PRIORITY-AWARE EXPLAINABLE CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Explainable clustering aims to represent a given clustering using an interpretable model, such as a decision tree, while approximately preserving the original assignment cost. For k-medians clustering under the ℓ_1 norm, prior work has shown that threshold decision trees achieve a competitive ratio of $\Theta(\log k)$, and that this bound is tight.

In this paper, we introduce *priority-aware clustering*, a refinement of explainable clustering in which different centers have different priorities. We start with the setting in which all centers are split into two groups: a small number m of high-priority centers and $k - m$ remaining centers. Our randomized algorithm outputs a threshold decision tree with $2k$ leaves that yields an improved competitive ratio of $O(\log m)$ for high-priority centers, while preserving an $O(\log k)$ guarantee overall. We further extend our results to settings with multiple priority classes or in which every center has its own priority. In the latter case, the competitive ratio for a center ranked i is $\tilde{O}(\log i)$.

Our results provide a more flexible framework for explainable clustering, enabling interpretable models that offer stronger fidelity guarantees for high-priority clusters while maintaining worst-case guarantees.

1 INTRODUCTION

Clustering is a fundamental tool for finding structure in unlabeled data (Jain, 2010). In k-medians clustering under the ℓ_1 norm, given a finite set of points $X \subseteq \mathbb{R}^d$ and an integer k, the goal is to find k centers $c_1, \ldots, c_k \in \mathbb{R}^d$ that minimize $\sum_{x \in X} \min_i \|x - c_i\|_1$. Standard algorithms for k-medians (Bradley et al., 1997; Arya et al., 2004) output these centers and assign every point to its closest center. While this rule is simple from a computational perspective, it remains nontrivial for humans to interpret and reason about: determining the assigned center for a point $x \in \mathbb{R}^d$ requires computing k distances, involving roughly kd arithmetic operations. This type of calculation can be difficult to perform mentally and does not easily lend itself to intuitive explanations. Interpretability is essential in high-stakes applications, such as healthcare, where users need to understand and verify how a model makes its decisions (Rudin, 2019).

Moshkovitz, Dasgupta, Rashtchian, and Frost (2020) proposed using decision trees to represent k-medians clusterings, thereby providing a more interpretable description of the clustering. Specifically, they proposed to use threshold decision trees to assign points to centers. A threshold decision tree is a binary tree in which each internal node is labeled with a rule of the form $x_i \leq \theta$, where $i \in \{1, \ldots, d\}$ and $\theta \in \mathbb{R}$, and each leaf is labeled with one of the centers $c_1, \ldots, c_k$. For example, at a given node, the rule may be: if $x_5 \leq 7$ then follow the left branch, and otherwise follow the right branch. Each point is routed by these rules from the root to a leaf and is assigned to the center of that leaf. Moshkovitz et al. (2020) considered trees with k leaves, one for each center.

The interpretability of threshold trees stems from two properties. First, the assignment of a point to its cluster is given by the conjunction of simple conditions on the path from the root to the leaf of the point, such as "$x_5 \leq 7$ and $x_2 > 3$". Second, a tree with k leaves has $k - 1$ internal nodes, so each cluster is described by at most $k - 1$ conditions, independently of the dimension d. This makes the explanations of threshold decision trees simple and concise.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Moshkovitz et al. (2020) also proposed evaluating the quality of the decision tree using the standard k-medians objective. The cost of the tree is

$$\sum_{x \in X} \|x - \mathcal{T}(x)\|_1,$$

where $\mathcal{T}(x)$ denotes the center returned by the decision tree for the point x. The cost of the unconstrained clustering with centers $c_1, \ldots, c_k$ is given by

$$\sum_{x \in X} \min_i \|x - c_i\|_1,$$

that is, an unconstrained k-medians assigns every point x to the closest center. The quality of a threshold decision tree is measured by its competitive ratio, defined as the ratio between these two quantities. Moshkovitz et al. (2020) provided an $O(k)$-competitive algorithm and an $\Omega(\log k)$ lower bound for explainable k-medians under the ℓ_1 norm; after a series of improvements (see Section 2), Gupta et al. (2023) established an upper bound of $1 + H_{k-1}$, where H_{k-1} denotes the $(k-1)$st harmonic number, together with a lower bound that matches it up to lower-order terms. In particular, the algorithm of Gupta et al. (2023) takes as input a set of reference centers $c_1, \ldots, c_k$ from a given k-medians solution and outputs a random threshold decision tree $\mathcal{T}$ such that, for every point $x \in \mathbb{R}^d$,

$$\mathbb{E}\big[\|x - \mathcal{T}(x)\|_1\big] \leq \alpha_k \min_i \|x - c_i\|_1, \tag{1}$$

where $\alpha_k = 1 + H_{k-1}$ is the competitive ratio of the algorithm, and the expectation is taken over the randomness in the choice of the decision tree $\mathcal{T}$.

Observe that the guarantee (1) is the same for every point $x \in \mathbb{R}^d$: the factor α_k depends only on the number of centers k, and not on the center to which x is assigned. In some applications, however, errors in the assignment of points from some clusters are more costly than errors for other clusters. For example, consider a clustering of breast tumors in which some clusters consist mainly of malignant tumors and the others mainly of benign tumors. Assigning a malignant tumor to a cluster of benign tumors is a more serious error than the reverse, since it can lead to a missed cancer diagnosis. Therefore, it is desirable that the threshold tree preserves the assignments of points in the clusters of malignant tumors more faithfully than the assignments of other points. This suggests the following question: can an interpretable decision tree provide stronger guarantees for points assigned to selected centers, while still retaining the standard $O(\log k)$ guarantee for all points?

We study this question in a setting that we call *priority-aware clustering*. The reference centers are fixed and are assigned priorities, and the decision tree is required to provide stronger guarantees for higher-priority centers.

Our results. First, we design an algorithm that is given a set of k reference centers C that is partitioned into 2 priority classes: a set H of high-priority centers and a set L of low-priority centers. Let $|H| = m$ and $|L| = k - m$, where m is much smaller than k. Our algorithm samples a decision tree with $2k$ leaves from a carefully designed distribution such that, for every $x \in \mathbb{R}^d$, the following guarantees hold:

$$\mathbb{E}\big[\|x - \mathcal{T}(x)\|_1\big] \leq O(\log m) \min_{c_i \in H} \|x - c_i\|_1,$$

and

$$\mathbb{E}\big[\|x - \mathcal{T}(x)\|_1\big] \leq O(\log k) \min_i \|x - c_i\|_1.$$

In fact, we obtain a stronger result that extends to the setting of $s \geq 2$ priority classes $P_1, P_2, \ldots, P_s$, where P_1 is the highest-priority class and P_i has higher priority than P_j if and only if $i < j$. We prove the following theorem.

Theorem 1.1. *There exists a polynomial-time algorithm that takes as input a set of k reference centers $C \subseteq \mathbb{R}^d$ partitioned into $s \geq 2$ priority classes $P_1, P_2, \ldots, P_s$ such that $|P_1| \leq |P_2| \leq \cdots \leq |P_s|$, and outputs a threshold tree $\mathcal{T}$ with sk leaves such that for every $x \in \mathbb{R}^d$, if c is the closest center to x in the reference clustering and $c \in P_i$, then*

$$\mathbb{E}\big[\|x - \mathcal{T}(x)\|_1\big] \leq O(\log s \log |P_i|)\, \|x - c\|_1.$$

We then turn to the fully-ordered setting in which all centers $c_1, \ldots, c_k$ are totally ordered by priority, with c_1 having the highest priority, followed by c_2, and so on. We prove the following theorem.

Theorem 1.2. *There exists a polynomial-time algorithm that, given an ordered list of reference centers $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$ in order of decreasing priority, returns a threshold tree $\mathcal{T}$ such that the following holds for every $\varepsilon > 0$. The number of leaves of $\mathcal{T}$ is at most*

$$O_\varepsilon\big(k \cdot (\log\log k)^{1+\varepsilon}\big)$$

with probability 1, and for every $x \in \mathbb{R}^d$ and every $i \in [k]$,

$$\mathbb{E}\big[\|x - \mathcal{T}(x)\|_1\big] \;\le\; O_\varepsilon\big(\log i \cdot (\log\log\log i)^{1+\varepsilon}\big)\, \|x - c_i\|_1.$$

For the case of two priority classes, the number of leaves is at most $2k$, while in the fully ordered setting it is $O_\varepsilon(k(\log\log k)^{1+\varepsilon})$. In Appendix D, we show that such an increase is in fact necessary: if the number of leaves is restricted to exactly k, then it is impossible to improve the competitive ratio even for a single high-priority center c. We note that the idea to design bi-criteria algorithms that use more than k leaves to improve the guarantees of the resulting explainable clustering is not new in the literature, as we discuss in the next section. Observe that a tree with $c(k) \cdot k$ leaves for a slowly growing function $c(k)$ still has depth that is independent of the dimension d.

2 Prior Work

There are two main approaches to interpretability in machine learning (Lipton, 2018; Murdoch et al., 2019; Rudin, 2019). The first explains the decisions of a black-box model after the model has been trained, for example, by approximating the model locally with an interpretable model (Ribeiro et al., 2016) or by assigning to each feature an importance value for a given prediction (Lundberg and Lee, 2017). The second uses models that are inherently interpretable, an approach that Rudin (2019) advocates for high-stakes decisions. Decision trees are a standard example of inherently interpretable models (Breiman et al., 1984; Quinlan, 1986). Their interpretability depends on their size, and several works study how to construct small or sparse trees (Bertsimas and Dunn, 2017; Hu et al., 2019; Souza et al., 2022).

Explainable clustering follows the second approach: the clustering is given by a threshold tree. Moshkovitz et al. (2020) gave an $O(k)$-competitive algorithm and an $\Omega(\log k)$ lower bound for explainable k-medians under the ℓ_1 norm. Their upper bound was improved to $O(\log^2 k)$ by Gamlath et al. (2021), to $\tilde{O}(\log k)$ by Makarychev and Shan (2021) and Esfandiari, Mirrokni, and Narayanan (2022), and to $O(\log k)$ by Gupta, Pittu, Svensson, and Yuan (2023) and Makarychev and Shan (2023). Gupta et al. (2023) further showed an upper bound of $1 + H_{k-1}$, which is tight up to lower-order terms. Explainable k-means and other variants of the problem were studied by Laber and Murtinho (2021); Charikar and Hu (2022); Bandyapadhyay et al. (2022); Laber et al. (2023).

Several works study bi-criteria algorithms, which use more than k leaves to obtain a better competitive ratio. Frost et al. (2020) proposed ExKMC, which expands a threshold tree with k leaves to a tree with $k' > k$ leaves in order to reduce the k-means cost. For explainable k-means, there are instances on which every threshold tree with k leaves has competitive ratio $\Omega(k)$ (Gamlath et al., 2021), and the best known upper bound for such trees is $O(k \log\log k)$ (Gupta et al., 2023). Makarychev and Shan (2022) showed that with $(1+\delta)k$ leaves, the competitive ratio improves to $O(\delta^{-1}\log^2 k \log\log k)$, and gave an almost matching lower bound of $\Omega(\delta^{-1}\log^2 k)$.

3 Preliminaries

Let $C = \{c_1, \ldots, c_k\} \subseteq \mathbb{R}^d$ be the set of reference centers. The center of a point $x \in \mathbb{R}^d$ in the reference clustering is its closest center in C (with ties broken arbitrarily).

A *threshold cut* is a pair $\omega = (i, \theta)$, where $i \in [d]$ and $\theta \in \mathbb{R}$. It divides $\mathbb{R}^d$ into the half-spaces $\{x : x_i \le \theta\}$ and $\{x : x_i > \theta\}$. The cut ω *separates* two points $y, z \in \mathbb{R}^d$ if they lie in different half-spaces, that is, if $\min(y_i, z_i) \le \theta < \max(y_i, z_i)$.

A *threshold tree* $\mathcal{T}$ is a rooted binary tree in which every internal node u is labeled with a threshold cut $\omega_u = (i_u, \theta_u)$ and every leaf is labeled with a center in C. Every node u is associated with a region $B_u \subseteq \mathbb{R}^d$: the region of the root is $\mathbb{R}^d$, and if u is an internal node with left child u_L and right child u_R, then $B_{u_L} = \{x \in B_u : x_{i_u} \le \theta_u\}$ and $B_{u_R} = \{x \in B_u : x_{i_u} > \theta_u\}$. Thus, B_u is

the axis-aligned rectangular region of all points that satisfy the conditions on the path from the root to u, and the regions of the leaves partition $\mathbb{R}^d$. For a point $x \in \mathbb{R}^d$, we denote by $\mathcal{T}(x)$ the center that labels the leaf v with $x \in B_v$; equivalently, x is routed from the root to v, and at every internal node u on this path it moves to u_L if $x_{i_u} \le \theta_u$ and to u_R otherwise. For every node u, we denote by $C_u = C \cap B_u$ the set of centers contained in u.

In priority-aware clustering, C is partitioned into priority classes $P_1, \dots, P_s$, where P_1 has the highest priority. For $s = 2$, we denote the high-priority class by H, the low-priority class by L, and let $m = |H|$. In the fully ordered setting, every center has its own priority, and we assume without loss of generality that $c_1 \preceq c_2 \preceq \cdots \preceq c_k$, where $c \preceq c'$ means that the priority of c is at least the priority of c'. The algorithms that we consider take the centers and their priorities as input and output a random threshold tree $\mathcal{T}$ whose leaves are labeled with centers in C. We say that $\mathcal{T}$ has competitive ratio α for a center $c \in C$ if $\mathbb{E}\big[\|x - \mathcal{T}(x)\|_1\big] \le \alpha \, \|x - c\|_1$ for every $x \in \mathbb{R}^d$ whose closest center in C is c. The goal is to obtain smaller competitive ratios for centers of higher priority.

Algorithm RANDOMCUTS

Input: k reference centers $C \subseteq [0,1]^d$
Output: A threshold tree $\mathcal{T}$ with k leaves

1: Create the root r of the tree $\mathcal{T}$ containing C.
2: Let $t = 1$.
3: **while** $\mathcal{T}$ has strictly less than k leaves **do**
4: Sample a coordinate i_t in $[d]$ uniformly at random.
5: Sample a threshold θ_t in $[0,1]$ uniformly at random.
6: **for every** leaf v of $\mathcal{T}$ that contains at least 2 centers **do**
7: **if** ω_t separates any 2 centers in C_v **then**
8: Attach two new children u_L, u_R to v with

$$C_{u_L} = \{c \in C_v : \; c_{i_t} \le \theta_t\} \text{ and } C_{u_R} = \{c \in C_v : \; c_{i_t} > \theta_t\}$$

9: Update $t = t + 1$.

Figure 1: Algorithm RANDOMCUTS for explainable k-medians in ℓ_1

Our priority aware algorithms are a refinement of the RANDOMCUTS algorithm (Algorithm 1). To simplify the exposition, we assume that all reference centers in C lie in the unit cube $[0,1]^d$.[1] The algorithm constructs a threshold tree in a top-down manner, starting from a root node that contains all centers in C, and iteratively applying random threshold cuts until the tree has exactly k leaves. At every iteration $t \ge 1$, while the current tree has fewer than k leaves, the algorithm samples a threshold cut $\omega_t = (i_t, \theta_t)$, where the coordinate $i_t \in [d]$ and the threshold $\theta_t \in [0,1]$ are drawn independently and uniformly at random. For every leaf v with $|C_v| \ge 2$, if ω_t separates two centers in C_v, the algorithm labels v with ω_t and splits it into two children. When the algorithm terminates, the tree has exactly k leaves, and every leaf contains exactly one center, which is its label.

This algorithm and its variants have been extensively studied in prior work (Gamlath et al., 2021; Makarychev and Shan, 2021; Gupta et al., 2023; Makarychev and Shan, 2023). RANDOMCUTS has competitive ratio $O(\log k)$ for every center, which is optimal up to a constant factor (Gupta et al., 2023; Makarychev and Shan, 2023). That is, the tree $\mathcal{T}$ returned by RANDOMCUTS satisfies, for every $x \in \mathbb{R}^d$, $\mathbb{E}\big[\|x - \mathcal{T}(x)\|_1\big] \le O(\log k) \min_{j \in [k]} \|x - c_j\|_1$.

[1] This assumption is without loss of generality: we can scale and recenter the instance so that it lies in the unit cube. Both RANDOMCUTS and our algorithms can also be implemented without rescaling or recentering.

4 PRIORITY-AWARE EXPLAINABLE CLUSTERING WITH MULTIPLE PRIORITY CLASSES

In this section, we present our algorithm for priority-aware explainable clustering with $s \geq 2$ priority classes that satisfies Theorem 1.1.

Our algorithm builds upon RANDOMCUTS. Although RANDOMCUTS has a strong approximation guarantee, it does not suffice for our purposes: the competitive ratio is the same for every data point ($O(\log k)$) and does not depend on the priority of the reference center. Before presenting our algorithm, we discuss two natural baseline approaches for obtaining priority-aware explainable clustering. While both approaches may appear promising, they lead to suboptimal trade-offs between competitive ratio guarantees and tree size, even in the case of two priority classes H (high priority) and L (low priority).

The first approach is to run RANDOMCUTS independently on the sets of high-priority and low-priority centers, obtaining trees $\mathcal{T}_H$ and $\mathcal{T}_L$. For each point x, we then select the better of $\mathcal{T}_H(x)$ and $\mathcal{T}_L(x)$. This approach achieves the desired approximation guarantees; however, the resulting decision tree has $\Theta(|H| \cdot |L|)$ leaves, which is significantly larger than the size achieved by our algorithm.

The second approach is to run RANDOMCUTS on the set of high-priority centers, producing a tree with m leaves that partitions the space into m rectangular regions. For each region, we identify the subset of low-priority centers it contains and apply RANDOMCUTS within that region. With a careful implementation, this yields a tree with $2k$ leaves and an $O(\log |H|)$ competitive ratio for high-priority centers. However, the competitive ratio for low-priority centers can be unbounded, as illustrated by the following example in $\mathbb{R}^2$.

For $M \gg 1$, we give an example where the competitive ratio is at least M. Consider two high-priority centers with coordinates $(M, 0)$ and $(M, 1)$, and one low-priority center c^* at $(0, 1)$. Let $x = (0, 0)$. The closest center to x is c^* at ℓ_1 distance 1. The RANDOMCUTS algorithm must first separate the high-priority centers $(M, 0)$ and $(M, 1)$. In doing so, it separates x from its closest center. The only remaining center in the region containing x is $(M, 0)$, which is at distance M from x. Thus, $\|x - \mathcal{T}(x)\|_1 / \|x - c^*\|_1 = M$.

Our algorithm. We now describe our algorithm PRIORITREE (see Algorithm 2). PRIORITREE takes as input priority classes $P_1, \ldots, P_s$ that partition the reference centers $C \subseteq [0, 1]^d$, and proceeds in three phases.

In the first phase (Line 1), we run RANDOMCUTS(C) to obtain a threshold tree $\mathcal{T}$ with exactly k leaves, each containing a single reference center. In the second phase, we expand each leaf of $\mathcal{T}$ so that it contains one representative center from each priority class. In particular, fix a leaf v of $\mathcal{T}$, and let c_j be the unique center it contains. Suppose that c_j belongs to class P_{i^*}. We initialize $\widetilde{C}_v = \{c_j\}$ and, for each priority class P_i with $i \neq i^*$, we add one representative center from P_i to $\widetilde{C}_v$ as follows (Line 10). Let $u_1, u_2, \ldots, u_T$ be the root-to-leaf path to v in $\mathcal{T}$, so that u_1 is the root and $u_T = v$. Let u be the deepest node along this path that satisfies $C_u \cap P_i \neq \emptyset$, i.e. $u = u_z$ for the largest $z \in [T]$ that satisfies this property. We sample $\bar{c}_i$ uniformly at random from $C_u \cap P_i$ and add it to $\widetilde{C}_v$. After this step, $|\widetilde{C}_v| = s$.

In the third phase, we construct a threshold tree on the s centers in $\widetilde{C}_v$ using RANDOMCUTS$(\widetilde{C}_v)$ (Line 11) and attach it to the leaf v. The resulting tree $\mathcal{T}$ has sk leaves.

In the next section, we analyze the PRIORITREE algorithm and prove Theorem 1.1.

We now consider the setting in which all reference centers are totally ordered by priority: $c_1 \preccurlyeq c_2 \preccurlyeq \cdots \preccurlyeq c_k$, where c_1 is the center with the highest priority, and c_k is the center with the lowest priority. We partition the centers $c_1, \ldots, c_k$ into $O(\log \log k)$ priority classes. The first class is $P_1 = \{c_1\}$, and for each $i \geq 2$, the class P_i consists of the next $2^{2^{i-1}}$ centers, except possibly for the final class, which contains all remaining centers. We then use PRIORITREE with the priority classes P_i to construct a decision tree.

Algorithm PRIORITREE

Input: Priority classes $P_1, P_2, \ldots, P_s$ that partition a set of reference centers $C \subseteq \mathbb{R}^d$, with P_1 being the highest-priority class.
Output: A threshold tree $\mathcal{T}$.

Run RANDOMCUTS(C) to obtain an initial tree $\mathcal{T}_1$.
Initialize $\mathcal{T} = \mathcal{T}_1$.
for every leaf v of $\mathcal{T}_1$ **do**
    Let c_j be the center assigned to v and let P_{i^*} be its priority class. Set $\bar{c}_{i^*} = c_j$.
    Let $\widetilde{C}_v = \{\bar{c}_{i^*}\}$.
    Let $u_1, u_2, \ldots, u_T$ be the root-to-leaf path in $\mathcal{T}_1$ with u_1 the root and $u_T = v$.
    for each $i \in [s] \setminus \{i^*\}$ **do**
        Let u be the node u_t with the largest index t such that $C_{u_t} \cap P_i \neq \varnothing$.
        Sample $\bar{c}_i$ uniformly at random from $C_u \cap P_i$.
        Add $\bar{c}_i$ to $\widetilde{C}_v$.
    Build a decision tree for the centers in $\widetilde{C}_v$ using RANDOMCUTS, and attach the resulting tree to the leaf v of $\mathcal{T}$.
return $\mathcal{T}$.

Figure 2: Priority-Aware Explainable k-medians clustering with s priority classes

The guarantees of this algorithm are given in Theorem 4.1 and are slightly weaker than those in Theorem 1.2. In particular, the competitive ratio α_i for points assigned to c_i depends not only on its rank i, but also on k; though, this dependence is very mild.

Theorem 4.1. *There exists a polynomial-time algorithm that, given an ordered list of reference centers $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$ in order of decreasing priority, returns a threshold tree $\mathcal{T}$ with the following guarantees. The tree $\mathcal{T}$ has $O(k \log\log k)$ leaves with probability 1, and for every $x \in \mathbb{R}^d$ and every $i \in [k]$,*

$$\mathbb{E}\big[\|x - \mathcal{T}(x)\|_1\big] \;\leq\; O\big(\log i \cdot \log\log\log k\big)\, \|x - c_i\|_1.$$

In Appendix C, we present a refined algorithm, RANKEDPRIORITREE, for the case of ranked centers and show that it satisfies the guarantees of Theorem 1.2. The algorithm constructs a decision tree by carefully defining priority classes recursively applying RANKEDPRIORITREE to these classes.

5 ANALYSIS OF THE PRIORITY-AWARE EXPLAINABLE CLUSTERING ALGORITHM

In this section, we prove Theorem 1.1 and introduce the notion of Set Elimination Games.

5.1 SET ELIMINATION GAME

To analyze the priority-aware explainable clustering algorithm, we define a variant of the Set Elimination Game introduced in Makarychev and Shan (2023). This variant of the game models the execution of our randomized priority-aware algorithm.

Let (Ω, μ) be a finite measure space. Consider k non-empty sets $S_1, S_2, \ldots, S_k \subseteq \Omega$ with $\mu(S_i) > 0$. There are two types of sets: m *high-priority* sets, with indices $I_H \subseteq [k]$, and $k - m$ *low-priority* sets, with indices $I_L \subseteq [k]$.

The Set Elimination Game evolves in rounds. In the first round, all k sets enter a *competition* and in each of the following rounds, some sets are eliminated, until we reach the end of the game, where some sets are declared *winners*. The sets that are not eliminated at the start of a round $n \geq 1$ are called the *remaining sets*, and the set of their indices is denoted by $\mathcal{R}_n$. The game starts with $\mathcal{R}_1 = [k]$ and in every round $n \geq 1$ an element ω is sampled from Ω according to the probability

measure $\mu/\mu(\Omega)$ i.e., $\Pr(\omega \in A) = \mu(A)/\mu(\Omega)$ for every measurable set $A \subseteq \Omega$. Then, the remaining sets are updated according to the following rules.

1. If for every $i \in \mathcal{R}_n$, $\omega \in S_i$, then ω is ignored and $\mathcal{R}_{n+1} = \mathcal{R}_n$.
2. If there exists $i \in \mathcal{R}_n$ such that $\omega \notin S_i$, then all remaining sets that contain ω are removed from the game and so $\mathcal{R}_{n+1} = \{i \in \mathcal{R}_n : \omega \notin S_i\}$.

The game ends in the first round n^* in which at most one high-priority set remains, i.e., $|\mathcal{R}_{n^*} \cap I_H| \leq 1$. The winners are determined at the start of round n^*. Formally, the *winners* of the Set Elimination Game form a set $W \subseteq [k]$ such that:

1. If no high-priority sets remain at the start of round n^*, then $W = \mathcal{R}_{n^*-1} \cap I_H$, i.e., all high-priority sets eliminated in the previous round.
2. If exactly one high-priority set remains at round n^*, then $W = \mathcal{R}_{n^*} \cap I_H$.

For any subset $I \subseteq [k]$, let $\bar{\mu}(I) = \frac{\sum_{i \in I} \mu(S_i)}{|I|}$ be the average measure of the sets in I. The *cost of the game* is defined as $\bar{\mu}(W)$. Although the game terminates at round n^*, for the sake of analysis we imagine that the elimination process continues for infinitely many rounds beyond n^*.

The Set Elimination Game in Makarychev and Shan (2023) follows the same elimination rules, but considers only one type of sets and always produces a single winner: the last remaining set. In contrast, our version stops at a stopping time and may produce multiple winners. Although the elimination dynamics are unchanged, this modification substantially changes the analysis. The main technical difficulty is that stopping the process before a unique surviving set remains creates dependencies among the winning sets that do not arise in the original game. Handling these dependencies requires a new argument tailored to the multi-winner setting.

In Appendix A, we prove the following theorem that upper bounds the cost of the game in expectation.

Theorem 5.1. *Consider a Set Elimination Game with $k \geq 1$ sets and $1 \leq m \leq k$ high-priority sets. Then*

$$\mathbf{E}[\bar{\mu}(W)] \leq O(\log m) \min_{j \in I_H} \mu(S_j).$$

5.2 Proof of Theorem 1.1

In this section, we prove Theorem 1.1 via a reduction to the Set Elimination Game. The proof relies on the following key lemma.

Lemma 5.2. *Let $C \subseteq \mathbb{R}^d$ be a set of k centers partitioned into two classes: the high-priority class H and the low-priority class L. Fix a center $c \in H$ and a point $x \in [0,1]^d$ whose closest center in C is c. Let $\mathcal{T}$ be the tree produced by* RandomCuts(C)*, and let v be the leaf of $\mathcal{T}$ to which x is routed. Consider the path from the root of $\mathcal{T}$ to v, and let u be the deepest node along this path such that $C_u \cap H \neq \varnothing$. Let $\bar{c}$ be a center sampled uniformly at random from $C_u \cap H$.*

Then, for some absolute constant $\beta > 0$, $\mathbb{E}[\|x - \bar{c}\|_1] \leq \beta \log |H| \cdot \|x - c\|_1$.

Before we prove Lemma 5.2, we show how it can be used to obtain a proof for Theorem 1.1.

Proof of Theorem 1.1. Let $\mathcal{T}_1$ be the initial threshold tree produced by RandomCuts(C) in Line 1.

We first verify that the final tree $\mathcal{T}$ has sk leaves. The tree $\mathcal{T}_1$ has exactly k leaves. After adding one representative from each priority class to every leaf of $\mathcal{T}_1$, each leaf is associated with exactly s centers. In the final phase (Line 11), each leaf is replaced by a subtree with s leaves. Therefore, the resulting tree $\mathcal{T}$ has sk leaves.

We now fix a point $x \in \mathbb{R}^d$ and bound $\mathbb{E}[\|x - \mathcal{T}(x)\|_1]$. Fix a priority class P_i, and let c be the closest center to x in P_i. Let v be the leaf of $\mathcal{T}_1$ to which x is routed, and let $\widetilde{C}_v$ be the set of centers associated with v after adding one representative from each priority class at Line 10. Let $\bar{c}_i \in \widetilde{C}_v$ denote the representative of class P_i in v.

Applying Lemma 5.2 with high-priority class $H = P_i$ and low-priority class $L = \bigcup_{j\in[s]\setminus\{i\}} P_j$, we obtain

$$\mathbb{E}[\|x - \bar{c}_i\|_1] \le \beta \log |P_i| \cdot \|x - c\|_1. \tag{2}$$

Since RANDOMCUTS is applied to the s centers in C_v at Line 11, the upper bound of $O(\log s)$ on its competitive ratio from Gupta et al. (2023); Makarychev and Shan (2023) implies that

$$\mathbb{E}[\|x - \mathcal{T}(x)\|_1 \mid \widetilde{C}_v] \le O(\log s) \min_{c'\in\widetilde{C}_v} \|x - c'\|_1 \le O(\log s)\|x - \bar{c}_i\|_1. \tag{3}$$

Taking expectations on both sides of (3) and combining with (2), we obtain

$$\mathbb{E}[\|x - \mathcal{T}(x)\|_1] \le O(\log s \, \log |P_i|)\|x - c\|_1.$$

□

We proceed to prove Lemma 5.2.

Proof of Lemma 5.2. Denote the centers in C by $c_1, c_2, \ldots, c_k$. For each iteration t of the RANDOMCUTS algorithm, let u_t be the leaf whose associated region in $\mathbb{R}^d$ contains x. Then u_1 is the root of $\mathcal{T}$ and $u_T = v$. Let $u^* = u_{t^*}$, where t^* is the largest index $t \in [T]$ such that $C_{u_t} \cap H \neq \varnothing$.

Then $\bar{c}$ is sampled uniformly at random from $C_{u^*} \cap H$. By conditioning on C_{u^*}, we obtain

$$\mathbb{E}\big[\|x - \bar{c}\|_1\big] = \mathbb{E}\big[\mathbb{E}[\|x - \bar{c}\|_1 \mid C_{u^*}]\big] = \mathbb{E}\left[\frac{\sum_{j:c_j\in C_{u^*}\cap H} \|x - c_j\|_1}{|C_{u^*} \cap H|}\right]. \tag{4}$$

To analyze this expression, we will construct an instance of the Set Elimination Game (see Section 5.1) with the following properties:

1. For every $j \in [k]$, $\mu(S_j) = \|x - c_j\|_1$.
2. The set of high-priority indices is $I_H = \{j \in [k] : c_j \in H\}$.
3. The random set $\{j \in [k] : c_j \in C_{u^*} \cap H\}$ has the same distribution as W, where W denotes the set of winners of the Set Elimination Game.

Set Elimination Game Instance. We now define the Set Elimination Game instance and show that it satisfies Properties 1, 2, and 3. Let (Ω, μ) be the measure space in which Ω is the set of all threshold cuts in $[d] \times [0, 1]$, namely

$$\Omega = \{(\ell, \theta) : \ell \in [d],\ \theta \in [0, 1]\}.$$

For a measurable set $S \subseteq \Omega$, let

$$\mu(S) = \sum_{\ell=1}^{d} \mu_L\big(\{\theta \in [0, 1] : (\ell, \theta) \in S\}\big),$$

where μ_L denotes Lebesgue measure. For every $j \in [k]$, let S_j be the set of cuts that separate x from c_j:

$$S_j = \{(\ell, \theta) \in \Omega : \operatorname{sgn}(x_\ell - \theta) \neq \operatorname{sgn}((c_j)_\ell - \theta)\}.$$

Here y_ℓ denotes the ℓ-th coordinate of a vector $y \in \mathbb{R}^d$. We define the set of indices corresponding to high-priority sets by $I_H = \{j \in [k] : c_j \in H\}$.

By construction, $\mu(S_j) = \|x - c_j\|_1$ for every $j \in [k]$, establishing Property 1. The set of high-priority indices is defined as in Property 2. It remains to prove Property 3, which will complete the proof of Lemma 5.2.

The set of winners W of the Set Elimination Game and the set of indices from which $\bar{c}$ is selected, namely $\{j \in [k] : c_j \in C_{u^*} \cap H\}$, are both random subsets of $[k]$. To show that they have the same distribution, we construct a coupling under which these two sets are equal with probability 1.

Consider an execution of the algorithm RANDOMCUTS(C), and let $(i_1, \theta_1), (i_2, \theta_2), \ldots$ be the sequence of threshold cuts generated during this execution. We define the element selected in round n of the Set Elimination Game to be $\omega_n = (i_n, \theta_n)$. Then each ω_n is distributed uniformly over $[d] \times [0, 1]$ and independently of all other ω_i, as required by the definition of the Set Elimination Game.

Let $Q_n = \{j \in [k] : c_j \in C_{u_n}\}$ be the set of indices of the centers that lie in the same leaf as x before the n-th threshold cut is sampled by the RANDOMCUTS algorithm. To prove Property 3, it suffices to show that $Q_{t^*} \cap I_H = W$, where $I_H = \{j : c_j \in H\}$ as in Property 2. We show that this fact follows from the following claim.

Claim 5.3. *For every $n \in [T]$, it holds that $\mathcal{R}_n = Q_n$, where $\mathcal{R}_n$ are the indices of the remaining sets of the Set Elimination Game at the start of round n.*

Proof of Claim 5.3. We prove the claim by induction on n.

For $n = 1$, observe that $Q_1 = \{j \in [k] : c_j \in C_{u_1}\} = [k] = \mathcal{R}_1$, where we used that u_1 is the root of $\mathcal{T}$ and so $C_{u_1} = C$. Now fix $n \in \{1, \ldots, T-1\}$ and assume that $\mathcal{R}_n = Q_n$. We show that $\mathcal{R}_{n+1} = Q_{n+1}$. Consider the following two cases.

Case I: For every $j \in \mathcal{R}_n = Q_n$, we have $\omega_n \in S_j$. Then, for every center $c_j \in C_{u_n}$, the cut (i_n, θ_n) separates all centers in C_{u_n} from x and so it is ignored by the RANDOMCUTS algorithm (see Line 7). Thus $C_{u_{n+1}} = C_{u_n}$ and so $Q_{n+1} = Q_n$. Also, by the rules of the Set Elimination Game, $\mathcal{R}_{n+1} = \mathcal{R}_n$ and thus $Q_{n+1} = \mathcal{R}_{n+1}$.

Case II. There exists a $j \in \mathcal{R}_n = Q_n$ such that $\omega_n \notin S_j$. Then, $\mathcal{R}_{n+1} = \mathcal{R}_n \setminus B$, where $B = \{j \in \mathcal{R}_n : \omega_n \in S_j\}$. If $B = \varnothing$, by definition of ω_n, the cut (i_n, θ_n) separates no centers in C_{u_n} from x, and thus it is ignored by the algorithm, so $\mathcal{R}_{n+1} = Q_{n+1}$. If $B \neq \varnothing$, then the centers $\{c_j : j \in B\}$ are separated from x and thus: $Q_{n+1} = Q_n \setminus B = \mathcal{R}_n \setminus B = \mathcal{R}_{n+1}$. □

We consider two cases: $t^* < T$ and $t^* = T$. If $t^* < T$, then by definition of t^*, we have $C_{u_{t^*}} \cap H \neq \varnothing$ and $C_{u_{t^*+1}} \cap H = \varnothing$. By Claim 5.3, it follows that $\mathcal{R}_{t^*+1} \cap I_H = Q_{t^*+1} \cap I_H = \varnothing$, whereas $\mathcal{R}_{t^*} \cap I_H = Q_{t^*} \cap I_H \neq \varnothing$. Therefore, $W = \mathcal{R}_{t^*} \cap I_H = Q_{t^*} \cap I_H$.

If $t^* = T$, then the leaf $u_{t^*} = v$ of $\mathcal{T}$ contains exactly one center from H; otherwise, RANDOMCUTS would not terminate. Hence, $|Q_{t^*} \cap I_H| = |\mathcal{R}_{t^*} \cap I_H| = 1$. By the definition of the Set Elimination Game, if exactly one high-priority set remains, then it is declared the winner. Therefore, $W = \mathcal{R}_{t^*} \cap I_H = Q_{t^*} \cap I_H$.

Final step – bounding the competitive ratio. We now express the right-hand side of (4) in terms of the Set Elimination Game:

$$\mathbb{E}[\|x - \bar{c}\|_1] = \mathbb{E}\left[\frac{\sum_{j:c_j \in C_{u^*} \cap H} \|x - c_j\|_1}{|C_{u^*} \cap H|}\right] = \mathbb{E}\left[\frac{\sum_{j \in W} \mu(S_j)}{|W|}\right] \leq O(\log |H|) \min_{c' \in H} \|x - c'\|_1.$$

The second equality follows from properties 1–3, and the inequality follows from Theorem 5.1. Since $c \in H$, we have $\min_{c' \in H} \|x - c'\|_1 \leq \|x - c\|_1$, which completes the proof. □

AI USE STATEMENT

In this work, we used generative AI tools to check proofs written by the authors for errors. We have not used generative AI tools to formulate mathematical claims, to develop theoretical models or conceptual frameworks, to provide critical ingredients for proofs, to propose or refine hypotheses, or to interpret results, and the remaining tasks with required disclosure (synthetic data, datasets, experiments, implementation, translation, and qualitative data analysis) are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on Computing*, 33(3):544–562, 2004. doi: 10.1137/S0097539702416402.

Sayan Bandyapadhyay, Fedor Fomin, Petr A. Golovach, William Lochet, Nidhi Purohit, and Kirill Simonov. How to find a good explanation for clustering? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3904–3912, 2022.

Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, 2017.

Paul S. Bradley, Olvi L. Mangasarian, and W. Nick Street. Clustering via concave minimization. In *Advances in Neural Information Processing Systems 9*, pages 368–374. MIT Press, 1997.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

Moses Charikar and Lunjia Hu. Near-optimal explainable k-means for all dimensions. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2580–2606. SIAM, 2022.

Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Almost tight approximation algorithms for explainable clustering. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2641–2663. SIAM, 2022.

Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. ExKMC: Expanding explainable k-means clustering. *arXiv preprint arXiv:2006.02399*, 2020.

Buddhima Gamlath, Xinrui Jia, Adam Polak, and Ola Svensson. Nearly-tight and oblivious algorithms for explainable clustering. *Advances in Neural Information Processing Systems*, 34: 28929–28939, 2021.

Anupam Gupta, Madhusudhan Reddy Pittu, Ola Svensson, and Rachel Yuan. The price of explainability for clustering. In *Proceedings of the 64th IEEE Annual Symposium on Foundations of Computer Science*, pages 1131–1148. IEEE, 2023.

Xiyang Hu, Cynthia Rudin, and Margo Seltzer. Optimal sparse decision trees. *Advances in Neural Information Processing Systems*, 32, 2019.

Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. doi: 10.1016/j.patrec.2009.09.011.

John Frank Charles Kingman. *Poisson processes*, volume 3. Clarendon Press, 1992.

Eduardo Laber, Lucas Murtinho, and Felipe Oliveira. Shallow decision trees for explainable k-means clustering. *Pattern Recognition*, 137:109239, 2023.

Eduardo S. Laber and Lucas Murtinho. On the price of explainability for some clustering problems. In *Proceedings of the International Conference on Machine Learning*, pages 5915–5925. PMLR, 2021.

Zachary C. Lipton. The mythos of model interpretability. *Communications of the ACM*, 61(10): 36–43, 2018. doi: 10.1145/3233231.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774, 2017.

Konstantin Makarychev and Liren Shan. Near-optimal algorithms for explainable k-medians and k-means. In *Proceedings of the International Conference on Machine Learning*, pages 7358–7367. PMLR, 2021.

Konstantin Makarychev and Liren Shan. Explainable k-means: Don't be greedy, plant bigger trees! In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1629–1642, 2022.

Konstantin Makarychev and Liren Shan. Random cuts are optimal for explainable k-medians. *Advances in Neural Information Processing Systems*, 36:66890–66901, 2023.

Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, 2017.

Michal Moshkovitz, Sanjoy Dasgupta, Cyrus Rashtchian, and Nave Frost. Explainable k-means and k-medians clustering. In *International conference on machine learning*, pages 7055–7065. PMLR, 2020.

W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. doi: 10.1073/pnas.1900654116.

J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986. doi: 10.1007/BF00116251.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. doi: 10.1038/s42256-019-0048-x.

Victor Feitosa Souza, Ferdinando Cicalese, Eduardo Laber, and Marco Molinaro. Decision trees with short explainable rules. *Advances in neural information processing systems*, 35:12365–12379, 2022.

A Bounding the Cost of the Set Elimination Game

In this section, we analyze the cost of the Set Elimination Game. We begin by slightly simplifying the setting: first, we discretize the measure space (Ω, μ), and then assign each round a continuous time using the exponential clock. We then proceed to the proof of Theorem 5.1 in Section A.4.

A.1 Discretizing the Measure Space

We first reduce the Set Elimination Game with an arbitrary finite measure space (Ω, μ) to an equivalent game on a discrete finite measure space. To this end, we introduce an equivalence relation $\sim$ on Ω. We say that $\omega', \omega'' \in \Omega$ are equivalent, and write $\omega' \sim \omega''$, if they belong to exactly the same subsets S_i. In other words, for every set S_i in the Set Elimination Game, either both ω' and ω'' belong to S_i, or neither does. Observe that this equivalence relation is consistent with the rules of the game: the game evolves identically if, at some round n, we select ω' or ω'' with $\omega' \sim \omega''$.

We now consider the quotient space $\Omega/\sim$. The number of equivalence classes is at most the number of distinct subsets of $\{1, \dots, k\}$, since each equivalence class is uniquely determined by the collection of sets S_i that contain its elements. Therefore, the quotient space $\Omega/\sim$ contains at most 2^k equivalence classes.

A.2 Set Elimination Game with Continuous Time

To analyze the cost of the Set Elimination Game, it is convenient to work in continuous time. Following Makarychev and Shan (2023), we associate each round of the discrete game with a continuous time obtained via an exponential clock.

In the continuous-time view, each element $\omega \in \Omega$ is assigned an independent clock whose arrival times follow an exponential distribution with parameter $\mu(\omega)$. Equivalently, one may view the process as assigning to each ω an independent Poisson process $N_\omega(t)$ with rate $\mu(\omega)$ and selecting ω at each arrival time of $N_\omega(t)$. In this way, we obtain a sequence (ω_i, τ_i), where ω_i is the i-th selected element of Ω and τ_i is the time at which ω_i is selected. We assume that the sequence is ordered by increasing values of τ_i.

We claim that the distribution of the sequence $\omega_1, \omega_2, \dots$ defined above is the same as in the Set Elimination Game. Specifically, the random variables $\omega_1, \omega_2, \dots$ are mutually independent, and each ω_i is distributed according to

$$\Pr(\omega_i = \omega) = \frac{\mu(\omega)}{\mu(\Omega)}.$$

This claim follows from the Coloring Theorem.

Theorem A.1 (Coloring Theorem, Kingman (1992); Mitzenmacher and Upfal (2017))**.** *Let $N(t)$ be a Poisson process with rate λ. Suppose that each arrival is independently assigned a color $c \in \{1, \dots, k\}$ with probabilities $p_1, \dots, p_k$, where $\sum_{i=1}^k p_i = 1$. Then the processes $N_1(t), \dots, N_k(t)$, where $N_i(t)$ counts arrivals of color i, are independent Poisson processes with rates $\lambda p_1, \dots, \lambda p_k$, respectively.*

In our setting, the colors correspond to the elements $\omega \in \Omega$. The Coloring Theorem implies that the sequence (ω_i) defined above has the same distribution as the sequence generated in the following way: consider a Poisson process with rate

$$\lambda = \sum_{\omega \in \Omega} \mu(\omega) = \mu(\Omega),$$

and at each arrival time independently select an element $\omega \in \Omega$ with probability $p_\omega = \frac{\mu(\omega)}{\mu(\Omega)}$. This is exactly the rule used to select ω in the Set Elimination Game. Therefore, by the Coloring Theorem, the two processes generate the same distribution over sequences (ω_i).

We extend the definition of the remaining sets $\mathcal{R}_n$ to continuous time as follows. For every time $t \geq 0$, let $\mathcal{R}(t) = \mathcal{R}_{n+1}$, where n is the largest index such that $\tau_n < t$.

We refer to the time at which an element $\omega \in \Omega$ is selected as the hitting time of ω, and denote it by $h(\omega)$. For a set $S \subseteq \Omega$, the hitting time $h(S)$ is the first time at which an element of S is selected, i.e.,

$$h(S) = \min_{\omega \in S} h(\omega).$$

Since the random variables $\{h(\omega)\}_{\omega \in \Omega}$ are independent exponential random variables with parameters $\mu(\omega)$, the random variable $h(S)$ is itself exponentially distributed with parameter

$$\sum_{\omega \in S} \mu(\omega) = \mu(S).$$

The *elimination time* of S_i is defined as

$$e(S_i) = \min\{\tau_n : i \notin \mathcal{R}_n\}.$$

Note that $h(S_i) \leq e(S_i)$, since at least one element of S_i must be selected before S_i can be eliminated.

A.3 Competitions

In the analysis of the game, we will use the notion of a *competition* restricted to a collection of sets I. Consider a realization of the Set Elimination Game, which is fully determined by the arrival sequence $\omega_1, \omega_2, \ldots$. Let $K = [k]$. For a subset $I \subseteq K$ with $I \cap H \neq \varnothing$, we define the *competition* on I to be the Set Elimination Game in which only the sets $\{S_i\}_{i \in I}$ participate. The elimination rules are the same as in the original game. Specifically, at round n, if every set remaining in the competition contains ω_n, then no set is eliminated; otherwise, all remaining sets S_i with $i \in I$ that contain ω_n are eliminated. We denote the set of winners of this competition by $W(I)$. Throughout the analysis, we assume that all competitions evolve simultaneously using the same arrival sequence $\omega_1, \omega_2, \ldots$.

Note that the winners of the original game are precisely the winners of the competition in which all sets participate, namely $W(K)$.

A.4 Proof of Theorem 5.1

We now prove Theorem 5.1. Throughout the proof, we assume that the measure space Ω is finite and discrete (see Section A.1) and that each $\omega \in \Omega$ is assigned a random hitting time $h(\omega)$ distributed as an exponential random variable with parameter $\mu(\omega)$. The random variables $\{h(\omega)\}_{\omega \in \Omega}$ are mutually independent (see Section A.2).

Proof of Theorem 5.1. Throughout the analysis, we assume that the $\arg\max$ function returns a unique element, with ties broken arbitrarily.

Let $j^* \in I_H$ be the index of the high priority set with the smallest measure, i.e.

$$\mu(S_{j^*}) = \min_{j \in I_H} \mu(S_j).$$

We say that a high priority set S_j with $j \in I_H \setminus \{j^*\}$ is a *surprise* set if

$$e(S_j) \geq h(S_{j^*}) \geq \frac{\lceil \ln m \rceil}{\mu(S_j)}.$$

We further denote the set of indices of the *surprise sets* as $\mathcal{S}$. The following lemma from Makarychev and Shan (2023) bounds the probability that a given set is a surprise set.

Lemma A.2. *[Lemma 3.3 in Makarychev and Shan (2023)] For every $j \in I_H \setminus \{j^*\}$*

$$\Pr(j \in \mathcal{S}) \leq \frac{1}{m} \frac{\mu(S_{j^*})}{\mu(S_j)}.$$

Note that the elimination rules considered in this paper are identical to those in Makarychev and Shan (2023), and the definition of a surprise set depends only on the hitting and elimination times, and not on the set of winners. Therefore, the lemma applies to our setting as well.

We show that, since high-priority sets are unlikely to be surprise sets, the contribution of surprise sets to the cost of the game is negligible. Let E^{bad} denote the event that $W([k]) \cap \mathcal{S} \neq \varnothing$, i.e., that at least one winner is a surprise set. Let E^{good} denote the complement of E^{bad}. We can thus decompose the expected cost of the game as

$$\begin{aligned}\mathbf{E}[\bar{\mu}(W([k]))] = \mathbf{E}\left[\bar{\mu}(W([k]))\,\mathbf{1}\left\{E^{\text{bad}}\right\}\right] + \\ \mathbf{E}\left[\bar{\mu}(W([k]))\,\mathbf{1}\left\{E^{\text{good}}\right\}\right].\end{aligned} \tag{5}$$

We now derive a bound on the first term. We begin with the following claim.

Claim A.3. *Let $i = \arg\max_{j \in W([k])} \mu(S_j)$. On the event E^{bad}, the set S_i is a surprise set.*

Proof of Claim A.3. By definition of the event E^{bad}, there exists a surprise set among the winners. Fix any $j \in W([k]) \cap \mathcal{S}$. Since $i \in W([k])$, the set S_i is not eliminated before S_{j^*}, and therefore $e(S_i) \geq h(S_{j^*})$. Because $j \in \mathcal{S}$, we have

$$h(S_{j^*}) \geq \frac{\lceil \ln m \rceil}{\mu(S_j)} \geq \frac{\lceil \ln m \rceil}{\mu(S_i)},$$

where the second inequality follows from the definition of i and the fact that $\mu(S_i) \geq \mu(S_j)$. Therefore $e(S_i) \geq h(S_{j^*}) \geq \frac{\lceil \ln m \rceil}{\mu(S_i)}$ and so $i \in \mathcal{S}$. □

Observe that

$$\bar{\mu}(W([k])) = \frac{\sum_{j \in W([k])} \mu(S_j)}{|W([k])|} \leq \max_{j \in W([k])} \mu(S_j).$$

Therefore,

$$\begin{aligned}\mathbf{E}\left[\bar{\mu}(W([k]))\,\mathbf{1}\{E^{\text{bad}}\}\right] &\leq \mathbf{E}\left[\max_{j \in W([k])} \mu(S_j)\;\mathbf{1}\{E^{\text{bad}}\}\right] \\ &= \sum_{j \in I_H} \mu(S_j)\;\Pr\Big(j = \arg\max_{j' \in W([k])} \mu(S_{j'}),\; E^{\text{bad}}\Big) \\ &\leq \sum_{j \in I_H} \mu(S_j)\;\Pr(j \in \mathcal{S}) \\ &\leq \mu(S_{j^*}),\end{aligned}$$

where the second inequality follows from Claim A.3, and the last inequality follows from Lemma A.2 together with the fact that $|I_H| = m$.

Next, we bound the second term in Equation 5. To this end, we define a cost associated with each $\omega \in S_{j^*}$:

$$\text{cost}(\omega) = \bar{\mu}(W([k]))\,\mathbf{1}\{E^{\text{good}}\} \cdot \mathbf{1}\{h(\omega) = h(S_{j^*})\}.$$

We decompose the second term in Equation 5 as follows:

$$\mathbf{E}\left[\bar{\mu}(W([k]))\,\mathbf{1}\{E^{\text{good}}\}\right] = \sum_{\omega \in S_{j^*}} \mathbf{E}[\text{cost}(\omega)]. \tag{6}$$

Here we use the fact that, with probability one, all hitting times $h(\omega)$ are distinct, and exactly one element $\omega \in S_{j^*}$ satisfies $h(\omega) = h(S_{j^*})$. Therefore, $\sum_{\omega \in S_{j^*}} \mathbf{1}\{h(\omega) = h(S_{j^*})\} = 1$.

Fix $\omega \in S_{j^*}$. Consider the hitting time $h(\omega)$ and the set of remaining sets at that time. We define three mutually exclusive and exhaustive events (see Figure 3):

- Let E^{+} be the event that $\omega \in S_j$ for all $j \in \mathcal{R}(h(\omega))$.
- Let E^{-} be the event that $\omega \notin S_j$ for some $j \in \mathcal{R}(h(\omega)) \cap I_H$.

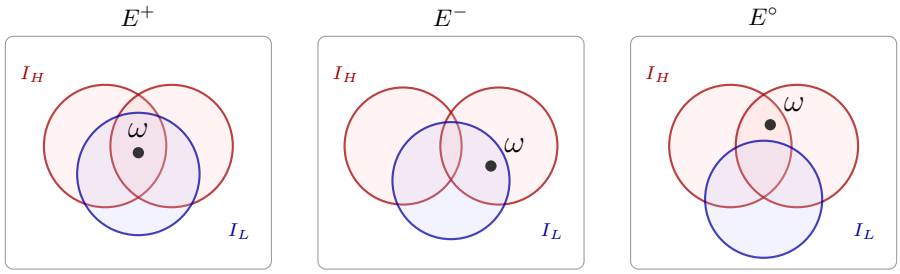


Figure 3: Illustration of the events E^+, E^-, and E°. Red circles denote high-priority sets (I_H), while blue circles denote low-priority sets (I_L). E^+**:** ω lies in every remaining set. E^-**:** ω lies outside at least one high-priority set. E°**:** ω belongs to every high-priority set but not to some low-priority set.

- Let E^o be the event that $\omega \in S_j$ for all $j \in \mathcal{R}(h(\omega)) \cap I_H$, but there exists $i \in \mathcal{R}(h(\omega)) \cap I_L$ such that $\omega \notin S_i$.

Then, define

$$\text{cost}^+(\omega) = \text{cost}(\omega)\,\mathbf{1}\{E^+\}, \quad \text{cost}^-(\omega) = \text{cost}(\omega)\,\mathbf{1}\{E^-\}, \quad \text{cost}^o(\omega) = \text{cost}(\omega)\,\mathbf{1}\{E^o\}.$$

We have

$$\mathbf{E}[\text{cost}(\omega)] = \mathbf{E}[\text{cost}^+(\omega)] + \mathbf{E}[\text{cost}^-(\omega)] + \mathbf{E}[\text{cost}^o(\omega)] \tag{7}$$

and proceed to bound each term separately.

Case 1: Bounding $\mathbf{E}[\text{cost}^o(\omega)]$. The number of remaining high-priority sets $|\mathcal{R}(t) \cap I_H|$ at time t is a non-increasing function of t. Therefore, we partition the timeline into phases $A_0, A_1, \ldots, A_{\lceil \ln m \rceil}$, where for $0 \le i \le \lceil \ln m \rceil - 1$,

$$A_i = \left\{ t \in [0,\infty) : \frac{m}{e^{i+1}} < |\mathcal{R}(t) \cap I_H| \le \frac{m}{e^i} \right\},$$

and

$$A_{\lceil \ln m \rceil} = \{ t \in [0,\infty) : \mathcal{R}(t) \cap I_H = \varnothing \}.$$

For each $0 \le i \le \lceil \ln m \rceil$, let E_i denote the event that $h(\omega) \in A_i$. Since $\bigcup_i A_i = [0,\infty)$, the events E_i are disjoint and exhaustive. Finally, let t_i denote the starting time of phase A_i, i.e.,

$$t_i = \begin{cases} \inf A_i & \text{if } A_i \neq \varnothing \\ +\infty & \text{otherwise} \end{cases}$$

Then each t_i is a stopping time. We decompose the cost as follows:

$$\mathbf{E}[\text{cost}^o(\omega)] \le \sum_{i=0}^{\lceil \ln m \rceil} \mathbf{E}\left[\frac{\sum_{j \in W([k])} \mu(S_j)}{|W([k])|}\, \mathbf{1}\{h(\omega) = h(S_{j^*})\}\, \mathbf{1}\{E^o\} \cdot \mathbf{1}\{E_i\} \right]. \tag{8}$$

On the event E^o, there exists a set S_i with $i \in \mathcal{R}(h(\omega)) \cap I_L$ such that $\omega \notin S_i$, while for every $j \in \mathcal{R}(h(\omega)) \cap I_H$, we have $\omega \in S_j$. Therefore, the winners of the competition on $[k]$ are exactly

$$W([k]) = \mathcal{R}(h(\omega)) \cap I_H,$$

and every winner S_j satisfies $\omega \in S_j$. Moreover, on the event E_i, we have

$$\frac{m}{e^{i+1}} < |\mathcal{R}(h(\omega)) \cap I_H| \le \frac{m}{e^i}. \tag{9}$$

Therefore

$$\frac{1}{|W([k])|}\, \mathbf{1}\{E_i\}\, \mathbf{1}\{E^o\} < \frac{e^{i+1}}{m}\, \mathbf{1}\{E_i\}\, \mathbf{1}\{E^o\}. \tag{10}$$

Combining Equations 8 and 10, we obtain

$$\mathbf{E}[\mathrm{cost}^o(\omega)] \le \sum_{i=0}^{\lceil \ln m \rceil} \frac{e^{i+1}}{m} \mathbf{E}\left[\sum_{j \in W([k])} \mu(S_j)\, \mathbf{1}\{h(\omega) = h(S_{j^*})\}\, \mathbf{1}\{E_i\}\, \mathbf{1}\{E^o\}\right]$$
$$\le \sum_{i=0}^{\lceil \ln m \rceil} \frac{e^{i+1}}{m} \mathbf{E}\left[\sum_{j \in \mathcal{R}(t_i) \cap I_H} \mu(S_j) \Pr(E_i, E^o, h(\omega) = h(S_{j^*}), j \in \mathcal{R}(h(\omega)) \cap I_H \mid \mathcal{F}_{t_i})\right], \quad (11)$$

where $(\mathcal{F}_t)_{t \ge 0}$ denotes the natural filtration of the process, and we used that the random set $\mathcal{R}(t_i)$ is $\mathcal{F}_{t_i}$-measurable.

We now consider the event inside the probability in Equation 11. As we established earlier, if $j \in \mathcal{R}(h(\omega)) \cap I_H = W([k])$, then $\omega \in S_j$. Moreover, ω must be the first element selected from S_j i.e., we must have $h(\omega) \le h(S_j \setminus \{\omega\})$; otherwise, some element $\omega' \in S_j \setminus \{\omega\}$ would be selected earlier. Since $h(\omega) = h(S_{j^*})$, the element ω' cannot belong to S_{j^*}. Consequently, the set S_j would be eliminated when ω' is selected, implying that $j \notin \mathcal{R}(h(\omega))$.Therefore

$$\Pr\Big(E_i, E^o, h(\omega) = h(S_{j^*}), j \in W([k]) \mid \mathcal{F}_{t_i}\Big) \le \Pr\Big(h(\omega) \le h(S_j \setminus \{\omega\}) \mid \mathcal{F}_{t_i}\Big)$$
$$= \frac{\mu(\omega)}{\mu(\omega) + \mu(S_j \setminus \{\omega\})} = \frac{\mu(\omega)}{\mu(S_j)}, \quad (12)$$

To compute $\Pr\big(h(\omega) \le h(S_j \setminus \{\omega\}) \mid \mathcal{F}_{t_i}\big)$, we use the memoryless property of the exponential distribution. Conditional on $\mathcal{F}_{t_i}$, the hitting times of ω and $S_j \setminus \{\omega\}$ are independent exponential random variables with parameters $\mu(\omega)$ and $\mu(S_j \setminus \{\omega\})$, respectively. Combining Equations 11 and 12 and using the upper bound on the number of high priority sets that contain ω during phase i (see Equation 9), we conclude that

$$\mathbf{E}[\mathrm{cost}^o(\omega)] \le \sum_{i=0}^{\lceil \ln m \rceil} \frac{e^{i+1}}{m} \mathbf{E}\Big[\sum_{j \in \mathcal{R}(t_i) \cap I_H} \mu(S_j) \cdot \frac{\mu(\omega)}{\mu(S_j)}\Big] \quad (13)$$
$$= \sum_{i=0}^{\lceil \ln m \rceil} \frac{e^{i+1}}{m} \cdot \mu(\omega) \cdot |\mathcal{R}(t_i) \cap I_H|$$
$$\le e\lceil \ln m + 1 \rceil\, \mu(\omega).$$

Case 2: Bounding $\mathbf{E}[\mathrm{cost}^-(\omega)]$. Let $I_\omega^- \subseteq [k]$ denote the set of indices $i \in [k]$ such that $\omega \notin S_i$. Similarly, let $I_\omega^+ \subseteq [k]$ denote the set of indices $i \in [k]$ such that $\omega \in S_i$. We consider the competitions on the sets I_ω^+ and I_ω^- (see Section A.3 for the definition of competitions).Recall that we denote the winners of these competitions by $W(I_\omega^+)$ and $W(I_\omega^-)$, respectively. We will use the following lemma, which we prove in Appendix B.

Lemma A.4. *The following equalities hold.*

- *On the event $E^- \cap \{h(\omega) = h(S_{j^*})\}$, we have $W([k]) = W(I_\omega^-)$.*
- *On the event $E^+ \cap \{h(\omega) = h(S_{j^*})\}$, we have $W([k]) = W(I_\omega^+)$.*

Moreover, the hitting time $h(\omega)$ is independent of both $W(I_\omega^-)$ and $W(I_\omega^+)$.

We have

$$\mathbf{E}\left[\mathrm{cost}^-(\omega)\right] \le \mathbf{E}\left[\max_{j \in W([k])} \mu(S_j)\, \mathbf{1}\{h(\omega) = h(S_{j^*})\}\, \mathbf{1}\{E^{\mathrm{good}}\}\, \mathbf{1}\{E^-\}\right]$$
$$= \sum_{j \in I_H} \mu(S_j) \Pr\Big(j = \arg\max_{i \in W([k])} \mu(S_i), h(\omega) = h(S_{j^*}), E^{\mathrm{good}}, E^-\Big). \quad (14)$$

Consider the event inside the probability in Equation 14, for a fixed $j \in I_H$. On the event E^{good}, no winner is a surprise set; hence, if $j \in W([k])$, then $j \notin \mathcal{S}$. Since $j \in W([k])$ implies $e(S_j) \geq h(S_{j^*})$, the condition $j \notin \mathcal{S}$ implies $h(\omega) = h(S_{j^*}) < \frac{\lceil \ln m \rceil}{\mu(S_j)}$. Therefore,

$$\Pr\Big(j = \operatorname*{arg\,max}_{i \in W([k])} \mu(S_i),\, h(\omega) = h(S_{j^*}),\, E^{\text{good}},\, E^-\Big) \leq \Pr\Big(j = \operatorname*{arg\,max}_{i \in W([k])} \mu(S_i),\, h(\omega) = h(S_{j^*}),\, h(\omega) < \frac{\lceil \ln m \rceil}{\mu(S_j)},\, E^-\Big).$$

By Lemma A.4, in the event $E^- \cap \{h(\omega) = h(S_{j^*})\}$, we have $W([k]) = W(I_\omega^-)$ and, moreover, $W(I_\omega^-)$ is independent of $h(\omega)$. This implies that

$$\begin{aligned}\Pr\Big(j = \operatorname*{arg\,max}_{i \in W([k])} \mu(S_i), h(\omega) = h(S_{j^*}),\, E_{\text{good}},\, E^-\Big) &\leq \Pr\Big(j = \operatorname*{arg\,max}_{i \in W(I_\omega^-)} \mu(S_i),\, h(\omega) < \tfrac{\lceil \ln m \rceil}{\mu(S_j)}\Big) \\ &= \Pr\Big(j = \operatorname*{arg\,max}_{i \in W(I_\omega^-)} \mu(S_i)\Big) \Pr\Big(h(\omega) < \tfrac{\lceil \ln m \rceil}{\mu(S_j)}\Big). \end{aligned} \tag{15}$$

Substituting Equation 15 into Equation 14, and using the fact that $h(\omega)$ is an exponential random variable with parameter $\mu(\omega)$, we obtain

$$\begin{aligned}\mathbf{E}[\text{cost}^-(\omega)] &\leq \sum_{j \in I_H} \mu(S_j) \Pr\Big(j = \operatorname*{arg\,max}_{i \in W(I_\omega^-)} \mu(S_i)\Big)\Big(1 - e^{-\lceil \ln m \rceil \mu(\omega)/\mu(S_j)}\Big) \\ &\leq \sum_{j \in I_H} \mu(S_j) \Pr\Big(j = \operatorname*{arg\,max}_{i \in W(I_\omega^-)} \mu(S_i)\Big) \cdot \lceil \ln m \rceil \frac{\mu(\omega)}{\mu(S_j)} \\ &= \lceil \ln m \rceil \mu(\omega). \end{aligned} \tag{16}$$

Case 3: Bounding $\mathbf{E}[\text{cost}^+(\omega)]$. This case is handled similarly to the previous one using Lemma A.4. We obtain

$$\begin{aligned}\mathbf{E}[\text{cost}^+(\omega)] &\leq \sum_{j \in I_H} \mu(S_j) \Pr\Big(j = \operatorname*{arg\,max}_{i \in W([k])} \mu(S_i), h(\omega) = h(S_{j^*}), E^{\text{good}}, E^+\Big) \\ &\leq \sum_{j \in I_H} \mu(S_j) \Pr\Big(j = \operatorname*{arg\,max}_{i \in W(I_\omega^+)} \mu(S_i)\Big) \Pr\Big(h(\omega) \leq \frac{\lceil \ln m \rceil}{\mu(S_j)}\Big) \\ &\leq \lceil \ln m \rceil \mu(\omega). \end{aligned} \tag{17}$$

Final calculation. Substituting Equations 13, 16, and 17 into Equation 7, we obtain

$$\mathbb{E}[\text{cost}(\omega)] \leq (2 + e)\, \lceil \ln m \rceil\, \mu(\omega). \tag{18}$$

Combining Equations 5 and 6, we conclude that

$$\begin{aligned}\mathbf{E}[\bar{\mu}(W([k]))] &\leq \mathbf{E}[\bar{\mu}(W([k]))\, \mathbf{1}\{E^{\text{bad}}\}] + \mathbf{E}[\bar{\mu}(W([k]))\, \mathbf{1}\{E^{\text{good}}\}] \\ &\leq \mu(S_{j^*}) + (2 + e)\, \lceil \ln m \rceil\, \mu(S_{j^*}) \\ &\leq (3 + e)\, \lceil \ln m \rceil\, \mu(S_{j^*}). \end{aligned}$$

□

B Proof of Lemma A.4

To prove Lemma A.4, we make use of a structural lemma about the Set Elimination Game proved in Makarychev and Shan (2023).

Lemma B.1 (Lemma A.1. in Makarychev and Shan (2023)). *Consider two set systems X and Y that are subsets of the set system $K = [k]$ and $X \subset Y$. Fix a realization of the game $\omega_1, \omega_2, \ldots$ Then, for any round n of the game, we must have*

$$\mathcal{R}_n(Y) \cap X = \mathcal{R}_n(X) \text{ or } \mathcal{R}_n(Y) \cap X = \varnothing.$$

We now prove Lemma A.4, which we restate for convenience.

Lemma A.4. *The following equalities hold.*

- *On the event $E^- \cap \{h(\omega) = h(S_{j^*})\}$, we have $W([k]) = W(I_\omega^-)$.*
- *On the event $E^+ \cap \{h(\omega) = h(S_{j^*})\}$, we have $W([k]) = W(I_\omega^+)$.*

Moreover, the hitting time $h(\omega)$ is independent of both $W(I_\omega^-)$ and $W(I_\omega^+)$.

Proof of Lemma A.4. The random variable $W(I_\omega^-)$ is a function of the Poisson Processes $N_{\omega'}(t)$ for every $\omega' \in \bigcup_{i \in I_\omega^-} S_i$. This is because every time an element $\omega'' \notin \bigcup_{i \in I_{\omega^-}} S_i$ is selected during the competition I_ω^-, the remaining sets are not updated, because no remaining set contains ω''. By definition, no set S_j with $j \in I_\omega^-$ contains ω and because the Poisson Processes are mutually independent for different elements in Ω, we conclude that $W(I_\omega^-)$ is independent of $N_\omega(t)$. Moreover, the competition I_ω^+ is independent of $N_\omega(t)$, because all sets in I_ω^+ contain ω which means that every time ω is selected, it is ignored and the remaining sets are not updated.

Let n^* be final round of the competition K, i.e. the first round where $|\mathcal{R}_n(K) \cap I_H| \leq 1$. Let q be the first round when ω is selected. Moreover, let $n' = n^*$ if $|\mathcal{R}_{n^*}(K) \cap I_H| = 1$ and $n' = n^* - 1$ otherwise, meaning that $W(K) = \mathcal{R}_{n'}(K) \cap I_H$.

In the event $E^- \cap \{h(\omega) = h(S_{j^*})\}$, at round q there is a high priority set that does not contain ω. The set S_{j^*} is also remaining at round q, because ω is the first element hit from S_{j^*}. All sets that contain ω are eliminated and $\mathcal{R}_{q+1}(K) \subseteq I_\omega^-$. Because there is a priority set still remaining at round $q+1$, we know that $n' \geq q+1$. For every subsequent round $q' \geq q+1$ we have $\mathcal{R}_{q'}(K) \subseteq \mathcal{R}_{q+1}(K) \subseteq I_\omega^-$ and thus $\mathcal{R}_{n'}(K) \subseteq I_\omega^-$. We have that $\mathcal{R}_{n'}(K) \cap I_\omega^- = \mathcal{R}_{n'}(K) \neq \varnothing$ and thus, by Lemma B.1,

$$\mathcal{R}_{n'}(I_\omega^-) = \mathcal{R}_{n'}(K) \cap I_\omega^- = \mathcal{R}_{n'}(K) \Rightarrow$$
$$\mathcal{R}_{n'}(I_\omega^-) \cap I_H = W(K).$$

If $|W(K)| = 1$, there is only 1 high priority set remaining in the competition I_ω^- at round n' and so it must be the winner of I_ω^-, that is $W(I_\omega^-) = \mathcal{R}_{n'}(I_\omega^-) \cap I_H = W(K)$. Otherwise, the element $\omega_{n'}$ eliminates all sets S_j with $j \in W(K)$, by the definition of round n'. Because the remaining sets for both competitions K and I_ω^- are exactly the same at round n', the cut $\omega_{n'}$ eliminates every set S_j with $j \in \mathcal{R}_{n'}(I_\omega^-) \cap I_H = W(K)$ and thus $W(I_\omega^-) = W(K)$. Either way, $W(K) = W(I_\omega^-)$.

In the event $E^+ \cap \{h(\omega) = h(S_{j^*})\}$, all remaining sets contain ω at round q and $j^* \in \mathcal{R}_q(K)$, because ω is the first element from S_{j^*} that is hit. If $q \geq n^*$, the only remaining high priority set at round n^* is S_{j^*}. This means that it is the winner of the competition K, i.e. $\mathcal{R}_{n'}(K) \cap I_H = W(K) = \{j^*\}$. Because $\omega \in S_{j^*}$, we have $j^* \in I_\omega^+$ and so $\mathcal{R}_{n'}(K) \cap I_\omega^+ \neq \varnothing$. By Lemma B.1, $\mathcal{R}_{n'}(I_\omega^+) = \mathcal{R}_{n'}(K) \cap I_\omega^+$ and therefore

$$\mathcal{R}_{n'}(I_\omega^+) \cap I_H = W(K) = \{j^*\}.$$

So at round n', there is exactly 1 high priority set remaining in the competition I_ω^+ and so $W(I_\omega^+) = W(K) = \{j^*\}$.

On the other hand, if $q < n^*$, it holds that $q \leq n'$. In the event $E^+ \cap \{h(\omega) = h(S_{j^*})\}$, all remaining sets contain ω at round q in the competition K, i.e. $\mathcal{R}_q(K) \subseteq I_\omega^+$. For the subsequent rounds $q' \geq q+1$ we have $\mathcal{R}_{q'}(K) \subseteq \mathcal{R}_q(K) \subseteq I_\omega^+$ and therefore $\mathcal{R}_{n'}(K) \subseteq I_\omega^+$. By Lemma B.1, we get that $\mathcal{R}_{n'}(I_\omega^+) = \mathcal{R}_{n'}(K)$ and following the exact same steps as in the analysis of the event $E^- \cap \{h(\omega) = h(S_{j^*})\}$, we conclude that $W(I_\omega^+) = W(K)$. □

Algorithm RANKEDPRIORITREE

Input: A set of k reference centers $C \subseteq \mathbb{R}^d$ and a total order $\preceq$ on C.
Output: A threshold tree $\mathcal{T}$.

Sort C under $\preceq$ and relabel so that $c_1 \preceq c_2 \preceq \cdots \preceq c_k$.
if $|C| = 1$ **then return** a single leaf, containing c_1.
Run RANDOMCUTS$(C \setminus \{c_1\})$ to obtain an initial tree $\mathcal{T}_1$.
Partition $c_2, \ldots, c_k$ into consecutive priority classes $P_1, \ldots, P_s$ in the priority order, with $P_1 = \{c_2, c_3\}$ and, for $i \geq 2$, P_i containing the next $2^{2^{i-1}}$ centers after P_{i-1}, except that the final block P_s contains all remaining centers and includes c_k.
for every leaf v of $\mathcal{T}_1$ **do**
    Let c_j be the center assigned to v and let P_{i^*} be its priority class. Set $\bar{c}_{i^*} = c_j$.
    Let $C'_v = \{\bar{c}_{i^*}\}$.
    Let $u_1, u_2, \ldots, u_N$ be the root-to-leaf path in $\mathcal{T}_1$ with u_1 the root and $u_N = v$.
    for each $i \in [s] \setminus \{i^*\}$ **do**
        Let u be the node u_t with the largest index t such that $C_{u_t} \cap P_i \neq \varnothing$.
        Sample $\bar{c}_i$ uniformly at random from $C_u \cap P_i$.
        Add $\bar{c}_i$ to C'_v.
    Replace v with the tree returned by the recursive call RANKEDPRIORITREE$(C'_v, \preceq)$.
Let $\mathcal{T}_2$ be the resulting tree.
for every leaf v' of $\mathcal{T}_2$ **do**
    Let c'_j be the single center assigned to leaf v'.
    Replace v' with the tree returned by RANDOMCUTS$(\{c'_j, c_1\})$
return the resulting tree $\mathcal{T}$.

Figure 4: Explainable k-medians clustering with ranked centers

C EXPLAINABLE CLUSTERING WITH RANKED CENTERS

In this section, we study the explainable clustering setting in which the reference centers are ordered by decreasing priority. Let $C = \{c_1, c_2, \ldots, c_k\} \subseteq \mathbb{R}^d$ and a total order $\preccurlyeq$ over $C \times C$. We say that for any $c, c' \in C$, the center c has *equal or higher priority* than center c' if and only if $c \preccurlyeq c'$. Without loss of generality, assume that $c_i \preccurlyeq c_j$ if and only if $i \leq j$. Our goal is to design an explainable clustering algorithm that prioritizes centers of higher priority. We prove the following theorem.

Theorem 1.2. *There exists a polynomial-time algorithm that, given an ordered list of reference centers $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$ in order of decreasing priority, returns a threshold tree $\mathcal{T}$ such that the following holds for every $\varepsilon > 0$. The number of leaves of $\mathcal{T}$ is at most*

$$O_\varepsilon\big(k \cdot (\log\log k)^{1+\varepsilon}\big)$$

with probability 1*, and for every $x \in \mathbb{R}^d$ and every $i \in [k]$,*

$$\mathbb{E}\big[\|x - \mathcal{T}(x)\|_1\big] \;\leq\; O_\varepsilon\big(\log i \cdot (\log\log\log i)^{1+\varepsilon}\big)\, \|x - c_i\|_1.$$

Before we prove Theorem 1.2, we need some definitions. For every non-empty $C' \subseteq C$, let $\mathcal{T}_{C'} =$ RANKEDPRIORITREE(C') be the output of Algorithm 4 with input a set of centers C'. We will also denote $\mathcal{T} = \mathcal{T}_C$. We define the *rank* of a center c in the set $C' \ni c$ as the number of centers $c' \in C'$ with $c' \preccurlyeq c$, denoted by $r_{C'}(c)$. The center $c \in C'$ with rank $r_{C'}(c) = i$ is denoted as $c_i(C')$.

If x coincides with any center in C, its cost in the final tree $\mathcal{T}$ will be 0 with probability 1 and will satisfy Theorem 1.2 trivially. So from now on, we assume that no center in C is equal to x. We can now define the function $\alpha : [k] \to \mathbb{R}_{>0}$ that will help us to bound the cost distortion of x.

Definition C.1. *Let function $\alpha : [k] \to \mathbb{R}_{>0}$ such that*

- $\alpha(i)$ *is the smallest number* a *such that every non-empty* $C' \subseteq C$ *with at least* i *centers, it holds that*
$$\frac{\mathbf{E}[\|x - \mathcal{T}_{C'}(x)\|_1]}{\|x - c_i(C')\|_1} \leq a$$
- $\ell(i)$ *is the smallest number* l *such that for every non-empty* $C' \subseteq C$ *with exactly* i *centers, the number of leaves of the tree* $\mathcal{T}_{C'}$ *is at most* l *with probability* 1.

Note that, α is well defined because each $C' \subseteq C$ is finite and the ratio is also finite for every C'.

By definition of the function α, for every $i \in [k]$:
$$\mathbf{E}[\text{cost}_{\ell_1}(x, \mathcal{T})] \leq \alpha(i)\|x - c_i\|_1.$$
Thus, to obtain the approximation ratio of x with respect to c_i, it suffices to show that for every $\varepsilon > 0$:
$$\alpha(i) = O_\varepsilon\big(\log i \cdot (\log\log\log i)^{1+\varepsilon}\big).$$

We now prove Theorem 1.2.

Proof of Theorem 1.2. Let $f : \{2, 3, \ldots, k\} \to [k]$ such that for every $i \in \{2, 3, \ldots, k\}$
$$f(i) = \min\left\{j \geq 1 : \sum_{z=1}^{j} 2^{2^{z-1}} \geq i - 1\right\}.$$
Observe that, for constant i, $f(i)$ is a constant, otherwise $f(i) = O(\log\log i)$. This function also satisfies the following property.

Claim C.2. *For every* $i \in \{2, 3, \ldots, k\}$, *it holds that*
$$f(i) < i.$$

Proof. We prove the claim by induction on i

For the base case, we have $f(2) = 1 < 2$.

Now assume that the claim holds for every $j \in \{2, 3, \ldots, i\}$ for some $2 \leq i < k$.

We have
$$\begin{aligned}\sum_{z=1}^{f(i)+1} 2^{2^{z-1}} &= \sum_{z=1}^{f(i)} 2^{2^{z-1}} + 2^{2^{f(i)}} \\ &\geq i - 1 + 4 > i\end{aligned}$$
where we used that $f(i) \geq 1$ and that $\sum_{z=1}^{f(i)} 2^{2^{z-1}} \geq i - 1$ by definition of f. Because $f(i+1)$ is the smallest $j \geq 1$ such that $\sum_{z=1}^{j} 2^{2^{z-1}} \geq i$, we
$$f(i+1) \leq f(i) + 1 < i + 1$$
where we used the inductive hypothesis $f(i) < i$. □

We show that the function α from Definition C.1 satisfies the following.
$$\begin{aligned}&\alpha(1) \leq O(1) \\ &\alpha(i) \leq O(\log i) \cdot \alpha(f(i)) \text{ for every } i \in \{2, \ldots, k\}.\end{aligned}$$

By Claim C.2, this is a valid recurrence relation. Then, by unrolling the recurrence, for constant i, we get that $\alpha(i)$ is a constant, and for large i, there exists a constant $A > 0$ such that
$$\alpha(i) \leq O\left(A^r \log(i) \cdot \log\log\log(i) \cdot \log^{(5)}(i) \cdots \log^{(2r+1)} i\right).$$
where $\log^{(n)}(i)$ is the composition of the $\log(\cdot)$ function n times and $r \geq 0$ is the largest number such that $\log^{(2r+1)}(i) \geq 1$. Here, $r = O(\log^* i)$, where $\log^* i$ is the iterated logarithm of i, i.e. the largest j such that $\log^{(j)} i \geq 1$. Thus, the term A^r multiplied by the product of $\log^{(2j+1)}(i)$ for $2 \leq j \leq r$ are dominated by $O_\varepsilon\left((\log\log\log i)^\varepsilon\right)$ for every $\varepsilon > 0$ and so
$$\alpha(i) \leq O_\varepsilon\big(\log i \cdot (\log\log\log i)^{1+\varepsilon}\big).$$
We now proceed to prove the recursive relation for α.

Case $i = 1$. Let C' be a non-empty subset of C.

If $|C'| = 1$, then $\mathcal{T}_{C'}$ consists of a single leaf containing $c_1(C')$, thus, with probability 1:

$$\|x - \mathcal{T}_{C'}\|_1 \leq O(1)\|x - c_1(C')\|_1 \tag{19}$$

If $|C'| \geq 2$, let v' be the leaf where x is assigned by the partially built tree $\mathcal{T}_2$ at line 14. Let c'_j be the single center in v'. Because the RANDOMCUTS algorithm is run on centers $\{c'_j, c_1(C')\}$, we have

$$\mathbb{E}[\|x - \mathcal{T}_{C'}(x)\|_1] \leq O(1)\|x - c_1(C')\|_1.$$

Since we established the above bounds for every non-empty subset $C' \subseteq C$, we have:

$$\alpha(1) \leq O(1).$$

Case $i \geq 2$. Let a not-empty subset $C' \subseteq C$ with $|C'| \geq 2$.

Let v be the leaf where x is assigned in $\mathcal{T}_1$. Observe that center $c_i(C')$ belongs to class $P_{f(i)}$. By Lemma 5.2 with high priority class $P_{f(i)}$ and low priority class $\bigcup_{j\in[s]\setminus f(i)} P_j$, we get that

$$\begin{aligned}\mathbb{E}[\|x - \bar{c}_{f(i)}\|_1] &\leq O(\log|P_i|) \min_{c'\in P_{f(i)}} \|x - c'\|_1 \\ &\leq O(\log i)\|x - c_i(C')\|_1 \end{aligned} \tag{20}$$

By construction of the priority classes, $\bar{c}_1 \preccurlyeq \bar{c}_2 \preccurlyeq \cdots \preccurlyeq c_s$ and so the rank of $\bar{c}_{f(i)}$ in C'_v is $r_{C'_v}(\bar{c}_{f(i)}) = f(i)$. By definition of the function α, because $C'_v \subseteq C$ and $|C'_v| \geq f(i)$,

$$\begin{aligned}\mathbb{E}[\|x - \mathcal{T}_{C'}(x)\|_1 \mid C'_v] &\leq \alpha(f(i))\|x - c_{f(i)}(C'_v)\|_1 \\ &= \alpha(f(i))\|x - \bar{c}_{f(i)}\|_1.\end{aligned}$$

Taking expectations on both sides yields:

$$\mathbb{E}[\|x - \mathcal{T}_{C'}(x)\|_1] \leq \alpha(f(i))\mathbb{E}[\|x - \bar{c}_{f(i)}\|_1]. \tag{21}$$

By combining Equations 20 and 21, we get

$$\mathbb{E}[\|x - \mathcal{T}_{C'}(x)\|_1] \leq O(\log i)\alpha(f(i))\|x - c_i(C')\|_1. \tag{22}$$

Equation 22 holds for every non empty $C' \subseteq C$ with $|C'| \geq 2$ and so, by definition of the function α:

$$\alpha(i) \leq O(\log i)\alpha(f(i))\|x - c_i(C')\|_1.$$

The function $\ell(k)$ satisfies a similar recursive relation. Specifically, the tree $\mathcal{T}_1$ has exactly $k-1$ leaves. Since the number of priority classes is $f(k)$, after copying centers in every leaf of $\mathcal{T}_1$, each leaf contains $f(k-1)$ centers. The algorithm RANKEDPRIORITREE is recursively applied to every leaf and by definition of $\ell(k)$, the tree $\mathcal{T}_2$ has at most:

$$(k-1)\ell(f(k-1))$$

leaves with probability 1. The final tree $\mathcal{T}$ has at most 2 times the number of leaves of $\mathcal{T}_2$ and so:

$$\ell(k) \leq 2(k-1)\ell(f(k-1)).$$

Because $\ell(1) = 1$, there exists a constant $A' > 0$ such that

$$\ell(k) \leq O\left((A')^m k \log\log k \cdot \log\log\log\log k \cdot \log^{(6)} k \cdots \log^{(2m)} k\right)$$

where $m = O(\log^*(k))$, and by a similar argument as for function α we have,

$$\ell(k) = O_\varepsilon\left(k(\log\log k)^{1+\varepsilon}\right).$$

□

D LOWER BOUNDS FOR PRIORITY-AWARE EXPLAINABLE CLUSTERING

In the standard non-priority-aware setting, explainable clustering decision trees typically have exactly k leaves, one for each center. The priority-aware decision trees constructed in this paper depart from this baseline and use more than k leaves. In this section, we show that this additional tree complexity is not merely an artifact of our algorithm, but is necessary for obtaining nontrivial priority-aware guarantees. Even if we aim to achieve an improved competitive ratio only for a *single* center c_1, we need to have at least $k + \Omega(\log k)$ leaves; otherwise, the competitive ratio α_1 for c_1 is $\Omega(\log k)$ matching the standard non-priority-aware guarantee up to constants and therefore yielding no asymptotic priority-aware improvement for c_1.

Let α_i be the competitive ratio for center c_i; specifically, for every data point $x \in X$ assigned to c_i in the reference clustering (that is, c_i is the closest center to x among $c_1, \ldots, c_k$), we have

$$\mathbb{E}\big[\|x - \mathcal{T}(x)\|_1\big] \le \alpha_i \|x - c_i\|_1. \tag{23}$$

Throughout this section, we assume that all α_i are finite.

We also show that even in the relaxed setting where the tree is allowed extra leaves, as long as the total number of leaves is $o(k \log k)$, the average competitive ratio cannot be improved below logarithmic. Specifically, if the number of leaves is $o(k \log k)$, as is the case in Theorem 1.2, then the average competitive ratio $\alpha = \frac{1}{k}(\alpha_1 + \cdots + \alpha_k)$ is at least $\Omega(\log k)$. Thus, the average competitive ratio we obtain in our paper is optimal up to lower-order factors in the regime where the number of leaves is $o(k \log k)$.

For a randomized decision tree $\mathcal{T}$, let $\mathrm{leaves}(\mathcal{T})$ denote the expected number of leaves in $\mathcal{T}$. As in the rest of the paper, let $\mathcal{T}(x)$ denote the center assigned to x by the decision tree. Moreover, in this section, we also write $\widehat{\mathcal{T}}(x)$ for the corresponding leaf of the decision tree.

Theorem D.1. *Let $\eta_1 = \frac{1}{500}$ and $\eta_2 = \frac{1}{100}$. For every $k \ge 3$, there exists an instance of k-medians such that, for every randomized priority-aware explainable clustering $\mathcal{T}$ with all $\alpha_i < \infty$, the following holds.*

- *If $\alpha_1 \le \eta_1 \ln k$, then*
$$\mathrm{leaves}(\mathcal{T}) \ge k + \eta_2 \ln k.$$
- *If $\mathrm{leaves}(\mathcal{T}) < \frac{\eta_2}{2} k \ln k$, then*
$$\alpha = \frac{\alpha_1 + \cdots + \alpha_k}{k} \ge \frac{\eta_1}{2} \ln k.$$

Proof. Our construction is a modification of that of Moshkovitz et al. (2020). We will need the following claim from their paper.

Claim D.2 (Claim A.2 in Moshkovitz et al. (2020))**.** *For every $k \ge 3$ and $d = k^3$, there are k points $\mathcal{C} \subseteq \{\pm 1\}^d$ that have the following properties for every $\varepsilon \ge \frac{\ln k}{\sqrt{k}}$:*

1. *For every distinct $c, c' \in \mathcal{C}$, the Hamming distance between c and c' is at least $\frac{d}{4}$.*
2. *For every $\ell \le L = \frac{\ln k}{50}$ indices in $[d]$, and every assignment to these indices, the number of points in $\mathcal{C}$ that have this assignment is at least $k\left(\frac{1}{2^\ell} - \varepsilon\right)$.*

We use the set of points $\mathcal{C}$ provided by this claim as the set of centers. Denote the centers in $\mathcal{C}$ by $c_1, \ldots, c_k$ in an arbitrary order. We now describe the set of data points. For each center c_i, we add c_i itself to the set of data points. Also, we add the d neighbors of c_i in the hypercube $\{-1, 1\}^d$ to the set of data points (each neighbor x of c_i is obtained by flipping the sign of a single coordinate). Point c_i and its neighbors are assigned to center c_i in the optimal clustering. Since $k \ge 3$, we have, for $\varepsilon = \frac{\ln k}{\sqrt{k}}$,

$$k\left(\frac{1}{2^L} - \varepsilon\right) > 1.05,$$

and thus item 2 in Claim D.2 guarantees that there are at least 2 points in $\mathcal{C}$ with a given assignment.

First, observe that for every center c_i, leaf $\widehat{\mathcal{T}}(x)$ of $\mathcal{T}$ is labeled c_i (almost surely over the random choice of $\mathcal{T}$), since for $x = c_i$, we have

$$\mathbb{E}\left[\|\mathcal{T}(x) - c_i\|_1\right] \leq \alpha_i \|x - c_i\|_1 = 0,$$

and thus $\mathcal{T}(x) = c_i$ almost surely. We have identified k distinct leaves of $\mathcal{T}$: $\widehat{\mathcal{T}}(c_1), \ldots, \widehat{\mathcal{T}}(c_k)$ labeled $c_1, \ldots, c_k$, respectively.

Fix a center c_t. We show that if $\alpha_t \leq \eta_1 \ln k$, then there are at least $\eta_2 \ln k$ additional leaves labeled c_t in $\mathcal{T}$, in expectation. The statement of the theorem will then immediately follow.

Lemma D.3. *If $\alpha_t \leq \eta_1 \ln k$, then there are at least $\eta_2 \ln k + 1$ leaves labeled c_t in $\mathcal{T}$ in expectation.*

Proof. For $j \in [d]$, let $\mathbf{x}^j \in X$ be the data point obtained from c_t by flipping the j-th coordinate. Choose one of the data points $\mathbf{x}^j$ uniformly at random and consider $\mathcal{T}(\mathbf{x}^j)$. We are guaranteed that

$$\mathbb{E}\left[\|\mathcal{T}(\mathbf{x}^j) - \mathbf{x}^j\|_1\right] \leq \alpha_t \|\mathbf{x}^j - c_t\|_1 = 2\alpha_t, \tag{24}$$

where the expectation is over the random choice of $\mathcal{T}$ and $\mathbf{x}^j$.

Since all data points have only ± 1 coordinates, we may assume that all comparisons in the decision tree $\mathcal{T}$ are of the form $x_r \geq 0$ for some $r \in [d]$. Further, we may assume that the same comparison does not appear twice along the same branch of the tree.

Consider the branch B of $\mathcal{T}$ connecting the root to $\widehat{\mathcal{T}}(c_t)$. Let I be the set of indices r corresponding to comparisons $x_r \leq 0$ used along B, and let $R = |I|$ be the depth of $\widehat{\mathcal{T}}(c_t)$ in $\mathcal{T}$. Note that B, I, and R depend on the random tree.

We show that $R > \frac{\ln k}{50}$ almost surely. Indeed, if $R \leq \frac{\ln k}{50}$, then by Claim D.2, item 2, applied to the set I, there are at least 2 centers $c_i \in \mathcal{C}$ that satisfy exactly the same threshold inequalities $x_r \geq 0$ along B as c_t; for all of them, $\mathcal{T}(c_i) = \mathcal{T}(c_t)$. However, we showed that $\mathcal{T}(c_i) \neq \mathcal{T}(c_t)$ for $c_i \neq c_t$. We obtain a contradiction. We conclude that $R \geq L$ almost surely.

Consider data points $\mathbf{x}^j$ with $j \in I$. We now show that all leaves $\widehat{\mathcal{T}}(\mathbf{x}^j)$ for $j \in I$ are all distinct and, furthermore, all of them are distinct from $\widehat{\mathcal{T}}(c_t)$. Fix $j \in I$. Consider the branch B_j of $\mathcal{T}$ that connects the root to $\widehat{\mathcal{T}}(\mathbf{x}^j)$. Branches B and B_j diverge at the threshold node $x_j \geq 0$. Therefore, $B_j \neq B$ and thus $\widehat{\mathcal{T}}(\mathbf{x}^j) \neq \widehat{\mathcal{T}}(c_t)$. Now consider distinct $j, j' \in I$. Assume that the threshold $x_j \geq 0$ appears before $x_{j'} \geq 0$ along B. Then at the threshold node $x_j \geq 0$, branch B_j diverges from B, but $B_{j'}$ does not. We conclude that all leaves $\widehat{\mathcal{T}}(\mathbf{x}^j)$ are distinct and different from $\widehat{\mathcal{T}}(c_t)$.

Let $Q \subseteq I$ be the subset of indices for which $\mathcal{T}(\mathbf{x}^j) = c_t$. By Claim D.2, item 1, for $j \in I \setminus Q$, we have

$$\|\mathcal{T}(\mathbf{x}^j) - \mathbf{x}^j\|_1 \geq \|\mathcal{T}(\mathbf{x}^j) - c_t\|_1 - \|\mathbf{x}^j - c_t\|_1 \geq 2 \cdot \frac{d}{4} - 2 = \frac{d}{2} - 2.$$

From (24), we get the following chain of inequalities for j sampled uniformly at random from $[d]$:

$$2\alpha_t \geq \mathbb{E}\left[\|\mathcal{T}(\mathbf{x}^j) - \mathbf{x}^j\|_1\right] \geq \frac{\mathbb{E}[|I \setminus Q|]}{d} \cdot \left(\frac{d}{2} - 2\right) = \left(1 - \frac{4}{d}\right) \frac{\mathbb{E}[R - |Q|]}{2}.$$

Using that $d = k^3 \geq 27$ and thus $4\frac{d}{d-4} < 5$, we get

$$\mathbb{E}[|Q|] \geq \mathbb{E}[R] - 5\alpha_t \geq L - 5\alpha_t \geq \frac{\ln k}{50} - \frac{\ln k}{100} = \frac{\ln k}{100} = \eta_2 \ln k.$$

Finally, we note that the leaves $\widehat{\mathcal{T}}(c_t)$ and $\widehat{\mathcal{T}}(\mathbf{x}^j)$ for all $j \in Q$ are labeled c_t and are distinct. Thus, the expected number of leaves labeled c_t is at least

$$\mathbb{E}[|Q|] + 1 \geq \eta_2 \ln k + 1.$$

□

We now apply this lemma to obtain the desired lower bounds. Letting $t = 1$, we get that, under the assumption $\alpha_1 \leq \eta_1 \ln k$ of item 1, there are at least $\eta_2 \ln k + 1$ leaves labeled c_1; additionally, for every $i > 1$, there is at least one leaf – namely $\widehat{\mathcal{T}}(c_i)$ – labeled c_i. Therefore, there are at least $k + \eta_2 \ln k$ leaves in $\mathcal{T}$ in expectation. This proves item 1.

For item 2, we prove the contrapositive: assume that

$$\alpha = \frac{\alpha_1 + \cdots + \alpha_k}{k} < \frac{\eta_1 \ln k}{2}.$$

By Markov's inequality, for at least half of the indices t, we have $\alpha_t \leq \eta_1 \ln k$. Let T be the set of such indices. Applying Lemma D.3 to each $t \in T$, we get that there are at least $1 + \eta_2 \ln k$ leaves labeled c_t, in expectation. Summing over all $t \in T$, we get that the total number of leaves is at least

$$\frac{k}{2}(1 + \eta_2 \ln k) > \frac{\eta_2}{2} k \ln k$$

in expectation. □