On Complexity of 1-Center in Various Metrics

Amir Abboud
Weizmann Institute of Science, Rehovot, Israel

MohammadHossein Bateni
Google Research, Mountain View, CA, USA

Vincent Cohen-Addad
Google Research, Zürich, Switzerland

Karthik C. S.
Rutgers University, New Brunswick, NJ, USA

Saeed Seddighin
Toyota Technological Institute at Chicago, IL, USA

Abstract

We consider the classic 1-center problem: Given a set P of n points in a metric space find the point in P that minimizes the maximum distance to the other points of P. We study the complexity of this problem in d-dimensional ℓ_p-metrics and in edit and Ulam metrics over strings of length d. Our results for the 1-center problem may be classified based on d as follows.

- **Small d.** Assuming the hitting set conjecture (HSC), we show that when $d = \omega(\log n)$, no subquadratic algorithm can solve the 1-center problem in any of the ℓ_p-metrics, or in the edit or Ulam metrics.
- **Large d.** When $d = \Omega(n)$, we extend our conditional lower bound to rule out sub*quartic* algorithms for the 1-center problem in edit metric (assuming Quantified SETH). On the other hand, we give a $(1+\epsilon)$-approximation for 1-center in the Ulam metric with running time $\widetilde{O_\varepsilon}(nd + n^2\sqrt{d})$.

We also strengthen some of the above lower bounds by allowing approximation algorithms or by reducing the dimension d, but only against a weaker class of algorithms which list all requisite solutions. Moreover, we extend one of our hardness results to rule out subquartic algorithms for the well-studied 1-median problem in the edit metric, where given a set of n strings each of length n, the goal is to find a string in the set that minimizes the sum of the edit distances to the rest of the strings in the set.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Facility location and clustering; Theory of computation → Problems, reductions and completeness; Theory of computation → Unsupervised learning and clustering

Keywords and phrases Center, Clustering, Edit metric, Ulam metric, Hamming metric, Fine-grained Complexity, Approximation

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2023.1

Category APPROX

Related Version *Full Version*: `https://arxiv.org/abs/2112.03222`

Funding *Amir Abboud*: This project has received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No 101078482). Additionally, the author is supported by an Alon scholarship and a research grant from the Center for New Scientists at the Weizmann Institute of Science.
Karthik C. S.: This work was supported by Subhash Khot's Simons Investigator Award and by a grant from the Simons Foundation, Grant Number 825876, Awardee Thu D. Nguyen and by the National Science Foundation under Grant CCF-2313372.
Saeed Seddighin: Supported by a Google research gift.

1 Introduction

Given a set of points P in a metric space, finding the point that "best" represents P is a fundamental question in both discrete and continuous optimization. Motivated by applications ranging from machine learning to computational biology, this question has naturally received a large amount of attention through the years.

The objective can be phrased in various ways: In the median problem, the goal is to find the point p that minimizes the sum of the distances to the points in P; in the mean problem, it is the point that minimizes the sum of distances squared; while in the center problem, it is the point p that minimizes the maximum distance from a point of P to p. When the metric is the ℓ_2 (Euclidean) metric, the question of computing the geometric median dates back to the 17th century, when Torricelli was looking for a solution for the case $|P| = 3$, and to whom Fermat described an explicit solution. More recently, the question of computing the center has also become central in applications arising, e.g., in machine learning [11], to compute the minimum enclosing ball of a set of points, or in computational biology, to find a good representative of a set of strings (representing molecular sequences) (e.g., [30]). This fundamental computational geometry problem which has applications to various domains, is the problem we consider in this paper.

Formally, in the (often referred to as the *discrete*) 1-center problem, the input is a set of points P in a metric space, and the goal is to find a point of P that minimizes the maximum distance to the points in P. When doing data summarization or compression, the discrete version often makes more sense: Given a set of, say n strings, taking the most representative string among the input strings, or at least in the set of grammatically (or semantically) meaningful strings is much more insightful than taking an arbitrary string as representative. This also applies more globally, outputting a data element that has been observed provides a better summary than a data element that has been forged by the algorithm and that may be unlikely to exist in the real-world. From a computational complexity standpoint, this problem can be easily solved in time $O(|P|^2 f(d))$ where $f(d)$ is the time required to compute the distance of two points. This can be done by enumerating all possible choices for the center; and for each choice computing the distance from each point in P; then outputting the best center. However, is this naïve algorithm the best we can do?

The computational geometry community has done extensive work on the above question since the 80s. For metrics such as ℓ_1 or ℓ_2, computing the center has received a large deal of attention. When the dimension is assumed to be a constant, there exist *barely subquadratic* algorithms for the ℓ_2 metric, while there exists near-linear time algorithms for the ℓ_1 case (for a discussion on this we refer the reader to [35]). For the case of string metrics, such as Ulam or Edit distance metrics, nothing better than the $O(|P|^2 f(d))$ (where d is the string length) "brute-force" algorithm is known.

Understanding how fast the 1-center problem can be solved in these different metrics is not only interesting from a computational complexity point of view, but also from the perspective of an improved understanding of the geometry of these metrics. For example, is the geometry of the ℓ_1 metric "more amenable" for designing algorithms than the ℓ_2 one? Is the Edit distance metric hard for such problems? We also believe that understanding the geometry of the Ulam and Edit distance metrics, which one may interpret as generalization of the Hamming metric, is not only a very basic computational geometry question, but also would likely lead to better algorithms for these widely-studied problems. We thus ask:

How fast can the 1-center problem be solved or approximated in ℓ_p-metrics and stringology metrics such as Ulam or Edit distance?

1.1 Our Results

In this paper, we take a step towards answering the above question by providing lower and upper bounds on solving the 1-center problem in ℓ_p, Ulam, and Edit distance metrics.

Assuming the Hitting Set Conjecture (HSC), we provide a strong conditional lower bound for the 1-center problem, in a pleathora of metrics.

▶ **Theorem 1** (see Theorem 9, Corollary 12, and Corollary 14 for formal statement). *Assuming* HSC*, no algorithm running in time $n^{2-o(1)}$ can given as input a set of points/strings P of dimension/length d solve the discrete 1-center problem in Edit/Ulam/ℓ_p metric, where $|P| = n$, $d = \widetilde{\Omega}(\log n)$, and $p \in \mathbb{R}_{\geq 1} \cup \{0\}$.*

Moreover, by assuming a stronger complexity theoretic assumption we can strengthen this lower bound in the case when $d = \text{poly}(n)$ for the Edit metric. For the sake of presentation, we state our result below when $d = n$.

▶ **Theorem 2** (see Theorem 15). *Assuming Quantified SETH, no algorithm running in time $n^{4-o(1)}$ can given as input a set of n strings of length $d := n$ each, solve the discrete 1-center problem in Edit metric.*

It's worth emphasizing that the above lower bound for the edit metric is a rare quartic lower bound in fine-grained complexity. It's true that, conceptually, it's not unexpected because there's a quadratic hardness from the 1-center problem and a quadratic hardness from edit distance, so we would expect the combined problem to be quartic. But we find it noteworthy that this actually works on the technical level because complexity theory is full of notorious examples where such "semi-stitching techniques" completely fail (for example KRW games [20]).

Note that we cannot expect such lower bounds for the 1-center problem in ℓ_p-metrics when $d = n$, as one can compute all pairwise distances within a point-set in subcubic time using fast matrix multiplication.

Next, we complement the lower bounds by the following subcubic approximation scheme for the 1-center in the Ulam metric.

▶ **Theorem 3.** *There exists a $1+\epsilon$ approximation algorithm for the 1-center under Ulam metric that runs in time $\widetilde{O}_\epsilon(nd + n^2\sqrt{d})$.*

It is worth emphasizing here that for the (discrete) 1-center problem in any metric space, an arbitrary point in the input is a 2-approximate solution. Also note that exact 1-center in Ulam metric can be solved in $O(n^2 d)$ time. It remains an open problem to show a conditional lower bound of $n^{3-o(1)}$ for computing the 1-center in the Ulam metric for n strings each of length n.

Finally, we strengthen some of the lower bounds above, but against a weaker class of algorithms, specifically, against algorithms which list all requisite solutions. Using the ideas in [35, 10], assuming HSC, we rule out subquadratic algorithms that can list all optimal solutions to the 1-center problem in the Euclidean metric even for very low $d = o(\log n)$ dimensions. At a high level, this result contrasts with both ℓ_1 and ℓ_∞ metrics where the 1-center in $o(\log n)$ dimensions can be solved in $n^{1+o(1)}$ time.

▶ **Theorem 4** (see Theorem 20 for formal statement). *Assuming* HSC*, there is no $n^{2-o(1)}$-time algorithm listing all optimal solutions to the 1-center problem in $7^{\log^* n}$ dimensions in the Euclidean metric.*

In the same spirit as above, by applying the distributed PCP framework [2, 31] we extend the lower bound in Theorem 1 against approximation algorithms which list all approximately optimal 1-centers.

▶ **Theorem 5** (see Theorem 22 for formal statement). *Assuming* HSC*, there is some $\delta > 0$, such that no $n^{2-o(1)}$-time algorithm can given as input a set of points/strings P of dimension/length d, list all $(1+\delta)$-approximate solutions to the 1-center problem in the Edit/Ulam/ℓ_p metric, where $|P| = n$, $d = \widetilde{\Omega}(\log n)$, and $p \in \mathbb{R}_{\geq 1} \cup \{0\}$.*

One may compute all pairwise distances in $\widetilde{O}(n^2)$ time for the inputs in Theorems 4 and 5, and then obtain the list of all optimal and approximately optimal solutions efficiently. Our theorems above say that one cannot do much better. It remains an intriguing open problem to extend the above two conditional lower bounds but against standard decision algorithms. We note that this involves breaking some technical barriers and in particular, developing techniques that go beyond the dimensionality reduction techniques of [35, 10] and the distributed PCP framework [2, 31] respectively.

We close this subsection by a short discussion about the Discrete 1-median problem in ℓ_p-metrics and string metrics. For the case when $d = n$, we can prove a result similar to Theorem 2 (see Remark 18). On the other hand for ℓ_p-metrics, one cannot prove a result similar to Theorem 1 for the 1-median problem, because the 1-median problem in Hamming and ℓ_1-metrics admits a near linear time algorithm and for the Euclidean metric, it is even unclear if the problem is in NP! (see discussion in [18].) Also note that by subsampling coordinates, we can approximate 1-median in all ℓ_p-metrics to $(1+\varepsilon)$ factor, for any $\varepsilon > 0$ in near linear time.

1.2 Related Work

We now review the related work on the 1-center problem, and the related 1-median problem. Both problems may be considered in the *discrete* or *continuous* settings. The discrete[1] version asks the center or median to be picked from an input set of points, while in the continuous version, the "center" is an arbitrary element of the metric. See [13] for a discussion on these two settings.

Below, we mainly discuss 1-center problem in stringology metrics as the literature on related work in ℓ_p metrics is too vast to survey (but the interested reader may look at [12, 25, 23, 16] and the references therein).

Metrics arising in stringology

We now review results on the 1-center problem in metric spaces arising from stringology applications. Let Σ be an alphabet, often the binary alphabet $\{0, 1\}$. Consider the set of strings $\Sigma^* = \Sigma^L$ of length L, with a metric distance $D : \Sigma^* \times \Sigma^* \mapsto \mathbb{R}$. Researchers have mainly considered the following metrics defined over this space:

- **Edit distance** (ED or Levenshtein distance): The minimum number of single-character insertions, deletions, and substitutions required to change one string to the other.
- **Hamming distance or ℓ_1 over binary alphabets** (HD): A special case of edit distance, where only substitutions are allowed.
- **Ulam distance** (UD): Same as edit distance with the restriction that the input strings may not contain any character more than once.

[1] Sometimes called the *medoid* problem in contrast to *median*, *generalized* median, or *Steiner* string.

For most of the above metrics, one need to incorporate into the running times obtained for simpler metric such as ℓ_1 or ℓ_2 the time it takes to compute the exact or approximate distance between any two points of the space. Naumovitz et al. [28] show how to approximate UD within factor $1+\varepsilon$ in time $\widetilde{O}(d/\eta+\sqrt{d})$ if the distance is η. This result is tight up to log factors.

Turning back to the 1-center and 1-median problem, note the *discrete* versions can be solved exactly via $O(n^2)$ distance computations, giving trivial $\widetilde{O}(n^2 d)$-time algorithms for the case of UD. In Section 4, we show that this can be improved to $\widetilde{O}(n^2\sqrt{d})$ for $1+\varepsilon$ approximation if we combine two algorithms [26, 28] for computations of UD. Note that a 2-approximation is trivial, as we can output any string as the hub in the case of 1-center or a random string in the case of 1-median.

Recently [9] made progress on obtaining better approximation algorithms for the continuous 1-Median problem in UD, where the median can be picked from anywhere in space, by presenting the first polynomial-time constant-factor approximation algorithm with approximation guarantee smaller than 2 as well as an exact algorithm for the case where the input contains *three* strings. They observe that if the average distance to median is $\Omega(d)$, picking the best string as the median already gives an approximation better than two. Now the problem is reduced to the above case if the total cost is mostly due to a small subset of characters. Otherwise, they argue that one can deduce the relative ordering of a good portion of the optimal median by looking at pairs of characters whose relative order is consistent in *most* input strings.

Note that in the continuous case, for constant d or constant n, the median and center problems are both solvable in polynomial time for string problems. De la Higuera and Casacuberta [15] prove that median and center are both NP-hard. Nicolas and Rivals [29] lift the restrictions and show that median and center are both NP-hard and $W[1]$-hard (when parameterized by n, the number of strings), even for binary alphabets. Prior to these works, NP-hardness of median was only known for ED when the substitutions have specific costs for each pair of characters [33]. Li et al. [24, 25] give a PTAS for the HD 1-center problem,[2] augmenting the LP-based PTAS for super-logarithmic d [5]. The HD 1-center problem is known to be NP-hard [16, 23].[3] Previously the best polynomial-time approximation ratio was $\frac{4}{3}+\varepsilon$ in general [23, 19], with an exact algorithm known for constant d (optimal value) [34].

1.3 Organization of the Paper

In Section 2, we provide the formal definition of 1-center problem and the various hypotheses used in the paper. In Section 3, we provide conditional lower bounds against exact algorithms that compute 1-center when $d=\omega(\log n)$. Next, in Section 4, we provide a subcubic approximation algorithm for 1-center in Ulam metric when $d=n$. Finally, in Section 5, we provide some hardness of approximation results (Theorem 5).

2 Preliminaries

▶ **Definition 6** (Discrete 1-center). *Let (X,Δ) be a metric space. Given a set of points $P\subseteq X$ in the metric space, find x in P which minimizes the maximum distance to every other point.*

[2] This is the *closest string* problem. They also give a PTAS for the *closest substring* problem, which assumes that the cost of deletions from the input strings (∞ for HD) is zero.

[3] Note that median is solvable exactly for HD.

Perhaps the most popular assumption for proving conditional lower bounds for polynomial time problems is the Orthogonal Vectors Hypothesis (OVH) that is implied by the Strong Exponential Time Hypothesis (SETH). Unfortunately, the logical structure of these problems makes reductions to our 1-center problems difficult. This was observed already by Abboud, Vassilevska Williams, and Wang [3] in the context of 1-center in *graphs* (known as the Graph Radius problem) and has lead them to introduce the hitting set conjecture (HSC): a stronger variant of OVH that facilitates reductions to problems with different structure. A formal barrier for establishing HSC (and similarly also any hardness results for 1-center) under SETH was presented by Carmosino et al. [8].

▶ **Definition 7** (HSC). *For every $\varepsilon > 0$ there exists $c > 1$ such that no algorithm running in time $n^{2-\varepsilon}$ can, given as input two collections of n-many subsets $\mathcal{A}$ and $\mathcal{B}$ of the universe $U := [c \log n]$, determine if there exists S in $\mathcal{A}$ which has non-empty intersection with every subset in $\mathcal{B}$.*

The difference between HSC and OVH is in the quantifiers: $\exists\forall$ versus $\exists\exists$. Studying the *polyline simplification* problem, Bringmann and Chaudhury [6] proposed a further strengthening with more quantifiers. Just like OVH is implied by SETH, an assumption about k-SAT, so too can HSC and its generalizations with more quantifiers be based on the hardness of a quantified version of k-SAT; an assumption called *Quantified-SETH*. Interestingly, the previous papers using Quantified-SETH [6, 1] only needed its special case where the quantifier structure is $\forall\exists$; whereas in this paper we benefit from a $\exists\forall\exists$ structure that has one more alternation.

The specific hardness assumption (implied by Quantified SETH) that we need is the following; we refer to [6, 1] for further discussion on Quantified-SETH and to [3, 36] for further discussion on HSC and on the need for assumptions with other quantifier structures.

▶ **Definition 8** ($\exists\forall\exists\exists$OVH). *For every $\varepsilon > 0$ there exists $c > 1$ such that no algorithm running in time $n^{4-\varepsilon}$ can, given as input four collections of n-many subsets $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$ of the universe $U := [c \log n]$, determine if there exists S_A in $\mathcal{A}$ such that for all S_B in $\mathcal{B}$ there exist $S_C \in \mathcal{C}$ and $S_D \in \mathcal{D}$ such that the intersection $S_A \cap S_B \cap S_C \cap S_D = \emptyset$ is empty.*

3 Exact Lower Bounds for 1-center

In this section, we prove conditional lower bounds for the 1-center problem. We start with some high-level remarks about the reductions and our contributions.

Previous work (for example [31, 14]) has already designed reductions from SETH and OVH to *closest pair* kinds of questions for the metrics we consider, and our work can be viewed as lifting these results to the 1-center question. As discussed in Section 2 this requires a new starting assumption (either Quantified SETH or the Hitting Set Conjecture) that has a different structure. Thus, technically, the main contribution is to adapt the gadgetry of previous work into new reductions with a different structure. In some cases, fundamental difficulties arise and we can only resolve them by requiring that the algorithm lists all solutions.

In all our reductions, we first reduce to the *Discrete 1-center with Facilities*, where given a set of clients $C \subseteq X$ and a set of facilities $F \subseteq X$ in the metric space, the goal is to find x in F which minimizes the maximum distance to every point in C. We then reduce a hard instance (F, C) of the Discrete 1-center with Facilities problem to an instance P of the standard Discrete 1-center problem (without facilities) by adding a few additional coordinates to points in $F \cup C$ and then introducing a new point/string s such that it is far

from every point in C (in comparison to its distance from the points/strings in F). Thus we ensured that the 1-center of $P := F \cup C \cup \{s\}$ must be from F. Nevertheless, for the sake of compactness, this two step reduction in the proofs of this section is sometimes written as a one step reduction.

This section is organized as follows. In Section 3.1, we show the conditional subquadratic time lower bounds for 1-center in various metrics (Theorem 1). Next, in Section 3.2, we show the conditional subquartic time lower bound for 1-center in edit metric (also Theorem 2) and explain how to adapt it for 1-median. Finally, in Section 3.3, we show that there are no subquadratic listing algorithms for Euclidean 1-center even in low dimensions (Theorem 4).

3.1 Subquadratic Lower Bounds for 1-center when $d = \omega(\log n)$ in String and ℓ_p-metrics

In this subsection, we show that subquadratic time algorithms for 1-center do not exist in ℓ_p-metrics, Ulam metic, and edit metric, when $d = \omega(\log n)$.

▶ **Theorem 9** (Subquadratic Hardness of 1-center in ℓ_p-metrics). *Let $p \in \mathbb{R}_{\geq 1} \cup \{0\}$. Assuming* HSC*, for every $\varepsilon > 0$, there exists $c > 1$ such that no algorithm running in time $n^{2-\varepsilon}$ can, given as input a point-set $P \subseteq \{0,1\}^d$, solve the discrete 1-center problem in ℓ_p-metric, where $|P| = n$ and $d = c \log n$.*

Proof. Let $(\mathcal{A} := (S_1, \ldots, S_n), \mathcal{B} := (T_1, \ldots, T_n), U)$ be an instance arising from HSC. We construct a point-set $P \subseteq \{0,1\}^d$ where $|P| = 2n + 1$ and $d = 5 \cdot |U| + 2$. We build the two maps $\tau_{\mathcal{A}} : \mathcal{A} \to \{0,1\}^d, \tau_{\mathcal{B}} : \mathcal{B} \to \{0,1\}^d$ and a special point $s \in \{0,1\}^d$ and the point-set P is then simply defined to be the union of $\{s\}$ and the images (range) of $\tau_{\mathcal{A}}$ and $\tau_{\mathcal{B}}$, i.e.,

$$P := \{\tau_{\mathcal{A}}(S) \mid S \in \mathcal{A}\} \cup \{\tau_{\mathcal{B}}(T) \mid T \in \mathcal{B}\} \cup \{s\}.$$

Let $U := \{u_1, \ldots, u_m\}$. We define our special point s as follows:

$$\forall i \in [5m+2],\ s_i := \begin{cases} 0 & \text{if } 1 \leq i \leq 3m \\ 1 & \text{if } 3m+1 \leq i \leq 5m+2 \end{cases}$$

For any $S \in \mathcal{A}$ we define $\tau_{\mathcal{A}}(S)$ as follows:

$$\forall i \in [5m+2],\ \tau_{\mathcal{A}}(S)_i := \begin{cases} 1 & \text{if } u_i \in S \text{ and } 1 \leq i \leq m \\ 0 & \text{if } u_i \notin S \text{ and } 1 \leq i \leq m \\ 0 & \text{if } u_{i-m} \in S \text{ and } m+1 \leq i \leq 2m \\ 1 & \text{if } u_{i-m} \notin S \text{ and } m+1 \leq i \leq 2m \\ 0 & \text{if } 2m+1 \leq i \leq 4m+1 \\ 1 & \text{if } 4m+2 \leq i \leq 5m+2 \end{cases}$$

For any $T \in \mathcal{B}$ we define $\tau_{\mathcal{B}}(T)$ as follows:

$$\forall i \in [5m+2],\ \tau_{\mathcal{B}}(T)_i := \begin{cases} 1 & \text{if } u_i \in T \text{ and } 1 \leq i \leq m \\ 0 & \text{if } u_i \notin T \text{ and } 1 \leq i \leq m \\ 0 & \text{if } m+1 \leq i \leq 2m \\ 0 & \text{if } u_{i-2m} \in T \text{ and } 2m+1 \leq i \leq 3m \\ 1 & \text{if } u_{i-2m} \notin T \text{ and } 2m+1 \leq i \leq 3m \\ 0 & \text{if } 3m+1 \leq i \leq 5m+2 \end{cases}$$

Notice that for any S, S' in $\mathcal{A}$ and T in $\mathcal{B}$, we have

$$\begin{aligned}\|\tau_{\mathcal{A}}(S) - \tau_{\mathcal{B}}(T)\|_p &= (|S| + |T| - 2 \cdot |S \cap T| + m - |S| + m - |T| + m + 1)^{1/p} \\ &= (3m + 1 - 2 \cdot |S \cap T|)^{1/p}.\end{aligned}$$

$$\|\tau_{\mathcal{A}}(S) - \tau_{\mathcal{A}}(S')\|_p \leq (2m)^{1/p}.$$

$$\|\tau_{\mathcal{A}}(S) - s\|_p = (2m + 1)^{1/p}.$$

$$\|\tau_{\mathcal{B}}(T) - s\|_p = (3m + 2)^{1/p}.$$

Suppose there exists S in $\mathcal{A}$ such that it intersects with every subset T in $\mathcal{B}$ then $\tau_{\mathcal{A}}(S)$ has distance strictly less than $(3m + 1)^{1/p}$ with $\tau_{\mathcal{B}}(T)$ for every T in $\mathcal{B}$. Additionally, $\tau_{\mathcal{A}}(S)$ has distance at most $(2m)^{1/p}$ with $\tau_{\mathcal{A}}(S')$ for any $S' \in \mathcal{A}$ and distance $(2m + 1)^{1/p}$ with s. Therefore, $\tau_{\mathcal{A}}(S)$ is at distance at most $(3m)^{1/p}$ from every point in P.

On the other hand, if for every S in $\mathcal{A}$ there exists T in $\mathcal{B}$ such that S and T are disjoint, then we show that for any point x in P there is a point y in P such that $\|x-y\|_p \geq (3m+1)^{1/p}$. Suppose $x := \tau_{\mathcal{B}}(T)$ for some $T \in \mathcal{B}$ then we have x is at distance $(3m+2)^{1/p}$ from s. Similarly if $x := s$ then it is at distance $(3m+2)^{1/p}$ from every $\tau_{\mathcal{B}}(T)$ for all $T \in \mathcal{B}$. Finally, if $x := \tau_{\mathcal{A}}(S)$ for some $S \in \mathcal{A}$ then from the soundness assumption we have that there exists T in $\mathcal{B}$ such that S and T are disjoint. Thus, x is at distance $(3m + 1)^{1/p}$ from $\tau_{\mathcal{B}}(T)$. ◀

▶ Remark 10. For the ℓ_∞-metric, we can solve Discrete 1-center problem in $O(nd^2)$ time as follows. Given input point-set P, for every coordinate $i \in [d]$, determine a farthest pair of points (a_i, b_i) in the point-set when restricted to that coordinate. Note that discrete 1-center cost of P is equal to the cost of the discrete 1-center of the point-set $\{a_1, ..., a_d, b_1, ..., b_d\}$ when the center can be picked anywhere in P. Thus we can solve discrete 1-center in the ℓ_∞-metric in $O(nd^2)$ time, which is near linear time as long as $d = n^{o(1)}$.

The quadratic lower bound for 1-center in Ulam metric follows from the below lemma.

▶ **Lemma 11.** *Let Π_d denote the set of all permutations over $[d]$. For every $d \in \mathbb{N}$, there is a function $\eta : \{0,1\}^d \to \Pi_{2d}$, such that for all $a, b \in \{0,1\}^d$ the following holds:*

$$\mathsf{ed}(\eta(a), \eta(b)) = 2 \cdot \|a - b\|_0.$$

Moreover, for any $a \in \{0,1\}^d$, $\eta(a)$ can be computed in $O(d)$ time.

Proof. Let $a \in \{0,1\}^d$. We define $\eta(a)$ as follows:

$$\forall i \in [2d],\ \eta(a)[i] = \begin{cases} i & \text{if } i = 2k-1 \text{ and } a_k = 0 \text{ for some } k \in \mathbb{N} \\ i & \text{if } i = 2k \text{ and } a_k = 0 \text{ for some } k \in \mathbb{N} \\ i+1 & \text{if } i = 2k-1 \text{ and } a_k = 1 \text{ for some } k \in \mathbb{N} \\ i-1 & \text{if } i = 2k \text{ and } a_k = 1 \text{ for some } k \in \mathbb{N} \end{cases}$$

Fix some $k \in [d]$ and $a, b \in \{0,1\}^d$. If $a_k = b_k$ then notice that $\eta(a)[2k] = \eta(b)[2k]$ and $\eta(a)[2k-1] = \eta(b)[2k-1]$. If $a_k \neq b_k$ then $\eta(a)[2k] = \eta(b)[2k-1]$ and $\eta(a)[2k-1] = \eta(b)[2k]$. Since the characters do not repeat, we have that the optimal distance is obtained by swapping which amounts to two edit operations. ◀

▶ **Corollary 12** (Subquadratic Hardness of 1-center in Ulam metric)**.** *Assuming* HSC*, for every $\varepsilon > 0$ there exists $c > 1$ such that no algorithm running in time $n^{2-\varepsilon}$ can given as input a set P of n many permutations of $[d]$, solve the discrete 1-center problem in Ulam metric, where $|P| = n$ and $d = c \log n$.*

The quadratic lower bound for 1-center in Edit metric follows from the below lemma.

▶ **Lemma 13.** *For every $d \in \mathbb{N}$, there is a function $\eta : \{0,1\}^d \to \{0,1\}^{d'}$, such that for all $a, b \in \{0,1\}^d$ the following holds:*

$$\mathsf{ed}(\eta(a), \eta(b)) = \|a - b\|_0.$$

Moreover, for any $a \in \{0,1\}^d$, $\eta(a)$ can be computed in $O(d \log d)$ time.

Proof. Let $l_1, l_2, \ldots, l_d$ be d strings of length $10 \log d$ each made by realizing $10 \log d$ 0/1 bits uniformly at random. It follows that with high probability, the hamming distance as well as the edit distance of each pair l_i, l_j $(i \neq j)$ is $\Omega(\log d)$ [22]. For a string a, we define $\eta(a)$ in the following way: we make a string of size $d(10 \log d + 1)$ which consists of d consecutive blocks. Each block i starts with a_i and is followed by l_i. By putting all the blocks next to each other we obtain a string of size $d(10 \log d + 1)$ which we denote by $\eta(a)$. We prove in the following that $\mathsf{ed}(\eta(a), \eta(b)) = \|a - b\|_0$ holds for each pair of strings a and b.

$\mathsf{ed}(\eta(a), \eta(b)) \leq \|a - b\|_0$ immediately follows from the fact that by only toggling the first characters of some block of $\eta(a)$ we can turn $\eta(a)$ into $\eta(b)$ and this transformation only costs $\|a - b\|_0$. Note that we only toggle the first characters of the blocks whose corresponding characters in a and b are not the same.

Now, assume for the sake of contradiction that $\mathsf{ed}(\eta(a), \eta(b)) < \|a - b\|_0$ holds. This implies that for at least $d - \|a - b\|_0 + 1$ many blocks of a, the transformation cost is 0. In other words, for each of these blocks, there is a substring of length $10 \log d + 1$ in $\eta(b)$ which is completely the same as that block. Since the blocks are generated randomly, this can only happen if for some i, the i'th block of $\eta(a)$ is transformed into the i'th block of $\eta(b)$ and $a_i = b_i$. Thus, this implies that for at least $d - \|a - b\|_0 + 1$ different values of i we have $a_i = b_i$ which is contradiction. ◀

▶ **Corollary 14** (Subquadratic Hardness of 1-center in Edit metric)**.** *Assuming* HSC*, for every $\varepsilon > 0$ there exists $c > 1$ such that no algorithm running in time $n^{2-\varepsilon}$ can given as input a point-set $P \subseteq \{0,1\}^d$ solve the discrete 1-center problem in edit metric, where $|P| = n$ and $d = c \log n \log \log n$.*

Next we prove much higher lower bounds for the Edit metric when d is larger.

3.2 Subquartic Lower Bound for 1-center when $d = n$ in Edit metric

We now present our lower bound under Quantified SETH which offers a conceptual novelty since as discussed in Section 2 it is the first time (to our knowledge) that more than two quantifier alternations are utilized.

▶ **Theorem 15** (Subquartic Hardness of 1-center in Edit metric)**.** *Assuming Quantified SETH, for every $\varepsilon > 0$ no algorithm running in time $n^{4-\varepsilon}$ can given as input a point-set $P \subseteq \{0,1\}^n$ solve the discrete 1-center problem in edit metric, where $|P| = n$.*

Proof. Let us first reduce to the 1-center problem with facilities where there are two sets of binary strings, a set of clients C and a set of facilities F and the goal is to decide if there is a string in F that has ED at most τ to all strings in C. Given an instance $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ of $\exists\forall\exists\exists$OVH we construct C and F as follows.

First we will use the following lemma that follows from the existing reductions from OVH to ED [4, 7] (the latter reference gets the alphabet size down to 2).

▶ **Lemma 16** ([7]). *There are two linear time algorithms such that: each algorithm takes a set (A or B) of n binary vectors of length d and constructs (independently of the other) a binary string (s_A or s_B) of length $O(nd)$ with the following property for a fixed value τ that only depends on n, d: $ED(s_A, s_B) < \tau$ if there is a pair of orthogonal vectors $v_A \in A, v_B \in B$ and $ED(s_A, s_B) \geq \tau$ otherwise.*

For a set $X \subseteq [d]$ let $v(X) \in \{0,1\}^d$ be the natural encoding of the set as a binary vector where $v(X)[i] = 1$ iff $i \in X$. Note that two vectors are orthogonal iff the two corresponding sets are disjoint.

Now, for each set $S_A \in \mathcal{A}$ define the set of n vectors $A = \{v(X) \mid S_C \in \mathcal{C}, X = S_A \cap S_C\}$ representing the n sets that result from intersecting S_A with any set in $\mathcal{C}$. Similarly, for each set $S_B \in \mathcal{B}$ define the set of n vectors $B = \{v(X) \mid S_D \in \mathcal{D}, X = S_B \cap S_D\}$.

It follows that there is an orthogonal pair $v_A \in A, v_B \in B$ iff there exist $S_C \in \mathcal{C}$ and $S_D \in \mathcal{D}$ such that $S_A \cap S_B \cap S_C \cap S_D = \emptyset$. Therefore, if we use the algorithms in the above lemma to encode each set A with a string s_A and add it to the set of facilities F, and also encode each set B with a string s_B and add it into the set of clients C we get the reduction we are after: By the definition of the $\exists\forall\exists\exists$OVH problem, there is a string $s_A \in F$ such that for all strings $s_B \in C$ we have $ED(s_A, s_B) < \tau$ if and only if the given $\exists\forall\exists\exists$OVH instance is a yes-instance.

Finally, we reduce to the basic 1-center problem (without facilities). Suppose that the strings in F, C have length m. We simply construct an instance $P \subseteq \{0,1\}^{4m}$ of 1-center as follows.

$$P := \{1^m \circ f \circ 0^{2m} \mid f \in F\} \cup \{1^m \circ c \circ 1^{2m} \mid c \in C\} \cup \{0^{4m}\}.$$

The following simple facts about the ED of the transformed strings show that the optimal center in P must be from $\{1^m \circ f \circ 0^{2m} \mid f \in F\}$ and its cost would be smaller than $2m + \tau$ iff the original $\exists\forall\exists\exists$OVH instance is a yes-instance.

▷ Claim 17. Let x, y be two binary strings of length m with ED exactly t.

- $ED(1^m \circ x \circ 0^{2m}, 1^m \circ y \circ 0^{2m}) \leq m$.
- $ED(1^m \circ x \circ 0^{2m}, 0^{4m}) \leq 2m$.
- $ED(1^m \circ x \circ 0^{2m}, 1^m \circ y \circ 1^{2m}) = 2m + t$.
- $ED(1^m \circ x \circ 1^{2m}, 0^{4m}) \geq 3m$.

The first and second items follow from the straightforward alignment of the strings. The fourth item follows because $ED(0^\ell, 1^\ell) = \ell$. The third item requires a bit more care. First, to see that the ED is at most $2m + t$ consider the alignment that maps x to y optimally at cost t and then maps the other parts in the straightforward way at cost $2m$. Now suppose for contradiction that there was a better alignment. This alignment must match one of the new letters (from the transformation) to x or y; otherwise it would yield an alignment between x, y at cost smaller than t. But any alignment that matches the 1 letters on the left to x or y can be corrected so that the 1^m parts on the left are matched to each other, without affecting the cost. Similarly, any matching between the letters on the right to x or y can be corrected without increasing the cost. Suppose that a 0 from the right is matched to y. This implies that one of the 1's to the right of y must be deleted (because there are no longer enough 0's in the other string to get substituted with all of them), and a corrected alignment that instead substitutes the 0 with a 1 (reducing the number of such deletions by one) and leaves the mate in y unmatched does not have a higher cost. We refer the reader to [7] for more formal proofs of such claims. ◀

▶ Remark 18. The above reduction to Edit also work for the 1-median problem but with two key differences. The first and main difference is that, since we take the sum instead of the max, the cost in the objective may now be affected by non-orthogonal pairs and it is no longer sufficient to have gadgets that give distance $< \tau$ or $\geq \tau$ depending on the orthogonality. Instead, we need gadgets that guarantee that the distance is either $< \tau$ or exactly τ. Fortunately, such requirements can be accomplished, see e.g. Theorem 4 in [4]. The second difference is that we do not need the $\forall$ quantifier in the starting assumption; the sum is powerful enough to support the (standard) $\exists\exists\exists\exists$ structure type. Therefore, the lower bounds for 1-median can be based on the standard SETH rather than the Quantified-SETH.

3.3 Subquadratic Lower Bounds for 1-center in Low dimensional Euclidean space

In this subsection, we show that an algorithm with subquadratic running time does not exist in the low dimensional Euclidean metric for the 1-center problem. Our proof essentially adopts ideas developed in [35, 10]. We note that this result is surprising as there is a near linear time algorithm for 1-center in the low dimensional ℓ_1-metric.

▶ Remark 19. For the ℓ_1-metric, we can solve Discrete 1-center problem in $O(n2^{2d})$ time by using the isometric embedding of the ℓ_1-metric to the ℓ_∞-metric [27], and then noting Remark 10.

▶ **Theorem 20.** *Assuming* HSC*, there exists a constant $\eta > 1$ such that for every $\varepsilon > 0$, no algorithm running in time $n^{2-\varepsilon}$ can given as input a point-set $P \subseteq \mathbb{R}^d$ and a positive real α output all points in P whose 1-center cost in the Euclidean metric is at most α, where $|P| = n$, $d = \eta^{\log^* n}$, and representing each vector requires at most $\widetilde{O}(\log n)$ bits.*

Proof of Theorem 20. We prove the theorem statement by contradiction. Suppose for some $\varepsilon > 0$ there is an algorithm $\mathcal{T}$ running in time $n^{2-\varepsilon}$ that can given as input a point-set $P \subseteq \mathbb{R}^d$ and a positive real α output all points in P whose 1-center cost in the Euclidean metric is at most α, where $|P| = n$, $d = \eta^{\log^* n}$, and each vector is of at most $k \log n$ bit entries (for some constant integer k).

Let $(\mathcal{A} := (S_1, \ldots, S_n), \mathcal{B} := (T_1, \ldots, T_n), U)$ be an instance arising from HSC, where $|U| = c \log n$. We think of each set in $\mathcal{A}$ and $\mathcal{B}$ as its characteristic vector in $\{0,1\}^{c \log n}$. We show how we can decide this instance in $n^{2-\frac{\varepsilon}{2}}$ time using $\mathcal{T}$, thus contradicting HSC.

We need the following theorem from Chen [10].

▶ **Theorem 21** (Chen [10])**.** *Let b, ℓ be two sufficiently large integers. There is a reduction $\psi_{b,\ell} : \{0,1\}^{b \cdot \ell} \to \mathbb{Z}^\ell$ and a set $V_{b,\ell} \subseteq \mathbb{Z}$, such that for every $x, y \in \{0,1\}^{b \cdot \ell}$,*

$$x \cdot y = 0 \Leftrightarrow \psi_{b,\ell}(x) \cdot \psi_{b,\ell}(y) \in V_{b,\ell}$$

and

$$0 \leq \psi_{b,\ell}(x)_i < \ell^{6^{\log^*(b)} \cdot b}$$

for all possible x and $i \in [\ell]$. Moreover, the computation of $\psi_{b,\ell}(x)$ takes $\mathrm{poly}(b \cdot \ell)$ time, and the set $V_{b,\ell}$ can be constructed in $O\left(\ell^{O(6^{\log^*(b)} \cdot b)} \cdot \mathrm{poly}(b \cdot \ell)\right)$ time.

We use the above theorem with $\ell = 7^{\log^* n}$ and $b = |U|/\ell$. Note that if $\ell = 7^{\log^* n}$ then $\log\left(\ell^{6^{\log^*(b)} \cdot b}\right) = o(\log n)$. All of the below construction details appears in [35, 10] and we skip many of the calculations and claim proofs hereafter. Our contributions are mainly in

using these previously known constructions in a new way to prove the theorem statement. In particular, for every $t \in V_{b,\ell}$ we create an instance $(P_t \subseteq \mathbb{R}^{(\ell+1)^2+3}, \alpha := \sqrt{2n^5-1})$ of 1-center as follows.

For every[4] $S_i \in \mathcal{A}$ (resp. $T_j \in \mathcal{B}$) we first define a point $p_i^t \in \mathbb{Z}^{\ell+1}$ (resp. $q_j^t \in \mathbb{Z}^{\ell+1}$) as follows:

$p_i^t := (\psi_{b,\ell}(S_i), t)$ (resp. $q_j^t := (\psi_{b,\ell}(T_j), -1)$).

It is then easy to verify that $S_i \cap T_j = \emptyset$ if and only if there exists some $t \in V_{b,\ell}$ such that $\langle p_i^t, q_j^t \rangle = 0$. Next for every $p_i^t \in \mathbb{Z}^{\ell+1}$ (resp. $q_j^t \in \mathbb{Z}^{\ell+1}$) we define $\widetilde{p}_i^t \in \mathbb{Z}^{(\ell+1)^2}$ (resp. $\widetilde{q}_j^t \in \mathbb{Z}^{(\ell+1)^2}$) as follows:

$\forall a, b \in [\ell+1],\ \widetilde{p}_i^t(a,b) := p_i^t(a) \cdot p_i^t(b)$ (resp. $\widetilde{q}_j^t(a,b) := -q_j^t(a) \cdot q_j^t(b)$).

It is then straightforward to verify that $\langle p_i^t, q_j^t \rangle = 0$ if and only if $\langle \widetilde{p}_i^t, \widetilde{q}_j^t \rangle \geq 0$.

Finally, we have our pointset $P_t \in \mathbb{R}^{(\ell+1)^2+3}$ defined as follows:

$$P_t := \underbrace{\left\{ \left(\widetilde{p}_i^t, \sqrt{n^5 - \|\widetilde{p}_i^t\|_2^2}, 0, 0 \right) \middle| S_i \in \mathcal{A} \right\}}_{P_t^{\mathcal{A}}} \bigcup \underbrace{\left\{ \left(-\widetilde{q}_j^t, 0, \sqrt{n^5 - \|\widetilde{q}_j^t\|_2^2}, \sqrt{n^5} \right) \middle| T_j \in \mathcal{B} \right\}}_{P_t^{\mathcal{B}}} \cup \left\{ \vec{0} \right\},$$

where $\vec{0} = (0, 0, \ldots, 0)$.

It can then be verified that $\langle \widetilde{p}_i^t, \widetilde{q}_j^t \rangle \geq 0$ if and only if the distance between $\left(\widetilde{p}_i^t, \sqrt{n^5 - \|\widetilde{p}_i^t\|_2^2}, 0 \right)$ and $\left(-\widetilde{q}_j^t, 0, \sqrt{n^5 - \|\widetilde{q}_j^t\|_2^2} \right)$ is at least $\sqrt{2n^5}$; otherwise their distance is at most $\sqrt{2n^5-1}$. Also note that any pair of points in $P_t^{\mathcal{A}}$ or any pair of points in $P_t^{\mathcal{B}}$ are at distance at most $\sqrt{2n^5-1}$ from each other. Finallt, note that the distance between any point in $P_t^{\mathcal{B}}$ and $\vec{0}$ is exactly $\sqrt{2n^5}$ and the distance between any point in $P_t^{\mathcal{A}}$ and $\vec{0}$ is exactly $\sqrt{n^5}$.

We run $\mathcal{T}$ on $(P_t, \alpha := \sqrt{2n^5-1})$ for every $t \in V_{b,\ell}$. Let $\mathcal{O}_t \subseteq P_t$ be the output of running $\mathcal{T}$ on (P_t, α). In other words for every $t \in V_{b,\ell}$ and every $p \in \mathcal{O}_t$ we have that for every $p' \in P_t$, $\|p - p'\|_2 \leq \sqrt{2n^5-1}$.

We claim that there exists S in $\mathcal{A}$ such that it intersects with every subset T in $\mathcal{B}$ if and only if there exists $i \in [n]$ such that for all $t \in V_{b,\ell}$, we have $\left(\widetilde{p}_i^t, \sqrt{n^5 - \|\widetilde{p}_i^t\|_2^2}, 0 \right) \in \mathcal{O}_t$.

Suppose there exists S_{i^*} in $\mathcal{A}$ such that it intersects with every subset T in $\mathcal{B}$. Fix $t \in V_{b,\ell}$. We have that $\left(\widetilde{p}_{i^*}^t, \sqrt{n^5 - \|\widetilde{p}_{i^*}^t\|_2^2}, 0 \right)$ is at distance at most $\sqrt{2n^5-1}$ from every other point in $P_t^{\mathcal{A}}$ just from construction. Suppose there is $\left(-\widetilde{q}_j^t, 0, \sqrt{n^5 - \|\widetilde{q}_j^t\|_2^2} \right) \in P_t^{\mathcal{B}}$ such that their distance is greater than $\sqrt{2n^5-1}$ then from the construction, their distance must be $\sqrt{2n^5}$, which implies that $S_{i^*} \cap T_j = \emptyset$, a contradiction.

On the other hand, if for every S in $\mathcal{A}$ there exists T in $\mathcal{B}$ such that S and T are disjoint then we show that for any point $\left(\widetilde{p}_i^t, \sqrt{n^5 - \|\widetilde{p}_i^t\|_2^2}, 0 \right)$ there exists $t \in V_{b,\ell}$ such that $\left(\widetilde{p}_i^t, \sqrt{n^5 - \|\widetilde{p}_i^t\|_2^2}, 0 \right) \notin \mathcal{O}_t$. Fix $i \in [n]$. Let $T_j \in \mathcal{B}$ such that $S_i \cap T_j = \emptyset$. Let $t^* := \psi_{b,\ell}(S_i) \cdot \psi_{b,\ell}(T_j)$. From Theorem 21 we have that $t^* \in V_{b,\ell}$. Thus $\left(\widetilde{p}_i^{t^*}, \sqrt{n^5 - \|\widetilde{p}_i^{t^*}\|_2^2}, 0 \right)$ and $\left(-\widetilde{q}_j^{t^*}, 0, \sqrt{n^5 - \|\widetilde{q}_j^{t^*}\|_2^2} \right)$ in P_{t^*} are at distance at least $\sqrt{2n^5}$ and thus $\left(\widetilde{p}_i^{t^*}, \sqrt{n^5 - \|\widetilde{p}_i^{t^*}\|_2^2}, 0 \right) \notin \mathcal{O}_{t^*}$.

Finally, note that the total run time was $O(n^{2-\varepsilon} \cdot |V_{b,\ell}|) = O(n^{2-\varepsilon} \log n) < n^{2-\frac{\varepsilon}{2}}$. ◀

[4] Recall that we think of S_i and T_j through their characteristic vector.

4 An $n^{2.5}$ time $1+\epsilon$ Approximation Algorithm for 1-Center in Ulam Metric when $d = n$

In this section, we consider the 1-center problem under Ulam metric. More precisely, we consider a problem where n strings $s_1, s_2, \ldots, s_n$ are given as input and our goal is to find a string s_k such that the maximum distance of s_k from the rest of the strings is minimized. Our focus here is on the Ulam metric.

We assume throughout this section that the length of all strings is equal to d. Our algorithm for this case is two-fold. Let o be the value of the solution (i.e., the maximum distance of the center of the strings to the rest of the strings is exactly equal to o). If o is lower bounded by $\sqrt{d}$, previous work on Ulam distance gives us a $1+\epsilon$ approximate solution for center in the following way: We iterate over all pairs of strings and each time we estimate their Ulam distance via the algorithm of Naumovitz, Saks, and Seshadhri [28] for approximating the Ulam distance of each pair. When the Ulam distance of two strings is equal to u, their algorithm takes time $\widetilde{O}_\epsilon(d/u+\sqrt{d})$ to $1+\epsilon$ approximate the solution. Thus, we only run their algorithm up to a runtime of $\widetilde{O}_\epsilon(\sqrt{d})$ to either obtain a $1+\epsilon$ approximate solution for the Ulam distance or verify that the Ulam distance is smaller than $\sqrt{d}$. It follows that if $o \geq \sqrt{d}$ this information is enough for us to approximate the 1-center problem within a factor $1+\epsilon$ and the runtime of the algorithm is bounded by $\widetilde{O}_\epsilon(n^2\sqrt{d})$. Thus, it only remains to design an algorithm for the low-distance regime.

From here on, we assume that $o \leq \sqrt{d}$. In this case, we take an arbitrary string (say s_1) and compute the Ulam distance of that string to all the other strings. In addition to this, we also keep track of the changes that convert s_1 into all the strings. It follows that since $o \leq \sqrt{d}$, the distance of s_1 to all the strings is bounded by at most $2\sqrt{d}$. Thus, via the transformations we compute in this step, we would be able to make a transformation from any s_i to any s_j with at most $4\sqrt{d}$ operations (we can combine the transformation from s_1 to s_i and the transformation from s_1 to s_j). Using this information, we can determine the exact Ulam distance of every string s_i to every string s_j in the following way:

We start with the non-optimal transformation from s_i to s_j that uses at most $4\sqrt{d}$ operations. We then split the characters from $[1, \ldots, d]$ into buckets such that in each bucket all the characters are next to each other and they appear in the same order in the two strings. To be more precise, consider the following procedure: color each character of s_i and s_j which is touched in the transformation (deleted, added, or changed) in red and the rest blue. Each set of consecutive blue characters and each single red character makes a bucket. It follows that because there is a transformation from s_i to s_j with at most $4\sqrt{d}$ operations, the total number of buckets would be bounded by $O(\sqrt{d})$. Moreover, there exists an optimal transformation wherein either all characters of each buckets are deleted/inserted or all characters of each bucket remain intact. This implies that we can compress the two strings into smaller strings by replacing each bucket with a single character. The insertion and deletion of these special characters then has a cost proportional to the size of the bucket. This way, the size of the two strings would be bounded by $O(\sqrt{d})$ and thus we can compute the Ulam distance of the two strings in time $\widetilde{O}(\sqrt{d})$. Therefore, we can compute the center of the strings in time $\widetilde{O}_\epsilon(nd + n^2\sqrt{d})$.

▶ **Theorem 3.** *There exists a $1+\epsilon$ approximation algorithm for the 1-center under Ulam metric that runs in time $\widetilde{O}_\epsilon(nd + n^2\sqrt{d})$.*

Proof of Theorem 3. The outline of the algorithm along with its runtime analysis is given earlier. Here we prove that the approximation factor of the algorithm is bounded by $1+\epsilon$. In case $o \geq \sqrt{d}$, we use the $1+\epsilon$ approximation algorithm of Naumovitz, Saks, and Seshadhri [28]

Algorithm 1 1-center of n strings under Ulam metric.

Data: $s_1, s_2, \ldots, s_n$
Result: 1-center
$o \leftarrow \infty$;
for $i \leftarrow 1$ *to* n **do**
  $mx \leftarrow -1$;
  for $j \leftarrow 1$ *to* n **do**
    Run [28] on s_i and s_j up to $\widetilde{O}_\epsilon(\sqrt{d})$ steps;
    if *the algorithm terminates* **then**
      $mx \leftarrow \max\{mx, \text{the output of the algorithm}\}$;
    end
  end
  $o \leftarrow \min\{o, mx\}$;
end
if $o \neq -1$ **then**
  return o;
end
else
  $o \leftarrow \infty$;
  for $i \leftarrow 1$ *to* n **do**
    $tr_i \leftarrow$ optimal transformation between s_1 and s_i;
  end
  for $i \leftarrow 1$ *to* n **do**
    $mx \leftarrow -1$;
    for $j \leftarrow 1$ *to* n **do**
      $(s_i^*, s_j^*) \leftarrow$ compressed versions of (s_i, s_j) based on tr_i and tr_j;
      $mx \leftarrow \max\{mx, \text{the Ulam distance of } s_i^* \text{ and } s_j^*\}$;
    end
    $o \leftarrow \min\{o, mx\}$;
  end
  return o;
end

for each pair of strings up to a runtime of $\widetilde{O}_\epsilon(\sqrt{d})$. If the algorithm gives an estimation before we terminate it, we take the value into account when determining the maximum distance for the strings involved. It follows that since $o \geq \sqrt{d}$, then for each string s_x, there is one string s_y whose distance to s_x is at least $\sqrt{d}$ and thus the maximum distance we determine for each string is a $1+\epsilon$ approximation of the optimal value.

Next, we show that in case $o \leq \sqrt{d}$, our algorithm determines the Ulam distance of each pair exactly and thus solves the 1-center problem correctly. In order to determine the Ulam distance between s_i and s_j, we begin with a transformation of size at most $4\sqrt{d}$ between the two strings. We then mark all the characters that are either deleted or inserted in this transformation and all the characters that are next to these characters. We then compress the two strings in the following way: each marked character becomes a single character with the same value. Each maximal interval of unmarked characters that are next to each other also become a single character whose value represents the entire interval. Therefore, the

compressed strings have $O(\sqrt{d})$ characters each. It follows that the Ulam distance of the compressed strings is exactly equal to the Ulam distance of the original strings. Moreover, even though for the compressed strings the operations have arbitrary costs, we can still solve Ulam distance in time proportional to the length of the strings which results in an algorithm with runtime $\widetilde{O}(\sqrt{d})$ for computing Ulam distance between each pair. ◀

5 Hardness of Approximation of 1-center in String and ℓ_p-metrics

In this section, we prove hardness of approximation results for the 1-center problem. A background on error correcting codes is detailed in Appendix A.

▶ **Theorem 22** (Subquadratic Inapproximability of 1-center in ℓ_p-metrics). *Let $p \in \mathbb{R}_{\geq 1} \cup \{0\}$. Assuming* HSC*, for every $\varepsilon > 0$ there exists $\delta > 0$ such that no algorithm running in time $n^{2-\varepsilon}$ can given as input a point-set $P \subseteq \{0,1\}^d$ and a positive real α output all points in P whose 1-center cost in the ℓ_p-metric is at most $\alpha \cdot (1+\delta)$, where $|P| = n$ and $d = O_\varepsilon(\log n)$.*

Proof. Fix $p \in \mathbb{R}_{\geq 1} \cup \{0\}$. We prove the theorem statement by contradiction. Suppose for some $\varepsilon > 0$ there is an algorithm $\mathcal{T}$ running in time $n^{2-\varepsilon}$ that can for every $\delta > 0$, given as input a point-set $P \subseteq \mathbb{R}^d$ and a positive real α output all points in P whose 1-center cost in the ℓ_p-metric is at most $\alpha \cdot (1+\delta)$, where $|P| = n$ and $d = O_\varepsilon(\log n)$ (dependency on ε will become clear later in the proof).

Let $(\mathcal{A} := (S_1, \ldots, S_n), \mathcal{B} := (T_1, \ldots, T_n), U)$ be an instance arising from HSC, where $|U| = c \log n$. We think of each set in $\mathcal{A}$ and $\mathcal{B}$ as its characteristic vector in $\{0,1\}^{c \log n}$. We show how we can decide this instance in $n^{2-\frac{\varepsilon}{2}}$ time using $\mathcal{T}$, thus contradicting HSC.

The construction below is exactly the same as the one suggested by Rubinstein [31]. We however, use this construction to fit our purposes of proving lower bound for the 1-center problem.

Algebrization. Fix $T = 2c/\varepsilon$. Let q be the smallest prime greater than T (i.e., $q < 2 \cdot T$). Let $m := c \log n$. Let $C^1_{m/T}$ and $C^2_{m/T}$ be the codes guaranteed in Theorem 26 over $\mathbb{F}_{q^2}$ with block length $\ell \leq \lambda m/T$.

Let $\widetilde{C} \subseteq C^2_{m/T}$ such that $\omega \in \widetilde{C}$ if and only if $\omega\ |_{[m/T]} = \vec{0}$ (i.e., ω has a zero entry in each of the first m/T coordinates). For every $\omega \in \widetilde{C}$ we define two functions $\tau^\omega_{\mathcal{A}}, \tau^\omega_{\mathcal{B}} : \{0,1\}^m \to \{0,1\}^r$, where $r := q^{4(T+2)} \times \ell$. We can thus index every $i \in [r]$ using elements in $\mathbb{F}^T_{q^2} \times \mathbb{F}^T_{q^2} \times [\ell]$.

Fix $x \in \{0,1\}^m$. Let $x = (x^1, \ldots, x^T) \in \{0,1\}^m$ where for all $i \in [T]$ we have $x^i \in \{0,1\}^{m/T}$. We define $\tau^\omega_{\mathcal{A}}(x)$ coordinate wise. Fix $\zeta \in [r]$ and we think of ζ as follows:

$$\zeta = \big((\mu^{\mathcal{A}}_1, \ldots, \mu^{\mathcal{A}}_{T+2}), (\mu^{\mathcal{B}}_1, \ldots, \mu^{\mathcal{B}}_{T+2}), j\big) \in \mathbb{F}^{T+2}_{q^2} \times \mathbb{F}^{T+2}_{q^2} \times [\ell]. \tag{1}$$

We define $\tau^\omega_{\mathcal{A}}(x)_\zeta$ to be 1 if and only if:

$$\sum_{i \in [T+2]} \mu^{\mathcal{A}}_i \cdot \mu^{\mathcal{B}}_i = \omega(j) \quad \text{and} \quad \forall i \in [T],\ \mu^{\mathcal{A}}_i = C^1_{m/T}(x^i)_j,\ \text{and } \mu^{\mathcal{A}}_{T+1} = 0,\ \mu^{\mathcal{A}}_{T+2} = C^1_{m/T}(\mathbb{1}^{m/T})_j.$$

Similarly, we define $\tau^\omega_{\mathcal{B}}(x)$ coordinate wise. Fix $\zeta \in [r]$ and we think of ζ as in (1). We define $\tau^\omega_{\mathcal{B}}(x)_\zeta$ to be 1 if and only if:

$$\sum_{i \in [T+2]} \mu^{\mathcal{A}}_i \cdot \mu^{\mathcal{B}}_i = \omega(j) \quad \text{and} \quad \forall i \in [T],\ \mu^{\mathcal{B}}_i = C^1_{m/T}(x^i)_j \text{ and } \mu^{\mathcal{B}}_{T+1} = C^1_{m/T}(\mathbb{1}^{m/T})_j,\ \mu^{\mathcal{B}}_{T+2} = 0.$$

Construction. For every $S_i \in \mathcal{A}$, we define $s_i^\omega := \tau_{\mathcal{A}}^\omega(S_i)$. Further, we define $\widetilde{s}_i^\omega = (s_i^\omega, \mathbb{1}^r - s_i^\omega, \mathbb{1}^{2r}) \in \{0,1\}^{4r}$. For every $T_j \in \mathcal{B}$, we define $t_j^\omega := \tau_{\mathcal{B}}^\omega(T_j)$. Further, we define $\widetilde{t}_j^\omega = (\mathbb{1}^r - t_j^\omega, t_j^\omega, 0^{2r}) \in \{0,1\}^{4r}$.

We define the point-set P_ω to be $P_\omega^{\mathcal{A}} := \{\widetilde{s}_i^\omega \mid S_i \in \mathcal{A}\} \cup P_\omega^{\mathcal{B}} := \{\widetilde{t}_j^\omega \mid T_j \in \mathcal{B}\} \cup \{\mathbb{1}^{4r}\}$. Let $\alpha := (2q^{4(T+2)} - 4q^{2(T+1)}) \cdot \ell + 2r + \ell$. Let $\delta := 1/(4q^{4T} - 4q^{2T-2} + 1)$.

Analysis. We run $\mathcal{T}$ on (P_ω, α) for every $\omega \in \widetilde{C}$. Let $\mathcal{O}_\omega \subseteq P_\omega$ be the output of running $\mathcal{T}$ on (P_ω, α). In other words for every $\omega \in \widetilde{C}$ and every $s \in \mathcal{O}_t$ we have that for every $s' \in P_\omega$, $\|s - s'\|_p \leq (1+\delta)^{1/p} \cdot \alpha^{1/p}$.

We claim that there exists S in $\mathcal{A}$ such that it intersects with every subset T in $\mathcal{B}$ if and only if there exists $i \in [n]$ such that for all $\omega \in \widetilde{C}$, we have $\widetilde{s}_i^\omega \in \mathcal{O}_\omega$.

Suppose there exists S_{i^*} in $\mathcal{A}$ such that it intersects with every subset T in $\mathcal{B}$. Fix $\omega \in \widetilde{C}$. We have that $\widetilde{s}_{i^*}^\omega$ is at distance at most $(2r)^{1/p}$ from every other point in $P_\omega^{\mathcal{A}}$ just from construction. Suppose there is $\widetilde{t}_j^\omega \in P_\omega^{\mathcal{B}}$ such that their distance is greater than $\alpha^{1/p}$ then from the construction, their distance must be at least $(1+\delta)^{1/p} \cdot \alpha^{1/p}$, which implies that $S_{i^*} \cap T_j = \emptyset$, a contradiction.

On the other hand, if for every S in $\mathcal{A}$ there exists T in $\mathcal{B}$ such that S and T are disjoint then we show that for any point $\widetilde{s}_i^\omega$ there exists $\omega \in \widetilde{C}$ such that $\widetilde{s}_i^\omega \notin \mathcal{O}_\omega$. Fix $i \in [n]$. Let $T_j \in \mathcal{B}$ such that $S_i \cap T_j = \emptyset$. Let $\omega^* := \sum_{e \in [T]} C^1_{m/T}(S_i^e) \cdot C^1_{m/T}(T_j^e)$. From Theorem 26 we have that $\omega^* \in \widetilde{C}$. Thus $\widetilde{s}_i^{\omega^*}$ and $\widetilde{t}_j^{\omega^*}$ in P_{ω^*} are at distance at least $(1+\delta)^{1/p} \cdot \alpha^{1/p}$ and thus $\widetilde{s}_i^{\omega^*} \notin \mathcal{O}_{\omega^*}$. Also, note that for all $\omega \in \widetilde{C}$, we have that every point in $P_\omega^{\mathcal{B}}$ is at distance at least $(3r)^{1/p}$ from $\mathbb{1}^{4r}$.

Finally, note that the total run time was $O(n^{2-\varepsilon} \cdot |\widetilde{C}|) = O(n^{2-\varepsilon+\frac{c}{T}}) < n^{2-\frac{\varepsilon}{2}}$. ◀

We remark that the above construction is the same as the one in [31] albeit for a different problem (nearest neighbors problem).

Theorem 22 readily extend to the edit metric from the below statement and to the Ulam metric from Lemma 11.

▶ **Lemma 23** (Rubinstein [31]). *For large enough $d \in \mathbb{N}$, there is a function $\eta : \{0,1\}^d \to \{0,1\}^{d'}$, where $d' = O(d \log d)$, such that for all $a, b \in \{0,1\}^d$ the following holds for some constant $\lambda > 0$:*

$$|\mathsf{ed}(\eta(a), \eta(b)) - \lambda \cdot \log d \cdot \|a - b\|_0| = o(d').$$

Moreover, for any $a \in \{0,1\}^d$, $\eta(a)$ can be computed in $2^{o(d)}$ time.

References

1 Amir Abboud, Karl Bringmann, Danny Hermelin, and Dvir Shabtay. Scheduling lower bounds via AND subset sum. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 4:1–4:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.4.

2 Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. Distributed PCP theorems for hardness of approximation in P. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 25–36, 2017. doi:10.1109/FOCS.2017.12.

3 Amir Abboud, Virginia Vassilevska Williams, and Joshua R. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 377–391. SIAM, 2016. doi:10.1137/1.9781611974331.ch28.

4 Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 51–58. ACM, 2015. doi:10.1145/2746539.2746612.

5 Amir Ben-Dor, Giuseppe Lancia, Jennifer Perone, and R. Ravi. Banishing bias from consensus sequences. In Alberto Apostolico and Jotun Hein, editors, *Combinatorial Pattern Matching, 8th Annual Symposium, CPM 97, Aarhus, Denmark, June 30 - July 2, 1997, Proceedings*, volume 1264 of *Lecture Notes in Computer Science*, pages 247–261. Springer, 1997. doi:10.1007/3-540-63220-4_63.

6 Karl Bringmann and Bhaskar Ray Chaudhury. Polyline simplification has cubic complexity. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA*, volume 129 of *LIPIcs*, pages 18:1–18:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.SoCG.2019.18.

7 Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 79–97. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.15.

8 Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 261–270. ACM, 2016. doi:10.1145/2840728.2840746.

9 Diptarka Chakraborty, Debarati Das, and Robert Krauthgamer. Approximating the median under the ulam metric. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 761–775. SIAM, 2021. doi:10.1137/1.9781611976465.48.

10 Lijie Chen. On the hardness of approximate and exact (bichromatic) maximum inner product. *Theory Comput.*, 16:1–50, 2020. doi:10.4086/toc.2020.v016a004.

11 Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff. Sublinear optimization for machine learning. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 449–457. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.50.

12 Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 9–21. ACM, 2016. doi:10.1145/2897518.2897647.

13 Vincent Cohen-Addad, Karthik C. S., and Euiwoong Lee. On approximability of clustering problems without candidate centers. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2635–2648. SIAM, 2021. doi:10.1137/1.9781611976465.156.

14 Roee David, Karthik C. S., and Bundit Laekhanukit. On the complexity of closest pair via polar–pair of point–sets. *SIAM J. Discrete Math.*, 33(1):509–527, 2019. doi:10.1137/17M1128733.

15 Colin de la Higuera and Francisco Casacuberta. Topology of strings: Median string is np-complete. *Theoretical Computer Science*, 230(1):39–48, 2000. doi:10.1016/S0304-3975(97)00240-5.

16 Moti Frances and Ami Litman. On covering problems of codes. *Theory Comput. Syst.*, 30(2):113–119, 1997. doi:10.1007/s002240000044.

17 Arnaldo Garcia and Henning Stichtenoth. On the asymptotic behaviour of some towers of function fields over finite fields. *Journal of Number Theory*, 61(2):248–273, 1996. doi:10.1006/jnth.1996.0147.

18 M. R. Garey, Ronald L. Graham, and David S. Johnson. Some np-complete geometric problems. In Ashok K. Chandra, Detlef Wotschke, Emily P. Friedman, and Michael A. Harrison, editors, *Proceedings of the 8th Annual ACM Symposium on Theory of Computing, May 3-5, 1976, Hershey, Pennsylvania, USA*, pages 10–22. ACM, 1976. doi:10.1145/800113.803626.

19 Leszek Gasieniec, Jesper Jansson, and Andrzej Lingas. Efficient approximation algorithms for the hamming center problem. In Robert Endre Tarjan and Tandy J. Warnow, editors, *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, 17-19 January 1999, Baltimore, Maryland, USA*, pages 905–906. ACM/SIAM, 1999. URL: http://dl.acm.org/citation.cfm?id=314500.315081.

20 Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via direct sum in communication coplexity. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference, Chicago, Illinois, USA, June 30 - July 3, 1991*, pages 299–304. IEEE Computer Society, 1991. doi:10.1109/SCT.1991.160273.

21 Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *J. ACM*, 66(5):33:1–33:38, 2019. doi:10.1145/3325116.

22 Marcos Kiwi and José Soto. On a speculated relation between chvátal–sankoff constants of several sequences. *Combinatorics, Probability and Computing*, 18(4):517–532, 2009.

23 J. Kevin Lanctôt, Ming Li, Bin Ma, Shaojiu Wang, and Louxin Zhang. Distinguishing string selection problems. *Inf. Comput.*, 185(1):41–55, 2003. doi:10.1016/S0890-5401(03)00057-9.

24 Ming Li, Bin Ma, and Lusheng Wang. Finding similar regions in many sequences. *J. Comput. Syst. Sci.*, 65(1):73–96, 2002. doi:10.1006/jcss.2002.1823.

25 Ming Li, Bin Ma, and Lusheng Wang. On the closest string and substring problems. *J. ACM*, 49(2):157–171, 2002. doi:10.1145/506147.506150.

26 Michael Mitzenmacher and Saeed Seddighin. Dynamic algorithms for LIS and distance to monotonicity. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 671–684. ACM, 2020. doi:10.1145/3357713.3384240.

27 Ashley Montanaro. Metric Embeddings. http://people.maths.bris.ac.uk/~csxam/presentations/embeddings.pdf, 2008. [Online; accessed 12-December-2008].

28 Timothy Naumovitz, Michael Saks, and C Seshadhri. Accurate and nearly optimal sublinear approximations to ulam distance. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2012–2031. SIAM, 2017.

29 François Nicolas and Eric Rivals. Complexities of the centre and median string problems. In *Proceedings of the 14th Annual Conference on Combinatorial Pattern Matching*, CPM'03, pages 315–327, Berlin, Heidelberg, 2003. Springer-Verlag.

30 François Nicolas and Eric Rivals. Hardness results for the center and median string problems under the weighted and unweighted edit distances. *Journal of discrete algorithms*, 3(2-4):390–415, 2005.

31 Aviad Rubinstein. Hardness of approximate nearest neighbor search. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1260–1268. ACM, 2018. doi:10.1145/3188745.3188916.

32 Kenneth W. Shum, Ilia Aleshnikov, P. Vijay Kumar, Henning Stichtenoth, and Vinay Deolalikar. A low-complexity algorithm for the construction of algebraic-geometric codes better than the Gilbert-Varshamov bound. *IEEE Trans. Information Theory*, 47(6):2225–2241, 2001. doi:10.1109/18.945244.

33 Jeong Seop Sim and Kunsoo Park. The consensus string problem for a metric is np-complete. *J. Discrete Algorithms*, 1(1):111–117, 2003. doi:10.1016/S1570-8667(03)00011-X.

34 Nikola Stojanovic, Piotr Berman, Deborah Gumucio, Ross C. Hardison, and Webb Miller. A linear-time algorithm for the 1-mismatch problem. In Frank K. H. A. Dehne, Andrew Rau-Chaplin, Jörg-Rüdiger Sack, and Roberto Tamassia, editors, *Algorithms and Data Structures, 5th International Workshop, WADS '97, Halifax, Nova Scotia, Canada, August 6-8, 1997, Proceedings*, volume 1272 of *Lecture Notes in Computer Science*, pages 126–135. Springer, 1997. doi:10.1007/3-540-63307-3_53.

35 Ryan Williams. On the difference between closest, furthest, and orthogonal pairs: Nearly-linear vs barely-subquadratic complexity. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1207–1215. SIAM, 2018. doi:10.1137/1.9781611975031.78.

36 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3447–3487. World Scientific, 2018.

A Error Correcting Codes

An error correcting code C over alphabet Σ is a function $C : \Sigma^m \to \Sigma^\ell$ where m and ℓ are positive integers which are referred to as the *message length* and *block length* of C respectively. Intuitively, the function C encodes an original message of length m to an encoded message of length ℓ. The *rate* of a code $\rho(C)$ is defined as the ratio between its message length and its block length, i.e., $\rho(C) = m/\ell$. The *relative distance* of a code, denoted by $\delta(C)$, is defined as $\min_{x \neq y \in \Sigma^m} \delta(C(x), C(y))$ where $\delta(C(x), C(y))$ is the *relative Hamming distance* between $C(x)$ and $C(y)$, i.e., the fraction of coordinates on which $C(x)$ and $C(y)$ disagree.

In this paper, we require our codes to have some special algebraic properties which have been shown to be present in algebraic geometric codes [17]. First, we will introduce a couple of additional definitions.

▶ **Definition 24** (Systematicity). *Given $s \in \mathbb{N}$, a code $C : \Sigma^m \to \Sigma^\ell$ is* s-systematic *if there exists a size-s subset of $[\ell]$, which for convenience we identify with $[s]$, such that for every $x \in \Sigma^s$ there exists $w \in \Sigma^m$ in which $x = C(w)\,|_{[s]}$.*

▶ **Definition 25** (Degree-t Closure). *Let Σ be a finite field. Given two codes $C : \Sigma^m \to \Sigma^\ell, C' : \Sigma^{m'} \to \Sigma^\ell$ and positive integer t, we say that C' is a degree-t closure of C if, for every $w_1, \ldots, w_r \in \Sigma^m$ and $P \in \mathbb{F}[X_1, \ldots, X_r]$ of total degree at most t, it holds that $\omega := P(C(w_1), \ldots, C(w_r))$ is in the range of C', where $\omega \in \Sigma^\ell$ is defined coordinate-wise by the equation $\omega_i := P(C(w_1)_i, \ldots, C(w_r)_i)$.*

Below we provide a self-contained statement of the result we need; it follows from Theorem 7 of [32], which gives an efficient construction of the algebraic geometric codes based on [17]'s explicit towers of function fields.

▶ **Theorem 26** ([17, 32]). *There is a constant $\lambda > 0$ such that for any prime $q \geq 7$, there are two code families $\mathcal{C}^1 = \{C_n^1\}_{n\in\mathbb{N}}$, $\mathcal{C}^2 = \{C_n^2\}_{n\in\mathbb{N}}$ such that the following holds for all $n \in \mathbb{N}$,*

- *C_n^1 and C_n^2 are n-systematic code with alphabet $\mathbb{F}_{q^2}$,*
- *C_n^1 and C_n^2 have block length less than λn.*
- *C_n^2 has relative distance $\geq 1/2$,*
- *C_n^2 is a degree-2 closure of C_n^1, and,*
- *Any codeword in C_n^1 or C_n^2 can be computed in poly(n) time.*

We point the interested reader to [21] for a proof sketch of the above theorem.