

Clustering Permutations Under the Ulam Metric: A Parameterized Complexity Study

Tian Bai 

University of Bergen, Norway

Fedor V. Fomin 

University of Bergen, Norway

Petr A. Golovach 

University of Bergen, Norway

Yash Hiren More 

University of Bergen, Norway

Simon Wietheger 

TU Wien, Austria

Abstract

Rank aggregation seeks a representative permutation for a collection of rankings and plays a central role in areas such as social choice, information retrieval, and computational biology. Two fundamental aggregation tasks are the *center* and *median* problems, which minimize the maximum and the total distance to the input permutations, respectively. While these problems are well understood under Kendall's tau and related distances, their parameterized complexity under the Ulam metric, an edit-distance-based metric on permutations, has remained largely unexplored.

In this work, we initiate a systematic study of the parameterized complexity of rank aggregation under the Ulam metric. We consider both the center and median problems, as well as their generalizations to the k -center and k -median clustering settings, parameterized by the number of centers k and the distance budget d (corresponding to the maximum distance for center variants and the total distance for median variants). Both problems are known to be NP-hard already for $k = 1$.

We show that the Ulam k -center problem remains NP-hard when $d = 1$, but is fixed-parameter tractable when parameterized by $k + d$. Our algorithm is based on a novel local-search framework tailored to the non-local nature of Ulam distances. We complement this by proving that no polynomial kernel exists for the $k + d$ parameterization unless $\text{NP} \subseteq \text{coNP}/\text{poly}$. For the Ulam k -median problem parameterized by the total distance d , we establish W[1]-hardness and provide an XP algorithm. We also provide a polynomial kernel for the parameter $k + d$, which in turn yields a fixed-parameter tractable algorithm.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Parameterized complexity and exact algorithms; Mathematics of computing $\rightarrow$ Permutations and combinations

Keywords and phrases parameterized complexity, Ulam distance, rank aggregation, clustering

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.19

Category Track A: Algorithms, Complexity and Games

Related Version Full Version: <http://arxiv.org/abs/2604.25734>

Funding Tian Bai: Funded by the European Union, GA#101126560; Bergen research and training program for future AI leaders across the disciplines, LEAD AI and by the Trond Mohn forskningsstiftelse (grant no. TMS2023TMT01).

Fedor V. Fomin: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (NewPC grant agreement No. 101199930).

Petr A. Golovach: Supported by the Research Council of Norway under the BWCA (grant no. 314528) and Extreme-Algorithms (grant no 355137) projects.



© Tian Bai, Fedor V. Fomin, Petr A. Golovach, Yash Hiren More, and Simon Wietheger; licensed under Creative Commons License CC-BY 4.0

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).

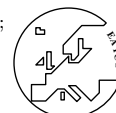
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;

Article No. 19; pp. 19:1–19:23



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Funded by
the European Union

Yash Hiren More: Supported by the Trond Mohn forskningsstiftelse (grant no. TMS2023TMT01).
Simon Wietheger: Supported by the Austrian Science Foundation (FWF, project 10.55776/Y1329).

1 Introduction

Rank aggregation is the problem of computing a representative ranking from a collection of permutations representing the preferences of individual voters. The roots for a mathematical investigation of this problem can be traced back to early studies of social choice theory and voting systems like Borda (1781) and Condorcet (1785), as well as later seminal works such as those by Arrow [4] or Diaconis and Graham [20]. Rank aggregation constitutes a fundamental task with broad applications in social choice theory, information retrieval, computational biology, and database systems [1, 4, 21, 30, 19]. Algorithmically, two canonical rank aggregation tasks are to compute the center and the median of permutations. Given a distance function $\text{dist}(\cdot, \cdot)$ on the space $\mathcal{S}_n$ of permutations over n symbols, the *center* of a permutation set $\Pi \subseteq \mathcal{S}_n$ is a permutation $\sigma \in \mathcal{S}_n$ minimizing

$$\max_{\pi \in \Pi} \text{dist}(\pi, \sigma),$$

whereas the *median* of Π is a permutation $\sigma \in \mathcal{S}_n$ minimizing

$$\sum_{\pi \in \Pi} \text{dist}(\pi, \sigma).$$

Among distance measures for rank aggregation, Kendall’s tau and the Ulam metric are the two most natural and fundamental metrics, extensively studied in both theory and applications [14]. Kendall’s tau distance counts the number of pairwise disagreements between two permutations, thereby capturing local inconsistencies in ranking order. In contrast, the Ulam metric is an edit distance that measures the minimum number of element move operations (deletion and insertion) required to transform one permutation into another. This metric captures global structural differences by isolating the longest common subsequence (the elements that are not moved) and quantifying how many elements must be relocated to achieve alignment.

The tasks of computing centers and medians under Kendall’s tau metric are well understood from different computational perspectives [2, 10, 21, 22, 27, 30, 33]. Finding the optimal center or median under Kendall’s tau is NP-hard already if there are just four input permutations [10, 21]. Additionally, the median under Kendall’s tau, also known as *Kemeny rankings*, admits a polynomial-time approximation scheme (PTAS) [27]. Both versions of rank aggregation under Kendall’s tau admit fixed-parameter tractable (FPT) algorithms parameterized by the solution cost [5, 7], and other natural parameters [8, 9, 16, 17, 23].

In contrast, the study of rank aggregation under the Ulam metric is relatively unexplored, especially from the perspective of parameterized complexity. A key reason is that Ulam distance is a global measure whose combinatorial structure is inherently harder to exploit algorithmically than the local nature of Kendall’s tau. Consequently, some algorithmic work has focused on estimating the distance itself. Andoni and Nguyen [3] and Naumovitz *et al.* [29] developed sublinear-time approximation schemes to estimate the Ulam distance between two permutations, using connections to calculate the longest common subsequence.

Finding an optimal center under the Ulam metric was shown to be NP-hard over a decade ago [5], while the median problem was only recently proven NP-hard [24]. A folklore 2-approximation exists for both the median and center variants under the Ulam metric. Recently, Chakraborty *et al.* [15] obtained a $(1.5 - \varepsilon)$ -approximation (with any $\varepsilon > 0$) for

the center problem with running time exponential in the number of input permutations m . At a similar time, the first approximation algorithm to break the barrier of 2 for the median problem appeared [12]. The subsequent work [13] extends the approximation framework to the setting of finding k medians, achieving a similar approximation guarantee in time $(k \log(mn))^{\mathcal{O}(k)} mn^3$, where n is the size of the alphabet and m is the number of input permutations. Very recently, Jaiswal *et al.* [26] improved on this work and provided a randomized algorithm running in $\tilde{O}((2k)^k nd)$ time. All three works on the median problem provide guarantees close to but below an approximation factor of 2.

To the best of our knowledge, no prior work has considered Ulam-based aggregation through parameterized exact algorithms, despite the metric's intimate connection to edit distance and longest common subsequence – domains where parameterized techniques have proven highly effective. The clustering problems under the Ulam metric seem more challenging, as the Ulam metric is a more complex and less local measure than Kendall's tau, which may suggest that the computational landscape of Ulam-based aggregation could be significantly different. One may notice that the Ulam distance is a smaller parameter than Kendall's tau, as every permutation can be transformed into the other by only moving the elements that are in a different order in the two permutations. It may happen that two permutations with Ulam distance 1 could differ in the relative order of an unbounded number of pairs of elements. This highlights a fundamental structural disparity between the two distance measures, which leads to the following natural question:

What is the parameterized complexity of rank aggregation under the Ulam metric?

1.1 Clustering Under the Ulam Metric: Definitions and Main Results

In this work, we initiate a systematic study of the parameterized complexity of rank aggregation under the Ulam metric. Beyond the classical parameter d , which captures the solution cost (the maximum distance or the sum of distances, respectively), we consider substantially more general variants of the problem: k -center and k -median clustering under the Ulam metric [13]. These extensions, denoted by ULAM METRIC k -CENTER and ULAM METRIC k -MEDIAN, are both natural and well established in the rank aggregation literature. In these problems, one seeks k representative permutations that minimize, respectively, the maximum distance to any assigned permutation (k -center) or the total sum of distances (k -median).

Formally, let $\mathcal{S}_n(\Sigma)$ be the set of all permutations over an alphabet Σ of size n , and let $\text{dist}_U(\pi, \sigma)$ be the Ulam distance between π and σ in $\mathcal{S}_n(\Sigma)$ (see Section 2 for the formal definitions). We define our problems of interest as follows.

ULAM METRIC k -CENTER

Input: An alphabet Σ of size n , a set of permutations $\Pi = \{\pi_1, \pi_2, \dots, \pi_m\} \subseteq \mathcal{S}_n(\Sigma)$, and integers k and d .

Task: Decide whether there exists a set $S = \{\sigma_1, \sigma_2, \dots, \sigma_k\} \subseteq \mathcal{S}_n(\Sigma)$ of k centers such that

$$\max_{\pi \in \Pi} \min_{\sigma \in S} \text{dist}_U(\pi, \sigma) \leq d.$$

ULAM METRIC k -MEDIAN

Input: An alphabet Σ of size n , a multiset of permutations $\Pi = \{\pi_1, \pi_2, \dots, \pi_m\} \subseteq \mathcal{S}_n(\Sigma)$, and integers k and d .

Task: Decide whether there exists a set $S = \{\sigma_1, \sigma_2, \dots, \sigma_k\} \subseteq \mathcal{S}_n(\Sigma)$ of k medians such that

$$\sum_{\pi \in \Pi} \min_{\sigma \in S} \text{dist}_U(\pi, \sigma) \leq d.$$

We remark that, in both clustering problems, the centers or medians are not required to belong to the input set of permutations Π . Hence, we consider variants of the problems that are referred to as *continuous* in some literature. Both problems are known to be NP-hard already for $k = 1$: hardness for the center variant was shown by Bachmaier *et al.* [5] and for the median by Fischer *et al.* [24].

We summarize our main results in Table 1. Most notably, we establish fixed-parameter tractability for both problems when parameterized simultaneously by k and d . Prior to our work, the parameterized complexity of these two clustering problems with respect to d was not even settled for the much simpler case $k = 1$.

We contrast this tractability result by establishing hardness for parameterization by d alone: ULAM METRIC k -CENTER is already NP-hard on instances with $d = 1$. In contrast, ULAM METRIC k -MEDIAN is in XP parameterized by d (that is, solvable in polynomial time for each fixed d), but is W[1]-hard for parameter d , ruling out fixed-parameter tractability under well-established complexity assumptions. Finally, we discover yet another difference in the tractability landscapes of ULAM METRIC k -CENTER and ULAM METRIC k -MEDIAN. While the latter admits a polynomial-sized kernel, we rule out the existence of such a kernel for ULAM METRIC k -CENTER under established complexity assumptions. Our results for the Ulam metric complement the mature theory developed for Kendall's tau. We also hope that they provide rigorous foundations for consensus ranking in applications where global structural coherence under the Ulam metric is essential.

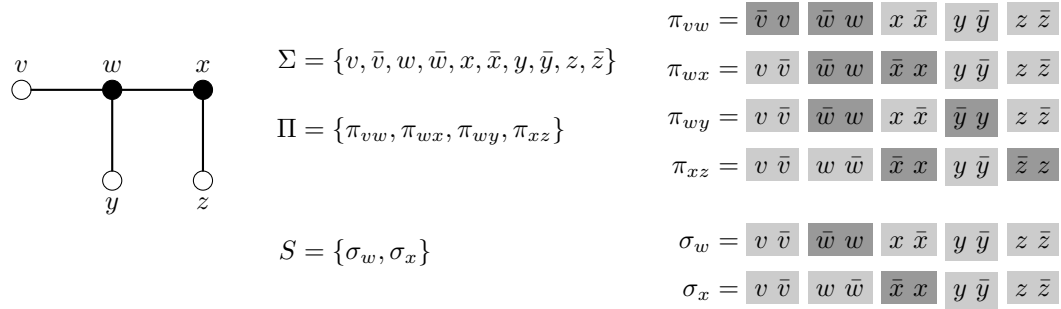
In the following, we provide a detailed overview of our results and the techniques employed. Formal proofs are deferred to Sections 3 and 4. Some details and proofs marked by $(\star)$ are omitted due to space constraints.

■ **Table 1** Summary of our results.

Parameters	k -center	k -median
k	NP-hard even if $k = 1$ [5]	NP-hard even if $k = 1$ [24]
d	NP-hard even if $d = 1$ (Thm. 1)	W[1]-hard, XP (Thms. 4 and 5)
$k + d$	FPT (Thm. 2)	FPT (Cor. 7)
	no polynomial kernel (Thm. 3)	polynomial kernel (Thm. 6)

1.2 Contributions and Techniques for Ulam Metric k -Center

The first group of results in our work concerns ULAM METRIC k -CENTER. As previous work already established that it is unlikely to obtain polynomial time algorithms even for instances aiming for $k = 1$ center, it is natural to ask for the tractability on instances where we instead restrict the radius of the centers to $d = 1$. We answer this question negatively by establishing the NP-hardness of ULAM METRIC k -CENTER for d is any positive constant at least 1. This hardness result is established via a reduction from the VERTEX COVER problem on triangle-free graphs.



■ **Figure 1** Exemplary vertex cover instance with solution $\{w, x\}$ (black vertices) and the constructed instance of ULAM METRIC k -CENTER with solution $S = \{\sigma_w, \sigma_x\}$.

► **Theorem 1** (*). *ULAM METRIC k -CENTER is NP-hard even if d is fixed to any positive constant at least 1.*

The construction for $d = 1$ is relatively straightforward but closes an important gap: it establishes that parameterization by d alone cannot yield fixed-parameter tractability under standard complexity assumptions. The main idea of the reduction is to construct an alphabet consisting of two symbols $v, \bar{v}$ per vertex v in the vertex cover instance. The input set Π contains one permutation per edge, all sharing a common base ordering except that the symbol pairs of the two endpoints of the edge are swapped in the respective permutation. A vertex cover then corresponds to a set of center permutations, each encoding a vertex by swapping the respective symbol pair in the base ordering. Then each permutation encoding an edge has Ulam distance 1 to only those permutations encoding vertices to which the edge is incident, which ensures an equivalence between the vertex cover instance and the ULAM METRIC k -CENTER instance. We defer a formal proof to Section 3, but provide an exemplary reduction in Figure 1.

A nontrivial challenge is that a single permutation – swapping symbol pairs of multiple vertices – might cover several edges without corresponding to any single vertex in a cover. We circumvent this problem by basing our reduction on triangle-free instances of VERTEX COVER, so this case will never occur. Furthermore, a direct reduction from the $d = 1$ case to general d is non-trivial; we overcome this by augmenting the $d = 1$ construction with additional corresponding symbols for each vertex.

The NP-hardness of ULAM METRIC k -CENTER even if $d = 1$ or if $k = 1$ naturally raises the question of the parameterized complexity of the problem with the combined parameter $k + d$. The following theorem establishes fixed-parameter tractability with respect to this combined parameter.

► **Theorem 2.** *ULAM METRIC k -CENTER can be solved in $2^{\mathcal{O}(d^2k(d+k))} \cdot mn^{1+o(1)}$ time, where $m := |\Pi|$ and $n := |\Sigma|$.*

On first glance, it may seem that the finite number of permutations in our space would allow for an approach based on some simple branching and *local search* around some permutation $\pi \in \Pi$. This approach is infeasible, however, as for a permutation π of length $n \gg d$, there are $n^{\Omega(d)}$ permutations within Ulam distance d from π , each of which is a potential candidate for the center. This lower bound follows from a simple counting argument. One can choose d symbols from the first half of π in $\binom{n/2}{d}$ ways and insert them into the second half in at least $(n/2)^d$ distinct ways, yielding a unique permutation at Ulam distance at most d from π .

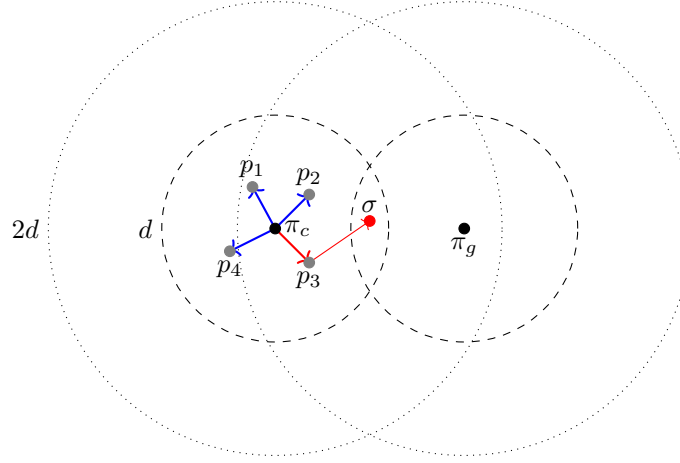
A natural alternative is to adapt techniques from string-based clustering, which typically perform bounded local search around a candidate solution and iteratively improve it by resolving “dissimilarities” with inputs that exceed the target distance d . Such a paradigm is well-known and has been highly successful for problems such as CLOSEST STRING [25] where the “dissimilarities” correspond to bits contributing to the Hamming distance. However, two permutations can admit $\Omega((n/d)^d)$ distinct LCSs of the same maximum length, making it impossible to pinpoint a canonical set of elements to guide local improvements. Thus, such techniques from string-based clustering cannot directly apply to the Ulam metric, and more involved methods are required. Our algorithm embarks on a similar approach as the one by Gramm *et al.* [25]. Rather sooner than later, however, the issue of not having a unique witness LCS requires much more involved techniques and branching procedures tailored to the Ulam metric to establish fixed-parameter tractability.

In our algorithm, we construct a family of *candidate permutations* that are at bounded Ulam distances from hypothetical centers. These permutations will eventually converge towards the center permutations. If $k = 1$, then such a candidate can be initially chosen as an arbitrary permutation from Π ; note that it should be at the Ulam distance at most d from a center in any YES-instance of the problem. For $k \geq 2$, we initiate the set of candidates using the idea behind the standard 2-approximation for the k -CENTER problem. On each step, we either conclude that the current set of candidates can serve as centers, that is, each permutation in Π is at distance at most d from one of the candidates, or find a *guide permutation* π_g at distance at least $d + 1$ from all the candidates. Then we can guess a candidate permutation π_c that should be transformed to a center for the cluster containing the guide. Our crucial result (Lemma 16) shows that, given π_c and π_g , we can efficiently (in FPT time) enumerate a set of permutations with distance one to π_c such that for any hypothetical center, there is a permutation in the set that is closer to the center than π_c was, see Figure 2 for a visualization of this idea. The proof of Lemma 16 is rather sophisticated and combines the *random separation* [11] and branching techniques. The random separation is used to highlight elements of π_c and π_g that should be moved to make these permutations closer to the center and the neighbors of these elements that are not moved. Then we identify an element of π_c that should be moved and, after that, find an appropriate place for this element using branching. Repeatedly invoking Lemma 16, we ensure that the candidate permutations converge to the centers.

The fixed-parameter tractability of ULAM METRIC k -CENTER with parameter $k + d$ established in Theorem 2 immediately leads to the question about the existence of a polynomial kernel for this problem. The following theorem provides a negative answer to this question.

► **Theorem 3 (★).** *ULAM METRIC k -CENTER does not admit a polynomial-sized kernel for the parameter $k + d$ unless $\text{NP} \subseteq \text{coNP/poly}$.*

The lower bound of Theorem 3 holds already for $k = 1$. The proof is based on a Polynomial Parameterized Transformation (PPT) reduction from the classical CLOSEST STRING problem to ULAM METRIC k -CENTER. To build such a transformation, we employ a *distance-preserving* transformation between binary strings and permutations, where for every pair of binary strings, their Hamming distance is the same as the Ulam distance between the permutations to which they translate. The reduction establishes that from the perspective of parameterized complexity with respect to the radius d , ULAM METRIC k -CENTER is at least as hard as CLOSEST STRING. Theorem 3 then follows by pipelining the PPT with established lower bounds on the kernelization of CLOSEST STRING by Basavaraju *et al.* [6].



■ **Figure 2** Illustration of the key idea behind our FPT algorithm for the k -center problem under Ulam distance. π_c denotes a candidate center, π_g a guide permutation with distance between d and $2d$ from π_c , and σ represents the true (optimal) center that covers both π_c and π_g . We enumerate in FPT time a set of permutations (shown as $\{p_1, p_2, p_3, p_4\}$) at distance 1 from π_c . Among these candidates, at least one candidate (shown as p_3) reduces the Ulam distance to σ and thus corresponds to a correct branch.

1.3 Contributions and Techniques for Ulam Metric k -Median

The second group of results is concerned with ULAM METRIC k -MEDIAN. Recall that the problem is NP-hard for $k = 1$ [5]. While ULAM METRIC k -CENTER is also NP-hard for $d = 1$, this is not the case for ULAM METRIC k -MEDIAN. It is easy to obtain a polynomial-time algorithm for this problem for each fixed value of d .

► **Theorem 4** ($\star$). *ULAM METRIC k -MEDIAN can be solved in $\mathcal{O}((mn)^{2d})$ time, where $m := |\Pi|$ and $n := |\Sigma|$.*

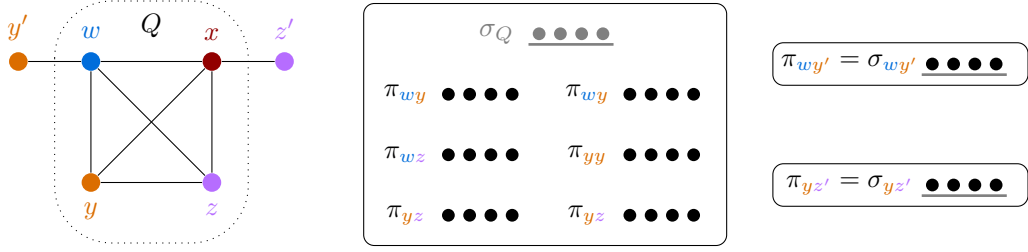
The algorithm builds on a simple branching technique, where we observe that the medians can be obtained by applying a total of d move-operations on the input set, and we branch on these d moves by brute-force. The result in Theorem 4 is tight in the sense that the existence of an FPT-algorithm for the parameterization by d is unlikely due to the following lower bound.

► **Theorem 5.** *ULAM METRIC k -MEDIAN is $W[1]$ -hard parameterized by d .*

Our proof of Theorem 5 relies on a careful reduction from MULTICOLORED CLIQUE. The instance is designed in a way that all but one of the medians in any hypothetical solution are already present in the input set Π . The challenge is to find that additional median in a way that it is sufficiently close to a number of input permutations. Crucially, the permutations are designed in such a way that this is only possible if all edges covered by the median form a clique, see Figure 3 for a schematic example. On a high level, we construct the instance such that there is one special symbol per color, and the position of that symbol in the additional median encodes which vertex of that color is selected for the multi-colored clique.

Interestingly, our proof indicates that the hardness of the problem does not so much depend on identifying which symbols need to be moved but also importantly on deciding where to move these symbols.

In contrast to the lower bound on the polynomial kernel for ULAM METRIC k -CENTER obtained in Theorem 3, ULAM METRIC k -MEDIAN admits a polynomial kernel.



■ **Figure 3** Illustration of the reduction from MULTICOLORED CLIQUE to ULAM METRIC k -MEDIAN. Left shows an instance of MULTICOLORED CLIQUE with four color classes ($\{w\}$, $\{x\}$, $\{y, y'\}$, and $\{z, z'\}$) and a colorful clique $Q = \{w, x, y, z\}$. Right shows the schematic solution to the constructed ULAM METRIC k -MEDIAN instance. Each edge is encoded as an input permutation; edges whose endpoints form a multicolored clique are covered by a single external median σ_Q (underlined), while all other edges each serve as their own median in singleton clusters (also underlined).

► **Theorem 6** ($\star$). *ULAM METRIC k -MEDIAN admits a kernel with $\mathcal{O}(d^2 + dk)$ permutations over an alphabet of size $\mathcal{O}(d^4 + d^2k^2)$, computable in $\mathcal{O}(m^2n \log n)$ time, where $m := |\Pi|$ and $n := |\Sigma|$.*

Our kernelization result is based on the observation that for any YES-instance of ULAM METRIC k -MEDIAN, Π contains at most $k + d$ distinct permutations. This allows us to reduce the size of Π . Then we reduce the size of the alphabet Σ by making replacements of identical subpermutations by shorter ones. Here, we use the fact that two permutations at Ulam distance at least $d + 1$ cannot belong to the same median in any solution. Thus, we can identify some sets of permutations such that every set of permutations that shares a median in a hypothetical solution is a subset to one of the sets. We show that the pairwise Ulam distance within each set are small. Combined with the fact that there are only few distinct permutations, this implies that in each set there are many (or long) identical subpermutations, which we can then shorten to reduce the size of the alphabet Σ .

Combining Theorems 4 and 6, we immediately obtain an FPT-algorithm with the following running time.

► **Corollary 7.** *ULAM METRIC k -MEDIAN can be solved in $(kd)^{\mathcal{O}(d)} + \mathcal{O}(m^2n \log n)$ time, where $m := |\Pi|$ and $n := |\Sigma|$.*

2 Preliminaries

Basic notation. We use $[n]$ to denote the set $\{1, 2, \dots, n\}$. Let Σ be a finite alphabet with $|\Sigma| = n$. A *permutation* of length n over Σ is a bijection $\pi := [n] \rightarrow \Sigma$. We write $\mathcal{S}_n(\Sigma)$ for the set of all permutations of length n over Σ . Thus, each $\pi \in \mathcal{S}_n(\Sigma)$ is an ordering of the symbols of Σ . We often write a permutation as a sequence $\pi = \pi(1)\pi(2) \cdots \pi(n)$.

A *subsequence* of a permutation π is obtained by deleting zero or more elements from π while maintaining the relative order of the remaining elements. A *common subsequence* of two permutations π_1 and π_2 is a subsequence that appears in both permutations. We denote by $\text{LCS}(\pi_1, \pi_2)$ the length of the longest common subsequence of π_1 and π_2 .

Ulam distance. A *move operation* on a permutation π removes one element from its current position and inserts it at another position, while maintaining the relative order of all other elements.

For two permutations $\pi, \sigma \in \mathcal{S}_n(\Sigma)$, the *Ulam distance* $\text{dist}_U(\pi, \sigma)$ is the minimum number of move operations needed to transform π into σ :

$$\text{dist}_U(\pi, \sigma) = \min\{d : \pi \text{ can be turned into } \sigma \text{ using } d \text{ moves}\}.$$

A classical identity characterizes this metric in terms of the longest common subsequence:

$$\text{dist}_U(\pi, \sigma) = n - \text{LCS}(\pi, \sigma).$$

Thus, the Ulam distance measures the minimum number of elements that must be relocated to make the permutations consistent in relative order. We remark that a longest common subsequence, and thereby the Ulam distance, between two permutations of length n can be computed in time in $O(n \log n)$ by relabeling the alphabet such that the first permutation has increasing order and then finding a longest increasing sequence in the other (relabelled) permutation. The Ulam distance on permutations is analogous to the Levenshtein edit distance on strings in the following sense: both distances identify a longest common subsequence and keep it untouched while performing edit operations on the remaining symbols.

For a set $\mathcal{S} \subseteq \mathcal{S}_n(\Sigma)$ of permutations and $\pi \in \mathcal{S}_n(\Sigma)$, we write $\text{dist}_U(\pi, \mathcal{S})$ to denote the minimum distance from π to any permutation in $\mathcal{S}$, that is, $\text{dist}_U(\pi, \mathcal{S}) = \min_{\sigma \in \mathcal{S}} \text{dist}_U(\pi, \sigma)$.

For two strings s, t of equal length over an alphabet, the *Hamming distance* $\text{dist}_H(s, t)$ counts the number of positions where the corresponding symbols differ.

Graphs. We work with simple undirected graphs. For a graph G , we denote by $V(G)$ its vertex set and by $E(G)$ its edge set. A *vertex cover* of a graph G is a subset $X \subseteq V(G)$ such that every edge in $E(G)$ has at least one endpoint in X . A *clique* in a graph G is a subset of vertices such that every pair of vertices in the subset is connected by an edge. A graph is *triangle-free* if it contains no clique of size 3. A *proper vertex coloring* of a graph G assigns colors to vertices such that no two adjacent vertices share the same color.

Parameterized Complexity. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$ where Σ^* is a set of strings over a finite alphabet Σ . An input of a parameterized problem is a pair (x, k) where x is a string over Σ and $k \in \mathbb{N}$ is a *parameter*. A parameterized problem is *fixed-parameter tractable* (or FPT) if it can be solved in $f(k) \cdot |x|^{\mathcal{O}(1)}$ time for some computable function f . The complexity class FPT consists of all fixed-parameter tractable parameterized problems. A parameterized problem is in the class XP if it can be solved in $|x|^{f(k)}$ time for a computable function $f(\cdot)$. The standard way to rule out the existence of an FPT-algorithm for a parameterized problem under the standard parameterized complexity assumption that $\text{FPT} \neq \text{W}[1]$ is to show that it is W[1]-hard, that is, at least as hard as any problem in the class of parameterized problems W[1]. A problem is *para-NP-hard* if it is NP-hard even when the parameter k is a constant.

A *kernelization algorithm* or *kernel* for a parameterized problem L is a polynomial-time algorithm that takes as its input an instance (x, k) of L and returns an instance (x', k') of the same problem such that (i) $(x, k) \in L$ if and only if $(x', k') \in L$ and (ii) $|x'| + k' \leq f(k)$ for some computable function $f: \mathbb{N} \rightarrow \mathbb{N}$. The function $f(\cdot)$ is the *size* of the kernel; a kernel is *polynomial* if f is a polynomial. A decidable parameterized problem is FPT if and only if it admits a kernel. However, up to some standard complexity-theoretic assumptions, there are

FPT problems that have no polynomial kernels. One of the ways to exclude a polynomial kernel for a parameterized problem is to show that there is a *polynomial parameter transformation* from a parameterized problem for which the existence of a polynomial kernel does not exist unless $\text{NP} \subseteq \text{coNP}/\text{poly}$, where a polynomial parameter transformation is a polynomial-time reduction such that the parameter in the new instance is bounded by a polynomial in the original parameter.

3 The Complexity of Ulam Metric k -Center

In this section, we study the ULAM METRIC k -CENTER problem. We first show that the problem is already NP-hard for $d = 1$. Our main result in this section is the fixed-parameter tractability of the problem parameterized by $k + d$. Nevertheless, under standard complexity assumptions, the problem does not admit a polynomial kernel parameterized by $k + d$.

3.1 Lower Bound for Constant Radius d

We prove that ULAM METRIC k -CENTER is NP-hard when the radius is fixed to the constant d , via a reduction from VERTEX COVER. Notice that VERTEX COVER remains NP-hard in triangle-free graphs [32]. This follows from a standard 2-subdivision transformation: replacing each edge by a path of length 3 eliminates all triangles and increases the size of a minimum vertex cover by exactly one per original edge. We may thus assume without loss of generality that the input graph is triangle-free, which suffices to prove Theorem 1. The idea of the reduction is to have two symbols per vertex. All permutations relevant to the reduction look almost entirely the same, except that only the pair of symbols for one or two vertices are swapped to encode that respective vertex or edge. If we let the input set of permutations encode all edges of the graph this way, then a vertex cover corresponds to a set of permutations encoding vertices. This holds as each permutation encoding an edge has Ulam distance 1 to only those permutations encoding vertices to which the edge is incident.

► **Theorem 1** ($\star$). *ULAM METRIC k -CENTER is NP-hard even if d is fixed to any positive constant at least 1.*

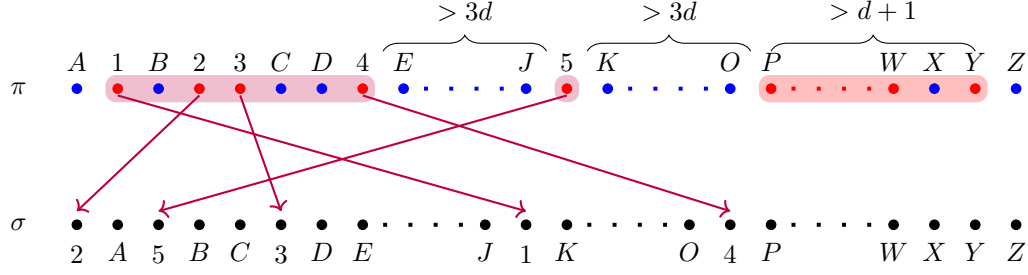
3.2 Fixed-Parameter Tractability by $k + d$

Since the problem is NP-hard even for constant d or k , neither parameter yields tractability on its own. We therefore consider the combined parameter $k + d$, and show that the Ulam center problem is fixed-parameter tractable under this parameterization.

Our algorithm relies on iteratively updating candidate permutations to eventually converge towards the center permutations by moving one symbol in a candidate in each step. Specifically, given a *candidate* permutation π_c and a *guide* permutation π_g , we can enumerate (in FPT time) a set P of permutations that differ from π_c by one move. Among them, at least one gets closer to any true center that covers both π_c and π_g . The key technical ingredient is captured in Lemma 16.

Our approach is based on the 2-colorings of the symbols in π_g and π_c . The 2-colorings encode which symbols are allowed to be moved toward a specific permutation $\sigma \in P$. We formalize when such a 2-coloring is fitting with respect to σ , an exemplary coloring is visualized in Figure 4.

► **Definition 8** (d -fitting coloring and witness). *Let $d \in \mathbb{N}$, $\pi \in \mathcal{S}_n(\Sigma)$, and $\text{col}: \Sigma \rightarrow \{\text{red}, \text{blue}\}$ be a coloring function of π . We say col is a d -fitting coloring (of π) with respect to some permutation $\sigma \in \mathcal{S}_n(\Sigma)$, if there is a subset of symbols $\Sigma_\pi \subseteq \Sigma$ such that*



■ **Figure 4** A permutation π with a d -fitting coloring with respect to a permutation σ , with *witness* $\Sigma_\pi = \{1, 2, 3, 4, 5\}$. Substrings highlighted in red form blocks. The rightmost block contains more than d red symbols, meaning that it cannot contain any moved symbol.

- $|\Sigma_\pi| = \text{dist}_U(\pi, \sigma)$ and σ can be obtained from π by moving the symbols in Σ_π ;
- for every symbol in $x \in \Sigma_\pi$, we have $\text{col}(x) = \text{red}$; and
- for every symbol $x \in \Sigma \setminus \Sigma_\pi$, if x appears in π within $3d$ symbols (before or after) of some symbol in Σ_π , then $\text{col}(x) = \text{blue}$.

Moreover, such a subset of symbols Σ_π is called a *witness* of the coloring col to be d -fitting.

Intuitively, red symbols are candidates for being moved (i.e., they may or may not actually be moved), but blue symbols are definitely not moved. Moreover, if a red symbol is indeed one of the moved symbols (i.e., it belongs to Σ_π), then all symbols within distance $3d$ of it must be correctly colored blue (if they are not moved) or red (if they are moved). In other words, whenever the coloring correctly identifies a moved symbol as red, it must also correctly color its entire $3d$ -neighborhood.

Consequently, the red symbols in a d -fitting coloring of π w.r.t. σ that are not separated by $3d$ consecutive blue symbols are tied to each other in the sense that either both or none of them belong to any witness Σ_π . This motivates the following definitions of blocks.

► **Definition 9** (Blocks). Let $d \in \mathbb{N}$, $\pi \in \mathcal{S}_n(\Sigma)$ and $\text{col}: \Sigma \rightarrow \{\text{red}, \text{blue}\}$. A block is a maximal substring (i.e., consecutive sequence) β of π such that

- the first and last symbols of β are colored red; and
- β contains no $3d$ consecutive blue symbols.

See also Figure 4 for a visualization of blocks. By definition, any two blocks are disjoint and separated by at least $3d$ blue vertices in π . We immediately obtain the following observation by recoloring all symbols in a block blue if it contains more than d red symbols.

► **Observation 10.** Let $d \in \mathbb{N}$, $\pi \in \mathcal{S}_n(\Sigma)$. Suppose col is a d -fitting coloring of π w.r.t. some permutation $\sigma \in \mathcal{S}_n(\Sigma)$. If $\text{dist}_U(\pi, \sigma) \leq d$, there is a d -fitting coloring col' of π w.r.t. σ such that there are at most d red symbols in each block. Moreover, such a coloring col' can be obtained from col in $\mathcal{O}(n)$ time.

We next argue how the notion of d -fitting colorings and that of witnesses provide a means to find a permutation that is one step closer to any true center σ than the candidate π_c .

► **Lemma 11.** Let $d \in \mathbb{N}$, $\pi_c, \pi_g \in \mathcal{S}_n(\Sigma)$. Let col_c and col_g be d -fitting colorings of π_c and π_g w.r.t. some permutation $\sigma \in \mathcal{S}_n(\Sigma)$ with $1 \leq \text{dist}_U(\pi_c, \sigma) \leq d$ and $\text{dist}_U(\pi_g, \sigma) \leq d$. Consider any symbol $x \in \Sigma$ such that there are witnesses Σ_c and Σ_g of col_c and col_g , respectively, with $x \in \Sigma_c \setminus \Sigma_g$. Then in time $\mathcal{O}(dn)$ we can compute and enumerate a set P with $|P| \leq 6d + 2$ and there is $\tilde{\pi} \in P$ with $\text{dist}_U(\tilde{\pi}, \sigma) = \text{dist}_U(\pi_c, \sigma) - 1$. The computation does not require the sets Σ_c and Σ_g as input.

19:12 Clustering Permutations Under the Ulam Metric

Proof. Consider an arbitrary pair Σ_c and Σ_g of witnesses. Intuitively, $x \in \Sigma_c$ means that by moving x to a specific new location in π_c , we make π_c closer to σ . As $x \notin \Sigma_g$, the position of x in π_g gives an indication for that location.

To see where to move x in π_c , let b_l be the closest symbol to the left of x in π_g that is still blue in both permutations, and let b_r be the closest such symbol to the right of x in π_g . If there is no such symbol on the right and/or left, consider b_l or b_r to be a *phantom* blue symbol before/after the first/last symbol of the permutation, respectively. As Σ_g does not contain any blue symbols, it does not contain any of x, b_l , and b_r . Thus, as x lies between b_l and b_r in π_g , it lies between them in σ as well. As b_l and b_r are blue in col_c as well, they do not change their position between π_c and σ . We add all permutations to P that are obtained by moving x in π_c to some location between b_l and b_r . This ensures that we consider all possible ways of moving x closer to its position in σ , leading to the fact that P always contains a permutation $\tilde{\pi}$ that is one step closer to σ than π_c .

We argue that we only need to add at most $6d + 2$ such permutations to P . Consider the next $2d + 1$ symbols to the left of x in π_g . Note that they belong to at most one block in π_g . In each block, there are at most d red symbols, and at least $d + 1$ of these symbols are blue in col_g . Thereby, these symbols do not appear in any witness Σ_g of col_g . Let L be the set of these blue symbols (select the closest $d + 1$ to x , if there are more than $d + 1$). As $\text{dist}_U(\pi_g, \sigma) \leq d$, the symbols in L appear as a subsequence of σ which is interrupted only by at most d other symbols. Call the set of these interrupting symbols I . As the symbols in $L \cup I$ occur consecutively in σ , the symbols in $(L \cup I) \setminus \Sigma_c$ appear consecutively in π_c , except some potential interruptions by symbols in Σ_c . Thus, the symbols in $L \setminus \Sigma_c$ have a pairwise maximal distance of at most $|L| + |I| + |\Sigma_c| - 1 \leq 3d$ in π_c . Thus, they belong to at most one block in π_c and so at most d of them are red in col_c . Hence, there is a symbol $b_l \in L$ which is blue in both col_g and col_c . The same line of argument shows that a suitable symbol b_r exists among the $2d + 1$ symbols to the right of x in π_g .

As the distance between b_l and b_r is at most $4d + 2$ in π_g and $\text{dist}_U(\pi_c, \pi_g) \leq 2d$, the distance between b_l and b_r in π_c is at most $6d + 2$, giving that many possibilities on where to insert x . If one or both of b_l and b_r are phantom symbols at the beginning/end of a permutation, the same upper bound applies.

We can find b_l and b_r in time $O(d)$ by starting a search outwards from x in π_c . Listing all obtained permutations takes time in $O(dn)$. ◀

It remains to show how to find such a symbol x . For technical reasons, we require a weak notion of consistency between two d -fitting colorings, which motivates the following definition.

► **Definition 12** (Consistent d -fitting colorings). *Let $d \in \mathbb{N}$, $\pi_c, \pi_g \in \mathcal{S}_n(\Sigma)$, and $\text{col}_c, \text{col}_g$ be d -fitting colorings of π_c and π_g with respect to some permutation $\sigma \in \mathcal{S}_n(\Sigma)$, respectively. We say that col_c and col_g are consistent d -fitting with respect to σ if there are respective witnesses $\Sigma_c, \Sigma_g \subseteq \Sigma$, such that for every symbol $x \in \Sigma_c \cup \Sigma_g$, we have $\text{col}_g(x) = \text{red}$ if and only if $x \in \Sigma_g$, and $\text{col}_c(x) = \text{red}$ if and only if $x \in \Sigma_c$.*

We first show how to enumerate a set of candidate pairs of colorings to be consistent d -fitting with respect to any σ .

► **Lemma 13.** *Let $d \in \mathbb{N}$ and $\pi_c, \pi_g \in \mathcal{S}_n(\Sigma)$. There is a set $\mathcal{C}$ of 2-coloring pairs such that*

- *the size of $\mathcal{C}$ is bounded by $|\mathcal{C}| \leq 2^{O(d^2)} \log n$; and*
- *for every $\sigma \in \mathcal{S}_n(\Sigma)$ with $\text{dist}_U(\pi_c, \sigma) \leq d$ and $\text{dist}_U(\pi_g, \sigma) \leq d$, $\mathcal{C}$ contains a consistent d -fitting coloring pair of π_c and π_g w.r.t. σ .*

Moreover, such a set can be computed and listed in $2^{O(d^2)} n \log n$ time, where $n := |\Sigma|$.

Proof. We construct the collection $\mathcal{C}$ using the notion of (a, b) -universality. An (a, b) -universal set $\mathcal{U} \subseteq \{0, 1\}^a$ is a set of bitstrings of length a , such that for each subset $\mathcal{S} \subseteq [a]$ of size b , the projection of $\mathcal{U}$ onto the coordinates in $\mathcal{S}$ yields all 2^b possible bitstrings of length b . Equivalently, if we restrict each string in $\mathcal{U}$ to any fixed set of b coordinates, all possible 2^b binary patterns appear. Naor, Schulman, and Srinivasan [28] showed that such a set exists with size at most $2^b b^{\mathcal{O}(\log b)} \log a$, and can be listed in time $2^b b^{\mathcal{O}(\log b)} a \log a$.

In our setting, we take $a = 2n$ and $b = 12d^2 + 4d$. Applying the above, we obtain an (a, b) -universal set $\mathcal{C} \subseteq \{0, 1\}^a$ of size

$$2^b b^{\mathcal{O}(\log b)} \log(a) = 2^{\mathcal{O}(d^2 + \log^2 d)} \log n = 2^{\mathcal{O}(d^2)} \log n,$$

which can be listed in time $2^{\mathcal{O}(d^2)} n \log n$. Each bitstring in $\{0, 1\}^{2n}$ naturally defines a pair of 2-colorings of the symbols in Σ : the first n bits define the coloring col_c of π_c and the last n bits define a coloring col_g of π_g .

Now, consider any permutation σ with $\text{dist}_U(\pi_c, \sigma) \leq d$ and $\text{dist}_U(\pi_g, \sigma) \leq d$. Let Σ_c and Σ_g be minimum-sized sets of moved symbols in π_c and π_g , respectively, to obtain σ . By Definitions 8 and 12, the 2-colorings col_c and col_g are consistent d -fitting w.r.t. σ if

- (i) every symbol in Σ_c (resp., Σ_g) is colored red in col_c (resp., col_g);
- (ii) every symbol in $\Sigma \setminus \Sigma_c$ (resp., $\Sigma \setminus \Sigma_g$) within distance $3d$ of a moved symbol in Σ_c (resp., Σ_g) is colored blue in col_c (resp., col_g); and
- (iii) every symbol in $\Sigma_g \setminus \Sigma_c$ (resp., $\Sigma_c \setminus \Sigma_g$) is colored blue in col_g (resp., col_c).

Recall that both sets have size at most d . The number of symbols involved in these conditions is at most

$$(|\Sigma_c| + |\Sigma_g|) + 2 \cdot 3d(|\Sigma_c| + |\Sigma_g|) + (|\Sigma_c| + |\Sigma_g|) \leq 12d^2 + 4d = b.$$

Because our (a, b) -universal set contains all possible pairs of colorings on any set of b positions, it must contain a pair that assigns suitable colors to all b involved symbols. Hence, this pair is consistent d -fitting w.r.t. σ . $\blacktriangleleft$

Given a guide π_g and the candidate π_c , Lemma 13 allows us to guess a consistent d -fitting coloring pair $(\text{col}_c, \text{col}_g)$ w.r.t. some center σ within distance d of both. We now use the colorings col_c and col_g to capture the structural relationship between the two permutations. To this end, we consider a certain vertex cover of their *permutation graph*. Here, a permutation graph G_{π_c, π_g} of two permutations $\pi_c, \pi_g \in \mathcal{S}_n(\Sigma)$ is the graph whose vertex set is the alphabet Σ and its edges are exactly the (unordered) pairs of vertices whose order is reversed between π_c and π_g . For each edge in G_{π_c, π_g} , exactly one of π_c and π_g has the two incident symbols in the same order as the center σ . A vertex cover of G_{π_c, π_g} thus gives an indication of which symbols need to be moved to transform π_g and π_c into σ .

► **Lemma 14.** *Let $d \in \mathbb{N}$, $\pi_c, \pi_g \in \mathcal{S}_n(\Sigma)$. Let col_c and col_g be consistent d -fitting colorings of π_c and π_g w.r.t. some permutation $\sigma \in \mathcal{S}_n(\Sigma)$. For every pair of witnesses Σ_c, Σ_g for col_c and col_g there is an inclusion-wise minimal vertex cover X of G_{π_c, π_g} with $X \subseteq \Sigma_c \cup \Sigma_g$.*

Proof. We begin by showing that $\Sigma_c \cup \Sigma_g$ is a vertex cover of G_{π_c, π_g} . Consider an edge xy in G_{π_c, π_g} . By definition of the permutation graphs, x and y appear in reversed order in π_c and π_g . Since σ is obtained from π_c (resp., π_g) by moving symbols in Σ_c (resp., Σ_g), at least one of x and y must be moved in one of the two permutations. Hence, at least one of x and y belongs to $\Sigma_c \cup \Sigma_g$, showing that $\Sigma_c \cup \Sigma_g$ is a vertex cover of G_{π_c, π_g} .

The lemma follows by exhaustively removing elements from $\Sigma_c \cup \Sigma_g$ upon whose removal all edges are still covered. $\blacktriangleleft$

Such a vertex cover indeed suffices to find a suitable vertex x as required for Lemma 11.

► **Lemma 15.** *Let $d \in \mathbb{N}$, $\pi_c, \pi_g \in \mathcal{S}_n(\Sigma)$. Let col_c and col_g be consistent d -fitting colorings of π_c and π_g w.r.t. some permutation $\sigma \in \mathcal{S}_n(\Sigma)$ with $\text{dist}_U(\pi_c, \sigma) \leq d$ and $\text{dist}_U(\pi_g, \sigma) \leq d$. Consider an inclusion-wise minimal vertex cover X of G_{π_c, π_g} with $X \subseteq \Sigma_c \cup \Sigma_g$ for a pair of witnesses Σ_c, Σ_g for col_c and col_g . Then in time $\mathcal{O}(|X|)$ we can find a symbol $x \in \Sigma$ with $x \in \Sigma_c \setminus \Sigma_g$. The computation does not require the sets Σ_c and Σ_g as input.*

Proof. Observe that π_g can be transformed into π_c by deleting and reinserting the symbols in X , so $\text{dist}_U(\pi_g, \pi_c) > d$ implies that $|X| > d$. As $\text{dist}_U(\pi_g, \sigma) = |\Sigma_g| \leq d$, we have that there is a symbol $x \in X$ such that $x \in \Sigma_c \setminus \Sigma_g$. As col_g and col_c are consistent d -fitting, we have that $\text{col}_c(x) = \text{red}$ and $\text{col}_g(x) = \text{blue}$. It suffices to consider any such colored symbol x' , as $\text{col}_g(x') = \text{blue}$ implies $x' \notin \Sigma_g$ and thus, by $X \subseteq \Sigma_c \cup \Sigma_g$ we have $x' \in \Sigma_c \setminus \Sigma_g$. ◀

With this, we are ready to forge our central tool:

► **Lemma 16.** *Let $d \in \mathbb{N}$, and $\pi_g, \pi_c \in \mathcal{S}_n(\Sigma)$ be two permutations with $d < \text{dist}_U(\pi_g, \pi_c) \leq 2d$. There exists a set of permutations $P \subseteq \mathcal{S}_n(\Sigma)$ satisfying*

- *the size of P is bounded by $|P| \leq 2^{\mathcal{O}(d^2)} \log n$; and*
- *for each permutation $\sigma \in \mathcal{S}_n(\Sigma)$ with $\text{dist}_U(\pi_g, \sigma) \leq d$ and $\text{dist}_U(\pi_c, \sigma) \leq d$, P contains a permutation $\tilde{\pi} \in P$ that is one step closer to σ than π_c , i.e., $\text{dist}_U(\tilde{\pi}, \sigma) = \text{dist}_U(\pi_c, \sigma) - 1$. Furthermore, such a set can be computed in $2^{\mathcal{O}(d^2)} n \log n$ time.*

Proof. We compute a collection $\mathcal{C}$ of pairs of 2-colorings of Σ according to Lemma 13. For each $\sigma \in \mathcal{S}_n(\Sigma)$ with $\text{dist}_U(\pi_g, \sigma) \leq d$ and $\text{dist}_U(\pi_c, \sigma) \leq d$, there exists a consistent d -fitting pair of colorings $(\text{col}_c, \text{col}_g) \in \mathcal{C}$ w.r.t. σ . Thus, for each pair in $\mathcal{C}$ we perform the following procedure, adding permutations to P as we go along.

Based on Observation 10, we can assume that both col_c and col_g have at most d red symbols in each block (if not, we recolor some blocks blue accordingly). We list all inclusion-wise minimal vertex covers of G_{π_c, π_g} that have size at most $2d$. We can do so by a simple and well-known branching technique: Start with an empty cover, iteratively take any uncovered edge vw and create two branches where you add v or w to the vertex cover, respectively, and repeat this procedure $2d$ times (or until all edges are covered). This list of at most $2^{2d} = 4^d$ covers may contain some that are not inclusion-wise minimal, but these can be filtered out in time $\mathcal{O}(4^d \cdot |V(G_{\pi_c, \pi_g})|) = \mathcal{O}(4^d n)$.

By Lemma 14 and as every witness for col_c or col_g has size at most d , at least one of the listed vertex covers consists only of vertices in $\Sigma_c \cup \Sigma_g$, where Σ_c and Σ_g are witnesses for the respective colorings. This holds true for *every* permutation σ for which the current colorings are consistent d -fitting. Using Lemma 15 on each of the vertex covers, we obtain a set of at most 4^d symbols, and for every suitable σ at least one of these symbols is included in $\Sigma_c \setminus \Sigma_g$ for respective witnesses. We note that applying Lemma 15 to vertex covers containing vertices which do not stem from union of witnesses might yield incorrect symbols, but for each of the vertex covers we still add at most one symbol. We apply Lemma 11 to each of the 4^d symbols, thereby producing a set of $4^d \cdot (6d + 2)$ permutations, which we all add to P . Note that the lemmas guarantee that for every σ for which the colorings are consistent d -fitting, we added at least one permutation which is one step closer to σ than the current candidate π_c .

The final size of P is bounded by

$$|P| \leq |\mathcal{C}| \cdot 4^d (6d + 2) \leq 2^{\mathcal{O}(d^2)} \log n.$$

The runtime estimate is based on combining the size of $\mathcal{C}$ from Lemma 13 with the time to find the unique vertex cover respecting a given pair of colorings. All other steps, like processing the blocks, are asymptotically dominated by that time. There are at most $2^{\mathcal{O}(d^2)} \log n$ pairs of colorings to enumerate. For each pair, the branching procedure on the vertex covers takes $\mathcal{O}(4^d \cdot (|E(G_{\pi_c, \pi_g})| + |V(G_{\pi_c, \pi_g})|)) = \mathcal{O}(4^d dn)$ time as $|V(G_{\pi_c, \pi_g})| = n$ and the existence of a vertex cover of size $2d$ implies that $|E(G_{\pi_c, \pi_g})| \leq 2dn$. Thus, the total time to compute P is in

$$2^{\mathcal{O}(d^2)} \log n \cdot \mathcal{O}(4^d dn) = 2^{\mathcal{O}(d^2)} n \log n. \quad \blacktriangleleft$$

Now we are ready to prove Theorem 2.

► **Theorem 2.** *ULAM METRIC k -CENTER can be solved in $2^{\mathcal{O}(d^2 k(d+k))} \cdot mn^{1+o(1)}$ time, where $m := |\Pi|$ and $n := |\Sigma|$.*

Proof. We may assume without loss of generality that any YES-instance admits a solution with exactly k centers. Indeed, if a solution exists with fewer than k centers, we can always add arbitrary permutations as centers until we reach exactly k centers. In addition, we exclude the trivial YES-instances where $k > m$. Now, consider a hypothetical solution $S^* = \{\sigma_1^*, \sigma_2^*, \dots, \sigma_k^*\}$ consisting of k centers. For each $i \in [k]$, let the i th cluster be the set of permutations in Π with distance at most d to σ_i^* . If a permutation would fit into multiple clusters, we assign it to only one of them arbitrarily.

We describe our algorithm as a branching procedure. It maintains a candidate set $S \subseteq \mathcal{S}_n(\Sigma)$ of size at most k along with a budget b_i for each candidate $\sigma_i \in S$. Here, S is intended to evolve into a solution to the ULAM METRIC k -CENTER instance, and each budget b_i indicates how many modifications to σ_i remain (or have been used so far). In each branch, the algorithm checks whether the current candidate set S covers all permutations in Π within distance d . The algorithm halts and accepts the instance if and only if at least one branch finds a valid solution. We now describe the procedure in detail; see Algorithm 1 for a high-level overview.

In the course of the algorithm, we one by one initialize each candidate σ_i as a permutation from Π , and evolve it into the i th center. Initially, pick the permutation $\pi_1 \in \Pi$ and set the first candidate σ_1 to equal this permutation: $\sigma_1 := \pi$. Then, the algorithm adds it as the first candidate to S and initializes its budget as $b_1 := d$.

Next, the algorithm branches on a permutation $\pi \in \Pi$ that has distance more than d to each of the candidates in S . If there is none, then the current candidates form a solution, and we accept the instance. Otherwise, the permutation π either belongs to one of the existing clusters or to a new cluster.

- **Existing cluster:** For each candidate $\sigma_i \in S$ with remaining budget $b_i > 0$ and distance at most $2d$ to π , we consider the case where π belongs to the i th cluster. We invoke Lemma 16 with d , guide $\pi_g = \pi$, and $\pi_c = \sigma_i$ to obtain the set P of permutations. For each permutation $\sigma'_i \in P$, we create a new branch where we replace σ_i by σ'_i and decrease its budget by one.
- **New cluster:** If $|S| < k$, we create a branch where π belongs to a new cluster. Specifically, we let $\sigma_{|S|+1} = \pi$ as a new candidate, initialize its budget as $b_{|S|+1} = d$, and add it to S . If none of the two above cases applies, we reject the branch. We accept the instance if and only if at least one branch accepts it. If none of the branches accepts the instance, we reject the branch.

Observe that if the algorithm accepts an instance, it indeed finds a solution. Conversely, consider a YES-instance of ULAM METRIC k -CENTER witnessed by some solution $S^* = \{\sigma_1^*, \sigma_2^*, \dots, \sigma_k^*\}$. When branching on a permutation $\pi \in \Pi$ not covered by the current

■ **Algorithm 1** Solving ULAM METRIC k -CENTER for radius d on permutations Π .

Input: An alphabet Σ of size n , a set of permutations Π of size m , integers k and d .
Output: Decide the whether there exist $S \subseteq \mathcal{S}_n(\Sigma)$ of size k such that for each $\pi \in \Pi$, there exists $\sigma \in S$ with $\text{dist}_U(\pi, \sigma) \leq d$.

```

1  $\sigma_1 \leftarrow \pi_1 \in \Pi$ ; // Initialize the first candidate center
2  $b_1 \leftarrow d$ ; // Initialize the budget of  $\sigma_1$ 
3 return  $\text{ALG}(\Sigma, \Pi, k, d, \{\sigma_1\}, \{b_1\})$ ;
4 Procedure  $\text{ALG}(\Sigma, \Pi, k, d, S = \{\sigma_1, \dots, \sigma_{|S|}\}, \{b_i\}_{i=1}^{|S|})$ 
5   if for every  $\pi \in \Pi$  there is  $\sigma_i \in S$  such that  $\text{dist}_U(\pi, \sigma_i) \leq d$  then accept
6     instance;
7   Pick arbitrary  $\pi \in \Pi$  with  $\text{dist}_U(\pi, \sigma_i) > d$  for all  $\sigma_i \in S$ ;
8   // Branching on how a permutation not yet covered by  $S$  is covered
9   for each  $\sigma_i \in \{\sigma_i \in S : b_i > 0, \text{dist}_U(\pi, \sigma_i) \leq 2d\}$  do
10     // The case that  $\pi$  is in the  $i$ th cluster.
11      $P \leftarrow$  invoke Lemma 16 with  $(d, \pi_g, \pi_c)$ , where  $\pi_g \leftarrow \pi$  and  $\pi_c \leftarrow \sigma_i$ ;
12     for each  $\sigma'_i \in P$  do
13        $\sigma_i \leftarrow \sigma'_i$ ; // Modify the candidate  $\sigma_i$ 
14        $b_i \leftarrow b_i - 1$ ; // Update the budget of  $\sigma_i$ 
15       accept instance if  $\text{ALG}(\Sigma, \Pi, k, d, S, \{b_i\}_{i=1}^{|S|})$  accepts;
16     end
17   end
18   if  $|S| < k$  then
19     // The case that  $\pi$  belongs to a new cluster.
20      $\sigma_j \leftarrow \pi$ , where  $j = |S| + 1$ ; // Add  $\pi$  as a new candidate center
21      $b_j \leftarrow d$ ; // Initialize the budget of  $\pi$ .
22     accept instance if  $\text{ALG}(\Sigma, \Pi, k, d, S \cup \{\sigma_j\}, \{b_i\}_{i=1}^j)$  accepts;
23   end
24   reject branch;
```

candidates S , there exists a branch where π is assigned to the cluster to which it belongs in S^* . Moreover, when π is assigned to the i th cluster from the current candidates S , by Lemma 16, there is a permutation $\sigma'_i \in P$ that is one step closer to σ_i^* than σ_i . Thus, in the respective branch, we replace σ_i by σ'_i , moving it one step closer to σ_i^* . Since each candidate starts with the distance at most d to its respective center in S^* , after receiving at most d updates, it will have converged to that center. From this point on, it will not receive any more updates since there are no permutations assigned to that cluster with a distance of more than d . Therefore, there is a branch in which the algorithm finds the solution S^* and accepts the instance.

Finally, we analyze the running time of the algorithm. Recall that the input of the branching procedure contains a set of candidates S along with their budgets $\{b_i : \sigma_i \in S\}$. Define the measure of the current input of a branch as

$$\mu := (k - |S|)(d + 1) + \sum_{i=1}^{|S|} b_i.$$

and denote by $T(\mu)$ the worst-case running time of the algorithm on inputs with measure μ .

Observe that each time we add a new candidate to S or modify an existing candidate in S , the measure μ decreases by one. On the one hand, when we add a new candidate to S , the size of S increases by one while the budget of the new candidate is initialized to d . Thus, the measure decreases by $(d+1) - d = 1$ and there is at most one such branch. On the other hand, when we modify a candidate to a permutation from P , its budget decreases by one while the size of S remains unchanged. Thus, the measure decreases by 1 as well, and there are at most $|S| \cdot |P| \leq k \cdot 2^{\mathcal{O}(d^2)} \log n$ such branches by Lemma 16. Moreover, computing P takes time in $2^{\mathcal{O}(d^2)} n \log n$, and the remaining time spend on this iteration is dominated by the computation of all Ulam distances to the current candidates (achievable in $\mathcal{O}(kmn \log n)$ time). Therefore, we have the recurrence

$$T(\mu) \leq 2^{\mathcal{O}(d^2)} n \log n + \mathcal{O}(kmn \log n) + (1 + k \cdot 2^{\mathcal{O}(d^2)} \log n) \cdot T(\mu - 1).$$

Notice that the measure satisfies $\mu \leq k(d+1)$ and the algorithm halts when $\mu = 0$. Solving the recurrence yields

$$\begin{aligned} T(k(d+1)) &= (1 + k \cdot 2^{\mathcal{O}(d^2)} \log n)^{k(d+1)} \cdot (2^{\mathcal{O}(d^2)} n \log n + kmn \log n) \\ &= 2^{\mathcal{O}(d^3 k)} \cdot k^{\mathcal{O}(dk)} \cdot (\log n)^{k(d+1)} \cdot \mathcal{O}(mn \log n). \end{aligned}$$

Observe that

$$(\log n)^{k(d+1)} = 2^{k(d+1) \log \log n} \leq 2^{k^2(d+1)^2 + (\log \log n)^2} = 2^{k^2(d+1)^2} \cdot n^{(\log \log n)^2 / \log n},$$

which bounds the total running time by

$$2^{\mathcal{O}(d^3 k + d^2 k^2)} \cdot k^{\mathcal{O}(dk)} \cdot \mathcal{O}(mn^{1 + (\log \log n)^2 / \log n} \log n) = 2^{\mathcal{O}(d^2 k(d+k))} \cdot mn^{1+o(1)}.$$

This shows that ULAM METRIC k -CENTER is in FPT by $k + d$. ◀

We prove that the fixed-parameter tractability of Ulam distance with respect to $k + d$ does not extend to a polynomial kernel via a reduction from a problem on binary strings.

► **Theorem 3** (*). *ULAM METRIC k -CENTER does not admit a polynomial-sized kernel for the parameter $k + d$ unless $\text{NP} \subseteq \text{coNP/poly}$.*

4 The Complexity of Ulam Metric k -Median

This section studies the parameterized complexity of ULAM METRIC k -MEDIAN with respect to the parameters d and $k + d$. We first prove that ULAM METRIC k -MEDIAN is $\text{W}[1]$ -hard parameterized by d but is in XP for this parameterization. Then we show a polynomial kernel parameterized by both k and d , implying fixed-parameter tractability for this parameterization.

We first rule out fixed-parameter tractability by d alone under established complexity assumptions. We then contrast it with a simple XP -algorithm for ULAM METRIC k -MEDIAN.

For the hardness result, we give a reduction from MULTICOLORED CLIQUE. An instance of MULTICOLORED CLIQUE consists of a graph G with a proper vertex coloring using k' colors, and asks whether G contains a clique of size k' . The problem is known to be $\text{W}[1]$ -hard when parameterized by k' [31]. For clarity, we here describe the reduction and then separately prove that it preserves YES-instances and NO-instances.

Let $\mathcal{C} = \{c_1, c_2, \dots, c_{k'}\}$ denote the color set, let $V(G) = \{v_1, v_2, \dots, v_{n'}\}$ be the vertex set of G and let $E(G) = \{e_1, e_2, \dots, e_{m'}\}$ be the edge set of G , where $n' = |V(G)|$ and $m' = |E(G)|$. Without loss of generality, we assume that $k' \geq 4$, as instances with smaller values can be solved in polynomial time by brute-force.

19:18 Clustering Permutations Under the Ulam Metric

We construct an instance (Π, k, d) of ULAM METRIC k -MEDIAN with $k = m' - \binom{k'}{2} + 1$. The idea of the reduction is to create one permutation for each edge in G . We build the instance in a way such that the existence of a multicolored clique $Q \subseteq V(G)$ allows the construction of a center permutation σ_Q such that edges with both endpoints in Q have small total distance to σ_Q , while every other edge can take itself as its center. Consequently, the number of centers required is $m' - \binom{|Q|}{2} + 1 = k$, and the median distance is entirely contributed by the edges inside Q . It remains to choose a suitable value for d .

Formally, let $q = \binom{k'}{2}(k' - 2) + 1$. Furthermore, we define the alphabet as the union of the color set, some v -symbols representing the vertices, and some x -, y -, and z -symbols representing the edges:

$$\Sigma = \mathcal{C} \cup \{v_j^i : i \in [q], v_j \in V(G)\} \cup \{x_j^i, y_j^i, z_j^i : i \in [q], e_j \in E(G)\}.$$

We have $n = |\Sigma| = k' + n'q + 3m'q$. We first construct a base permutation $\pi_{\text{base}} \in S_n(\Sigma)$ that starts with alternating between the x - and y -symbols, followed by the v - and the z -symbols, and finally the color-symbols:

$$\begin{aligned} \pi_{\text{base}} := & \boxed{x_1^1 y_1^1 \cdots x_1^q y_1^q} \boxed{x_2^1 y_2^1 \cdots x_2^q y_2^q} \cdots \boxed{x_{m'}^1 y_{m'}^1 \cdots x_{m'}^q y_{m'}^q} \\ & \boxed{v_1^1 v_1^2 \cdots v_1^q} \boxed{v_2^1 v_2^2 \cdots v_2^q} \cdots \boxed{v_{n'}^1 v_{n'}^2 \cdots v_{n'}^q} \\ & \boxed{z_1^1 z_1^2 \cdots z_1^q} \boxed{z_2^1 z_2^2 \cdots z_2^q} \cdots \boxed{z_{m'}^1 z_{m'}^2 \cdots z_{m'}^q} \\ & c_1 c_2 \cdots c_{k'}. \end{aligned}$$

For each edge $e_j = v_a v_b \in E(G)$ with $a < b$, we introduce a permutation π_{e_j} obtained from π_{base} as follows. Suppose v_a has color c_α and v_b has color c_β . Swap x_j^i and y_j^i for each $i \in [q]$; move c_α to the right of v_a^q ; move c_β to the right of v_b^q ; and move the remaining color-symbols to the right of z_j^q without changing their relative order. That is,

$$\begin{aligned} \pi_{e_j} := & \boxed{x_1^1 y_1^1 \cdots x_1^q y_1^q} \cdots \boxed{y_j^1 x_j^1 \cdots y_j^q x_j^q} \cdots \boxed{x_{m'}^1 y_{m'}^1 \cdots x_{m'}^q y_{m'}^q} \\ & \boxed{v_1^1 \cdots v_1^q} \cdots \boxed{v_a^1 \cdots v_a^q} \boxed{c_\alpha} \cdots \boxed{v_b^1 \cdots v_b^q} \boxed{c_\beta} \cdots \boxed{v_{n'}^1 \cdots v_{n'}^q} \\ & \boxed{z_1^1 \cdots z_1^q} \cdots \boxed{z_j^1 \cdots z_j^q} \boxed{\chi_{\alpha, \beta}} \boxed{z_{j+1}^1 \cdots z_{j+1}^q} \cdots \boxed{z_{m'}^1 \cdots z_{m'}^q}, \end{aligned}$$

where $\chi_{\alpha, \beta} = \boxed{c_1 \cdots c_{\alpha-1} c_{\alpha+1} \cdots c_{\beta-1} c_{\beta+1} \cdots c_{k'}}$. Finally, recall that $k = m' - \binom{k'}{2} + 1$. We set $d = \binom{k'}{2}(q + (k' - 2))$, and let the set of permutations consist of all permutations for edges in the graph, i.e., $\Pi = \{\pi_{e_j} : e_j \in E(G)\}$.

► **Lemma 17.** *The stated reduction from MULTICOLORED CLIQUE to ULAM METRIC k -MEDIAN preserves YES-instances.*

Proof. Assume that G contains a clique Q of size at least k' , which contains exactly one vertex of each color. We prove that there is a set S of size k that is a solution for the ULAM METRIC k -MEDIAN instance (Π, k, d) . For each vertex $v_j \in Q$, let $c_{\varphi(j)}$ be the color of v_j . Define σ_Q to be the permutation obtained from π_{base} by moving each symbol $c_{\varphi(j)} \in \mathcal{C}$ right after the symbol $v_j^q \in Q$. (There is no color-symbol after v_j^q if $v_j \notin Q$.) That is, if $Q = \{v_{j_1}, v_{j_2}, \dots, v_{j_{k'}}\}$ with $j_1 < j_2 < \cdots < j_{k'}$, then

$$\begin{aligned}
\sigma_Q := & \boxed{x_1^1 y_1^1 \cdots x_1^q y_1^q} \boxed{x_2^1 y_2^1 \cdots x_2^q y_2^q} \cdots \boxed{x_{m'}^1 y_{m'}^1 \cdots x_{m'}^q y_{m'}^q} \\
& \boxed{v_1^1 \cdots v_1^q} \cdots \boxed{v_{j_1-1}^1 \cdots v_{j_1-1}^q} \boxed{v_{j_1}^1 \cdots v_{j_1}^q} \boxed{c_{\varphi(j_1)}} \boxed{v_{j_1+1}^1 \cdots v_{j_1+1}^q} \\
& \cdots \boxed{v_{j_i-1}^1 \cdots v_{j_i-1}^q} \boxed{v_{j_i}^1 \cdots v_{j_i}^q} \boxed{c_{\varphi(j_i)}} \boxed{v_{j_i+1}^1 \cdots v_{j_i+1}^q} \\
& \cdots \boxed{v_{j_{k'}-1}^1 \cdots v_{j_{k'}-1}^q} \boxed{v_{j_{k'}}^1 \cdots v_{j_{k'}}^q} \boxed{c_{\varphi(j_{k'})}} \boxed{v_{j_{k'}+1}^1 \cdots v_{j_{k'}+1}^q} \\
& \cdots \boxed{v_{n'}^1 \cdots v_{n'}^q} \boxed{z_1^1 \cdots z_1^q} \boxed{z_2^1 \cdots z_2^q} \cdots \boxed{z_{m'}^1 \cdots z_{m'}^q}.
\end{aligned}$$

We then let $S = \{\sigma_Q\} \cup \Pi \setminus \Pi_Q$, where $\Pi_Q \subseteq \Pi$ is the set of $\binom{k'}{2}$ permutations associated with an edge that has both endpoints covered by Q . Observe that $|S| = m' - \binom{k'}{2} + 1 = k$ and the sum of distances of each permutation to its closest median is at most $\sum_{\pi \in \Pi_Q} \text{dist}_U(\pi, \sigma_Q)$. Any permutation in Π_Q can be transformed into σ_Q by moving the $k' - 2$ color-symbols that are not yet at the correct position as well as the q symbols of the form x_j^i with $i \in [q]$. Thus, the total sum of distances is at most $\binom{k'}{2}(q + (k' - 2)) = d$. $\blacktriangleleft$

► **Lemma 18.** *The stated reduction from MULTICOLOREDCLIQUE to ULAM METRIC k -MEDIAN preserves NO-instances.*

Proof. We prove by showing that the existence of a solution to the constructed ULAM METRIC k -MEDIAN instance implies the existence of a solution to the MULTICOLOREDCLIQUE instance.

Assume the ULAM METRIC k -MEDIAN instance is solved by a set S of permutation medians which minimize the sum of distances. Consider the clustering of Π described by S , where we have one cluster per median permutation in S and each permutation in Π is assigned to the cluster of the closest median under Ulam metric (breaking ties arbitrarily). For ease of presentation, we partition the alphabet Σ into two parts: Σ' is the set of x - and y -symbols; and Σ'' is the set of v -, z -, and color-symbols. By construction, any permutation π_e in Π is a concatenation of two substrings $\pi'_e \in \mathcal{S}_{2m'q}(\Sigma')$ and $\pi''_e \in \mathcal{S}_{k'+n'q+m'q}(\Sigma'')$. It is easy to see that $\text{dist}_U(\pi_{e_1}, \pi_{e_2}) = \text{dist}_U(\pi'_{e_1}, \pi'_{e_2}) + \text{dist}_U(\pi''_{e_1}, \pi''_{e_2})$ for any pair of permutations $\pi_{e_1}, \pi_{e_2} \in \Pi$. We further define $C' = \{\pi'_e : \pi_e \in C\}$ and $C'' = \{\pi''_e : \pi_e \in C\}$ for any cluster C in the clustering.

Clearly, there exists a cluster of size at least two as $k < m'$. Consider a cluster C of size $|C| \geq 2$. We first argue that C is the unique cluster of size more than one. Let σ be its median with optimal total distance to the permutations in C . By the optimality of σ , the center σ starts with a substring $\sigma' \in \mathcal{S}_{2m'q}(\Sigma')$ and end with a substring $\sigma'' \in \mathcal{S}_{k'+n'q+m'q}(\Sigma'')$. It is easy to see that $\text{dist}_U(\pi_e, \sigma) = \text{dist}_U(\pi'_e, \sigma') + \text{dist}_U(\pi''_e, \sigma'')$ for any permutation $\pi_e \in C$. This means that we can optimize σ' and σ'' independently to minimize the total distance. First, restrict our attention to the x - and y -symbols. For any pair of permutations $\pi'_{e_1}, \pi'_{e_2} \in C'$, we have $\text{dist}_U(\pi'_{e_1}, \pi'_{e_2}) \geq 2q$ as they have swapped the x - and y -symbols for distinct edges. Thus, we can derive that

$$\begin{aligned}
\sum_{\pi'_e \in C'} \text{dist}_U(\pi'_e, \sigma') &= \frac{1}{|C'| - 1} \sum_{\{\pi'_{e_1}, \pi'_{e_2}\} \subseteq C'} (\text{dist}_U(\pi'_{e_1}, \sigma') + \text{dist}_U(\pi'_{e_2}, \sigma')) \\
&\geq \frac{1}{|C'| - 1} \sum_{\{\pi'_{e_1}, \pi'_{e_2}\} \subseteq C'} \text{dist}_U(\pi'_{e_1}, \pi'_{e_2}) \\
&\geq \frac{1}{|C'| - 1} \cdot \binom{|C'|}{2} \cdot 2q = |C'|q = |C|q.
\end{aligned}$$

19:20 Clustering Permutations Under the Ulam Metric

Suppose for the contraction that there were r clusters $C_1, C_2, \dots, C_r$ ($r \geq 2$) of size at least two with respective medians $\sigma'_1, \sigma'_2, \dots, \sigma'_r$. Summing over all these clusters yields

$$(|C'_1| + |C'_2| + \dots + |C'_r|)q \leq \sum_{i \in [r]} \sum_{\pi'_e \in C'_i} \text{dist}_U(\pi'_e, \sigma'_i) \leq d = \binom{k'}{2}(q + (k' - 2)) < \left(\binom{k'}{2} + 1\right)q,$$

which implies that $|C_1| + |C_2| + \dots + |C_r| \leq \binom{k'}{2}$. By the definition of k we have

$$m' + r - (|C_1| + |C_2| + \dots + |C_r|) \leq k = m' - \binom{k'}{2} + 1,$$

leading to $r \leq 1$, a contradiction. Hence, we conclude that C is the unique cluster of size more than one, which indicates $|C| \geq m' - k' + 1 = \binom{k'}{2}$.

Since (Π, k, d) is a YES-instance, we have d is at least

$$\sum_{\pi_e \in C} \text{dist}_U(\pi_e, \sigma) = \sum_{\pi'_e \in C'} \text{dist}_U(\pi'_e, \sigma') + \sum_{\pi''_e \in C''} \text{dist}_U(\pi''_e, \sigma'') \geq |C'|q + \sum_{\pi''_e \in C''} \text{dist}_U(\pi''_e, \sigma'').$$

Since $d = \binom{k'}{2}(q + (k' - 2))$ and $|C'| = |C| \geq \binom{k'}{2}$, we derive that

$$\sum_{\pi''_e \in C''} \text{dist}_U(\pi''_e, \sigma'') \leq \binom{k'}{2}(k' - 2) = q - 1.$$

This implies that only color-symbols can be moved from σ'' to each permutation π''_e in C'' : there are q copies of each v - and z -symbol in Σ'' and they are ordered the same way in all permutations. Suppose $\bar{C} \subseteq C$ is the set of colors located before z_1^1 in σ'' , and let $\bar{k} = |\bar{C}|$. Recall that each permutation π_e has two of the $k' \geq 4$ color-symbols placed before z_1^1 ; one each after the symbols for its incident vertices. Thus, for at most $\binom{\bar{k}}{2}$ of the permutations in C'' , both of these color-symbols are in $\bar{C}$. All other permutations have to move at least one of the two symbols, which requires a total of at least $\binom{k'}{2} - \binom{\bar{k}}{2}$ units of the budget. Additionally, for each permutation in C'' , all $k' - 2$ color-symbols after z_1^1 have to be moved, except a total of at most $k' - \bar{k}$ symbols across all permutations. As a result, the total number of required move operations is at least

$$\begin{aligned} \sum_{\pi''_e \in C''} \text{dist}_U(\pi''_e, \sigma'') &\geq \binom{k'}{2} - \binom{\bar{k}}{2} + \binom{k'}{2}(k' - 2) - (k' - \bar{k}) \\ &= \binom{k'}{2}(k' - 2) + \frac{1}{2}(k'(k' - 3) - \bar{k}(\bar{k} - 3)) \geq \binom{k'}{2}(k' - 2). \end{aligned}$$

Combine this with the bound on $\sum_{\pi'_e \in C'} \text{dist}_U(\pi'_e, \sigma')$, and we finally have

$$\begin{aligned} \sum_{\pi_e \in C} \text{dist}_U(\pi_e, \sigma) &= \sum_{\pi'_e \in C'} \text{dist}_U(\pi'_e, \sigma') + \sum_{\pi''_e \in C''} \text{dist}_U(\pi''_e, \sigma'') \\ &\geq \binom{k'}{2}q + \binom{k'}{2}(k' - 2) = d. \end{aligned}$$

Hence, all inequalities above are in fact equalities, which yields $|C'| = k'$ and $\bar{k} = k'$. In other words, all color-symbols are placed somewhere before z_1^1 in σ . Note further that the budget d is only sufficient if for every permutation $\pi_e \in C$ both color-symbols “incident” to e are located between the same substrings of v -symbols as in σ . Hence, the union of all endpoints of the $\binom{k'}{2}$ distinct edges described by C has size $|C'| = k'$, implying that these edges form a k' -clique in G . $\blacktriangleleft$

The $W[1]$ -hardness of ULAM METRIC k -MEDIAN immediately follows from Lemmas 17 and 18 and the facts that the described reduction can be computed in polynomial time and determines d by a computable function of k' .

► **Theorem 5.** *ULAM METRIC k -MEDIAN is $W[1]$ -hard parameterized by d .*

We contrast this lower bound by designing a simple XP-algorithm for ULAM METRIC k -MEDIAN.

► **Theorem 4 (★).** *ULAM METRIC k -MEDIAN can be solved in $\mathcal{O}((mn)^{2d})$ time, where $m := |\Pi|$ and $n := |\Sigma|$.*

Last, we find a polynomial-sized kernel with respect to $k + d$, which also establishes the fixed-parameter tractability ULAM METRIC k -MEDIAN by $k + d$.

► **Theorem 6 (★).** *ULAM METRIC k -MEDIAN admits a kernel with $\mathcal{O}(d^2 + dk)$ permutations over an alphabet of size $\mathcal{O}(d^4 + d^2k^2)$, computable in $\mathcal{O}(m^2n \log n)$ time, where $m := |\Pi|$ and $n := |\Sigma|$.*

5 Conclusion

We initiated the study of the parameterized complexity of the ULAM METRIC k -CENTER and ULAM METRIC k -MEDIAN problems. Prior to our work, even the parameterized complexity for the case $k = 1$ with parameter d was unknown for these problems.

It is known [18] that CLOSEST STRING cannot be solved in $2^{o(d \log d)} \cdot (nm)^{\mathcal{O}(1)}$ time unless the Exponential Time Hypothesis (ETH) fails (we refer to [18] for the formal definition of ETH). Then the Polynomial Parameter Transformation from Theorem 3 implies the same computational lower bound for ULAM METRIC 1-CENTER. Is it possible to solve ULAM METRIC 1-CENTER for $k = 1$ in $d^{\mathcal{O}(d)} \cdot (nm)^{\mathcal{O}(1)}$ time similarly to CLOSEST STRING?

For ULAM METRIC k -MEDIAN, it is shown that the problem is $W[1]$ -hard when parameterized by d but is in XP under the same parameterization. Furthermore, the problem admits a polynomial kernel for the parameterization by both k and d . The latter result together with an XP-algorithm implies that ULAM METRIC k -MEDIAN can be solved in $(kd)^{\mathcal{O}(d)} \cdot (mn)^{\mathcal{O}(1)}$ time. Is it possible to improve this running time? Similarly to ULAM METRIC k -CENTER, we find that this question is interesting even for $k = 1$. Is there a single-exponential (or even subexponential) in d algorithm?

References

- 1 Alok Aggarwal and Jeffrey S. Vitter. The input/output complexity of sorting and related problems. *Communications of the ACM*, 31(9):1116–1127, 1988. doi:10.1145/48529.48535.
- 2 Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):23:1–23:27, 2008. doi:10.1145/1411509.1411513.
- 3 Alexandr Andoni and Huy L. Nguyen. Near-optimal sublinear time algorithms for ulam distance. In *Proceedings of the ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pages 76–86. SIAM, 2010. doi:10.1137/1.9781611973075.8.
- 4 Kenneth J. Arrow. *Social Choice and Individual Values*. Wiley, 1951. URL: <http://www.jstor.org/stable/j.ctt1nqb90>.
- 5 Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, and Andreas Hofmeier. On the hardness of maximum rank aggregation problems. *Journal of Discrete Algorithms*, 31:2–13, 2015. doi:10.1016/j.jda.2014.10.002.

- 6 Manu Basavaraju, Fahad Panolan, Ashutosh Rai, M. S. Ramanujan, and Saket Saurabh. On the kernelization complexity of string problems. *Theoretical Computer Science*, 730:21–31, 2018. doi:10.1016/j.tcs.2018.03.024.
- 7 Nadja Betzler, Michael R. Fellows, Jiong Guo, Rolf Niedermeier, and Frances A. Rosamond. Fixed-parameter algorithms for kemeny rankings. *Theoretical Computer Science*, 410(45):4554–4570, 2009. Algorithmic Aspects in Information and Management. doi:10.1016/j.tcs.2009.08.033.
- 8 Nadja Betzler, Jiong Guo, and Rolf Niedermeier. Parameterized computational complexity of Dodgson and Young elections. *Information and Computation*, 208(2):165–177, 2010. doi:10.1016/j.ic.2009.10.001.
- 9 Nadja Betzler, Jiong Guo, Rolf Niedermeier, and Johannes Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science*, 410:5425–5442, 2009. doi:10.1016/j.tcs.2009.05.029.
- 10 Therese Biedl, Franz-Josef Brandenburg, and Xiaotie Deng. On the complexity of crossings in permutations. *Discrete Mathematics*, 309(7):1813–1823, 2009. doi:10.1016/j.disc.2007.12.088.
- 11 Leizhen Cai, Siu Man Chan, and Siu On Chan. Random separation: A new method for solving fixed-cardinality optimization problems. In *Second International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 4169 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2006. doi:10.1007/11847250_22.
- 12 Diptarka Chakraborty, Debarati Das, and Robert Krauthgamer. Approximating the median under the Ulam metric. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 761–775. SIAM, 2021. doi:10.1137/1.9781611976465.48.
- 13 Diptarka Chakraborty, Debarati Das, and Robert Krauthgamer. Clustering permutations: New techniques with streaming applications. In *Innovations in Theoretical Computer Science Conference (ITCS)*, 2023. doi:10.4230/LIPIcs.ITCS.2023.31.
- 14 Diptarka Chakraborty, Syamantak Das, Arindam Khan, and Aditya Subramanian. Fair rank aggregation. In *NeurIPS*, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/974309ef51ebd89034adc64a57e304f2-Abstract-Conference.html.
- 15 Diptarka Chakraborty, Kshitij Gajjar, and Agastya Vibhuti Jha. Approximating the center ranking under ulam. In *Foundations of Software Technology and Theoretical Computer Science*, 2021. doi:10.4230/LIPIcs.FSTTCS.2021.12.
- 16 Ronald Christian, Michael R. Fellows, Frances Rosamond, and Arkadii Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11:217–224, 2007. doi:10.1007/s10058-007-0028-1.
- 17 Luís Cunha, Ignasi Sau, and Uéverton S. Souza. On the parameterized complexity of the median and closest problems under some permutation metrics. *Algorithms Mol. Biol.*, 19(1):24, 2024. doi:10.1186/s13015-024-00269-z.
- 18 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 19 Debarati Das and Amit Kumar. Breaking the two approximation barrier for various consensus clustering problems. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 323–372. SIAM, 2025. doi:10.1137/1.9781611978322.10.
- 20 Persi Diaconis and R. L. Graham. Spearman’s footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268, 1977. URL: <http://www.jstor.org/stable/2984804>.
- 21 Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web (WWW)*, pages 613–622. ACM, 2001. doi:10.1145/371920.372165.

- 22 Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 301–312. ACM, 2003. doi:10.1145/872757.872795.
- 23 Henning Fernau, Fedor V. Fomin, Daniel Lokshtanov, Matthias Mnich, Geevarghese Philip, and Saket Saurabh. Social choice meets graph drawing: How to get subexponential time algorithms for ranking and drawing problems. *Tsinghua Science and Technology*, 19(4):374–386, 2014. doi:10.1109/TST.2014.6867519.
- 24 Nick Fischer, Elazar Goldenberg, Mursalin Habib, and Karthik C. S. Hardness of Median and Center in the Ulam Metric. In Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman, editors, *33rd Annual European Symposium on Algorithms (ESA 2025)*, volume 351 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 111:1–111:17, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2025.111.
- 25 Jens Gramm, Rolf Niedermeier, and Peter Rossmanith. Fixed-parameter algorithms for CLOSEST STRING and related problems. *Algorithmica*, 37(1):25–42, 2003. doi:10.1007/s00453-003-1028-3.
- 26 Ragesh Jaiswal, Amit Kumar, and Jatin Yadav. Robust-Sorting and Applications to Ulam-Median. In Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis, editors, *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, volume 334 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 100:1–100:19, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2025.100.
- 27 Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 95–103. ACM, 2007. doi:10.1145/1250790.1250806.
- 28 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *Foundations of Computer Science (FOCS)*, pages 182–191. IEEE Computer Society, 1995. doi:10.1109/SFCS.1995.492475.
- 29 Timothy Naumovitz, Michael E. Saks, and C. Seshadhri. Accurate and nearly optimal sublinear approximations to ulam distance. In *Proceedings of the ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pages 2012–2031. SIAM, 2017. doi:10.1137/1.9781611974782.131.
- 30 Pavel A. Pevzner. *Computational molecular biology: An algorithmic approach*. MIT Press, 2000.
- 31 Krzysztof Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *J. Comput. Syst. Sci.*, 67(4):757–771, 2003. doi:10.1016/S0022-0000(03)00078-3.
- 32 Svatopluk Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 015(2):307–309, 1974. URL: <http://eudml.org/doc/16622>.
- 33 Anke van Zuylen and David P. Williamson. Deterministic algorithms for rank aggregation and other ranking and clustering problems. In *Approximation and Online Algorithms*, pages 260–273. Springer, 2007. doi:10.1007/978-3-540-77918-6_21.