.

[GSZ11] Michael T. Goodrich, Nodari Sitchinava, and Qin Zhang. Sorting, searching, and simulation in the MapReduce framework. In *ISAAC*, volume 7074 of *Lecture Notes in Computer Science*, pages 374–383. Springer, 2011.

[GU19] Mohsen Ghaffari and Jara Uitto. Sparsifying distributed algorithms with ramifications in massively parallel computation and centralized local computation. In *SODA*, pages 1636–1653. SIAM, 2019.

[HLW24] Lingxiao Huang, Jian Li, and Xuan Wu. On optimal coreset construction for Euclidean (k, z)-clustering. In *STOC*, pages 1594–1604. ACM, 2024.

[HM04] Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *STOC*, pages 291–300. ACM, 2004.

[HV20] Lingxiao Huang and Nisheeth K. Vishnoi. Coresets for clustering in Euclidean spaces: importance sampling is nearly optimal. In *STOC*, pages 1416–1429. ACM, 2020.

[HZ23] Alireza Haqi and Hamid Zarrabi-Zadeh. Almost optimal massively parallel algorithms for k-center clustering and diversity maximization. In *SPAA*, pages 239–247. ACM, 2023.

[IBY+07] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, and Dennis Fetterly. Dryad: distributed data-parallel programs from sequential building blocks. In *EuroSys*, pages 59–72. ACM, 2007.

[JKPS25] Hongyan Ji, Kishore Kothapalli, Sriram V. Pemmaraju, and Ajitanshu Singh. Fast deterministic massively parallel ruling sets algorithms. In *ICDCN*, pages 152–160. ACM, 2025.

[JMM+03] Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM*, 50(6):795–824, 2003.

[JMNZ24] Rajesh Jayaram, Vahab Mirrokni, Shyam Narayanan, and Peilin Zhong. Massively parallel algorithms for high-dimensional Euclidean minimum spanning tree. In *SODA*, pages 3960–3996. SIAM, 2024.

[JMS02] Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *STOC*, pages 731–740. ACM, 2002.

[JV01] Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.

[KSV10] Howard J. Karloff, Siddharth Suri, and Sergei Vassilvitskii. A model of computation for MapReduce. In *SODA*, pages 938–948. SIAM, 2010.

[LS16] Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. *SIAM J. Comput.*, 45(2):530–547, 2016.

[MP03] Ramgopal R. Mettu and C. Greg Plaxton. The online median problem. *SIAM J. Comput.*, 32(3):816–832, 2003.

[MW26] Nicolas Menand and Erik Waingarten. Streaming and massively parallel algorithms for Euclidean Max-Cut. In *SODA*, pages 3101–3161. SIAM, 2026.

[NS22] Danupon Nanongkai and Michele Scquizzato. Equivalence classes and conditional hardness in massively parallel computations. *Distributed Comput.*, 35(2):165–183, 2022.

[Ona18] Krzysztof Onak. Round compression for parallel graph algorithms in strongly sublinear space. *CoRR*, abs/1807.08745, 2018.

[RVW18] Tim Roughgarden, Sergei Vassilvitskii, and Joshua R. Wang. Shuffles and circuits (on lower bounds for modern parallel computation). *J. ACM*, 65(6):41:1–41:24, 2018.

[Tho04] Mikkel Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM J. Comput.*, 34(2):405–432, 2004.

[Whi15] Tom White. *Hadoop - The Definitive Guide: Storage and Analysis at Internet Scale*. O'Reilly, 4th edition, 2015.

[ZCF+10] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In *HotCloud*. USENIX Association, 2010.