Improved Lower Bounds for Approximating Parameterized Nearest Codeword and Related Problems Under ETH

Shuangle Li
State Key Laboratory for Novel Software Technology, Nanjing University, China

Bingkai Lin
State Key Laboratory for Novel Software Technology, Nanjing University, China

Yuwei Liu
BASICS, Shanghai Jiao Tong University, China

Abstract

In this paper we present a new gap-creating randomized self-reduction for the parameterized Maximum Likelihood Decoding problem over $\mathbb{F}_p$ (k-MLD$_p$). The reduction takes a k-MLD$_p$ instance with $k \cdot n$ d-dimensional vectors as input, runs in $O(d2^{O(k)}n^{1.01})$ time for some computable function f, outputs a $(3/2-\varepsilon)$-Gap-k'-MLD$_p$ instance for any $\varepsilon > 0$, where $k' = O(k^2 \log k)$. Using this reduction, we show that assuming the randomized Exponential Time Hypothesis (ETH), no algorithms can approximate k-MLD$_p$ (and therefore its dual problem k-NCP$_p$) within factor $(3/2-\varepsilon)$ in $f(k) \cdot n^{o(\sqrt{k/\log k})}$ time for any $\varepsilon > 0$.

We then use reduction by Bhattacharyya, Ghoshal, Karthik and Manurangsi (ICALP 2018) to amplify the $(3/2-\varepsilon)$-gap to any constant. As a result, we show that assuming ETH, no algorithms can approximate k-NCP$_p$ and k-MDP$_p$ within γ-factor in $f(k) \cdot n^{o(k^{\varepsilon_\gamma})}$ time for some constant $\varepsilon_\gamma > 0$. Combining with the gap-preserving reduction by Bennett, Cheraghchi, Guruswami and Ribeiro (STOC 2023), we also obtain similar lower bounds for k-MDP$_p$, k-CVP$_p$ and k-SVP$_p$.

These results improve upon the previous $f(k) \cdot n^{\Omega(\text{poly} \log k)}$ lower bounds for these problems under ETH using reductions by Bhattacharyya et al. (J.ACM 2021) and Bennett et al. (STOC 2023).

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Error-correcting codes; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Nearest Codeword Problem, Hardness of Approximations, Fine-grained Complexity, Parameterized Complexity, Minimum Distance Problem, Shortest Vector Problem

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.107

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2402.09825

Acknowledgements We thank the anonymous reviewers for their detailed comments.

1 Introduction

The study of linear error correcting codes has drawn attention to two dual fundamental computational problems called Nearest Codeword Problem (NCP) and Maximum Likelihood Decoding (MLD). Given a matrix $A \in \mathbb{F}_p^{m \times n}$ and a vector $\vec{t} \in \mathbb{F}_p^m$, the Nearest Codeword Problem (NCP) asks for a vector $\vec{x} \in \mathbb{F}_p^n$ such that $||A\vec{x} - \vec{t}||_0$ is minimized. Here $|| \cdot ||_0$ denotes the Hamming weight. While in the Maximum Likelihood Decoding (MLD), we are given a matrix $A \in \mathbb{F}_p^{m \times n}$ and a vector $\vec{t} \in \mathbb{F}_p^m$, the goal is to minimize $||\vec{x}||_0$ subject to $A\vec{x} = \vec{t}$. Another fundamental problem related to a linear code is the homogeneous version of NCP, known as Minimum Distance Problem (MDP), where the task is to find a non-zero vector $\vec{x}$ such that $||A\vec{x}||_0$ is minimized.

© Shuangle Li, Bingkai Lin, and Yuwei Liu;
licensed under Creative Commons License CC-BY 4.0

The computational complexity of MLD, NCP and MDP has been studied with great effort throughout the past several decades. It is known that MLD, NCP and MDP are not only NP-hard [12, 48], but also NP-hard to approximate within any constant ratio [6, 7, 18, 22, 42, 47]. Moreover, the variant of MLD that allows the code being preprocessed by unbounded computational resource is also NP-hard to approximate within a factor of $(3 - \varepsilon)$ [24, 44]. Also it is proven that assuming $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{\mathsf{poly}(\log n)})$, no polynomial time algorithm can approximate NCP up to $2^{\log^{1-\epsilon} n}$ factor for any $\epsilon > 0$ [6, 43] and no polynomial time algorithm can approximate MDP up to $2^{\log^{1-\epsilon} n}$ for any $\epsilon > 0$ [7, 18, 22, 42]. For some specific codes, MLD is also shown to be NP-hard, e.g. product code [8], Reed-Solomon code [28], algebraic geometry code [17]. On the algorithmic side, it is known that NCP can be approximate to $O(n/\log n)$ in polynomial time [5].

The lattice version of NCP and MDP are known as Closest Vector Problem (CVP) and Shortest Vector Problem (SVP). In these problems, a lattice $\mathcal{L}$ is given instead of a linear code. For CVP a target $\vec{t}$ is additionally given and the goal is to find a vector $\vec{v} \in \mathcal{L}$ such that $||\vec{v} - \vec{t}||_p$ is minimized, where $||\cdot||_p$ denotes the ℓ_p-norm. And for SVP the goal is to find a non-zero vector $\vec{v} \in \mathcal{L}$ with minimum ℓ_p norm. The study for CVP and SVP also has long history [4, 6, 20, 25, 29, 33, 40–42, 47]. For CVP, it is NP-hard to approximate within factor $n^{c/\log\log n}$ for some constant $c > 0$ [20]. As for SVP, it was shown that no polynomial time algorithm can approximate SVP within any constant factor assuming $\mathsf{NP} \not\subseteq \mathsf{RP}$ [33], and no polynomial time algorithm can approximate SVP up to $2^{\log^{1-\epsilon} n}$ factor assuming $\mathsf{NP} \not\subseteq \mathsf{RTIME}(n^{\mathsf{poly}(\log n)})$ [29]. Lattice problems have many applications in cryptography [45, 46]. Due to their importance, lattice problems are also extensively studied in the fine-grained complexity area, see, e.g., [1–3, 11] and a very recent survey by Bennett [9] for more details on hardness of SVP.

Over the past three decades, parameterized complexity, a new framework to address NP-hard problems, has been rapidly developed and drawing growing attention. The study in the field of parameterized complexity focuses on whether a problem can be solved in $f(k) \cdot n^{O(1)}$ time (FPT time), where k is a parameter given along with the instance. In the parameterized version of k-MLD, k-NCP, k-MDP, k-CVP and k-SVP, an integer k is additionally given and the task is to decide whether the optimal value is no greater than k. Downey, Fellows, Vardy and Whittle [21] showed that k-MLD (and therefore k-NCP) is W[1]-hard and belongs to W[2]. They asked if k-CVP and k-SVP (in ℓ_2 norm) is W[1]-hard. 20 years later in recent breakthroughs [10, 13], the parameterized intractability of k-NCP, k-MDP, k-CVP and k-SVP are settled. Notably they ruled out not only exact FPT algorithms, but also FPT approximation algorithms as well. Specifically, [13] first presented a gap-creating reduction for k-NCP and then showed gap-preserving reductions from k-NCP towards k-MDP, k-CVP and k-SVP. Soon afterwards, Bennett, Cheraghchi, Guruswami and Ribeiro [10] improved the gap-preserving reductions for more general cases (general fields and general ℓ_p norm). These two works jointly showed that it is W[1]-hard to approximate k-NCP and k-MDP within any constant factor over any finite field $\mathbb{F}_p$, and it is W[1]-hard to approximate k-CVP in the ℓ_p norm within any constant factor for any $p \geq 1$. And they showed hardness for k-SVP to approximate within any constant factor in the ℓ_p norm for any $p > 1$, and some constant approaching 2 for $p = 1$.

After obtaining FPT-inapproximability results, it is natural to study fine-grained time lower bounds for parameterized approximability of these problems. Assuming Gap-ETH [19, 39], Manurangsi [38] showed that no $f(k) \cdot n^{o(k)}$ time algorithm can approximate k-NCP and k-CVP to any constant factor. With the gap-preserving reduction in [10], one can further show that no $f(k) \cdot n^{o(k)}$ time algorithm can approximate k-MDP and k-SVP to

any constant under the randomized Gap-ETH. All these results are based on an assumption with a gap. This raises the following open question:

(1) Can we establish similar lower bounds for these problems under the weaker and gap-free assumption of ETH?

We note that the gap-preserving reduction in [10] from Gap-k-NCP (Gap-k-CVP) to Gap-k'-MDP (Gap-k'-SVP) has $k' = O(k)$. So, it suffices to prove constant Gap-k-NCP (Gap-k-CVP) has no $f(k) \cdot n^{o(k)}$-time algorithm assuming ETH [30]. Unfortunately, the gap-creating reduction in [13] causes an exponential growth of the parameter and only gives an $\Omega(n^{(\log k)^{1/(2+\epsilon)}})$-time lower bound for constant Gap-k-NCP under ETH (See the analysis in Section 1.3). Therefore, finding better reductions for Gap-k-NCP and Gap-k-CVP is the crux of improving lower bounds for Gap-k-MDP and Gap-k-SVP.

1.1 Our Contributions

We take a step forward on closing the gap between results under gap-free assumption (ETH) and gap assumption (Gap-ETH). Our main result is a new direct gap-creating self reduction for k-MLD, which is the dual problem of k-NCP, with polynomial growth of the parameter.

▶ **Theorem 1** (informal; See Theorem 20 for a formal statement)**.** *For any constant $1 < \gamma < \frac{3}{2}$ and prime power $p > 1$, there is a reduction runs in $O_k(n^{O(1)})$ that on input a k-MLD$_p$ instance $(V, \vec{t})$, output a Gap-k'-MLD$_p$ instance $(V', \vec{t}')$ satisfies:*

- *(Completeness) If there exists k vectors in V with their sum [1] being $\vec{t}$, then there exists k' vectors in V' with their sum being $\vec{t}'$.*
- *(Soundness) If for any set $S \subseteq V$ with size at most k, $\vec{t} \notin Span(S)$, then for any set $S' \subseteq V'$ with size at most $\gamma k'$, $\vec{t}' \notin Span(S')$.*
- *Polynomial parameter growth $k' = O(k^2 \log k)$. (And $k' = O(k^3)$ if not allowing randomness).*

Combining this gap-creating reduction with the $f(k)n^{\Omega(k)}$-time ETH lower bound for k-MLD in [36, Theorem 11], we obtain improved lower bounds for Gap-k-NCP assuming ETH and randomized ETH.

▶ **Corollary 2.** *Assuming randomized ETH, for any prime power $p > 1$ and real number $\gamma \in (1, \frac{3}{2})$, no $f(k)n^{o(\sqrt{k/\log k})}$ time algorithm can solve γ-Gap-k-NCP$_p$.*

▶ **Corollary 3.** *Assuming ETH, for any prime power $p > 1$ and real number $\gamma \in (1, \frac{3}{2})$, no $f(k)n^{o(k^{1/3})}$ time algorithm can solve γ-Gap-k-NCP$_p$.*

By applying the gap amplification procedure in [14] ($\gamma \to \Omega(\gamma^2), k \to O(k^2)$, see Theorem 22 for a formal statement) sufficiently many (but still constant) times, we obtain a reduction for Gap-k-MLD with any constant gap with still polynomial growth of parameter. Therefore we obtain the following improved ETH lower bound for k-NCP.

▶ **Corollary 4.** *Assuming ETH, for any prime power $p > 1$ and real number $\gamma > 1$, no $f(k)n^{o(k^\epsilon)}$ time algorithm can solve γ-Gap-k-NCP$_p$ where $\epsilon = \frac{1}{\text{polylog}(\gamma)}$ is a constant.*

Combining our results of Gap-k-NCP$_p$ with the gap-preserving reductions in [13] and [10], we obtain improved ETH lower bounds for constant approximating k-NCP, k-CVP, k-MDP and k-SVP. The summarize of corollaries are present in Table 1.

[1] The definition of k-MLD used in our proof is a slightly different variant, where the vectors directly sum to the target in the YES case, but they are essentially equivalent, see Section 2.3 for more details.

Table 1 The $f(k)n^{\Omega(k^\epsilon)}$-time lower bound for k-NCP and k-CVP are based on ETH. The other lower bounds are based on randomized ETH.

Summarize of Corollaries

Problem	Inapprox Factor	Lower Bound	Dependency	Specification
k-NCP	any $\gamma \in (1, \frac{3}{2})$	$f(k)n^{\Omega(\sqrt{k/\log k})}$		any finite field $\mathbb{F}_p$
k-NCP	any $\gamma > 1$	$f(k)n^{\Omega(k^\epsilon)}$	$\epsilon = \frac{1}{\mathsf{polylog}(\gamma)}$	any finite field $\mathbb{F}_p$
k-MDP	any $\gamma > 1$	$f(k)n^{\Omega(k^\epsilon)}$	$\epsilon = \frac{1}{p\log\gamma\cdot\mathsf{polylog}(p)}$	any finite field $\mathbb{F}_p$
k-CVP	any $\gamma > 1$	$f(k)n^{\Omega(k^\epsilon)}$	$\epsilon = \Theta(\frac{1}{\mathsf{polylog}(\gamma)})$	in any ℓ_p norm, $p \geq 1$
k-SVP	any $\gamma > 1$	$f(k)n^{\Omega(k^\epsilon)}$	$\epsilon = \epsilon(p,\gamma)^2$	in any ℓ_p norm, $p > 1$
k-SVP	any $\gamma \in [1, 2)$	$f(k)n^{\Omega(k^\epsilon)}$	$\epsilon = \epsilon(p,\gamma)^3$	in any ℓ_p norm, $p \geq 1$

1.2 Technical Overview of Gap Creation Step

We implicitly use the threshold graph composition method [15, 34, 35, 37] to construct a $(3/2-\varepsilon)$-gap producing reduction for the k-MLD problem. This technique was first introduced in [34] to prove the W[1]-hardness of k-Biclique problem. A threshold graph is a bipartite graph that has a "threshold property", meaning that there is a significant gap in the number of common neighbors between any k vertices and any $k+1$ vertices on the left side. Threshold graph and its variants have been widely used to show hardness of approximation for various parameterized problems, such as k-DominatingSet [16], k-SetCover [32, 35], k-SetIntersection [15] or to create gap for subsequent reductions, e.g. [13].

Let $\dot\cup$ denotes for union set of multiple disjoint sets. In this paper, we implicitly use the strong threshold graphs in [37], which are bipartite graphs $T = (A \dot\cup B, E_T)$ with the following properties:

- **(i)** $A = A_1 \dot\cup A_2 \dot\cup \cdots \dot\cup A_k$.
- **(ii)** $B = B_1 \dot\cup B_2 \dot\cup \cdots \dot\cup B_m$.
- **(iii)** For any $a_1 \in A_1, \ldots, a_k \in A_k$ and $i \in [m]$, $a_1, \ldots, a_k$ have a common neighbor in B_i.
- **(iv)** For any $X \subseteq A$ and $I \subseteq [m]$ with $|I| \geq \varepsilon m$, if for every $i \in I$, there exists $b_i \in B_i$ has $k+1$ neighbors in X, then $|X| > h$.

These strong threshold graphs are constructed from error-correcting codes with large relative distance $(1-o(1))$, and such "threshold" properties essentially come from the following intuition of ECC: *If there is a collection of codewords* (X)*, and a constant fraction of entries of these codewords* $(I \subseteq [m], |I| \geq \varepsilon m)$ *such that, for each entry* $(i \in I)$*, there exists two distinct codewords in the collection having same content in it. Then, the collection must have huge size* (*at least* h). To characterize the aforementioned property, previous works [32, 37] introduced the definition of (ε-)*Collision Number* of an error-correcting code C, $\mathrm{Col}_\varepsilon(C)$, which is the minimum size of X mentioned above.

Diving into coding-based threshold graph. Our construction deeply relies on the collision number of an ECC, so we only use threshold graph as an intuitive illustration for readers, and we directly use the error-correcting codes in our formal analysis.

Below we illustrate the idea of our reduction. For simplicity, here we consider k-MLD problem on d-length vectors over binary field. Given k vectors sets $V_1, \ldots, V_k \subseteq \mathbb{F}_2^d$, a target vector $\vec{t}$ and a strong threshold graph $T = (A \dot\cup B, E_T)$, we first identify V_i with A_i for every

$i \in [k]$. Our goal is to construct a one-to-one mapping $f : A \cup B \to \mathbb{F}_2^D$ and a new target vector $\vec{t}' \in \mathbb{F}_2^D$ for some $D = \mathsf{poly}(d, k)$ such that in order to pick vectors from $f(A \cup B)$[4] with their sum being $\vec{t}'$, one has to pick a set $f(X)$ of vectors from $f(A)$ for some $X \subseteq A$ with $\sum_{\vec{a} \in X} \vec{a} = \vec{t}$ and a set $f(Y)$ of vectors from $f(B)$ for some $Y \subseteq B$ such that for every $i \in [m]$,

(a) either $|Y \cap B_i| \geq 2$,

(b) or $|Y \cap B_i| = 1$ and there exists $b_i \in B_i$ with one of following properties:

(b.1) $|X| = k$ and b_i is the common neighbors of vertices in X.

(b.2) b_i has at least $k + 1$ neighbors in X.

Then we argue that these properties imply a constant gap between the solution sizes in the (YES) and (NO) cases of the k-MLD problem.

(YES) Suppose there are $a_1 \in A_1, \ldots, a_k \in A_k$ such that $\sum_{i \in [k]} a_i = \vec{t}$. By the property (iii) of threshold graphs, $a_1, \ldots, a_k$ have a common neighbor $b_i \in B_i$ for every $i \in [m]$. Then according to (b), the sum of $f(a_1), \ldots, f(a_k)$ and $f(b_1), \ldots, f(b_m)$ is $\vec{t}'$.

(NO) On the other hand, if there are no $a_1 \in A_1, \ldots, a_k \in A_k$ such that $\sum_{i \in [k]} a_i = \vec{t}$, then one should pick either at least $(1 - \varepsilon)2m$ vectors from $f(B)$ and $k + 1$ vectors from $f(A)$, or pick a subset of vectors $f(X)$ from $f(A)$ and a subset of vectors $f(Y)$ from $f(B)$ for some $Y \subseteq B$ with $|\{i \in [m] : |Y \cap B_i| = 1\}| \geq \varepsilon m$. Let $I = \{i \in [m] : |Y \cap B_i| = 1\}$. According to the property (b.2), each vertex in $Y \cap B_i$ ($i \in I$) has $k + 1$ neighbors in X. Since $|I| \geq \varepsilon m$, by the property (iv) of threshold graphs, we have that $|X| > h$. Thus, either $(1 - \varepsilon)2m$ vectors in $f(B)$ and $k + 1$ vectors in $f(A)$ or m vectors in $f(B)$ and h vectors in $f(A)$ must be picked in this case.

To obtain a constant gap, we duplicate each vector in $f(A)$ m/k times and let $h = ck$ where c is some constant to be chosen. In the (yes) case, there are $2m$ vectors with their sum being $\vec{t}'$. In the (no) case, no $\min\{2(1 - \varepsilon)m + m, m + cm\}$ vectors from $f(A \cup B)$ can have sum $\vec{t}'$.

The proof framework above has two problems to be solved.

(P1) How to combine the threshold graph and the k-MLD instance to produce vectors $f(A \cup B)$ with the properties (a) and (b)?

(P2) The smaller parameter blow-up we create in reduction, the tighter running time lower bound we obtain. So how to construct a threshold graph with $h > ck$ and m as small as possible?

Our approach to solve Problem (P1). Problem (P1) is related to the composition step in the threshold graph composition method. For the k-SetCover problem, we can use the hypercube partition system [23] to solve this problem. Unfortunately, this does not apply to the k-MLD problem. To solve problem (P1), we exploit an additional property from the construction of strong threshold graph using error correcting codes. More precisely, we can assume that there is a encoding function $C : A \to \Sigma^m$ and each $b_i \in B_i$ can be written as a k-tuple in $(b_{i,1}, \ldots, b_{i,k}) \in \Sigma^k$ such that b_i is adjacent to $a_j \in A_j$ in the threshold graph if and only if $b_{i,j} = C(a_j)[i]$. Informally speaking, we choose the target vector $\vec{t}'$ and the one-to-one mapping $f : A \cup B \to \mathbb{F}_2^D$ such that any subset of vectors in $f(A \cup B)$ summing up to $\vec{t}'$ must contains, for each $i \in [m]$, at least one vector $f(b_i)$ for some $b_i \in B_i$. And if it contains exactly only one such vector $f(b_i)$, then one need to pick at least k vectors $f(a_1) \in f(A_1), \ldots, f(a_k) \in f(A_k)$ to cancel out the parts corresponding to $b_{i,1}, \ldots, b_{i,k}$ in the vector $f(b_i)$. A careful analysis shows that this construction has the properties (a) and (b).

[4] Here we let $f(X)$ denote the set $\{f(x) : x \in X\}$.

Our approach to solve Problem (P2). The construction of strong threshold graph in [37] was based on the idea of Karthik and Navon [32]. Karthik and Navon [32] observed that the "collision number" of an error-correcting code can be directly used to show the threshold property. Intuitively speaking, a set C of strings with high ε-collision number indicates that if there is some mechanism forces us to choose some strings in C that collides on at least ε fraction of entries, then we must choose at least $\mathrm{Col}_\varepsilon(C)$ strings.

Known analysis of collision number in [10,32] starts from the distance of an error-correcting code. For a code with relative distance δ, previous analysis shows that its ε-collision number is $\mathrm{Col}_\varepsilon(C) = \sqrt{\frac{2\varepsilon}{1-\delta}}$. Note that $\delta = 1 - \Theta(\frac{r}{m})$ for Reed-Solomon codes used in the previous works. To obtain a gap, we require $\mathrm{Col}_\varepsilon(C) \geq \Theta(k)$, which leads to $m = \Omega(k^2)r$. In our reduction, we additionally require $\Sigma^r \geq n$ to fit the input size, which requires $r \geq \frac{\log n}{\log |\Sigma|}$, then we have $m \geq k^2 \log n / \log |\Sigma|$. On the other hand, our reduction needs to enumerate every k-tuples in Σ^k, concerning the running time we require $|\Sigma|^k \leq n^{O(1)}$. Putting all together, we must have $m \geq \Omega(k^3)$. In fact, we showed that the Singleton bound of codes implies such construction must have parameter growth $\Omega(k^3)$.

To obtain a better parameter, we find the analysis by Karthik and Navon [32, Section 3.1] can be modified to show better lower bound for the ε-collision number of a random code. Following their idea, we show a random code $C_R : \Sigma^r \to \Sigma^m$ with superconstant-sized alphabet and $m = \Omega(|\Sigma|^{1/3} \log |\Sigma| r)$ would have ε-collision number $\mathrm{Col}_\varepsilon(C_R) \geq |\Sigma|^{1/3}$, with high probability. Setting $|\Sigma| = \Theta(k^3)$, we have $\mathrm{Col}_\varepsilon(C) \geq \Theta(k)$. But now the parameter $m = \Omega(|\Sigma|^{1/3} \log |\Sigma| r) \geq k \log n$ is too large. Our solution is to consider a new error correcting code with small dimension by increasing the alphabet size and show that this new code has the same collision number. More precisely, we partition the m bits into g blocks, each containing m/g bits and treat the code words as strings in Σ'^g where $\Sigma' = \Sigma^{m/g}$. Since $|\Sigma'^k| \leq n^{O(1)}$, we have $m/g \leq O(\frac{\log n}{k \log |\Sigma|}) = O(\frac{\log n}{3k \log k})$. Thus, $g \geq \Theta(\frac{mk \log k}{\log n}) \geq \Theta(k^2 \log k)$. This reduces the parameter growth from k^3 to $k^2 \log k$, and the (randomized) ETH-based running time lower bound can be improved to $n^{O(\sqrt{k/\log k})}$. We hope to see whether some better construction of threshold graph leads to better lower bound of problems we discuss.

1.3 Previous Work

The parameterized complexity of k-MDP had been open for many years. This problem was first resolved by [13]. Interestingly, the reduction in [13] also ruled out constant FPT-approximation algorithm for k-MDP over binary field. In addition, they also ruled out any constant FPT-approximation algorithm for k-CVP in all ℓ_p norms. Recent work by Bennett, Cheraghchi, Guruswami and Ribeiro [10] proved parameterized inapproximability for k-MDP over all finite fields and k-SVP in all ℓ_p norms and arbitrary constant gap. These results are all based on the W[1]-hardness of constant Gap-k-NCP or Gap-k-CVP in [13].

Unfortunately, the gap-creating reduction from k-Clique to constant Gap-k'-NCP or Gap-k'-CVP in [13] has a long reduction chain and causes a significant increase in the parameter. For example, the reduction from k-Clique to constant Gap-k'-NCP contains the following steps (the reduction for Gap-k'-CVP is similar):

- The first step is to reduce k-Clique to the One-Sided Gap Biclique problem. In this step, the reduction outputs a bipartite graph $H = (L \cup R, E)$ and three integers $s = k(k-1)/2$, $\ell = (k+1)!$ and $h > \ell$ on input a graph G and an integer k such that if G contains a k-clique, then there are s vertices in L with h common neighbors. On the other hand, if G contains no k-clique, then every s-vertex set of L has at most ℓ common neighbors in R.

- The second step is to reduce the One-Sided Gap Biclique problem to Gap-k'-Linear Dependent Set problem (Gap-k'-LDS)[5]. On input the bipartite graph $H = (L \cup R, E)$ and three positive integers $s, \ell, h \in \mathbb{N}$, the reduction outputs a set W of vectors and an integer $k' = hs$ such that, if H contains a $K_{s,h}$-subgraph, then there are k' vectors in W that are linearly dependent. If every s-vertex set in L has at most ℓ common neighbors, then any linearly dependent set of W must have size at least $(h/\ell)^{1/s}$. To create a constant gap, one must choose a large parameter h such that $(h/\ell)^{1/s} \geq \gamma hs$ for some $\gamma > 1$. Hence in [13], the authors have to set $h = (k+6)! \cdot (\gamma k^2)^{k^2}$ and $k' = hs \geq k^{\Omega(k^2)} = 2^{\Omega(k^2 \log k)}$.
- The next step is to reduce the Gap-k'-Linear Dependent Set problem (Gap-k'-LDS) to Gap-k''-Maximum Likelihood Decoding problem (Gap-k''-MLD)[6]. This reduction preserves the parameter i.e., $k'' = k$.
- The remaining step gives a reduction from constant Gap-k''-MLD to constant Gap-k''-NCP.

Combining this with the $f(k) \cdot n^{\Omega(k)}$-time lower bound for the k-Clique problem, we only get a $g(k) \cdot n^{\Omega((\log k)^{1/(2+\epsilon)})}$-time lower bound for Gap-k-NCP using the reduction from [13].

Under a stronger gap assumption (Gap-ETH), Manurangsi [38] showed a tight $n^{\Omega(k)}$ time lower bound for constant approximating problems discussed in this article. His approach is to show an $n^{\Omega(k)}$ time lower bound for constant approximating LaberCover, then reduce it to k-UniqueSetCover, then reduce k-UniqueSetCover to gap problems we discuss using reduction in [6]. The key step in his proof is to establish hardness result for approximating k-UniqueSetCover. To our best knowledge, there is no hardness of approximation result for the parameterized k-UniqueSetCover under gap-free assumptions, e.g. ETH and W[1] $\neq$ FPT.

Very recently, Guruswami, Ren and Sandeep [26] showed constant FPT-inapproximability of k-UniqueSetCover under the assumption that Average Baby PIH holds even for 2CSP instance having rectangular relations. It's interesting whether their result and method can shed some light on showing ETH-based $n^{\Omega(k)}$ time lower bound for k-UniqueSetCover. We remark that the ETH-based $n^{\Omega(k)}$ time lower bound for constant approximating k-UniqueSetCover is still an open problem, and so does its FPT-inapproximability assuming W[1] $\neq$ FPT.

1.4 Paper Organization

In Section 2, we give preliminary of this paper. In Section 3, we give a new analysis on collision number of random code, this section can be skipped if readers wants to see the reduction directly. In Section 4, we present our gap-creating reduction for k-MLD$_p$. In Section 5, we show how to apply our reduction to other results and show inapproximability of other problems. For self-containment, we give a proof of equivalence between k-MLD$_p$ and k-NCP$_p$ in Appendix A of our full version.

2 Preliminaries

For integer $m > 0$, let $[m] = \{1, 2, \cdots, m\}$. For prime power $p > 1$, we let $\mathbb{F}_p = \{0, 1 \cdots, p-1\}$ denote the p-sized finite field. We denote $\mathbb{F}_p^+$ as $\mathbb{F}_p \backslash \{0\}$. For a vector $\vec{v} \in \Sigma^m$ and $i \in [m]$, let $\vec{v}[i] \in \Sigma$ denote the i-th entry of $\vec{v}$. For two vectors $\vec{u}$, $\vec{v}$, let $\vec{u} \circ \vec{v}$ denote their concatenation.

[5] In fact, the reduction in [13] from One-Sided Gap Biclique to Gap-k-LDS goes though an intermediate problem called gap bipartite subgraph with minimum degree (GapBSMD).

[6] Again, they introduced a color-coding technique to Gap-k-LDS (Gap-k-Colored-LDS) and used it as an intermediate problem between Gap-k-LDS and Gap-k-MLD, for details see [13, Lemma 4.8, Theorem 5.4].

The symbol $\dot{\cup}$ denotes for the union set of multiple disjoint sets. As a supplement of big-O notation, we let $f(k,n) = O_k(g(n))$ denote there exists constant $c > 0$ and computable function $h : \mathbb{N} \to \mathbb{N}$ such that for any fixed $k > 0$, $f(k,n) < c \cdot h(k)g(n)$ holds for all sufficiently large n.

For alphabet Σ and vector $\vec{u}, \vec{v} \in \Sigma^m$, the relative distance of them is defined as $\mathsf{dist}(\vec{u}, \vec{v}) = \frac{|\{i \in [m] : \vec{u}[i] \neq \vec{v}[i]\}|}{m}$. In this article, we sometimes use "distance" as shorthand of relative distance. For vector $\vec{v} \in \mathbb{Z}^m$ and $p \geq 1$, let the ℓ_p norm of $\vec{v}$ be $\ell_p(\vec{v}) = (\Sigma_{1 \leq i \leq m} |\vec{v}[i]|^p)^{1/p}$.

2.1 Error-correcting Codes

Error-correcting code plays a fundamental role in computer science and information theory. The problem we mainly discuss in this article and the construction we use are closely related to them. We give a general definition of error-correcting code. A detailed and systematic introduction to coding theory can be found at [27].

▶ **Definition 5** (Error-correcting Codes). *Fix an alphabet Σ, an error-correcting code with length m and relative distance $\delta > 0$ is a subset $\mathcal{C} \subseteq \Sigma^m$ satisfying for all $\vec{x}, \vec{y} \in \mathcal{C}$, if $\vec{x} \neq \vec{y}$, $\mathsf{dist}(\vec{x}, \vec{y}) \geq \delta$.*

▶ **Definition 6** (Linear Codes). *Fix an alphabet Σ such that Σ^r and Σ^m being linear spaces, a linear code is an error-correcting code $\mathcal{C} \subseteq \Sigma^m$ associated with a linear function $f : \Sigma^r \to \Sigma^m$ that for all $x \in \Sigma^r$, $f(x) \in \mathcal{C}$.*

2.2 Hypothesis

We introduce the Exponential Time Hypothesis in this section.

▶ **Definition 7** (3-SAT). *Given a 3-CNF formula (conjunctive formal form, each clause contains exactly 3 literals) φ with n variables and m clauses, decide if there exists a boolean assignment $z \in \{0,1\}^n$ that satisfies φ, i.e., $\varphi(z) = 1$.*

▶ **Hypothesis 8** (Exponential Time Hypothesis [30, 31]). *There exists constant $\delta > 0$ such that 3-SAT with n variable and $O(n)$ clauses cannot be solved in time $O(2^{\delta n})$.*

▶ **Hypothesis 9** (Randomized Exponential Time Hypothesis). *There exists constant $\delta > 0$ such that 3-SAT with n variable and $O(n)$ clauses cannot be solved by randomized algorithm in time $O(2^{\delta n})$.*

2.3 Problems

We first give the definition of general parameterized Maximum Likelihood Decoding problem.

γ-Gap-k-MLD$_p$

Instance:	A vector multi-set $V \subseteq \mathbb{F}_p^d$ with size n and a target vector $\vec{t} \in \mathbb{F}_p^d$.
Parameter:	k.
Problem:	Distinguish between the following two cases:
(YES)	There exists k distinct vectors (with respect to multi-set), $\vec{v}_1, \cdots, \vec{v}_k \in V$ and $\alpha_1, \ldots, \alpha_l \in \mathbb{F}_p^+$ such that $\alpha_1\vec{v}_1 + \cdots + \alpha_k\vec{v}_k = \vec{t}$.
(NO)	Any $\ell \leq \gamma k$, l vectors $\vec{v}_1, \ldots, \vec{v}_l \in V$ and $\alpha_1, \ldots, \alpha_l \in \mathbb{F}_p^+$ satisfies $\alpha_1\vec{v}_1 + \cdots + \alpha_l\vec{v}_l \neq \vec{t}$.

To fit requirements in our reduction, we start from a special restricted type of parameterized Maximum Likelihood Decoding problem that vectors are partitioned into k different sets, and the YES case asks for selecting one vector from each set such that they directly add up to the target vector.

The other problems we study includes parameterized Nearest Codeword Problem, which asks if there is a codeword of the given linear code having distance no more than k to a given target vector; Minimum Distance Problem, which asks if the minimum distance of given linear code does not exceed k; Closest Vector Problem, asking if there is a vector in the given linear lattice having ℓ_p distance no more than k to a given target vector; Shortest Vector Problem, asking if the shortest non-zero vector of given linear lattice does not exceed k. Formal definitions of these problems are referred to the full version.

2.4 Probability Inequality

▶ **Theorem 10** (Chernoff Bound). *Consider independent random variables $X_1, \ldots, X_n \in \{0,1\}$ with $X = \sum_{i=1}^{m} X_i$ and $\mu = \mathbb{E}[X]$. For any $0 < \delta < 1$ we have*

$$\Pr[X \leq (1-\delta)\mu] \leq \exp\left(-\frac{\mu\delta^2}{2}\right).$$

3 Collision Number of Error Correcting Codes

In this section, we introduce the definition of collision number of a code, which is key to our gap-creating reductions. Given a collection of strings $S \subseteq \Sigma^m$, we say that S "collides" on the i-th coordinate if there are distinct $x, y \in S$ such that $x[i] = y[i]$. Following the work of [32, 37], we define the collision number of a set of strings as follows.

▶ **Definition 11** (ε-Collision Number). *For a set $C \subseteq \Sigma^m$ and $0 < \varepsilon < 1$, the ε-collision number of C, denote as $\mathrm{Col}_\varepsilon(C)$, is the smallest integer $s \in \mathbb{N}^+$ such that there exists $S \subseteq C$ with $|S| = s$ and S collides on more than εm coordinates, i.e.,*

$$|\{i \in [m] \mid \exists x, y \in S, x \neq y \text{ s.t. } x[i] = y[i]\}| > \varepsilon m.$$

To create a gap for the k-MLD$_p$ problem, we need to construct codes $C \subseteq \Sigma^m$ with collision number $\mathrm{Col}_\varepsilon(C) \geq \Omega(k)$ and m depends only on k. We sketch two constructions (Theorem 13 and Lemma 17).

▶ **Lemma 12** ([32], See also Theorem 10 in [37]). *For any constant $0 < \varepsilon \leq 1$, an error correcting code $C : \Sigma^r \to \Sigma^m$ with relative distance $0 < \delta < 1$ has $\mathrm{Col}_\varepsilon(C) \geq \sqrt{\frac{2\varepsilon}{1-\delta}}$.*

▶ **Theorem 13** ([32, 37]). *Fix any Reed-Solomon code $\mathcal{C}^{RS} : \Sigma^r \to \Sigma^m$ with $r < m \leq |\Sigma|$. For any $0 < \varepsilon < 1$, $\mathrm{Col}_\varepsilon(\mathcal{C}^{RS}) \geq \sqrt{\frac{2\varepsilon m}{r}}$.*

To fit the requirement in our reduction, i.e., $|\Sigma|^r \geq n$, we choose $|\Sigma| = n^{1/k}$ and $r = \Omega(k)$. To fit the requirement that $\mathrm{Col}_\varepsilon(C) = \Omega(k)$ in Lemma 19, the Reed-Solomon code here must satisfy $m = \Omega(k^2 r) = \Omega(k^3)$. Seeking for a shorter code with high ε-collision number, we turn to randomized construction of codes, and we show the following lemma. The proof is similar to [32, Claim 3.4] by showing each coordinate has collision with low probability in a small set, then apply Chernoff bound and union bound to show a small set can hardly have large collision number. Details see the full version.

▶ **Lemma 14.** *For any constant $0 < \varepsilon < 1$ and any random code $C_R : \Sigma^r \to \Sigma^m$ where each codeword is selected uniformly at random in Σ^m, if $m \geq 16\frac{1}{\varepsilon^2}|\Sigma|^{1/3} \ln |\Sigma| r$ and $|\Sigma| = \omega(1)$, then with high probability, $\mathrm{Col}_\varepsilon(C_R) > |\Sigma|^{1/3}$.*

By instantiating Lemma 14 with appropriate parameter, we have:

▶ **Lemma 15.** *For any constant $c > 0$ and $0 < \varepsilon < 1$, there is a randomized algorithm that given integers $n, k \in \mathbb{N}^+$, constructs a code $C \subseteq \Sigma^m$ with parameters $|C| = n, |\Sigma| = O(k^3)$ and $m = O(k \log n)$ such that with high probability, $\mathrm{Col}_\varepsilon(C) > ck$. Moreover, the running time of this algorithm is $O(nm|\Sigma|)$.*

▶ Remark 16. We remark that using an almost identical argument, Lemma 14 can be extended to the case that for each integer $t \geq 3$, if $m > \Omega(|\Sigma|^{1/t} \log |\Sigma| r)$ and $|\Sigma| = \omega(1)$, then w.h.p., $\mathrm{Col}_\varepsilon(C_R) > |\Sigma|^{1/t}$. For constant $t > 3$, setting $|\Sigma| = \Omega(k^t)$, Lemma 15 can be extended to the case with same parameter but larger code alphabet.

By merging the number of code blocks over small alphabet from Lemma 15, we obtain:

▶ **Lemma 17.** *For any constant $c > 1$ and $0 < \varepsilon < 1$, there is a randomized algorithm that given integers $n, k \in \mathbb{N}^+$, constructs a code $C \subseteq \Sigma^m$ with parameters $|C| = n, |\Sigma| = O(n^{1/k})$ and $m = O(k^2 \log k)$ such that with high probability $\mathrm{Col}_\varepsilon(C) > ck$. Moreover, the running time of this algorithm is $O(k^2 \log k n^{1+1/k})$.*

3.1 Limitation of Collision Analysis in [32, 37]

We have already shown the a direct collision analysis yields $m' = O(k^2 \log k)$. Below, we argue that collision analysis from distance must cause a cubic increase in the parameter.

To obtain a constant gap using Lemma 19 in Section 4, we require the collision number of code C to be $\mathrm{Col}_\varepsilon(C) = \Omega(k)$. Combining with Lemma 12, we immediately have the relative distance of code C must satisfy

$$\delta \geq 1 - 1/\Omega(k^2).$$

On the other hand, we have the following Singleton bound from coding theory, whose proof can be found in [27, Section 4.3].

▶ **Theorem 18** (Singleton Bound)**.** *For every code $C : \Sigma^r \to \Sigma^m$ with relative distance δ must have $r \leq m - \delta m + 1$.*

We apply the bound to parameter we choose and obtain $m - (1 - \frac{1}{\Omega(k^2)}))m + 1 \geq r$, i.e.,

$$m \geq \Omega(k^2) r.$$

Our reduction for MLD associates each input vector with a unique codeword, which requires $|C| = |\Sigma|^r \geq n$, leading to

$$r \geq (\log n)/(\log |\Sigma|).$$

Since our reduction runs in time $\Omega(|\Sigma|^k)$, we need $|\Sigma| \leq n^{O(1/k)}$. Thus $r \geq (\log n)/(\log |\Sigma|) \geq \Omega(k)$ and $m \geq \Omega(k^2) r \geq \Omega(k^3)$. Note that we've shown the Reed-Solomon code already achieves $m = O(k^3)$.

4 Gap-creating Reduction for k-MLD$_p$

In this section we present our gap-creation reduction for k-MLD$_p$. First we present a construction that illustrates our main idea and is the crux of our reduction. This construction produces an "unbalanced gap" k'-MLD$_p$ instance in the sense that the output instance is divided into two parts (with different sizes), any solution must contain an amount of vectors in each part. Further, for the NO case, any solution must contain constant fraction more vectors in at least one part. This construction still needs to be modified later to convert into an actual reduction.

▶ **Lemma 19.** *There is an algorithm which on input k sets of length-d vectors $V_1, \cdots, V_k \subseteq \mathbb{F}_p^d$ each of size n, a target vector $\vec{t} \in \mathbb{F}_p^d$ and a code $C \subseteq |\Sigma|^m$ with $|C| = n$ and $\mathrm{Col}_\varepsilon(C) \geq ck$ outputs $A = A_1 \dot\cup \cdots \dot\cup A_k \subseteq \mathbb{F}_p^D$ and $B = B_1 \dot\cup \ldots \dot\cup B_m \subseteq \mathbb{F}_p^D$ with $D = O(d + km|\Sigma|)$ and a target vector $\vec{t}' \in \mathbb{F}_p^D$ in $O(dm^2k^2|\Sigma|(n + |\Sigma|^k))$-time such that*

- **(i)** *If there exist $\vec{v}_1 \in V_1, \ldots, \vec{v}_k \in V_k$ such that $\sum_{i\in[k]} \vec{v}_i = \vec{t}$, then there exists $\vec{a}'_1 \in A_1, \cdots, \vec{a}'_k \in A_k$ and $\vec{b}'_1 \in B_1, \cdots, \vec{b}'_m \in B_m$ with their sum being $\vec{t}'$.*
- **(ii)** *If for any $\vec{v}_1 \in V_1, \ldots, \vec{v}_k \in V_k$ and $\alpha_1, \ldots \alpha_k \in \mathbb{F}_p^+$ it holds that $\alpha_1\vec{v}_1 + \cdots + \alpha_k\vec{v}_k \neq \vec{t}$, then any $X \subseteq A \dot\cup B$ and $\lambda : X \to \mathbb{F}_p^+$ such that $\sum_{\vec{x}\in X} \lambda(\vec{x})\vec{x} = \vec{t}'$ must satisfy at least one of the following:*
  - *$|X \cap A| \geq ck$ and $|X \cap B| \geq m$,*
  - *$|X \cap A| \geq k$ and $|X \cap B| \geq 2(1-\varepsilon)m$.*

Proof. The resulting dimension is $D = d + mk|\Sigma| + k + m$. We break the resulting dimension into 4 blocks respectively of size $d, mk|\Sigma|, k$ and m. To be precise, for any vector $\vec{x} \in \mathbb{F}_p^D$, let

- $\vec{x}^{(1)} \in \mathbb{F}_p^d$ be the first block,
- $\vec{x}^{(2)} \in \mathbb{F}_p^{mk|\Sigma|}$ be second block,
- $\vec{x}^{(3)} \in \mathbb{F}_p^k$ be the third block,
- $\vec{x}^{(4)} \in \mathbb{F}_p^m$ be the fourth block.

We further break the second block into m sub-blocks each of size $k|\Sigma|$, i.e., $\vec{x}^{(2)} = \vec{x}^{(2,1)} \circ \cdots \circ \vec{x}^{(2,m)}$.

We let $\vec{e}_i$ be the indicator vector of which the i-th entry is 1 and the other entries are 0. To be convenient, the dimension of $\vec{e}_i$ depends on the context. Specially we let $\iota : \Sigma \to [|\Sigma|]$ be an arbitrary bijection, and for every $\sigma \in \Sigma$ we let

$$\vec{e}_\sigma = (\overbrace{0, \ldots, 0}^{\iota(\sigma)-1}, 1, \underbrace{0 \ldots, 0}_{|\Sigma|}).$$

Construction of A. For every V_i, associate each $\vec{v} \in V_i$ a distinct codeword of C, denoted by $C(\vec{v})$. For every $i \in [k]$ and $\vec{v} \in V_i$, introduce a vector $\vec{a}_{i,\vec{v}}$ as

- $\vec{a}^{(1)}_{i,\vec{v}} = \vec{v}$,
- $\vec{a}^{(2,j)}_{i,\vec{v}} = (\underbrace{\overbrace{\vec{0}, \ldots, \vec{0}}^{(i-1)}, \vec{e}_{C(\vec{v})[j]}, \vec{0}, \ldots, \vec{0}}_{k})$, for every $j \in [m]$,
- $\vec{a}^{(3)}_{i,\vec{v}} = \vec{e}_i$,
- $\vec{a}^{(4)}_{i,\vec{v}} = \vec{0}_m$.

And we let $A_i = \{\vec{a}_{i,\vec{v}} \mid \vec{v} \in V_i\}$ and $A = A_1 \cup \cdots \cup A_k$.

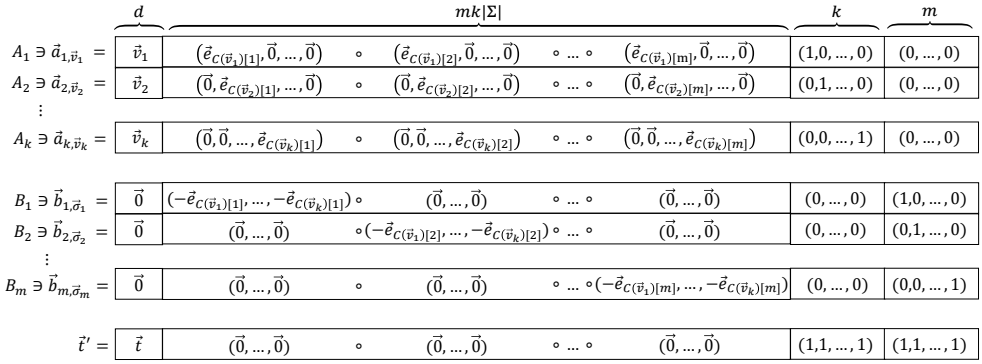

Figure 1 Illustration for the vectors of Lemma 19 in the completeness setting. We can choose each $\vec{b}_{j,\vec{\sigma}_j}$ as $\vec{\sigma}_j = (C(\vec{v}_1)[j], \cdots, C(\vec{v}_k)[j])$.

Construction of B. For every $j \in [m]$ and $\vec{\sigma} = (\sigma_1, \dots, \sigma_k) \in \Sigma^k$, introduce a vector $\vec{b}_{j,\vec{\sigma}}$ as

- $\vec{b}^{(1)}_{j,\vec{\sigma}} = \vec{0}_d$,
- $\vec{b}^{(2,j)}_{j,\vec{\sigma}} = (-\vec{e}_{\sigma_1}, \cdots - \vec{e}_{\sigma_k})$, $\vec{b}^{(2,j')}_{j,\vec{\sigma}} = \vec{0}_k$ for every $j' \in [m] \backslash \{j\}$,
- $\vec{b}^{(3)}_{j,\vec{\sigma}} = \vec{0}_k$,
- $\vec{b}^{(4)}_{j,\vec{\sigma}} = \vec{e}_j$.

We let $B_j = \{\vec{b}_{j,\vec{\sigma}} \mid \vec{\sigma} \in \Sigma^k\}$ and $B = B_1 \cup \cdots \cup B_m$.

Finally we set the target vector $\vec{t}'$ as

- $\vec{t}'^{(1)} = \vec{t}$,
- $\vec{t}'^{(2)} = \vec{0}_{mk|\Sigma|}$,
- $\vec{t}'^{(3)} = \vec{1}_k$,
- $\vec{t}'^{(4)} = \vec{1}_m$.

Time complexity. Producing each vector in A requires $O(d + mk|\Sigma| + km) = O(d + mk|\Sigma|)$ time, so the total time cost producing A is $O(dkn + mk^2 n|\Sigma|)$. Producing each vector in B also requires $O(d + mk|\Sigma|)$ time, and the total time cost producing B is $O(dm|\Sigma|^k + m^2 k|\Sigma|^{k+1})$. So the total time cost of this reduction is $O(dm^2k^2|\Sigma|(n + |\Sigma|^k))$.

Proof of (i). Suppose there exist $\vec{v}_1 \in V_1, \cdots, \vec{v}_k \in V_k$ satisfying $\sum_{i\in[k]} \vec{v}_i = \vec{t}$. For every $i \in [k]$ we choose a vector $\vec{a}_{i,\vec{v}_i} \in A_i$. And for every $j \in [m]$ we choose a vector $\vec{b}_{j,\vec{\sigma}_j} \in B_j$, where $\vec{\sigma}_j = (C(\vec{v}_1)[j], \dots, C(\vec{v}_m)[j]) \in \Sigma^k$. We now examine that $\sum_{i\in[k]} \vec{a}_{i,\vec{v}_i} + \sum_{j\in[m]} \vec{b}_{j,\vec{\sigma}_j} = \vec{t}'$ as:

- For the first block,

$$\sum_{i\in[k]} \vec{a}^{(1)}_{i,\vec{v}_i} + \sum_{j\in[m]} \vec{b}^{(1)}_{j,\vec{\sigma}_j} = \sum_{i\in[k]} \vec{v}_i + \sum_{j\in[m]} \vec{0}_d = \vec{t} = \vec{t}'^{(1)}.$$

- For every $j \in [m]$ the $(2,j)$-th block,

$$\begin{aligned}
\sum_{i\in[k]} \vec{a}^{(2,j)}_{i,\vec{v}_i} + \sum_{j'\in[m]} \vec{b}^{(2,j)}_{j',\vec{\sigma}_{j'}} &= \sum_{i\in[k]} \vec{a}^{(2,j)}_{i,\vec{v}_i} + \vec{b}^{(2,j)}_{j,\vec{\sigma}_j} \\
&= \sum_{i\in[k]} (\overbrace{\vec{0}, \dots, \vec{0}}^{i-1}, \vec{e}_{C(\vec{v}_i)[j]}, \vec{0}, \dots, \vec{0}) + (-\vec{e}_{C(\vec{v}_1)[j]}, \dots, -\vec{e}_{C(\vec{v}_k)[j]}) \\
&= (\vec{e}_{C(\vec{v}_1)[j]}, \dots, \vec{e}_{C(\vec{v}_k)[j]}) + (-\vec{e}_{C(\vec{v}_1)[j]}, \dots, -\vec{e}_{C(\vec{v}_k)[j]}) \\
&= \vec{0}_{k|\Sigma|} = \vec{t}'^{(2,j)}.
\end{aligned}$$

- For the third block,

$$\sum_{i\in[k]} \vec{a}^{(3)}_{i,\vec{v}_i} + \sum_{j\in[m]} \vec{b}^{(3)}_{j,\vec{\sigma}_j} = \sum_{i\in[k]} \vec{e}_i + \sum_{j\in[m]} \vec{0}_k = \vec{1}_k = \vec{t}'^{(3)}.$$

- For the fourth block,

$$\sum_{i\in[k]} \vec{a}^{(4)}_{i,\vec{v}_i} + \sum_{j\in[m]} \vec{b}^{(4)}_{j,\vec{\sigma}_j} = \sum_{i\in[k]} \vec{0}_m + \sum_{j\in[m]} \vec{e}_j = \vec{1}_m = \vec{t}'^{(4)}.$$

Proof of (ii). Suppose $X \subseteq A \dot{\cup} B$ and $\lambda : X \to \mathbb{F}_p^+$ such that $\sum_{\vec{x}\in X} \lambda(\vec{x})\vec{x} = \vec{t}'$. Observe the third block of the equation:

$$\sum_{\vec{x}\in X} \lambda(\vec{x})\vec{x}^{(3)} = \sum_{i\in[k]} \sum_{\vec{x}\in X\cap A_i} \lambda(\vec{x})\vec{e}_i = \vec{1}_m = \vec{t}'^{(3)}.$$

For every $i \in [k]$, $X \cap A_i$ must not be empty since $\sum_{\vec{x}\in X\cap A_i} \lambda(\vec{x}) = 1$. Also similarly by observing the fourth block it holds that $X \cap B_j$ must not be empty for every $j \in [m]$. Therefore $|X \cap A| \geq k$ and $|X \cap B| \geq m$.

Further suppose that any $\vec{v}_1 \in V_1, \ldots, \vec{v}_k \in V_k$ and $\alpha_1, \ldots \alpha_k \in \mathbb{F}_p^+$ must satisfy $\alpha_1\vec{v}_1 + \cdots + \alpha_k\vec{v}_k \neq \vec{t}$, we show that either $|X \cap A| \geq ck$ or $|X \cap B| \geq 2(1-\varepsilon)m$.

We let $I \subseteq [m]$ be the set of indices j that $X \cap B_j$ contains only one vector, i.e.,

$$I = \{j \in [m] : |X \cap B_j| = 1\}.$$

Since $|X \cap B_j| \geq 1$ for every $j \in [m]$, if $|I| \leq \varepsilon m$ then

$$|X \cap B| \geq \sum_{j\in[m]\setminus I} |X \cap B_j| \geq 2(1-\varepsilon)m$$

as desired. It remains to show that if $|I| > \varepsilon m$ then $|X \cap A| \geq ck$.

First we claim that there must be an $i \in [k]$ such that $X \cap A_i$ contains more than one vector. Otherwise suppose that $|X \cap A_i| = 1$ for every $i \in [k]$, let $\vec{a}_{i,\vec{v}_i} \in X \cap A_i$ be the unique vector in $X \cap A_i$. Recall that in the first block, vectors in $X \cap B$ are all zero, so the sum of vectors in X in the first block is

$$\sum_{\vec{x}\in X} \lambda(\vec{x})\vec{x}^{(1)} = \sum \lambda(\vec{a}_{i,\vec{v}_i})\vec{a}'^{(1)}_{i,\vec{v}_i} = \sum_{i\in[k]} \lambda(\vec{a}_{i,\vec{v}_i})\vec{v}_i = \vec{t} = \vec{t}'^{(1)}$$

This contradicts to our assumption that for all $\vec{v}_1 \in V_1, \ldots, \vec{v}_k \in V_k$ and $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_p^+$, $\sum_{i\in[k]} \alpha_i\vec{v}_i \neq \vec{t}$. Therefore, there must be such an index $i^* \in [k]$ that $|A'_{i^*}| > 1$.

Let $l > 1$ be the size of $X \cap A_{i^*}$, we next show that $l \geq ck$. Suppose that $X \cap A_{i^*} = \{\vec{a}_{i^*,\vec{v}_1}, \ldots, \vec{a}_{i^*,\vec{v}_l}\}$ where $\vec{v}_1, \ldots \vec{v}_l \in V_{i^*}$. We show in the following that the codeword set $\{C(\vec{v}_1), \ldots, C(\vec{v}_l)\}$ must collide on every $j \in I$. Fix any $j \in I$, let $\vec{b}_{j,\vec{\sigma}}$ be the unique vector in $X \cap B_j$, where $\vec{\sigma} = (\sigma_1, \ldots, \sigma_k)$. Recall that the $(2, j)$-th block of the resulting dimension consists of $k|\Sigma|$ coordinates, here we further break it down into k sub-blocks each of size $|\Sigma|$, and we focus on the $(2, j, i^*)$-th sub-block:

$$\begin{aligned}
\sum_{\vec{x}\in X} \lambda(\vec{x})\vec{x}^{(2,j,i^*)} &= \lambda(\vec{a}_{i^*,\vec{v}_1})\vec{a}^{(2,j,i^*)}_{i^*,\vec{v}_1} + \cdots + \lambda(\vec{a}_{i^*,\vec{v}_l})\vec{a}^{(2,j,i^*)}_{i^*,\vec{v}_l} + \lambda(\vec{b}_{j,\vec{\sigma}})\vec{b}^{(2,j,i^*)}_{j,\vec{\sigma}} \\
&= \lambda(\vec{a}_{i^*,\vec{v}_1})\vec{e}_{C(\vec{v}_1)[j]} + \cdots + \lambda(\vec{a}_{i^*,\vec{v}_l})\vec{e}_{C(\vec{v}_l)[j]} - \lambda(\vec{b}_{j,\vec{\sigma}})\vec{e}_{\sigma_{i^*}} \\
&= \vec{0}_{|\Sigma|} = \vec{t}'^{(2,j,i^*)}.
\end{aligned}$$

If $C(\vec{v}_1)[j], \ldots, C(\vec{v}_l)[j]$ are all distinct, the equation $\lambda(\vec{a}_{i^*,\vec{v}_1})\vec{e}_{C(\vec{v}_1)[j]} + \cdots + \lambda(\vec{a}_{i^*,\vec{v}_l})\vec{e}_{C(\vec{v}_l)[j]} - \lambda(\vec{b}_{j,\vec{\sigma}})\vec{e}_{\sigma_{i^*}} = \vec{0}_{|\Sigma|}$ must not be satisfied since $l > 1$ and the λ's are nonzero. Therefore $\{C(\vec{v}_1), \ldots, C(\vec{v}_l)\}$ must collide on the j-th coordinate.

If $|I| > \varepsilon m$ then $\{C(\vec{v}_1), \ldots, C(\vec{v}_l)\}$ collide on more than εm coordinates, by the definition of collision number, it holds that $|\{C(\vec{v}_1), \ldots, C(\vec{v}_l)\}| \geq \mathrm{Col}_\varepsilon(C) \geq ck$. And thus $|X \cap A| \geq |X \cap A_{i^*}| \geq ck$. ◀

Since the codes (with good collision number) we construct has codeword length $m = O(k^2 \log k)$ (Lemma 17) much greater than k, the above construction cannot directly lead to a gap-creating reduction for k-MLD. To settle this, intuitively we further duplicate the vector sets $A_1, \ldots, A_k$ several times into m vector sets. This leads to our gap creating reduction as follows.

▶ **Theorem 20.** *For any $0 < \varepsilon < 1$, there is a randomized reduction which on input k sets of length-d vectors $V_1, \cdots, V_k \subseteq \mathbb{F}_p^d$ each of size n and a target vector $\vec{t} \in \mathbb{F}_p^d$ outputs k' vector sets $U_1, \ldots, U_{k'} \subseteq \mathbb{F}_p^D$ and a target vector $\vec{t}' \in \mathbb{F}_p^D$ with $k' = O(k^2 \log k)$ and $D = O(k'd + k'^2 n^{1/k})$ in $O(d2^{O(k)}n^{1.01})$ time such that*

- **(i)** *If there exist $\vec{v}_1 \in V_1, \ldots, \vec{v}_k \in V_k$ such that $\sum_{i\in[k]} \vec{v}_i = \vec{t}$, then there exists $\vec{u}_1 \in U_1, \ldots, \vec{u}_{k'} \in U_{k'}$ with their sum being $\vec{t}'$.*
- **(ii)** *If any $\vec{v}_1 \in V_1, \ldots, \vec{v}_k \in V_k$ and $\alpha_1, \ldots \alpha_k \in \mathbb{F}_p^+$ must satisfy $\alpha_1\vec{v}_1 + \cdots + \alpha_k\vec{v}_k \neq \vec{t}$, then any $X \subseteq \bigcup_{i\in[k']} U_i$ and $\lambda : X \to \mathbb{F}_p^+$ such that $\sum_{\vec{x}\in X} \lambda(\vec{x})\vec{x} = \vec{t}'$ must satisfy $|X| \geq (\frac{3}{2} - \varepsilon)k'$.*

▶ Remark 21. Consider the k-VectorSum$_q$ problem in [36], whose definition is identical to k-MLD$_q$ except that it requires all the coefficients being 1. A closer look at our reduction shows that it can directly create a gap of almost $(q+1)/2$ for k-VectorSum$_q$ rather than almost $\frac{3}{2}$ in the k-MLD$_q$ case.

5 Lower Bounds for Gap-k-NCP and Other Problems

In this section, we show the reduction described in the previous sections implies improved running time lower bounds for various problems under ETH.

5.1 Maximum Likelihood Decoding and Nearest Codeword Problem

In [14], Bhattacharyya, Ghoshal, Karthik and Manurangsi presented a gap amplification procedure for Gap-k-MLD$_p$. Although they only discussed the procedure on the binary field, it's straightforward to see the procedure also works for Gap-k-MLD$_p$ instances over all $\mathbb{F}_p$. Formally,

▶ **Theorem 22** (Generalization of Lemma 4.5 in [14])**.** *For integers $k_1, k_2 > 0$, $k' = k_2 + k_1k_2$ and reals $\gamma_1, \gamma_2 > 1$, $\gamma' \geq \gamma_1\gamma_2(1 - \frac{1}{k_1})$, there is a polynomial time algorithm that on input 2 vector sets $U \subseteq \mathbb{F}_p^{m_1}, V \subseteq \mathbb{F}_p^{m_2}$, $|U| = n_1, |V| = n_2$, two target vectors $\vec{t} \in \mathbb{F}_p^{m_1}, \vec{s} \in \mathbb{F}_p^{m_2}$, outputs a vector set $W \subseteq \mathbb{F}_p^{m_2+n_1m_1}$ and a target vector $\vec{t}' \in \mathbb{F}_p^{m_2+n_1m_1}$ satisfies:*

- *If $(U, \vec{t})$ is a YES instance of γ_1-Gap-k_1-MLD$_p$ instance and $(V, \vec{s})$ is a YES instance of γ_2-Gap-k_2-MLD$_p$ instance, then $(W, \vec{t}')$ is a YES instance of γ'-Gap-k'-MLD$_p$.*
- *If $(U, \vec{t})$ is a NO instance of γ_1-Gap-k_1-MLD$_p$ instance and $(V, \vec{s})$ is a NO instance of γ_2-Gap-k_2-MLD$_p$ instance, then $(W, \vec{t}')$ is a NO instance of γ'-Gap-k'-MLD$_p$.*

Readers seeking for a formal proof is referred to [14, Section 4.2].

5.1.1 ETH-based Running Time Lower Bound

Taking a closer look at the reduction from 3-SAT to k-VectorSum in [36, Theorem 11], we observe that by applying a minor modification, their reduction can actually have soundness condition as:

- If ϕ is not satisfiable, then for any $\vec{v}_1 \in V_1, \ldots, \vec{v}_k \in V_k$ and $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_p^+$, $\Sigma_{i=1}^k \alpha_i \vec{v}_i \neq \vec{t}$.

The modification is simply appending a vector $(0^{i-1} \circ 1 \circ 0^{k-i})$ to each vector in V_i, for all $1 \leq i \leq k$. Then, the target vector is changed from a zero vector to $\vec{t} = 0^d \circ 1^k$. Completeness of their reduction is trivially preserved. For soundness we claim, we note that for any $\vec{v}_1 \in V_1, \cdots, \vec{v}_k \in V_k$ and $\alpha_1, \cdots, \alpha_k \in \mathbb{F}_p$, if $\Sigma_{i=1}^k \alpha_i \vec{v}_i = \vec{t}$, then $\alpha_1 = \cdots = \alpha_k = 1$.

By strengthening the soundness condition in [36], we obtain exactly the restricted version of k-MLD$_p$ in the previous sections. Combining with their soundness for k-VectorSum, we obtain the following hardness result for k-MLD$_p$ as:

▶ **Theorem 23** (Theorem 11 in [36]). *Assuming ETH, for any constant integer p, k-MLD$_p$ has no $n^{o(k)}$-time algorithm.*

The parameterized MLD and NCP are equivalent in the sense that there exist reductions preserving the solution size in both direction. Recall that Theorem 20 showed a reduction from k-MLD$_p$ to $(3/2-\varepsilon)$-Gap-k'-MLD$_p$ with $k' = k^2 \log k$ and $\varepsilon > 0$. Combining running time lower bound in Theorem 23, we have:

▶ **Theorem 24.** *Assuming randomized ETH, for any constant integer p, constant $1 < \gamma < \frac{3}{2}$, γ-Gap-k-MLD$_p$ and γ-Gap-k-NCP$_p$ has no $O_k(n^{o(\sqrt{k/\log k})})$-time algorithm.*

By applying Theorem 22 to the gap instance itself $O(\log\log\gamma)$ times, we can obtain the ETH-based time lower bound for approximating parameterized MLD and NCP to any constant factor.

▶ **Corollary 25.** *Assuming ETH, for any constant integer p and constant $\gamma > 1$, γ-Gap-k-MLD$_p$ and γ-Gap-k-NCP$_p$ have no $O_k(n^{o(k^\epsilon)})$ time algorithm, where $\epsilon = \frac{1}{\mathit{polylog}(\gamma)}$ is a constant.*

5.2 Minimum Distance Problem

The reduction from Gap-k-NCP to Gap-k-MDP in [10] is as follows.

▶ **Theorem 26** ([10], Theorem 3.1 and 3.3). *For any prime power $p \geq 2$ there is a randomized reduction from $(4p)$-Gap-k-NCP$_p$ to $\frac{4p}{4p-1}$-Gap-k'-MDP$_p$ runs in polynomial time with $k' = O(k)$.*

Use our gap-creating reduction for Gap-k-MLD, and apply the gap amplification for a constant number of times, then use Theorem 26 to reduce to Gap-k'-MDP and self-tensoring the instance for a constant number of times, we have:

▶ **Corollary 27.** *Assuming randomized ETH, for any prime power $p \geq 2$ and real number $\gamma > 1$, γ-Gap-k-MDP$_p$ has no $O_k(n^{o(k^\epsilon)})$ time algorithm, where $\epsilon = \Theta(\frac{1}{p \log \gamma \cdot \mathsf{polylog}(p)})$.*

5.3 Closest Vector Problem

We need a reduction from (2γ)-Gap-k-MLD$_q$ to γ-Gap-$2k$-CVP$_p$ from [13].

▶ **Theorem 28** ([13], Theorem 7.2). *For any real numbers $\gamma, p \geq 1$ and a prime number $q > 2\gamma$, there is a reduction from (2γ)-Gap-k-MLD$_q$ to γ-Gap-k'-CVP$_p$ runs in polynomial time, where $k' = 2k$.*

Use our gap-creating reduction for Gap-k-MLD, and apply gap amplification for a constant number of times to obtain 2γ-gap, then use Theorem 28 to reduce to Gap-k'-CVP, we have:

▶ **Corollary 29.** *Assuming ETH, there exists constant $c > 0$, for any real numbers $p, \gamma \geq 1$, γ-Gap-k-CVP$_p$ has no $O_k(n^{o(k^\epsilon)})$ time algorithm, where $\epsilon = \Theta(\frac{1}{\gamma^c})$.*

5.4 Shortest Vector Problem

Combining our work with [10], we show two ways of obtaining running time lower bound for γ-Gap-k-SVP$_p$. The first way reduces from Gap-k-CVP$_p$, obtaining lower bound for only a fixed constant ratio and all l_p norms where $p \geq 1$. The second way reduces from Gap-k-NCP$_q$, obtaining lower bound for all constant ratio and all l_p norms except for l_1.

5.4.1 Reduction From Gap-k-CVP$_p$

▶ **Theorem 30** ([10], Theorem 4.1 and 4.3, modified). *For any real numbers $p \geq 1$ and $\gamma' \in [1, 2)$ there exist a real number $\gamma \geq 1$ [7] and a reduction from γ-Gap-k-CVP$_p$ to γ'-Gap-k'-SVP$_p$ runs in polynomial time, where $k' \leq \gamma k$.*

Use Corollary 29 to obtain a γ_0-gap CVP instance, where γ_0 fits requirement in Theorem 30, then use Theorem 30 we have:

▶ **Corollary 31.** *Assuming randomized ETH, for any real numbers $p \geq 1$ and $\gamma \in [1, 2)$, γ-Gap-k-SVP$_p$ has no $O_k(n^{o(k^\epsilon)})$ time algorithm,where $0 < \epsilon < 1$ is some constant that depends on p and γ.*

5.4.2 Reduction From Gap-k-NCP$_2$

▶ **Theorem 32** ([10], Lemma 5.1 and Theorem 5.2, modified). *There exists a constant real $\mu \geq 1$ such that, for any real numbers $p > 1$ and $\gamma' \geq 1$, there exists a reduction from μ-Gap-k-NCP$_2$ to γ'-Gap-k'-SVP$_p$ runs in polynomial time, where $k' = O(k^c)$, $c > 1$ is a constant only depends on p and γ'*[8].

The reduction in Theorem 32 in fact proceeds in two steps: first reduces μ-Gap-k-NCP$_2$ to γ'-Gap-k'-SVP$_p$ for some fixed $\gamma' > 1$ with $k' < \mu k$ (while having some additional properties for the second step), then use a tensor technique to amplify the gap to any constant.

▶ **Corollary 33.** *Assuming randomized ETH, for any real numbers $p > 1$ and $\gamma \geq 1$, γ-Gap-k-SVP$_p$ has no $O_k(n^{o(k^\epsilon)})$ time algorithm, where $0 < \epsilon < 1$ is some constant that depends on p and γ.*

---

[7] $\gamma = (\max\left(12/\varepsilon, \frac{1}{(1+\varepsilon/2)^{1/p}-1}\right))^p$ where $\varepsilon = (\gamma')^{-1} - 1/2 > 0$.

[8] There are two problems here about the parameter blow-up, one is that $k' \leq (\mu k)^{O(1)}$ due to the Haviv-Regev "tensoring" step of SVP, the other is that to achieve final gap γ', the gap μ of NCP needs to satisfy $\frac{\mu}{2^p+1+\alpha\mu} > \gamma'$ for some $1/2 + 2^{-p} < \alpha < 1$, causing a polynomial blow-up of parameter to achieve such μ.

6 Conclusion

We have presented new ETH-based lower bounds for approximating parameterized nearest codeword problem and its related problems, improving upon the previous results from [10,13]. Our reduction technique is also simpler and more straight forward than the one used in [13]. However, our results still do not match the lower bound for constant Gap-k-NCP based on Gap-ETH [38]. A natural open problem is to close this gap by proving a stronger lower bound under an assumption that is weaker than Gap-ETH, such as constant Gap-k-Clique has no $n^{o(k)}$-time algorithm. This would be a key step towards understanding the fine-grained complexity of parameterized nearest codeword problem and its variants.

▶ **Open Problem 34.** *Prove $n^{o(k)}$ time lower bound of approximating k-NCP$_p$ or its related problems to any constant factor under assumptions weaker than Gap-ETH.*

To show such a result, as pointed out in [38], one might need to come up with a better "one-shot proof" that gives arbitrary constant factors without tensoring, and with linear parameter growth.

In this paper, we give a new method of composing *threshold graph* with vector problems to yield hardness of approximation results. We showed the limitation of analyzing collision number of a code from its relative distance in [32,37], and improved the analysis to bypass the limitation above. It might be interesting to consider whether this result can be further improved to yield threshold graph with better parameters, or some limitations of our method can be discovered, formally:

▶ **Open Problem 35.** *Give a better construction of strong threshold graph in Section 1.2 with $h = \Omega(k)$ and $m = O(k)$, or show that such graphs do not exist.*

References

1 Divesh Aggarwal, Huck Bennett, Zvika Brakerski, Alexander Golovnev, Rajendra Kumar, Zeyong Li, Spencer Peters, Noah Stephens-Davidowitz, and Vinod Vaikuntanathan. Lattice problems beyond polynomial time. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1516–1526. ACM, 2023. doi:10.1145/3564246.3585227.

2 Divesh Aggarwal, Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. Fine-grained hardness of CVP(P) - everything that we can prove (and nothing else). In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021*, pages 1816–1835. SIAM, 2021. doi:10.1137/1.9781611976465.109.

3 Divesh Aggarwal and Noah Stephens-Davidowitz. (gap/s)eth hardness of SVP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 228–238. ACM, 2018. doi:10.1145/3188745.3188840.

4 Miklós Ajtai. The shortest vector problem in L_2 is *NP*-hard for randomized reductions (extended abstract). In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 10–19. ACM, 1998. doi:10.1145/276698.276705.

5 Noga Alon, Rina Panigrahy, and Sergey Yekhanin. Deterministic approximation algorithms for the nearest codeword problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009. Proceedings*, volume 5687 of *Lecture Notes in Computer Science*, pages 339–351. Springer, 2009. doi:10.1007/978-3-642-03685-9_26.

6 Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. doi:10.1006/jcss.1997.1472.

7 Per Austrin and Subhash Khot. A simple deterministic reduction for the gap minimum distance of code problem. *IEEE Trans. Inf. Theory*, 60(10):6636–6645, 2014. doi:10.1109/TIT.2014.2340869.

8 S Barg. Some new np-complete coding problems. *Problemy Peredachi Informatsii*, 30(3):23–28, 1994.

9 Huck Bennett. The complexity of the shortest vector problem. *SIGACT News*, 54(1):37–61, 2023. doi:10.1145/3586165.3586172.

10 Huck Bennett, Mahdi Cheraghchi, Venkatesan Guruswami, and João Ribeiro. Parameterized inapproximability of the minimum distance problem over all fields and the shortest vector problem in all ℓ_p norms. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023*, pages 553–566. ACM, 2023. doi:10.1145/3564246.3585214.

11 Huck Bennett, Chris Peikert, and Yi Tang. Improved hardness of BDD and SVP under gap-(s)eth. In *13th Innovations in Theoretical Computer Science Conference, ITCS 2022*, volume 215 of *LIPIcs*, pages 19:1–19:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.19.

12 Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Trans. Inf. Theory*, 24(3):384–386, 1978. doi:10.1109/TIT.1978.1055873.

13 Arnab Bhattacharyya, Édouard Bonnet, László Egri, Suprovat Ghoshal, Karthik C. S., Bingkai Lin, Pasin Manurangsi, and Dániel Marx. Parameterized intractability of even set and shortest vector problem. *J. ACM*, 68(3):16:1–16:40, 2021. doi:10.1145/3444942.

14 Arnab Bhattacharyya, Suprovat Ghoshal, Karthik C. S., and Pasin Manurangsi. Parameterized intractability of even set and shortest vector problem from gap-eth. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, volume 107 of *LIPIcs*, pages 17:1–17:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.ICALP.2018.17.

15 Boris Bukh, Karthik C. S., and Bhargav Narayanan. Applications of random algebraic constructions to hardness of approximation. In *62nd IEEE Annual Symposium on Foundations of Computer Science*, pages 237–244. IEEE, 2021. doi:10.1109/FOCS52979.2021.00032.

16 Yijia Chen and Bingkai Lin. The constant inapproximability of the parameterized dominating set problem. *SIAM J. Comput.*, 48(2):513–533, 2019. doi:10.1137/17M1127211.

17 Qi Cheng. Hard problems of algebraic geometry codes. *IEEE Trans. Inf. Theory*, 54(1):402–406, 2008. doi:10.1109/TIT.2007.911213.

18 Qi Cheng and Daqing Wan. A deterministic reduction for the gap minimum distance problem. *IEEE Trans. Inf. Theory*, 58(11):6935–6941, 2012. doi:10.1109/TIT.2012.2209198.

19 Irit Dinur. Mildly exponential reduction from gap-3sat to polynomial-gap label-cover. In *Electronic colloquium on computational complexity ECCC; research reports, surveys and books in computational complexity*, 2016.

20 Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is np-hard. *Comb.*, 23(2):205–243, 2003. doi:10.1007/s00493-003-0019-y.

21 Rodney G. Downey, Michael R. Fellows, Alexander Vardy, and Geoff Whittle. The parametrized complexity of some fundamental problems in coding theory. *SIAM J. Comput.*, 29(2):545–570, 1999. doi:10.1137/S0097539797323571.

22 Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Inf. Theory*, 49(1):22–37, 2003. doi:10.1109/TIT.2002.806118.

23 Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.

24 Uriel Feige and Daniele Micciancio. The inapproximability of lattice and coding problems with preprocessing. *J. Comput. Syst. Sci.*, 69(1):45–67, 2004. doi:10.1016/j.jcss.2004.01.002.

25 Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inf. Process. Lett.*, 71(2):55–61, 1999. doi:10.1016/S0020-0190(99)00083-6.

26 Venkatesan Guruswami, Xuandi Ren, and Sai Sandeep. Baby pih: Parameterized inapproximability of min csp. *Electron. Colloquium Comput. Complex.*, TR23-155, 2023. arXiv:TR23-155.

27 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory. *Draft available at* https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/, 2(1), 2023.

28 Venkatesan Guruswami and Alexander Vardy. Maximum-likelihood decoding of reed-solomon codes is np-hard. *IEEE Trans. Inf. Theory*, 51(7):2249–2256, 2005. doi:10.1109/TIT.2005.850102.

29 Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. *Theory Comput.*, 8(1):513–531, 2012. doi:10.4086/toc.2012.v008a023.

30 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

31 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

32 Karthik C. S. and Inbal Livni Navon. On hardness of approximation of parameterized set cover and label cover: Threshold graphs from error correcting codes. In *4th Symposium on Simplicity in Algorithms*, pages 210–223. SIAM, 2021. doi:10.1137/1.9781611976496.24.

33 Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *J. ACM*, 52(5):789–808, 2005. doi:10.1145/1089023.1089027.

34 Bingkai Lin. The parameterized complexity of the k-biclique problem. *J. ACM*, 65(5):34:1–34:23, 2018. doi:10.1145/3212622.

35 Bingkai Lin. A simple gap-producing reduction for the parameterized set cover problem. In *46th International Colloquium on Automata, Languages, and Programming*, volume 132 of *LIPIcs*, pages 81:1–81:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.81.

36 Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. On lower bounds of approximating parameterized k-clique. In *49th International Colloquium on Automata, Languages, and Programming*, volume 229 of *LIPIcs*, pages 90:1–90:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.90.

37 Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. Constant approximating parameterized k-setcover is w[2]-hard. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms*, pages 3305–3316. SIAM, 2023. doi:10.1137/1.9781611977554.ch126.

38 Pasin Manurangsi. Tight running time lower bounds for strong inapproximability of maximum k-coverage, unique set cover and related problems (via t-wise agreement testing theorem). In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, pages 62–81. SIAM, 2020. doi:10.1137/1.9781611975994.5.

39 Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense csps. *arXiv preprint arXiv:1607.02986*, 2016.

40 Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput.*, 30(6):2008–2035, 2000. doi:10.1137/S0097539700373039.

41 Daniele Micciancio. The hardness of the closest vector problem with preprocessing. *IEEE Trans. Inf. Theory*, 47(3):1212–1215, 2001. doi:10.1109/18.915688.

42 Daniele Micciancio. Locally dense codes. In *IEEE 29th Conference on Computational Complexity*, pages 90–97. IEEE Computer Society, 2014. doi:10.1109/CCC.2014.17.

43 Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. doi:10.1137/S0097539795280895.

44 Oded Regev. Improved inapproximability of lattice and coding problems with preprocessing. *IEEE Trans. Inf. Theory*, 50(9):2031–2037, 2004. doi:10.1109/TIT.2004.833350.

45 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009. doi:10.1145/1568318.1568324.

46 Oded Regev. The learning with errors problem (invited survey). In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010*, pages 191–204. IEEE Computer Society, 2010. doi:10.1109/CCC.2010.26.

47 Jacques Stern. Approximating the number of error locations within a constant ratio is np-complete. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 10th International Symposium, AAECC-10, 1993, Proceedings*, volume 673 of *Lecture Notes in Computer Science*, pages 325–331. Springer, 1993. doi:10.1007/3-540-56686-4_54.

48 Alexander Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. Inf. Theory*, 43(6):1757–1766, 1997. doi:10.1109/18.641542.