The Computational Hardness of Estimating Edit Distance
[Extended Abstract]

Alexandr Andoni*
MIT
andoni@mit.edu

Robert Krauthgamer
Weizmann Institute and IBM Almaden
robert.krauthgamer@weizmann.ac.il

Abstract

We prove the first non-trivial communication complexity lower bound for the problem of estimating the edit distance (aka Levenshtein distance) between two strings. A major feature of our result is that it provides the first setting in which the complexity of computing the edit distance is provably larger than that of Hamming distance.

Our lower bound exhibits a trade-off between approximation and communication, asserting, for example, that protocols with $O(1)$ bits of communication can only obtain approximation $\alpha \geq \Omega(\log d / \log\log d)$, where d is the length of the input strings. This case of $O(1)$ communication is of particular importance, since it captures constant-size sketches as well as embeddings into spaces like L_1 and squared-L_2, two prevailing algorithmic approaches for dealing with edit distance. Furthermore, the bound holds not only for strings over alphabet $\Sigma = \{0, 1\}$, but also for strings that are permutations (called the Ulam metric).

Besides being applicable to a much richer class of algorithms than all previous results, our bounds are near-tight in at least one case, namely of embedding permutations into L_1. The proof uses a new technique, that relies on Fourier analysis in a rather elementary way.

1 Introduction

The *edit distance* (aka *Levenshtein distance*) between two strings is the number of insertions, deletions, and substitutions needed to transform one string into the other. This distance is of key importance in several fields such as computational biology and text processing, and consequently computational problems involving the edit distance were studied quite extensively. The most basic problem is that of computing the edit distance between two strings of length d over alphabet Σ. The fastest algorithm known for the case of constant-size alphabet remains the algorithm of Masek and Paterson [20] from 1980, that runs in time $O(d^2/\log d)$. Unfortunately, such near-quadratic time is prohibitive when working on large datasets, which is a common case in areas such as computational biology, and a possible approach is to trade accuracy for speed and employ faster algorithms that compute the edit distance approximately (possibly as a preliminary filtering step). Currently, the best quasi-linear time algorithm, due to Batu, Ergün, and Sahinalp [6], achieves $d^{1/3+o(1)}$ approximation.

Another major algorithmic challenge is to design a scheme for Nearest Neighbor Search (NNS) under the edit distance. In this problem, we wish to design a data structure that preprocesses a dataset of n strings of length d each, so that when a query string is given, the query's nearest neighbor (i.e., a dataset string with the smallest edit distance to the query string) can be reported quickly. However, no efficient solutions for this problem are known, even if one is content with a small approximation. All known algorithms with fast query time (polynomial in d and $\log n$) either require large space or have large approximation – Indyk [14] achieves constant approximation using $n^{d^{\Omega(1)}}$ space, and Ostrovsky and Rabani [23] obtain $2^{O(\sqrt{\log d \log\log d})}$ approximation using space that is polynomial in d and n.

It is thus natural to ask: is it really "hard" to design algorithms for the edit distance? A natural benchmark is the Hamming distance, in which substitutions are allowed, but not insertions and deletions. For Hamming distance, much better algorithms are known: (i) the distance between two strings can clearly be computed in time $O(d)$, and (ii) NNS schemes by Indyk and Motwani [15] and by Kushilevitz, Ostrovsky and Rabani [19] achieve $1+\epsilon$ approximation using space

*Part of this work was done while the author was visiting IBM Almaden.

that is polynomial in d and in n^{1/ϵ^2}. Empirically, edit distance appears to be more difficult than Hamming distance, and the reason is quite clear – insertions and deletions cause portions of the string to move and create an alignment problem – but there is no rigorous evidence that supports this intuition. In particular, we are not aware of a computational model in which the complexity of edit distance is provably larger than that of Hamming distance.

We give the first rigorous evidence for the *computational* hardness of the edit distance. In fact, we show a computational model in which the complexity of estimating edit distance is significantly larger than that of Hamming distance, and this is the first setting for such a separation. Our results hold for two important metrics:

1. *standard edit metric*, i.e. edit distance on $\{0,1\}^d$;
2. the *Ulam metric*, which is the edit distance on permutations of length d.

Here and throughout, a *permutation* is a string consisting of distinct characters. Our results immediately imply lower bounds for sketching algorithms and nonembeddability statements, areas that received a lot of attention; we will discuss these implications in more detail after stating our main results.

1.1 Main Results

Our main result is stated in terms of communication complexity of the *distance threshold estimation problem (DTEP)* and holds for both edit metric over $\Sigma = \{0,1\}$, and for the Ulam metric. In DTEP [26], for a threshold R and an approximation α fixed as parameters, we are given inputs x, y and we want to decide whether $\mathrm{ed}(x,y) > R$ or $\mathrm{ed}(x,y) \le R/\alpha$.

In the communication protocols setting, Alice and Bob, who have access to a common source of randomness, receive strings x and y respectively as their inputs, and their goal is to solve DTEP by exchanging messages. The communication complexity of the protocol is then defined as the minimum number of bits Alice and Bob need to exchange, to succeed with probability at least $2/3$. When x, y come from the standard edit metric, we denote the communication complexity by $\mathrm{CC}^{\{0,1\}}_{\alpha,R}$. Similarly, when x, y come from the Ulam metric, we denote the communication complexity by $\mathrm{CC}^{\mathrm{Ulam}}_{\alpha,R}$. Our main theorem provides a lower bound on the latter, exhibiting a trade-off between communication and approximation.

Theorem 1.1. *Let $d > 1$ and $\alpha = \alpha(d) > 1$. Then, there exist constants $c > 0$ and $0 < c_1 < c_2 < 1$, such that for all R satisfying $d^{c_1} \le R \le d^{c_2}$,*

$$c \cdot \mathrm{CC}^{\mathrm{Ulam}}_{\alpha,R} + \log(\alpha \log \alpha) \ge \log \log d.$$

We extend this result from the Ulam metric to the standard edit metric by reducing the latter to the former. The key idea, which may be of independent interest, is that substituting every alphabet symbol with an independent random bit preserves the edit distance, up to a constant factor (with high probability), as stated in the following theorem.

Theorem 1.2. *Let $P, Q \in \Sigma^d$ be two permutations, and let $\pi : \Sigma \mapsto \{0,1\}$ be a random function. Then*

$$\Pr[\mathrm{ed}(\pi(P), \pi(Q)) \le \mathrm{ed}(P,Q)] = 1; \text{ and}$$
$$\Pr[\mathrm{ed}(\pi(P), \pi(Q)) \ge \Omega(1) \cdot \mathrm{ed}(P,Q)] \ge 1 - 2^{-\Omega(\mathrm{ed}(P,Q))}.$$

Using the last two theorems, we obtain the following.

Corollary 1.3. *Let $d > 1$ and $\alpha = \alpha(d) > 1$. Then, there exist constants $c > 0$ and $0 < c_1 < c_2 < 1$, such that for all R satisfying $d^{c_1} \le R \le d^{c_2}$,*

$$c \cdot \mathrm{CC}^{\{0,1\}}_{\alpha,R} + \log(\alpha \log \alpha) \ge \log \log d.$$

The only lower bounds known previously for $\mathrm{CC}^{\{0,1\}}_{\alpha,R}$ and $\mathrm{CC}^{\mathrm{Ulam}}_{\alpha,R}$ are obtained by a straightforward reduction from the same problem on Hamming metric. These bounds assert that the communication complexity for $\alpha = 1 + \epsilon$ is $\Omega(1/\epsilon)$, and in the case of sketching protocols $\Omega(1/\epsilon^2)$ [27], and both are clearly uninformative for (say) $\alpha \ge 2$. See also [25] for other related results.

Comparison with Hamming distance. The next proposition, proved (implicitly) by Kushilevitz, Ostrovsky, and Rabani [19], upper bounds the communication complexity of DTEP over the Hamming metric. Let $\mathrm{H}(x,y)$ be the Hamming distance between x and y.

Proposition 1.4 ([19])**.** *Let $d > 1$, $R > 1$ and $\epsilon > 0$. Then there exists a communication protocol (in fact, a sketching algorithm) that given inputs $x, y \in \Sigma^d$ distinguishes whether $\mathrm{H}(x,y) > R$ or $\mathrm{H}(x,y) \le R/(1+\epsilon)$, using $O(1/\epsilon^2)$ communication.*

Observe that for a constant approximation factor α (namely, independent of d), the complexity of the Hamming metric is $O(1)$, while that of edit metric is $\Omega(\log \log d)$. It thus follows that edit distance is indeed provably harder to compute than Hamming, for communication protocols.

1.2 Implications and Related Work

Two promising approaches to designing algorithms for the edit metrics are via metric embeddings and via sketching, and our results preclude good approximation algorithms obtained via either of these approaches.

Embedding of edit distance into normed metrics. A current line of attack on edit distance is by embedding it into a computationally easier metric, for which efficient algorithms are known. An *embedding* is a mapping f from the strings into, say, ℓ_1 metric, such that for all strings x, y,

$$\mathrm{ed}(x,y) \le \|f(x) - f(y)\|_1 \le D \cdot \mathrm{ed}(x,y),$$

and $D \ge 1$ is called the embedding's *distortion* (approximation factor). An embedding with low distortion would have major consequences since it allows porting a host of existing algorithms for ℓ_1 metric to the case of edit distance. For example, an (efficiently computable) embedding with distortion D gives an efficient nearest neighbor data structure for approximation (say) $2D$, by applying the embedding and reverting to [15, 19].

Naturally, researchers were keen to find the least distortion for an embedding into ℓ_1 – the problem is cited in Matoušek's list of open problems [21], as well as in Indyk's survey [13]. Table 1.2 summarizes the previously known upper and lower bounds, as well as the implications of our theorems. The reader may find more background, including on some variations of the edit distance, in [24].

It is readily seen from the table that the only previous super-constant distortion lower bound is $\Omega(\log d)$ for embedding of edit distance into ℓ_1, due to Krauthgamer and Rabani [18], building on a technique of Khot and Naor [17]. Although an important lower bound, one can potentially overcome such a lower bound by, say, embedding edit distance into a richer space, such as $(\ell_2)^2$, the square of ℓ_2, with a possibly smaller distortion – the major implications of an embedding into $(\ell_2)^2$ are precisely the same as when embedding into ℓ_1. Unfortunately, on this front, much weaker lower bounds are known – the previous lower bound is only 3/2 [1]. To further stress how little is known, we note that one can consider even richer metrics, such as any fixed power of ℓ_2 (essentially equivalent to embedding a fixed root of edit distance into ℓ_2), which also has an efficient nearest neighbor data structure. For sufficiently high (but fixed) power of ℓ_2, the 3/2 bound of [1] gets weaker and becomes arbitrarily close to 1.

Our results rule out all such embeddings indirectly, by targeting a richer class of metrics – metrics that admit protocols with $O(1)$ communication and $O(1)$ approximation. (Proposition 1.4 shows that this class of metrics is indeed richer.) Observe that every embedding of edit distance (either one of the two mentioned earlier) into a metric in that richer class must incur distortion $D \ge \Omega(\frac{\log d}{\log\log d})$, as otherwise it would contradict our communication lower bounds. (Note that the embedding does not have to be efficiently computable.)

Corollary 1.5. *For every $p \ge 1$, embedding the standard edit metric or the Ulam metric into $(\ell_2)^p$ requires distortion $\Omega\left(\frac{\log d}{\log\log d}\right)$. The same is true also for embedding into ℓ_1.*

For the Ulam metric, this distortion lower bound is near-optimal, since the metric embeds into ℓ_1 with $O(\log d)$ distortion [8]. The previous distortion lower bound was 4/3 [9].

Sketching of edit distance. The sketch of a string x is a (randomized) mapping of x into a short "fingerprint" $\mathbf{sk}(x)$, such that sketches of two strings, $\mathbf{sk}(x)$ and $\mathbf{sk}(y)$, are sufficient to distinguish between the case where x, y are at edit distance $\mathrm{ed}(x,y) \le R/\alpha$, and the case where $\mathrm{ed}(x,y) > R$, for approximation factor $\alpha > 1$ and parameter $R > 1$. The main parameter of a sketching algorithm is its *sketch size*, the length of $\mathbf{sk}(x)$.

The sketching model can also be described as a simultaneous communication protocol, as follows. Alice receives x and computes $\mathbf{sk}(x)$, Bob receives y and computes $\mathbf{sk}(y)$, and then they send their computed values to a "referee", who needs to decide whether x, y are close or far based only on the sketches. By letting either Alice or Bob play the role of the referee in this simultaneous protocol, one easily sees that the sketch size required by a sketching algorithm is always no smaller than the number of communication bits required by a (general) protocol. The following corollary thus follows immediately from our preceding communication lower bounds.

Corollary 1.6. *Every $O(1)$-size sketching algorithm of the standard edit metric or of the Ulam metric can only achieve approximation of $\Omega\left(\frac{\log d}{\log\log d}\right)$.*

Sketching with constant sketch size can be viewed as a generalization of the "embeddings approach" presented above, by using Proposition 1.4, albeit with a small constant factor loss in the approximation factor. An important observation is that this more general approach suffices for the purpose of designing an NNS scheme with efficient query time (assuming that computing the sketch can be done efficiently) and in

Metric	Reference	ℓ_1-embedding	$(\ell_2)^2$-embedding	$O(1)$-size sketch
Edit on $\{0,1\}^d$	[23]	$2^{O(\sqrt{\log d \log\log d})}$	$\longrightarrow$	$\longrightarrow$
	[17, 18]	$\Omega(\log d)$		
	[1]	$\longleftarrow$	$\geq 3/2$	
	This paper	$\longleftarrow$	$\longleftarrow$	$\Omega(\frac{\log d}{\log\log d})$
Ulam (edit on permutations)	[8]	$O(\log d)$	$\longrightarrow$	$\longrightarrow$
	[9]	$\longleftarrow$	$\geq 4/3$	
	This paper	$\longleftarrow$	$\longleftarrow$	$\Omega(\frac{\log d}{\log\log d})$
Block edit distance	[11, 22, 9]	$O(\log d \log^* d)$	$\longrightarrow$	$\longrightarrow$
Edit distance with moves	[10]	$O(\log d \log^* d)$	$\longrightarrow$	$\longrightarrow$

Figure 1: Known bounds on distortion of embedding variants of edit distance into ℓ_1, $(\ell_2)^2$, and the approximation for achieving $O(1)$-size sketch. Since $\ell_1 \subset (\ell_2)^2$ and $(\ell_2)^2$ has $O(1)$-size sketch for 2-approximation, the upper bounds transfer from left to right, and the lower bounds transfer from right to left (as suggested by the arrows). Grayed cell mean no implied result for the corresponding column.

polynomial space.[1] Indeed, the nearest neighbor data structure for Hamming metric of [19] could be viewed as an instantiation of the last step. In addition, sketching can be useful for the original goal of quickly computing the distance.

The sketching model is also important as a basic computational notion for massive data sets, and in recent years, an intensive research effort has led to many sketching algorithms for DTEP over different metrics. Prior to our work, there were essentially three metrics for which a sketch size lower bound is known: for ℓ_1 [27] (equivalently, for ℓ_p, $p \in (1,2]$), for ℓ_∞ [26, 4] (implying lower bounds for ℓ_p, $p > 2$), and for the Earth-mover distance over $\{0,1\}^d$ [2].

Sketching of edit distance was studied in [5, 3, 23, 8], but the only lower bound known for sketching of edit distance is trivial in the sense that it follows immediately from Hamming distance (by a straightforward reduction). This lower bound on the sketch size is $\Omega(1/\epsilon^2)$ for approximation $\alpha = 1+\epsilon$ [27], which becomes uninformative for even a 2-approximation. In fact, Bar-Yossef et al. [3] write that "The state of affairs indicates that proving sketching lower bounds for edit distance may be quite hard."

[1] In particular, one can first amplify the sketching's probability of success to $1 - n^{-\Omega(1)}$, where n is the number of points in the dataset, using sketch size $O(\log n)$. Then, the data structure pre-indexes all possible sketches in the amplified protocol, using only $2^{O(\log n)} = n^{O(1)}$ space. For each possible value of the amplified sketch, data structure stores the answer that the sketching referee would conclude from the sketches of the query and each dataset point. Note that, in fact, s-sized sketch imply $n^{O(s)}$-size NN data structure.

1.3 Our Techniques

Our proof of Theorem 1.1 consists of three steps. Generally speaking, we design two input distributions: $\tilde{\mu}_0$ over "far" pairs (x,y) (i.e. $\mathrm{ed}(x,y) > R$), and $\tilde{\mu}_1$ over "close" pairs (i.e. $\mathrm{ed}(x,y) \leq R/\alpha$). The goal then becomes to show that these distributions are indistinguishable by protocols with low communication complexity. By Yao's minimax principle, it suffices to consider deterministic protocols.

The first step reduces the problem to proving that the two distributions $\tilde{\mu}_0, \tilde{\mu}_1$ are indistinguishable by boolean functions over $\mathbb{Z}_p^d$. Roughly speaking, we show that if there is a protocol using at most l bits of communication, then there exists a (deterministic) sketching protocol that uses sketch size of 1 bit and achieves an advantage of at least $2^{-O(l)}$ in distinguishing between the two distributions. Let $\mathcal{H}^A, \mathcal{H}^B : \mathbb{Z}_p^d \to \{-1,+1\}$ be the boolean functions that Alice and Bob, respectively, use as their sketch functions. We can then further restrict the sketching protocol such that the referee decides by checking whether $\mathcal{H}^A(x) = \mathcal{H}^B(y)$ or not. This step follows the approach employed earlier in [2], with some minor technical differences.

The second step's main goal is to further characterize the advantage achieved by $\mathcal{H}^A, \mathcal{H}^B$ in terms of a carefully crafted measure of statistical distance between the two input distributions $\tilde{\mu}_0, \tilde{\mu}_1$. For this approach to be effective, it is important that the functions $\mathcal{H}^A, \mathcal{H}^B$ depend only on a few coordinates, and in order to guarantee this (indirectly), we include in $\tilde{\mu}_0, \tilde{\mu}_1$ a random noise component, which effectively destroys any dependence on many coordinates. Specifically, this step assumes that, each distribution $\tilde{\mu}_t$, $t \in \{0,1\}$, has the following structure: choose $x \in \mathbb{Z}_p^d$ uniformly at

random, and then generate y from x via a sequence of two randomized operations. The first operation is a random noise with rate $\rho > 0$, i.e., each coordinate is modified independently with probability $1-\rho$ into a randomly chosen value. The second operation permutes the coordinates according to a permutation drawn from a distribution $\mathcal{D}_t$. Given this $\mathcal{D}_t$, consider the following derived distribution: take a vector $u \in \mathbb{Z}_p^d$ with λ non-zero positions (called a λ-*test*) and apply a random permutation $\pi \in \mathcal{D}_t$ to it; let $A_u^{(t,\lambda)}$ be the resulting distribution of vectors. (Note that the support of $A_u^{(t,\lambda)}$ contains only vectors with precisely λ non-zero entries.) Our measure Δ_λ, *called λ-test distinguishability*, is the maximum, over all such λ-tests u, of the total variation distance between $A_u^{(0,\lambda)}$ and $A_u^{(1,\lambda)}$. It captures to what extent one can distinguish $\mathcal{D}_0$ from $\mathcal{D}_1$ (and thus $\tilde{\mu}_0$ from $\tilde{\mu}_1$) by inspecting only λ positions of y. Our upper bound on the advantage achieved by $\mathcal{H}^A, \mathcal{H}^B$ takes roots in the following dichotomy. If $\mathcal{H}^B$ essentially depends on many coordinates of y (e.g., a linear function with many terms), then the advantage is bounded by ρ^λ (i.e., the random noise N_ρ destroys almost all the information), and if $\mathcal{H}^B$ essentially depends on a few, say λ, coordinates, then the advantage is bounded by Δ_λ. To prove this dichotomy, we rely on Fourier analysis which expands $\mathcal{H}^A, \mathcal{H}^B$ into linear functions at different levels λ.

In the third step, we complete the description of μ_0, μ_1 by detailing the construction of $\mathcal{D}_0, \mathcal{D}_1$, and give an upper bound the λ-test distinguishability Δ_λ for these distributions. In a simplified view, each distribution $\mathcal{D}_t$ is generated by a block rotation operation, namely, choosing a random block of length L and applying to it $\epsilon_t L$ cyclic shifts. The difference between the two distributions is in the magnitude of the rotation (namely, ϵ_t).

Our use of Fourier analysis is elementary, and does not involve the KKL theorem [16] or Bourgain's noise sensitivity theorem [7], which were used in the previous non-embeddability results for edit distance [17, 18]. We also note that our hard distribution is notably different from the distributions of [18] or [17], which do admit efficient communication protocols.

To prove Theorem 1.2, we give a new characterization of the Ulam distance between two strings. In particular, building on the work of [12, 25], we prove that if two strings (permutations) P, Q are at distance $k = \mathrm{ed}(P,Q)$, then there exist $\Theta(k)$ pairs of characters in P, all characters at distinct positions, such that for each pair (a,b), their order in P is opposite to that in Q (if they appear in Q at all). Once we have this characterization, a careful counting of the number of the possible alignments between P and Q finishes the proof of the theorem.

Organization. The proof of Theorem 1.1 appears in Section 3. Due to space constraints, for some lemmas we give only proof sketches. The proof of Theorem 1.2 is deferred to the full version of the paper.

2 Preliminaries

We use notation $[d] = \{1, 2 \dots d\}$, and $\mathbb{Z}_p = \{0, 1 \dots p-1\}$. For a vector $u \in \mathbb{Z}_p^d$, define the *weight* of u, denoted $\mathrm{wt}(u)$, to be the number of coordinates in u that are non-zero.

Definition 2.1. *For matrix $A \in M_{n,n}(\mathbb{R})$ and $p \in [1, \infty]$, the p-*norm *of A is defined by* $\|A\|_p = \max\{\|Av\|_p : \ v \in \mathbb{C}^n, \|v\|_p = 1\}$.

2.1 Fourier Analysis over $\mathbb{Z}_p^d$

We review basic Fourier Analysis over $\mathbb{Z}_p^d$ for a prime $p \geq 2$.

The collection of functions $f : \mathbb{Z}_p^d \to \mathbb{C}$ is a vector space of dimension p^d, equipped with an inner product given by $\langle f, g \rangle = \mathbb{E}_{x \in \mathbb{Z}_p^d}\left[f(x) \cdot \overline{g(x)}\right]$. For $u \in \mathbb{Z}_p^d$, define a character $\chi_u(x) = e^{\frac{2\pi i}{p}(x \cdot u)}$, where $x \cdot u$ is the scalar product of $x, u \in \mathbb{Z}_p^d$. The set of characters $\{\chi_u \mid u \in \mathbb{Z}_p^d\}$ forms an orthonormal basis, called the Fourier basis. Thus every function $f : \mathbb{Z}_p^d \to \mathbb{C}$ admits a Fourier expansion $f = \sum_{u \in \mathbb{Z}_p^d} \hat{f}_u \chi_u$, where $\hat{f}_u = \langle f, \chi_u \rangle$ is called the Fourier coefficient of f corresponding to u. Parseval's equality states that $\mathbb{E}_{x \in \mathbb{Z}_p^d}\left[f(x)\overline{g(x)}\right] = \sum_{u \in \mathbb{Z}_p^d} \hat{f}_u \overline{\hat{g}_u}$.

We let N_ρ stand for a *random noise* vector over in $\mathbb{Z}_p^d$, namely, a vector where each coordinate is set independently at random as follows: with probability ρ it is set to zero, and with probability $1-\rho$ it is set to a random value from $\mathbb{Z}_p$.

The *noise operator* T_ρ (also called Bonami-Beckner operator) operates on functions $f : \mathbb{Z}_p^d \to \mathbb{R}$, and is defined by $(T_\rho f)(x) = \mathbb{E}_{N_\rho}[f(x + N_\rho)]$. The following standard fact relates the Fourier coefficients of f with those of $T_\rho f$.

Fact 2.2. *For every vector $u \in \mathbb{Z}_p^d$, $\widehat{(T_\rho f)}_u = \hat{f}_u \cdot \rho^{\mathrm{wt}(u)}$.*

Note that, for $p = 2$, i.e. Fourier expansion over $\{0,1\}^d$, this is equivalent to having $\widehat{(T_\rho f)}_S = \hat{f}_S \rho^{|S|}$ for every $S \subseteq [d]$.

2.2 Edit metric and Ulam metric

Let Σ be the alphabet; we mostly consider $\Sigma = \{0,1\}$ or $\Sigma = \mathbb{Z}_p = \{0,1,\dots p-1\}$ for $p \in \mathbb{N}$. For $x \in \Sigma^d$, we let x_i denote the i^{th} position in x whenever $i \in [d]$, and extend the notation to $i \notin [d]$ by defining $x_i = x_j$ where $i \equiv j \pmod d$ and $j \in [d]$.

Definition 2.3 (Edit metrics)**.** *Let $d > 0$. The* edit metric over Σ *is the space Σ^d endowed with distance function* $\mathrm{ed}(x,y)$*, which equals to the minimum number of character substitutions/insertions/deletions to transform x into y.*

When $|\Sigma| \geq d$, we call Ulam metric *the space over permutations $x \in \Sigma^d$, where x is called a permutation if no symbol $c \in \Sigma$ appears more than once in x. The space is endowed with the same distance function* $\mathrm{ed}(x,y)$*.*

We will also use the following operation on strings.

Definition 2.4 (Rotation operations)**.** *Fix $d > 1$ and an alphabet Σ. For $s, L \in [d]$, define the* right rotation operation $\overrightarrow{R}_{s,L} : \Sigma^d \to \Sigma^d$ *as follows. When applied to a string x, it takes the substring of x of length L starting at position s (with wrap-around) and performs on it a cyclic shift to the right (by 1 position); the rest of x remains unchanged. A left rotation $\overleftarrow{R}_{s,L}$ is defined similarly.*

L is called the length of the rotation operation.

For example, $\overrightarrow{R}_{j,2}$ swaps positions j and $j+1$ in x. Note that $\overrightarrow{R}_{s,L}$ works as a permutation (i.e. it is a bijection). Also, for $i \in [L]$, $(\overrightarrow{R}_{s,L})^i$ is a rotation of the same block but by i positions to the right. Note that a rotation operation can be simulated by at most two deletions and two insertions (and only one of each when the rotation block does not wrap-around).

3 Proof of the Main Theorem

In this section we prove Theorem 1.1. Fix the values of d and R, and let us use the alphabet $\Sigma = \mathbb{Z}_p$ for p being the smallest prime greater than d^3. For the rest of this section, define $\tilde{\mu}$ (our hard distribution) as $\tilde{\mu} = \frac{\tilde{\mu}_0 + \tilde{\mu}_1}{2}$, where $\tilde{\mu}_0$ will be a distribution over *far* pairs of points (x,y) and $\tilde{\mu}_1$ will be a distribution over *close* pairs (x,y), i.e., $\mathrm{ed}(x,y) > R$ and $\mathrm{ed}(x,y) \leq R/\alpha$, respectively.

We will follow the steps outlined in Section 1.3, and eventually put all the pieces together in Section 3.5. It is worth noting that the definition of the hard distribution $\tilde{\mu}$ is quite technical, and that we will mention (and use) a few simple properties of it even before specifying it in full detail in Section 3.3.

3.1 Reduction to Boolean Functions

Our first lemma says that if there is an efficient communication protocol, then there are boolean functions with a non-negligible advantage in distinguishing the distribution $\tilde{\mu}_0$ from $\tilde{\mu}_1$. This lemma is based on the ideas from [2], although the current proof is simpler than in [2].

Lemma 3.1. *If $\mathrm{CC}^{\mathrm{Ulam}}_{\alpha,R} \leq l$ for some $l \geq 1$, then there exist boolean functions $\mathcal{H}^A, \mathcal{H}^B : \mathbb{Z}_p^d \to \{-1,+1\}$, such that*

$$\Pr_{\tilde{\mu}_0}[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)] - \Pr_{\tilde{\mu}_1}[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)] \geq 2^{-O(l)}.$$

Proof sketch. The main idea is to reduce the communication to a simultaneous (i.e. sketching) protocol where Alice and Bob each send a sketch of one bit only, and the referee performs an equality test on these two bits. Then, using Yao's minimax principle, we easily obtain two deterministic boolean functions $\mathcal{H}^A$ and $\mathcal{H}^B$ that complete the proof.

To accomplish the reduction, assume an l-bit protocol and construct a one-bit sketching protocol as follows: Alice and Bob guess the entire transcript of length l using public coins (the guess is independent of the actual inputs). Each of them then checks whether the guessed transcript describes the messages they would send using the assumed l-bit protocol, using the guessed transcript to simulate the other party's messages. If the transcript turns out to be incompatible, they send a bit chosen independently at random. Otherwise, Alice always outputs 1, and Bob outputs the outcome of the guessed transcript. Observe that if the guessed transcript is not correct, then at least one of the two bits output by Alice and Bob is completely random. Thus, for inputs from $\tilde{\mu}_1$, Alice and Bob's bits are equal with probability at least $2^{-l} \cdot \frac{2}{3} + (1-2^{-l}) \cdot \frac{1}{2}$, and for inputs from $\tilde{\mu}_0$, that probability is at most $2^{-l} \cdot \frac{1}{3} + (1-2^{-l}) \cdot \frac{1}{2}$. □

The rest of the proof of the Theorem 1.1 focuses on these boolean functions $\mathcal{H}^A, \mathcal{H}^B$.

3.2 From Boolean Functions to λ-Tests

Next we provide a method to lower bound the advantage achieved by the boolean functions $\mathcal{H}^A, \mathcal{H}^B$, by relating it to a certain statistical property of the hard distribution $\tilde{\mu}$. The lemma below will use the general structure of the hard distribution $\tilde{\mu} = \frac{\tilde{\mu}_0 + \tilde{\mu}_1}{2}$, which we describe next. For each $t \in \{0,1\}$, the distribution $\tilde{\mu}_t$ will be formed via a small modification of another distribution μ_t, which has a structure that is easier for

analysis. We analyze below (in Lemma 3.4) the distributions μ_0 and μ_1, but eventually the total variation distance between $\tilde{\mu}_t$ and μ_t for each t will be shown to be extremely small, and the lemma will immediately extend to $\tilde{\mu}_0$,$\tilde{\mu}_1$ as well.

The distribution μ_t consists of pairs (x, y) chosen as follows: $x \in \mathbb{Z}_p^d$ is chosen uniformly at random, and y is constructed from x in two steps. In the first step, let $z \triangleq x + N_\rho$, where N_ρ is random noise of some rate $\rho \in (0,1)$ (that does not depend on t). In the second step, y is obtained from z by permuting the coordinates of z according to a distribution $\mathcal{D}_t$ over permutations. Formally, $\mathcal{D}_t$ is a distribution over permutation operations, where a permutation operation is a function $\pi : \mathbb{Z}_p^d \to \mathbb{Z}_p^d$ for which there exists a permutation $\hat{\pi}$ over $[d]$ such that $\pi(x) = (x_{\hat{\pi}(1)}, \ldots x_{\hat{\pi}(d)})$. We will require that $\mathcal{D}_t$ is *symmetric* in the sense that every two permutation operations π and π^{-1} are equi-probable (in it).

We next quantify the "difference" between the distributions $\mathcal{D}_0, \mathcal{D}_1$ from the perspective of what we call λ-*tests*. For $\lambda \in [d]$, we define a λ-*test* to be a vector $u \in \mathbb{Z}_p^d$ with precisely λ non-zero entries, i.e., $\mathrm{wt}(u) = \lambda$. For a distribution $\mathcal{D}_t$ and $\lambda \in [d]$, let the matrix $A^{(t,\lambda)}$ be the transition matrix a Markov chain whose states are all the λ-tests, and whose transitions are according to $\mathcal{D}_t$, i.e., at a λ-test u, the process picks $\pi \in \mathcal{D}_t$ and moves to state $\pi(u)$ (which is also a λ-test). In other words, a row u of $A^{(t,\lambda)}$ is a vector, that has, for every λ-test w, a coordinate of value $\Pr_{\pi \in \mathcal{D}_t}[w = \pi(u)]$. We denote this row by $A_u^{(t,\lambda)}$. Note that the matrix $A^{(t,\lambda)}$ is symmetric (since $\mathcal{D}_t$ is symmetric) and thus it is doubly-stochastic.

Definition 3.2. *The* λ-test distinguishability *of* $\mathcal{D}_0, \mathcal{D}_1$*, denoted* Δ_λ*, is the maximum, over all* λ*-tests* u*, of the total variation distance between the distributions* $A_u^{(0,\lambda)}$ *and* $A_u^{(1,\lambda)}$*.*

Fact 3.3. $\Delta_\lambda = \|A^{(0,\lambda)} - A^{(1,\lambda)}\|_\infty / 2$.

The following lemma bounds the advantage achieved by $\mathcal{H}^A, \mathcal{H}^B$ using the λ-test distinguishability Δ_λ.

Lemma 3.4. *Consider* $\mathcal{H}^A, \mathcal{H}^B : \mathbb{Z}_p^d \to \{-1, +1\}$ *and* $\rho \in (0,1)$*. If each* μ_t*, for* $t \in \{0,1\}$*, is defined as above from a symmetric distributions* $\mathcal{D}_t$ *over permutation operations, then*

$$\Pr_{\mu_0}[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)] - \Pr_{\mu_1}[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)] \leq \max_{\lambda \in [d]} \Delta_\lambda \rho^\lambda.$$

Proof. For $t \in \{0,1\}$, define $C^{(t)} \triangleq \mathbb{E}_{\mu_t}\left[\mathcal{H}^A(x)\mathcal{H}^B(y)\right]$ to be the *correlation* between the two boolean functions. Note that, $\mathbb{E}_{\mu_t}\left[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)\right] = \frac{1}{4}\mathbb{E}_{\mu_t}\left[(\mathcal{H}^A(x) - \mathcal{H}^B(y))^2\right] = \mathbb{E}_x\left[(\mathcal{H}^A(x))^2\right]/4 + \mathbb{E}_x\left[(\mathcal{H}^B(x))^2\right]/4 - C^{(t)}/2 = 1/2 - C^{(t)}/2$. Thus,

$$\Pr_{\mu_0}[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)] - \Pr_{\mu_1}[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)] = \frac{C^{(1)} - C^{(0)}}{2}.$$

We will show that $C^{(1)} - C^{(0)} \leq 2 \max_{\lambda \in [d]} \Delta_\lambda \rho^\lambda$. For this purpose, it is more convenient to express each $C^{(t)}$ in terms of the Fourier coefficients of $\mathcal{H}^A$ and $\mathcal{H}^B$. Recall that μ_t is generated by picking a random x, and constructing y from x using a random noise N_ρ and a random permutation drawn from $\mathcal{D}_t$, namely, $y = \pi(x + N_\rho)$, where $\pi \in \mathcal{D}_t$. Let $\mu_t|x$ denote the distribution μ_t conditioned on the value of x. Thus,

$$\mathbb{E}_{\mu_t}\left[\mathcal{H}^A(x)\mathcal{H}^B(y)\right] = \mathbb{E}_{x \in \mathbb{Z}_p^d}\left[\mathcal{H}^A(x) \cdot \mathbb{E}_{\mu_t|x}\left[\mathcal{H}^B(y)\right]\right]$$

Define $f^{(t)}(x) \triangleq \mathbb{E}_{\mu_t|x}\left[\mathcal{H}^B(y)\right]$. Then

$$f^{(t)}(x) = \mathbb{E}_{N_\rho}\left[\mathbb{E}_{\pi \in \mathcal{D}_t}\left[\mathcal{H}^B(\pi(x + N_\rho))\right]\right].$$

Since $C^{(t)} = \mathbb{E}_x\left[\mathcal{H}^A(x) f^{(t)}(x)\right]$, we can switch to the Fourier basis by applying Parseval's equality, and get

$$C^{(t)} = \sum_{u \in \mathbb{Z}_p^d} \widehat{(\mathcal{H}^A)}_u \overline{\widehat{(f^{(t)})}_u}, \tag{1}$$

where $\widehat{(\mathcal{H}^A)}_u$ and $\widehat{(f^{(t)})}_u$ are the Fourier coefficients of $\mathcal{H}^A$ and $f^{(t)}$ respectively.

The next proposition, which we shall prove shortly, expresses the level λ Fourier coefficients of $f^{(t)}$ in terms of those of $\mathcal{H}^B$. Let $\left(\widehat{(f^{(t)})}_u\right)_{u:\mathrm{wt}(u)=\lambda}$ be the vector of the Fourier coefficients of $f^{(t)}$ indexed by u's of weight $\mathrm{wt}(u) = \lambda$. Define $\left(\widehat{(\mathcal{H}^B)}_u\right)_{u:\mathrm{wt}(u)=\lambda}$ similarly.

Proposition 3.5. *For all* $\lambda \in [d]$ *and* $\mathcal{H}^B : \mathbb{Z}_p^d \to \mathbb{C}$*,*

$$\left(\widehat{(f^{(t)})}_u\right)_{u:\mathrm{wt}(u)=\lambda} = \rho^\lambda A^{(t,\lambda)} \cdot \left(\widehat{(\mathcal{H}^B)}_u\right)_{u:\mathrm{wt}(u)=\lambda} \tag{2}$$

This proposition naturally leads us to break each $C^{(t)}$ into the terms corresponding to each Fourier level λ. Define the λ^{th}*-correlation* to be

$$C_\lambda^{(t)} \triangleq \sum_{u \in \mathbb{Z}_p^d : \mathrm{wt}(u)=\lambda} \widehat{(\mathcal{H}^A)}_u \overline{\widehat{(f^{(t)})}_u}. \tag{3}$$

Then, $C^{(1)} - C^{(0)} = \sum_{\lambda=0}^d \left(C_\lambda^{(1)} - C_\lambda^{(0)}\right)$. We can now bound each $C_\lambda^{(1)} - C_\lambda^{(0)}$ in terms of Δ_λ and ρ.

Let $\omega_\lambda^A = \left\| \left(\widehat{(\mathcal{H}^A)}_u\right)_{u:\mathrm{wt}(u)=\lambda} \right\|_2$ be the ℓ_2-weight of the level λ Fourier coefficients of $\mathcal{H}^A$, and define similarly ω_λ^B. By Parseval's identity, $\sum_{\lambda=0}^d \left(\omega_\lambda^A\right)^2 = \mathbb{E}_x\left[\mathcal{H}^A(x) \cdot \overline{\mathcal{H}^A(x)}\right] = 1$, and similarly $\sum_{\lambda=0}^d \left(\omega_\lambda^B\right)^2 = 1$.

Proposition 3.6. *For all $\lambda \in [d]$,*

$$C_\lambda^{(1)} - C_\lambda^{(0)} \leq 2\Delta_\lambda \rho^\lambda \cdot \omega_\lambda^A \omega_\lambda^B.$$

We will prove the proposition shortly by a straightforward calculation. Given this proposition, we have

$$C^{(1)} - C^{(0)} = \sum_{\lambda=0}^{d} \left(C_\lambda^{(1)} - C_\lambda^{(0)} \right) \leq \sum_{\lambda=1}^{d} 2\Delta_\lambda \rho^\lambda \cdot \omega_\lambda^A \omega_\lambda^B$$

$$\leq \sum_{\lambda=1}^{d} 2\Delta_\lambda \rho^\lambda \cdot \frac{\left(\omega_\lambda^A\right)^2 + \left(\omega_\lambda^B\right)^2}{2} \leq 2 \max_{\lambda \in [d]} \Delta_\lambda \rho^\lambda.$$

thereby proving Lemma 3.4. □

It remains to prove Propositions 3.5 and 3.6.

Proof of Proposition 3.5. Define a new function $g^{(t)} : \mathbb{Z}_p^d \to \mathbb{R}$ as

$$g^{(t)}(z) \triangleq \mathbb{E}_{\pi \in \mathcal{D}_t} \left[\mathcal{H}^B(\pi(z)) \right].$$

Then $f^{(t)} = T_\rho g^{(t)}$, and thus $\widehat{\left(f^{(t)}\right)}_u = \widehat{\left(g^{(t)}\right)}_u \cdot \rho^{\mathrm{wt}(u)}$ for all $u \in \mathbb{Z}_p^d$ (by Fact 2.2). It remains to prove that

$$\left(\widehat{\left(g^{(t)}\right)}_u \right)_{u:\mathrm{wt}(u)=\lambda} = A^{(t,\lambda)} \cdot \left(\widehat{\left(\mathcal{H}^B\right)}_u \right)_{u:\mathrm{wt}(u)=\lambda} \quad (4)$$

Similarly to the operator T_ρ, we define the operator $\mathcal{O}_t$ as $(\mathcal{O}_t \mathcal{H}^B)(x) \triangleq \mathbb{E}_{\pi \in \mathcal{D}_t} \left[\mathcal{H}^B(\pi(x)) \right]$. Since $g^{(t)} = \mathcal{O}_t \mathcal{H}^B$, we proceed to analyze how the operator $\mathcal{O}_t$ works on the Fourier coefficients of a function $\mathcal{H}^B$.

Fact 3.7. *For a permutation operation π, define $\mathcal{P}_\pi$ to be an operator on functions $\psi : \mathbb{Z}_p^d \to \mathbb{R}$, given by $(\mathcal{P}_\pi \psi)(x) \triangleq \psi(\pi(x))$. Then, $\widehat{(\mathcal{P}_\pi \psi)}_u = \hat{\psi}_{\pi(u)}$.*

Now, the operator $\mathcal{O}_t$ defined earlier is simply a convex combination of several $\mathcal{P}_\pi$, where π is drawn from $\mathcal{D}_t$. Thus, with the above fact, for every $u \in \mathbb{Z}_p^d$,

$$\widehat{\left(g^{(t)}\right)}_u = \widehat{(\mathcal{O}_t \mathcal{H}^B)}_u = \mathbb{E}_{\pi \in \mathcal{D}_t} \left[\widehat{\left(\mathcal{H}^B\right)}_{\pi(u)} \right]. \quad (5)$$

Consequently, the vector of level λ Fourier coefficients of $g^{(t)}$ can be written as a product of the matrix $A^{(t,\lambda)}$ and the vector of the (same) level λ Fourier coefficients of $\mathcal{H}^B$, which proves Proposition 3.5. □

We will need the following fact for the proof of Proposition 3.6.

Fact 3.8. *Let $B \in M_{n,n}(\mathbb{R})$ be a symmetric matrix. Then, $\|B\|_2 \leq \|B\|_\infty$.*

Proof. It is known that $\|B\|_1 = \max_{j \in [n]} \sum_{i \in n} |B_{ij}|$ and $\|B\|_\infty = \max_{i \in [n]} \sum_{j \in [n]} |B_{ij}|$, and since B is symmetric, these two norms are equal. By Riesz-Thorin interpolation theorem, $\|B\|_2 \leq \|B\|_\infty$. □

Proof of Proposition 3.6. For every λ, the matrix $A^{(t,\lambda)}$ is symmetric, and so is $A^{(1,\lambda)} - A^{(0,\lambda)}$. Thus,

$$\begin{aligned}
C_\lambda^{(1)} - C_\lambda^{(0)} &= \\
&= \sum_{u \in \mathbb{Z}_p^d : \mathrm{wt}(u)=\lambda} \widehat{\left(\mathcal{H}^A\right)}_u \cdot \left(\overline{\widehat{\left(f^{(1)}\right)}}_u - \overline{\widehat{\left(f^{(0)}\right)}}_u \right) \\
&\leq \left\| \left(\widehat{\left(\mathcal{H}^A\right)}_u \right)_{u:\mathrm{wt}(u)=\lambda} \right\|_2 \cdot \left\| \left(\overline{\widehat{\left(f^{(1)}\right)}}_u - \overline{\widehat{\left(f^{(0)}\right)}}_u \right)_{u:\mathrm{wt}(u)=\lambda} \right\|_2 \\
&= \omega_\lambda^A \cdot \left\| \rho^\lambda \left(A^{(1,\lambda)} - A^{(0,\lambda)} \right) \left(\overline{\widehat{\left(\mathcal{H}^B\right)}}_u \right)_{u:\mathrm{wt}(u)=\lambda} \right\|_2 \\
&\leq \rho^\lambda \cdot \omega_\lambda^A \cdot \left\| A^{(1,\lambda)} - A^{(0,\lambda)} \right\|_2 \left\| \left(\widehat{\left(\mathcal{H}^B\right)}_u \right)_{u:\mathrm{wt}(u)=\lambda} \right\|_2 \\
&\leq \rho^\lambda \cdot \omega_\lambda^A \omega_\lambda^B \cdot \left\| A^{(1,\lambda)} - A^{(0,\lambda)} \right\|_\infty \\
&= 2\Delta_\lambda \cdot \rho^\lambda \cdot \omega_\lambda^A \omega_\lambda^B;
\end{aligned}$$

where we used (3), Cauchy-Schwarz, Proposition 3.5, Definition 2.1, Fact 3.8, and Fact 3.3, respectively. □

3.3 The Hard Distribution

Our hard distribution construction follows the general outline given in Section 3.2. In particular, we first define the intermediary distributions μ_0 and μ_1, for which we need to specify the value of ρ and the distributions $\mathcal{D}_0, \mathcal{D}_1$ over permutation operators. The description of the latter will form the bulk of the construction. Also, we will describe how to finally construct $\tilde{\mu}_t$ from μ_t, for each $t \in \{0, 1\}$.

Fix $\epsilon_0 \triangleq 1/2$ and select $\epsilon_1 = \Theta(\frac{1}{\alpha})$ as follows. Let $\beta \triangleq \frac{1-\epsilon_1}{1-\epsilon_0} = 2(1-\epsilon_1)$, and $\xi_1 \triangleq \lceil \log_2(C_1 \alpha) \rceil$, for a sufficiently large constant $C_1 > 0$ to be determined later. Set ϵ_1 to be the solution[2] to the equation $(1 - \epsilon_1) = \epsilon_1 \beta^{\xi_1}$; one can indeed verify that $\epsilon_1 = \Theta(\frac{1}{\alpha})$. Then, by construction,

$$\epsilon_0 = (1 - \epsilon_0) = (1 - \epsilon_1)\beta^{-1} = \epsilon_1 \beta^{\xi_1 - 1}. \quad (6)$$

For each $t \in \{0, 1\}$, we define the distribution μ_t over (x, y) such that $\mathrm{ed}(x, y)$ is likely to be $\Theta(\epsilon_t R)$, as follows. Choose $x \in \Sigma^d = \mathbb{Z}_p^d$ at random. Then set $z \triangleq x + N_\rho$ where $N_\rho \in \mathbb{Z}_p^d$ is a random noise of rate $\rho \triangleq 1 - \epsilon_1 R/d$. To obtain y, we apply a number of random rotation operations to z, each picked

[2] A proof of the existence is deferred to the full version.

independently from a specific distribution. We use the following notation:

- $m \triangleq 0.01 \cdot \log_\beta d = \Theta(\log d)$ is the number of possible lengths of a rotation operation;
- $L \triangleq \beta^l L_{\min}$, where $l \in [m]$, is the random variable defining the length of a rotation operation, where $L_{\min}$ is defined next;
- $L_{\min} \triangleq \Theta(d^{0.01}/\epsilon_0\epsilon_1)$ is the minimum length of a rotation operation, divided by β. $L_{\min}$ is chosen such that $\epsilon_0 L$ and $\epsilon_1 L$ are integers[3] for all $l \in [m]$;
- $w \triangleq C_2 \cdot \frac{R}{m \cdot L_{\min}}$ is the number of rotation operations that we apply, where $C_2 > 0$ is a large constant.

Generate a sequence $(r_1, r_2, \ldots, r_w)$ of w rotations by picking each r_i i.i.d. according to the following distribution $\mathcal{D}_t^{rot}$:

1. pick $l_i \in [m]$ randomly so that $\Pr[l_i = l] = \frac{\beta^{-l}}{\zeta}$ for $l \in [m]$, where $\zeta = \sum_{l=1}^{m} \beta^{-l}$ is the normalization constant;
2. pick a starting position $s_i \in [d]$ uniformly at random, and rotate the block that starts at position s_i and has length $L_i = \beta^{l_i} L_{\min}$ by $\epsilon_t \cdot L_i$ positions randomly either to the right or to the left. That is, r_i is chosen at random from the set $\left\{(\tilde{R}_{s,L_i})^{\epsilon_t L_i} \mid s \in [d], \tilde{R} \in \{\overrightarrow{R}, \overleftarrow{R}\}\right\}$.

To obtain y, apply to z the sequence of rotations $(r_1, \ldots, r_w)$, i.e.,

$$y \triangleq r_w(r_{w-1}(\ldots r_1(z)\ldots)) = (r_w \circ \ldots \circ r_2 \circ r_1)(x + N_\rho).$$

In the language of the Section 3.2, the distribution $\mathcal{D}_t$ of permutation operations is simply the distribution of $\pi = r_w \circ r_{w-1} \circ \ldots \circ r_1$, where $r_1, \ldots r_w$ are drawn independently from $\mathcal{D}_t^{rot}$.

Finally, we need to construct $\tilde{\mu}_t$ for $t \in \{0,1\}$. We note that we cannot set $\tilde{\mu}_t$ to be exactly μ_t because the latter may sometime generate pairs (x, y) that are not far or close, respectively. We thus define $\tilde{\mu}_0$ to be the distribution μ_0 restricted to (i.e. conditioned on) pairs (x, y) with that $\mathrm{ed}(x, y) > R$, and similarly $\tilde{\mu}_1$ is the distribution μ_1 restricted to pairs with $\mathrm{ed}(x, y) \le R/\alpha$. As we will see, the resulting distributions are very close to μ_0, μ_1, respectively. The total variation distance between $\tilde{\mu}_0$ and μ_0 is at most $\Pr_{\mu_0}[\mathrm{ed}(x, y) \le R] + O(1/d)$, where the second summand is upper bound on the event that either x or y is not a permutation. Similarly, total variation distance between $\tilde{\mu}_1$ and μ_1 is at most $\Pr_{\mu_1}[\mathrm{ed}(x, y) > R/\alpha] + O(1/d)$. The next lemma proves that μ_0 and μ_1 are very likely to generate pairs that are far and close, respectively.

Lemma 3.9. *For the above distributions,* $\Pr_{\mu_0}[\mathrm{ed}(x, y) \le R] \le d^{-\Omega(1)}$ *and* $\Pr_{\mu_1}[\mathrm{ed}(x, y) > R/\alpha] \le d^{-\Omega(1)}$. *Hence, for every* $t \in \{0, 1\}$, *the total variation distance between* $\tilde{\mu}_t$ *and* μ_t *is at most* $d^{-\Omega(1)}$.

The proof of this lemma relies on a claim that each individual rotation induces an expected distance of $\epsilon_t L$, and that over several rotation operations (and similarly for the noise) the resulting distance has a high concentration around its expectation. Full details are deferred to the full version.

3.4 λ-Test Indistinguishability

Having constructed the hard distribution, we now wish to apply to it Lemma 3.4, and we thus need to prove an upper bound on Δ_λ, the λ-test distinguishability of $\mathcal{D}_0, \mathcal{D}_1$.

Lemma 3.10. *Let* $\mathcal{D}_0, \mathcal{D}_1$ *be defined as in Section 3.3. Then for all* $\lambda \ge 1$, *we have* $\Delta_\lambda \le O\left(\lambda \frac{\log 1/\epsilon_1}{\log d} \cdot \frac{R}{d}\right)$.

Proof sketch. Fix a λ-test u and let δ_λ be the total variation distance between the distributions $r_0(u)$ and $r_1(u)$ where each $r_t \in \mathcal{D}_t^{rot}$. The heart of this lemma is the bound:

$$\delta_\lambda \le O\left(\lambda \log \frac{1}{\epsilon_1} \cdot \frac{L_{\min}}{d}\right). \tag{7}$$

The lemma would then follow by the union bound: $\Delta_\lambda \le \delta_\lambda w = O\left(\lambda \log \frac{1}{\epsilon_1} \cdot \frac{L_{\min}}{d} w\right) = O\left(\lambda \frac{\log 1/\epsilon_1}{\log d} \cdot \frac{R}{d}\right)$.

We prove the bound (7) on δ_λ in two steps. The first step proves the bound for $\lambda = 1$, which already illustrates the intuition why this distribution is hard. The second step builds on the first step to show the bound for general $\lambda \ge 2$.

Step 1: $\lambda = 1$. We prove that $\delta_1 \le O(\log \frac{1}{\epsilon_1} \cdot \frac{L_{\min}}{d})$ next.

Since $\lambda = 1$, we have only one non-zero entry in u, say at position j. For $t \in \{0, 1\}$, let $j^{(t)}$ be the random variable denoting the position of the symbol u_j in the vector $r(u)$ obtained by applying the random rotation $r \in \mathcal{D}_t^{rot}$ on u.

The total variation distance between the distributions of $j^{(0)}$ and of $j^{(1)}$ can be computed as the complement of their "common" weight, i.e., as $1 - \sum_{i \in [d]} \min_{t \in \{0,1\}} \Pr_{j^{(t)}}[j^{(t)} = i]$. It therefore suffices to show that the common weight is $\ge 1 - O(\log \frac{1}{\epsilon_1} \frac{L_{\min}}{d})$.

[3] It becomes more delicate when ϵ_1 is an irrational number, though there are many standard tricks to deal with this. We leave the details for the full version of the paper, and assume that $\epsilon_0 L$ and $\epsilon_1 L$ are integers in this extended abstract.

First, for both distributions $t \in \{0,1\}$, the symbol u_j remains at position j with probability

$$\Pr[j^{(t)} = j] = \sum_{l=1}^{m} \frac{\beta^{-l}}{\zeta} \frac{d-L}{d} = 1 - \frac{mL_{\min}}{\zeta d}.$$

Next, consider any $k = \epsilon_1 L = \epsilon_1 \cdot \beta^l L_{\min}$ for $l \in \{\xi_1 + 1, \ldots m\}$. We now prove that, for *all* $t \in \{0,1\}$,

$$\Pr_{j^{(t)}}[j^{(t)} = j+k] = \Pr_{j^{(t)}}[j^{(t)} = j-k] = \frac{L_{\min}}{\zeta \cdot 2d}.$$

Indeed, by the choice of ϵ_0, ϵ_1 (Eqn. (6)),

$$k = \epsilon_1 \cdot \beta^l L_{\min} = (1-\epsilon_1) \cdot \beta^{l-\xi_1} L_{\min}$$

as well as

$$\begin{aligned} k &= \epsilon_0 \cdot \beta^{l-(\xi_1-1)} L_{\min} \\ &= (1-\epsilon_0) \cdot \beta^{l-(\xi_1-1)} L_{\min}. \end{aligned}$$

The event $\{j^{(1)} = j+k\}$, i.e., the symbol u_j moves k positions to right under the distribution $\mathcal{D}_1^{rot}$, happens when either:

- the symbol u_j falls into the rotation block, the block is of length $\frac{k}{\epsilon_1} = \beta^l L_{\min}$, the rotation is to the right, and the symbol does not wrap-around; or
- the symbol u_j falls into a the rotation block, the block is of length $\frac{k}{1-\epsilon_1} = \beta^{l-\xi_1} L_{\min}$ that is rotated, the rotation is to the left, and the symbol is in the wrap-around part.

We thus obtain

$$\begin{aligned} &\Pr_{j^{(1)}}[j^{(1)} = j+k] = \\ &= \frac{\beta^{-l}}{\zeta} \cdot \frac{(1-\epsilon_1)\beta^l L_{\min}}{d} \cdot \frac{1}{2} + \frac{\beta^{-(l-\xi_1)}}{\zeta} \cdot \frac{\epsilon_1 \beta^{l-\xi_1} L_{\min}}{d} \cdot \frac{1}{2} \\ &= \frac{L_{\min}}{\zeta \cdot 2d}, \end{aligned}$$

and one can similarly prove that $\{j^{(1)} = j-k\}$, $\{j^{(0)} = j+k\}$ and $\{j^{(0)} = j-k\}$ all have the same probability, $\frac{L_{\min}}{\zeta \cdot 2d}$. We note that this "match-up" happens only for $l \geq \xi_1 + 1$, since for $l \leq \xi_1$ the second type of movement of the symbol u_j cannot happen anymore.

We can now sum the total weight that we identified as common in the two distributions to be

$$\left(1 - \frac{mL_{\min}}{\zeta d}\right) + 2 \cdot \sum_{l=\xi_1+1}^{m} \frac{L_{\min}}{\zeta \cdot 2d} = 1 - \xi_1 \cdot \frac{L_{\min}}{d}.$$

And thus, $\delta_1 \leq \xi_1 \frac{L_{\min}}{d} = O\left(\log \frac{1}{\epsilon_1} \cdot \frac{L_{\min}}{d}\right)$.

Step 2: $\lambda \geq 2$. When we have $\lambda \geq 2$ non-zero entries in u, the intuition is to group these non-zero entries into one or more segments and then reduce to the $\lambda = 1$ case with the role of "symbol u_j" being replaced by a segment. For example, when, say, there are $\lambda = 2$ non-zero entries in u, most of the block lengths L fall into two categories:

- L is much larger than the distance between the positions of the two non-zero entries – in which case, the two non-zero symbols from u move jointly most of the time, and thus the segment connecting the two symbols roughly behaves as the "symbol u_j" in the $\lambda = 1$ scenario;
- L is much smaller than the distance between the two positions – in which case, each of the two non-zero entries can be treated separately as in $\lambda = 1$ case.

Furthermore, we can bound the number of values of L that do not satisfy one of the above properties. A relatively straight-forward bound is roughly $O(\lambda^2)$ (all pair-wise distances between the non-zero entries), times $O(\log 1/\epsilon_1)$ (i.e., same factor as in $\lambda = 1$ case). This already gives a bound of $\delta_\lambda \leq O(\lambda^3 \log \frac{1}{\epsilon_1} \cdot \frac{L_{\min}}{d})$. To get the final bound (7), we employ a much more careful global analysis, that takes into consideration the fact that the same value of L is "good" for some segments but "bad" for other segments. The complete proof for this case appears in the full version of the paper. □

3.5 Putting it all together

Proof of Theorem 1.1. Let $\tilde{\mu} = \frac{\tilde{\mu}_0 + \tilde{\mu}_1}{2}$ be the distribution defined in Section 3.3. By Lemma 3.1, there must exist functions $\mathcal{H}^A, \mathcal{H}^B$ such that

$$\Pr_{\tilde{\mu}_0}[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)] - \Pr_{\tilde{\mu}_1}[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)] \geq 2^{-O(\mathrm{CC}^{\mathrm{Ulam}}_{\alpha,R})}.$$

Applying Lemma 3.4 to the distributions μ_0, μ_1, and using the fact that $\tilde{\mu}_0$ and $\tilde{\mu}_1$ are statistically close to μ_0 and μ_1 respectively (Lemma 3.9), we deduce that

$$\begin{aligned} &\Pr_{\tilde{\mu}_0}[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)] - \Pr_{\tilde{\mu}_1}[\mathcal{H}^A(x) \neq \mathcal{H}^B(y)] \\ &\qquad \leq \max_{\lambda \in [d]} \Delta_\lambda \rho^\lambda + d^{-\Omega(1)}. \end{aligned}$$

Plugging in the bound on Δ_λ (Lemma 3.10) and the value $\rho = 1 - \epsilon_1 R/d$ (Section 3.3), we obtain

$$\begin{aligned} 2^{-O(\mathrm{CC}^{\mathrm{Ulam}}_{\alpha,R})} &\leq \max_{\lambda \in [d]} O\left(\lambda \tfrac{\log(1/\epsilon_1)}{\log d} \cdot \tfrac{R}{d}\right) \cdot \left(1 - \tfrac{\epsilon_1 R}{d}\right)^\lambda + d^{-\Omega(1)} \\ &\leq O\left(\frac{1}{\epsilon_1} \cdot \frac{\log(1/\epsilon_1)}{\log d}\right) + d^{-\Omega(1)} \\ &= O\left(\frac{\alpha \log \alpha}{\log d}\right), \end{aligned}$$

which concludes the proof of Theorem 1.1. □

Acknowledgments

We thank Parikshit Gopalan, Piotr Indyk, T.S. Jayram, Ravi Kumar, Ilan Newman, and Yuri Rabinovich for numerous early discussions on nonembeddability of the Ulam metric, which undoubtedly were a precursor of the current work.

References

[1] A. Andoni, M. Deza, A. Gupta, P. Indyk, and S. Raskhodnikova. Lower bounds for embedding edit distance into normed spaces. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 523–526, 2003.

[2] A. Andoni, P. Indyk, and R. Krauthgamer. Earth mover distance over high-dimensional spaces. *ECCC Report TR07-048*, May 2007.

[3] Z. Bar-Yossef, T. S. Jayram, R. Krauthgamer, and R. Kumar. Approximating edit distance efficiently. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 550–559, Oct. 2004.

[4] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.

[5] T. Batu, F. Ergün, J. Kilian, A. Magen, S. Raskhodnikova, R. Rubinfeld, and R. Sami. A sublinear algorithm for weakly approximating edit distance. In *Proceedings of the Symposium on Theory of Computing*, pages 316–324, 2003.

[6] T. Batu, F. Ergün, and C. Sahinalp. Oblivious string embeddings and edit distance approximations. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 792–801, 2006.

[7] J. Bourgain. On the distributions of the Fourier spectrum of Boolean functions. *Israel J. Math.*, 131:269–276, 2002.

[8] M. Charikar and R. Krauthgamer. Embedding the ulam metric into ℓ_1. *Theory of Computing*, 2(11):207–224, 2006.

[9] G. Cormode. *Sequence Distance Embeddings*. Ph.D. Thesis. University of Warwick, 2003.

[10] G. Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. *ACM Trans. Algorithms*, 3(1):2, 2007.

[11] G. Cormode, M. Paterson, S. C. Sahinalp, and U. Vishkin. Communication complexity of document exchange. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 197–206, 2000.

[12] P. Gopalan, T. S. Jayram, R. Krauthgamer, and R. Kumar. Estimating the sortedness of a data stream. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[13] P. Indyk. Tutorial: Algorithmic applications of low-distortion geometric embeddings. *Proceedings of the Symposium on Foundations of Computer Science*, pages 10–33, 2001.

[14] P. Indyk. Approximate nearest neighbor under edit distance via product metrics. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 646–650, 2004.

[15] P. Indyk and R. Motwani. Approximate nearest neighbor: towards removing the curse of dimensionality. *Proceedings of the Symposium on Theory of Computing*, pages 604–613, 1998.

[16] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 68–80, 1988.

[17] S. Khot and A. Naor. Nonembeddability theorems via fourier analysis. *Mathematische Annalen*, 334(4):821–852, 2006.

[18] R. Krauthgamer and Y. Rabani. Improved lower bounds for embeddings into l_1. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 1010–1017, 2006.

[19] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *Proceedings of the 30th ACM Symposium on Theory of Computing*, pages 614–623, 1998.

[20] W. J. Masek and M. Paterson. A faster algorithm computing string edit distances. *J. Comput. Syst. Sci.*, 20(1):18–31, 1980.

[21] J. Matoušek. Collection of open problems on low-distortion embeddings of finite metric spaces. March 2007. Available online. Last access in August, 2007.

[22] S. Muthukrishnan and C. Sahinalp. Approximate nearest neighbors and sequence comparison with block operations. *Proceedings of the Symposium on Theory of Computing*, pages 416–424, 2000.

[23] R. Ostrovsky and Y. Rabani. Low distortion embedding for edit distance. In *Proceedings of the Symposium on Theory of Computing*, pages 218–224, 2005.

[24] C. Sahinalp. *Edit distance under block operations*. Encyclopedia of Algorithms (Ming Yang Kao, ed.). Springer. Forthcoming, available online. Last access in August, 2007.

[25] C. Sahinalp and A. Utis. Hardness of string similarity search and other indexing problems. In *Proceedings of International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1080 – 1098, 2004.

[26] M. Saks and X. Sun. Space lower bounds for distance approximation in the data stream model. In *Proceedings of the Symposium on Theory of Computing*, pages 360–369, 2002.

[27] D. Woodruff. Optimal space lower bounds for all frequency moments. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 167–175, 2004.