Electronic Colloquium on Computational Complexity, Report No. 115 (2019)

Parameterized Intractability of Even Set and Shortest Vector Problem

Arnab Bhattacharyya[*] Édouard Bonnet[†] László Egri[‡] Suprovat Ghoshal[§]

Karthik C. S.[¶] Bingkai Lin[||] Pasin Manurangsi[**] Dániel Marx[††]

September 4, 2019

Abstract

The *k-Even Set* problem is a parameterized variant of the *Minimum Distance Problem* of linear codes over $\mathbb{F}_2$, which can be stated as follows: given a generator matrix $\mathbf{A}$ and an integer k, determine whether the code generated by $\mathbf{A}$ has distance at most k, or in other words, whether there is a nonzero vector $\mathbf{x}$ such that $\mathbf{A}\mathbf{x}$ has at most k nonzero coordinates. The question of whether k-Even Set is fixed parameter tractable (FPT) parameterized by the distance k has been repeatedly raised in literature; in fact, it is one of the few remaining open questions from the seminal book of Downey and Fellows (1999). In this work, we show that k-Even Set is W[1]-hard under randomized reductions.

We also consider the parameterized *k-Shortest Vector Problem (SVP)*, in which we are given a lattice whose basis vectors are integral and an integer k, and the goal is to determine whether the norm of the shortest vector (in the ℓ_p norm for some fixed p) is at most k. Similar to k-Even Set, understanding the complexity of this problem is also a long-standing open question in the field of Parameterized Complexity. We show that, for any $p > 1$, k-SVP is W[1]-hard to approximate (under randomized reductions) to some constant factor.

[*] National University of Singapore. Email: arnabb@nus.edu.sg

[†] CNRS, École Normale Supérieure de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668. Email: edouard.bonnet@ens-lyon.fr

[‡] Indiana State University, Terre Haute. Email: laszlo.egri@mail.mcgill.ca

[§] Indian Institute of Science. Email: suprovat@iisc.ac.in

[¶] Weizmann Institute of Science. Email: karthik.srikanta@weizmann.ac.il

[||] Nanjing University. Email: lin@nju.edu.cn

[**] University of California, Berkeley. Email: pasin@berkeley.edu

[††] Institute for Computer Science and Control, Hungarian Academy of Sciences. Email: dmarx@cs.bme.hu

ISSN 1433-8092

1 Introduction

The study of error-correcting codes gives rise to many interesting computational problems. One of the most fundamental among these is the problem of computing the distance of a linear code. In this problem, which is commonly referred to as the *Minimum Distance Problem (MDP)*, we are given as input a generator matrix $\mathbf{A} \in \mathbb{F}_2^{n \times m}$ of a binary[1] linear code and an integer k. The goal is to determine whether the code has distance at most k. Recall that the distance of a linear code is $\min_{\mathbf{0} \neq \mathbf{x} \in \mathbb{F}_2^m} \|\mathbf{A}\mathbf{x}\|_0$ where $\| \cdot \|_0$ denote the 0-norm (aka the Hamming norm).

To see the fundamental nature of MDP, let us discuss two other natural ways of arriving at (equivalent formulations of) this problem. MDP has the following well-known dual formulation: the minimum distance of the code generated by $\mathbf{A} \in \mathbb{F}_2^{n \times m}$ can be also expressed as $\min_{\mathbf{0} \neq \mathbf{y} \in \mathbb{F}_2^m, \mathbf{A}^\perp \mathbf{y} = 0} \|\mathbf{y}\|_0$, where $\mathbf{A}^\perp$ is the orthogonal complement of $\mathbf{A}$. In other words, finding the minimum distance of the code is equivalent to finding the minimum set of linearly dependent vectors among the column vectors of $\mathbf{A}^\perp$. Thus, MDP is equivalent to the the Linear Dependent Set problem on vectors over $\mathbb{F}_2$ or, using the language of matroid theory, solving the Shortest Circuit problem on a represented binary matroid.

One can arrive at a more combinatorial formulation of the problem as a variant of the Hitting Set problem. Given a set system $\mathcal{S}$ over a universe U and an integer k, the Hitting Set problem asks for a k-element subset X of U such that $|S \cap X| \neq 0$ for every $S \in \mathcal{S}$. Hitting Set is a basic combinatorial optimization that is well studied (often under the dual formulation Set Cover) in the approximation algorithms and the parameterized complexity literature. More restrictive versions of the problem are the Exact Hitting Set problem, where we require $|S \cap X| = 1$, and the Odd Set problem, where we require $|S \cap X|$ to be odd. By analogy, we can define the Even Set problem, where we require $|S \cap X|$ to be even, but in this case we need to add the requirement $X \neq \emptyset$ to avoid the trivial solution. While Hitting Set, Exact Hitting Set, and Odd Set are known to be W[1]-hard parameterized by k, Even Set can be easily seen to be equivalent to MDP (in the dual formulation of MDP, the rows of $\mathbf{A}^\perp$ play the same role as the sets in $\mathcal{S}$).

The study of this problem dates back to at least 1978 when Berlekamp et al. [BMvT78] conjectured that it is NP-hard. This conjecture remained open for almost two decades until it was positively resolved by Vardy [Var97a, Var97b]. Later, Dumer et al. [DMS03] strengthened this intractability result by showing that even *approximately* computing the minimum distance of the code is hard. Specifically, they showed that, unless NP $=$ RP, no polynomial time algorithm can distinguish between a code with distance at most k and one whose distance is greater than $\gamma \cdot k$ for any constant $\gamma \geqslant 1$. Furthermore, under stronger assumptions, the ratio can be improved to superconstants and even almost polynomial. Dumer et al.'s result has been subsequently derandomized by Cheng and Wan [CW12] and further simplified by Austrin and Khot [AK14] and Micciancio [Mic14].

While the aforementioned intractability results rule out not only efficient algorithms but also efficient approximation algorithms for MDP, there is another popular technique in coping with NP-hardness of problems which is not yet ruled out by the known results: *parameteriza-*

[1] Note that MDP can be defined over larger fields as well; we discuss more about this in Section 8.

tion.

In parameterized problems, part of the input is an integer that is designated as the parameter of the problem, and the goal is now not to find a polynomial time algorithm but a *fixed parameter tractable (FPT)* algorithm. This is an algorithm whose running time can be upper bounded by some (computable) function of the parameter in addition to some polynomial in the input length. Specifically, for MDP, its parameterized variant[2] k-MDP has k as the parameter and the question is whether there exists an algorithm that can decide if the code generated by $\mathbf{A}$ has distance at most k in time $f(k) \cdot \text{poly}(mn)$ where f can be any computable function that depends only on k.

Note that k-MDP can be solved in $n^{O(k)}$ time. This can be easily seen in the dual formulation, as we can enumerate through all vectors $\mathbf{y}$ with Hamming norm at most k and check whether $\mathbf{A}^{\perp}\mathbf{y} = \mathbf{0}$. In Parameterized Complexity language, this means that k-MDP belongs to the class XP.

The parameterized complexity of k-MDP was first questioned by Downey et al. [DFVW99], who showed that parameterized variants of several other coding-theoretic problems, including the Nearest Codeword Problem and the Nearest Vector Problem[3] which we will discuss in more details in Section 1.1.1, are W[1]-hard. Thereby, assuming the widely believed W[1] $\neq$ FPT hypothesis, these problems are rendered intractable from the parameterized perspective. Unfortunately, Downey et al. fell short of proving such hardness for k-MDP and left it as an open problem:

Open Question 1.1. *Is k-MDP fixed parameter tractable?*

Although almost two decades have passed, the above question remains unresolved to this day, despite receiving significant attention from the community. In particular, the problem was listed as an open question in the seminal 1999 book of Downey and Fellows [DF99] and has been reiterated numerous times over the years [DGMS07, FGMS12, GKS12, FM12, DF13, CFJ+14, CFK+15, BGGS16, CFHW17, Maj17]. This problem is one of the few questions that remained open from the original list of Downey and Fellows [DF99]. In fact, in their second book [DF13], Downey and Fellows even include this problem as one of the six[4] "most infamous" open questions in the area of Parameterized Complexity.

Another question posted in Downey et al.'s work [DFVW99] that remains open is the parameterized *Shortest Vector Problem (k-SVP)* in lattices. The input of k-SVP (in the ℓ_p norm) is an integer $k \in \mathbb{N}$ and a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$ representing the basis of a lattice, and we want to determine whether the shortest (non-zero) vector in the lattice has length at most k, i.e., whether $\min_{\mathbf{0} \neq \mathbf{x} \in \mathbb{Z}^m} \|\mathbf{A}\mathbf{x}\|_p \leqslant k$. Again, k is the parameter of the problem. It should also be noted here that, similar to [DFVW99], we require the basis of the lattice to be integer valued, which is sometimes not enforced in literature (e.g. [vEB81, Ajt98]). This is because, if $\mathbf{A}$ is allowed to be any matrix in $\mathbb{R}^{n \times m}$, then parameterization is meaningless because we can simply scale $\mathbf{A}$ down by a large multiplicative factor.

[2] Throughout Sections 1 and 2, for a computational problem Π, we denote its parameterized variant by k-Π, where k is the parameter of the problem.

[3] The Nearest Vector Problem is also referred to in the literature as the Closest Vector Problem.

[4] So far, two of the six problems have been resolved: that of parameterized complexity of k-Biclique [Lin18] and that of parameterized approximability of k-Dominating Set [KLM19].

The (non-parameterized) Shortest Vector Problem (SVP) has been intensively studied, motivated partly due to the fact that both algorithms and hardness results for the problem have numerous applications. Specifically, the celebrated LLL algorithm for SVP [LLL82] can be used to factor rational polynomials, and to solve integer programming (parameterized by the number of unknowns) [Len83] and many other computational number-theoretic problems (see e.g. [NV10]). Furthermore, the hardness of (approximating) SVP has been used as the basis of several cryptographic constructions [Ajt98, AD97, Reg03, Reg05]. Since these topics are out of scope of our paper, we refer the interested readers to the following surveys for more details: [Reg06, MR09, NV10, Reg10].

On the computational hardness side of the problem, van Emde-Boas [vEB81] was the first to show that SVP is NP-hard for the ℓ_∞ norm, but left open the question of whether SVP on the ℓ_p norm for $1 \leqslant p < \infty$ is NP-hard. It was not until a decade and a half later that Ajtai [Ajt96] showed, under a randomized reduction, that SVP for the ℓ_2 norm is also NP-hard; in fact, Ajtai's hardness result holds not only for exact algorithms but also for $(1 + o(1))$-approximation algorithms as well. The $o(1)$ term in the inapproximability ratio was then improved in a subsequent work of Cai and Nerurkar [CN99]. Finally, Micciancio [Mic00] managed to achieve a factor that is bounded away from one. Specifically, Micciancio [Mic00] showed (again under randomized reductions) that SVP on the ℓ_p norm is NP-hard to approximate to within a factor of $\sqrt[p]{2}$ for every $1 \leqslant p < \infty$. Khot [Kho05] later improved the ratio to any constant, and even to $2^{\log^{1/2-\varepsilon}(nm)}$ under a stronger assumption. Haviv and Regev [HR07] subsequently simplified the gap amplification step of Khot and, in the process, improved the ratio to almost polynomial. We note that both Khot's and Haviv-Regev reductions are also randomized and it is still open to find a deterministic NP-hardness reduction for SVP in the ℓ_p norms for $1 \leqslant p < \infty$ (see [Mic12]); we emphasize here that such a reduction is not known even for the exact (not approximate) version of the problem. For the ℓ_∞ norm, the following stronger result due to Dinur [Din02] is known: SVP in the ℓ_∞ norm is NP-hard to approximate to within $n^{\Omega(1/\log\log n)}$ factor (under a *deterministic* reduction).

Very recently, fine-grained studies of SVP have been initiated [BGS17, AS18]. The authors of [BGS17, AS18] showed that SVP for any ℓ_p norm cannot be solved (or even approximated to some constant strictly greater than one) in subexponential time assuming the existence of a certain family of lattices[5] and the (randomized) *Gap Exponential Time Hypothesis (Gap-ETH)* [Din16, MR16], which states that no randomized subexponential time algorithm can distinguish between a satisfiable 3-CNF formula and one which is only 0.99-satisfiable.

As with MDP, Downey et al. [DFVW99] were the first to question the parameterized tractability of k-SVP (for the ℓ_2 norm). Once again, Downey and Fellows included k-SVP as one of the open problems in both of their books [DF99, DF13]. As with Open Question 1.1, this question remains unresolved to this day:

Open Question 1.2. *Is k-SVP fixed parameter tractable?*

We remark here that, similar to k-MDP, k-SVP also belongs to XP, as we can enumerate over all vectors with norm at most k and check whether it belongs to the given lattice. There are only $(mn)^{O(k^p)}$ such vectors, and the lattice membership of a given vector can be decided

[5] This additional assumption is only needed for $1 \leqslant p \leqslant 2$. For $p > 2$, their hardness is conditional only on Gap-ETH.

in polynomial time (e.g., see page 18 [MG12]). Hence, this is an $(nm)^{O(k^p)}$-time algorithm for k-SVP.

1.1 Our Results

The main result of this paper is a resolution to the previously mentioned Open Questions 1.1 and 1.2: more specifically, we prove that k-MDP and k-SVP (on ℓ_p norm for any $p > 1$) are W[1]-hard under randomized reductions. In fact, our result is stronger than stated here as we rule out not only exact FPT algorithms but also FPT *approximation* algorithms as well. In particular, all of our results use the W[1]-hardness of approximating the k-BICLIQUE problem recently proved by Lin [Lin18] as a starting point.

With this in mind, we can state our results starting with the parameterized intractability of k-MDP, more concretely (but still informally), as follows:

Theorem 1.3 (Informal; see Theorem 6.1). *For any $\gamma \geqslant 1$, given input $(\mathbf{A}, k) \in \mathbb{F}_2^{n \times m} \times \mathbb{N}$, it is W[1]-hard (under randomized reductions) to distinguish between*

- *the distance of the code generated by $\mathbf{A}$ is at most k , and,*
- *the distance of the code generated by $\mathbf{A}$ is more than $\gamma \cdot k$.*

Notice that our above result rules out FPT approximation algorithms with *any* constant approximation ratio for k-MDP. In contrast, we can only prove FPT inapproximability with *some* constant ratio for k-SVP in ℓ_p norm for $p > 1$. These are stated more precisely below.

Theorem 1.4 (Informal; see Theorem 7.1). *For any $p > 1$, there exists a constant $\gamma_p > 1$ such that given input $(\mathbf{A}, k) \in \mathbb{Z}^{n \times m} \times \mathbb{N}$, it is W[1]-hard (under randomized reductions) to distinguish between*

- *the ℓ_p norm of the shortest vector of the lattice generated by $\mathbf{A}$ is $\leqslant k$, and,*
- *the ℓ_p norm of the shortest vector of the lattice generated by $\mathbf{A}$ is $> \gamma_p \cdot k$.*

We remark that our results do not yield hardness for SVP in the ℓ_1 norm and this remains an interesting open question. Section 8 contains discussion on this problem. We also note that, for Theorem 7.1 and onwards, we are only concerned with $p \neq \infty$; this is because, for $p = \infty$, the problem is NP-hard to approximate even when $k = 1$ [vEB81]!

1.1.1 Nearest Codeword Problem and Nearest Vector Problem

As we shall see in Section 2, our proof proceeds by first showing FPT hardness of approximation of the non-homogeneous variants of k-MDP and k-SVP called the *k-Nearest Codeword Problem* (k-NCP) and the *k-Nearest Vector Problem* (k-NVP) respectively. For both k-NCP and k-NVP, we are given a target vector $\mathbf{y}$ (in $\mathbb{F}_2^n$ and $\mathbb{Z}^n$, respectively) in addition to $(\mathbf{A}, k)$, and the goal is to find whether there is any $\mathbf{x}$ (in $\mathbb{F}_2^m$ and $\mathbb{Z}^m$, respectively) such that the (Hamming and ℓ_p, respectively) norm of $\mathbf{Ax} - \mathbf{y}$ is at most k. Note that their homogeneous counterparts, namely k-MDP and k-SVP, explicitly require the coefficient vector $\mathbf{x}$ to be non-zero, and hence they cannot be interpreted as special cases of k-NCPand k-NVP respectively.

As an intermediate step of our proof, we show that the k-NCP and k-NVP problems are hard to approximate[6] (see Theorem 5.1 and Theorem 7.2 respectively). This should be compared to Downey et al. [DFVW99], in which the authors show that both problems are W[1]-hard to solve exactly. Therefore our inapproximability result significantly improves on their work to rule out any polylog(k) factor FPT-approximation algorithm (assuming W$[1] \neq$ FPT) and are also the first inapproximability results for these problems.

We end this section by remarking that the computational complexity of both (non-parameterized) NCP and NVP are also thoroughly studied (see e.g. [Mic01, DKRS03, Ste93, ABSS97, GMSS99] in addition to the references for MDP and SVP), and indeed the inapproximability results of these two problems form the basis of hardness of approximation for MDP and SVP. We would like to emphasize that while W[1]-hardness results were known for k-NCP and k-NVP, it does not seem easy to transfer them to W[1]-hardness results for k-MDP and k-SVP; we really need parameterized *inapproximability* results for k-NCP and k-NVP to be able to transfer them to (slightly weaker) inapproximability results for k-MDP and k-SVP. There are other parameterized problems that resisted all efforts at proving hardness so far, and we believe that it may be the case for these problems as well that building a chain of inapproximability results is more feasible than building a chain of W[1]-hardness results.

1.2 Organization of the paper

In the next section, we give an overview of our reductions and proofs. After that, in Section 3, we define additional notation and preliminaries needed to fully formalize our proofs. In Section 4, we show the inapproximability of *k-Linear Dependent Set* (k-LDS), a problem naturally arising from linear algebra, that would be used as the base step for all future inapproximability results in this paper. In Section 5 we show the inapproximability of k-NCP. Next, in Section 6, we establish the constant inapproximability of k-MDP. Section 7 provides the inapproximability of k-NVP and k-SVP. Finally, in Section 8, we conclude with a few open questions and research directions.

2 Proof Overview

In the non-parameterized setting, all the aforementioned inapproximability results for both MDP and SVP are shown in two steps: first, one proves the inapproximability of their inhomogeneous counterparts (i.e. NCP and NVP), and then reduces them to MDP and SVP. We follow this general outline. That is, we first show, that both k-NCP and k-NVP are W[1]-hard to approximate. Then, we reduce k-NCP and k-NVP to k-MDP and k-SVP respectively. In this second step, we employ an adaptation of Dumer et al.'s reduction [DMS03] for k-MDP and Khot's reduction [Kho05] for k-SVP. While the latter reduction works almost immediately in the parameterized regime, there are several technical challenges in adapting Dumer et al.'s reduction

[6] While our k-MDP result only applies for $\mathbb{F}_2$, it is not hard to see that our intermediate reduction for k-NCP actually applies for every finite field $\mathbb{F}_q$ too.

to our setting. The remainder of this section is devoted to presenting all of our reductions and to highlight such technical challenges and changes in comparison with the non-parameterized setting.

As mentioned before, the starting point of all the hardness results in this paper is the W[1]-hardness of approximating the k-BICLIQUE problem. In Subsection 2.2, we show a gap-retaining reduction from the gap k-BICLIQUE problem to gap k-Linear Dependent Set (referred to hereafter as k-LDS), an intermediate problem that we introduce which might be of independent interest. We show a gap-retaining reduction from gap k-LDS to gap k-NCP in Subsection 2.3, and then a randomized reduction from gap k-NCP to k-MDP in Subsection 2.4. Finally, in Subsection 2.5, we show a gap-retaining reduction from gap k-LDS to gap k-NVP, and then a randomized reduction from gap k-NVP to k-SVP.

In the next subsection, we first give an overview of Dumer et al.'s reduction [DMS03] and highlight the challenges in extending their reduction to the parameterized setting, following which we give a sketch of the various steps involved in the actual reduction to k-MDP.

2.1 The Dumer-Micciancio-Sudan reduction

We start this subsection by describing the Dumer et al.'s (henceforth DMS) reduction [DMS03]. The starting point of the DMS reduction is the NP-hardness of approximating NCP to any constant factor [ABSS97]. Let us recall that in NCP we are given a matrix $\mathbf{A} \in \mathbb{F}_2^{n \times m}$, an integer k, and a target vector $\mathbf{y} \in \mathbb{F}_2^n$, and the goal is to determine whether there is any $\mathbf{x} \in \mathbb{F}_2^m$ such that $\|\mathbf{Ax} - \mathbf{y}\|_0$ is at most k. Arora et al. [ABSS97] shows that for any constant $\gamma \geqslant 1$, it is NP-hard to distinguish the case when there exists $\mathbf{x}$ such that $\|\mathbf{Ax} - \mathbf{y}\|_0 \leqslant k$ from the case when for all $\mathbf{x}$ we have that $\|\mathbf{Ax} - \mathbf{y}\|_0 > \gamma k$.

Dumer et al. introduce the notion of "locally dense codes" to enable a gadget reduction from NCP to MDP. Informally, a locally dense code is a linear code $\mathbf{L}$ with minimum distance d admitting a ball $\mathcal{B}(\mathbf{s}, r)$ centered at $\mathbf{s}$ of radius[7] $r < d$ and containing a large (exponential in the dimension) number of codewords. Moreover, for the gadget reduction to MDP to go through, we require not only the knowledge of the code, but also the center $\mathbf{s}$ and a linear transformation $\mathbf{T}$ used to index the codewords in $\mathcal{B}(\mathbf{s}, r)$, i.e., $\mathbf{T}$ maps $\mathcal{B}(\mathbf{s}, r) \cap \mathbf{L}$ onto a smaller subspace. Given an instance $(\mathbf{A}, \mathbf{y}, k)$ of NCP, and a locally dense code $(\mathbf{L}, \mathbf{T}, \mathbf{s})$ whose parameters (such as dimension and distance) we will fix later, Dumer et al. build the following matrix:

$$\mathbf{B} = \begin{bmatrix} \mathbf{ATL} & -\mathbf{y} \\ \vdots & \vdots \\ \mathbf{ATL} & -\mathbf{y} \\ \mathbf{L} & -\mathbf{s} \\ \vdots & \vdots \\ \mathbf{L} & -\mathbf{s} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} a \text{ copies} \\ \\ \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} b \text{ copies} \end{matrix} , \tag{1}$$

where a, b are some appropriately chosen positive integers. If there exists $\mathbf{x}$ such that $\|\mathbf{Ax} - \mathbf{y}\|_0 \leqslant k$ then consider $\mathbf{z}'$ such that $\mathbf{TLz}' = \mathbf{x}$ (we choose the parameters of $(\mathbf{L}, \mathbf{T}, \mathbf{s})$, in

[7] Note that for the ball to contain more than a single codeword, we must have $r \geqslant d/2$.

particular the dimensions of $\mathbf{L}$ and $\mathbf{T}$ such that all these computations are valid). Let $\mathbf{z} = \mathbf{z}' \circ 1$, and note that $\|\mathbf{Bz}\|_0 = a\|\mathbf{Ax} - \mathbf{y}\|_0 + b\|\mathbf{Lz} - \mathbf{s}\|_0 \leqslant ak + br$. In other words, if $(\mathbf{A}, \mathbf{y}, k)$ is a YES instance of NCP then $(\mathbf{B}, ak + br)$ is a YES instance of MDP. On the other hand if we had that for all $\mathbf{x}$, the norm of $\|\mathbf{Ax} - \mathbf{y}\|_0$ is more than γk for some constant[8] $\gamma > 2$, then it is possible to show that for all $\mathbf{z}$ we have that $\|\mathbf{Bz}\|_0 > \gamma'(ak + br)$ for any $\gamma' < \frac{2\gamma}{2+\gamma}$. The proof is based on a case analysis of the last coordinate of $\mathbf{z}$. If that coordinate is 0, then, since $\mathbf{L}$ is a code of distance d, we have $\|\mathbf{Bz}\|_0 \geqslant bd > \gamma'(ak + br)$; if that coordinate is 1, then the assumption that $(\mathbf{A}, \mathbf{y}, k)$ is a NO instance of NCP implies that $\|\mathbf{Bz}\|_0 > a\gamma k > \gamma'(ak + br)$. Note that this gives an inapproximability for MDP of ratio $\gamma' < 2$; this gap is then further amplified by a simple tensoring procedure.

We note that Dumer et al. were not able to find a deterministic construction of locally dense code with all of the above described properties. Specifically, they gave an efficient deterministic construction of a code $\mathbf{L}$, but only gave a randomized algorithm that finds a linear transformation $\mathbf{T}$ and a center $\mathbf{s}$ w.h.p. Therefore, their hardness result relies on the assumption that NP $\neq$ RP, instead of the more standard NP $\neq$ P assumption. Later, Cheng and Wan [CW12] and Micciancio [Mic14] provided constructions for such (families of) locally dense codes with an explicit center, and thus showed the constant ratio inapproximability of MDP under the assumption of NP $\neq$ P.

Trying to follow the DMS reduction in order to show the parameterized intractability of k-MDP, we face the following three immediate obstacles. First, there is no inapproximability result known for k-NCP, for any constant factor greater than 1. Note that to use the DMS reduction, we need the parameterized inapproximability of k-NCP, for an approximation factor which is greater than two. Second, the construction of locally dense codes of Dumer et al. only works when the distance is linear in the block length (which is a function of the size of the input). However, we need codes whose distance are bounded above by a function of the parameter of the problem (and not dependent on the input size). This is because the DMS reduction converts an instance $(\mathbf{A}, \mathbf{y}, k)$ of k-NCP to an instance $(\mathbf{B}, ak + br)$ of $(ak + br)$-MDP, and for this reduction to be an FPT reduction, we need $ak + br$ to be a function only depending on k, i.e., d, the distance of the code $\mathbf{L}$ (which is at most $2r$), must be a function only of k. Third, recall that the DMS reduction needs to identify the vectors in the ball $\mathcal{B}(\mathbf{s}, r) \cap \mathbf{L}$ with all the potential solutions of k-NCP. Notice that the number of vectors in the ball is at most $(nm)^{O(r)}$ but the number of potential solutions of k-NCP is exponential in m (i.e. all $\mathbf{x} \in \mathbb{F}_2^m$). However, this is impossible since $r \leqslant d$ is bounded above by a function of k!

We overcome the first obstacle by proving the inapproximability of k-NCP upto poly-logarithmic factors under W[1] $\neq$ FPT (see Subsection 2.2). Note that in order to follow the DMS reduction, it suffices to just show the inapproximability of k-NCP for some constant factor greater than 2; nonetheless the hardness of approximating k-NCP up to poly-logarithmic factors is of independent interest.

We overcome the third obstacle by introducing an intermediate problem in the DMS reduction, which we call the *sparse nearest codeword problem*. The sparse nearest codeword problem is a promise problem which differs from k-NCP in the following way: the objective here considers the distance of the target vector $\mathbf{y}$ to the nearest codeword $\mathbf{Ax}$ as well as the Ham-

[8] Note that in the described reduction, we need the inapproximability of NCP to a factor greater than two, even to just reduce to the *exact* version of MDP.

ming weight of the coefficient vector $\mathbf{x}$ which realizes the nearest codeword. We show the inapproximability of the sparse nearest codeword problem (See Subsection 2.3).

Finally, we overcome the second obstacle by introducing a variant of locally dense codes, which we call *locally suffix dense codes*. Roughly speaking, we show that any systematic code which nears the sphere-packing bound (aka Hamming bound) in the high rate regime is a locally suffix dense code. Then we follow the DMS reduction with the new ingredient of locally suffix dense codes (replacing locally dense codes) to reduce the sparse nearest codeword problem to k-MDP.

The full reduction goes through several intermediate steps, which we will describe in more detail in the coming subsections. The high-level summary of these steps is also provided in Figure 1. Throughout this section, for any gap problem, if we do not specify the gap in the subscript, then it implies that the gap can be any arbitrary constant (or even super constant).

2.2 Parameterized Inapproximability of k-LDS

To prove the inapproximability of MDP we first consider its dual problem LDS. Given a set $\mathbf{A}$ of n vectors over a finite field $\mathbb{F}_q$ and an integer k, the goal of k-LDS problem is to decide if there are k vectors in $\mathbf{A}$ that are linearly dependent. The gap version of this problem (GAPLDS) is to distinguish the case when there are k vectors in $\mathbf{A}$ that are linearly dependent from the case when any γk ($\gamma \geqslant 1$) vectors in $\mathbf{A}$ are linearly independent. As briefly touched upon in the introduction, k-LDS is closely related to k-MDP: one might think of $\mathbf{A}$ as a matrix in $\mathbb{F}_q^{n\times m}$ and the goal of k-LDS is to find a vector $\mathbf{y} \in \mathbb{F}_q^m$ with $\|\mathbf{y}\|_0 \leqslant k$ and $\mathbf{A}\mathbf{y} = \mathbf{0}$, then k-LDS is a yes-instance if and only if $\min_{\mathbf{0}\neq\mathbf{x}\in\mathbb{F}_q^{n'}} \|\mathbf{A}^{\perp}\mathbf{x}\|_0 \leqslant k$, where $\mathbf{A}^{\perp} \in \mathbb{F}_q^{m\times n'}$ is a matrix with maximum number of linearly independent column vectors such that $\mathbf{A}\mathbf{A}^{\perp} = \mathbf{0}$. Note that the parameterized inapproximability of k-MDP follows by the parameterized intractability of GAPLDS over the binary field.

However, we cannot prove the hardness of GAPLDS over the binary field directly. Instead, we tackle this problem in three steps. Our first step is to show the parameterized intractability of GAPLDS over large fields by giving a reduction from the ONE-SIDED BICLIQUE problem to GAPLDS.

It will be more convenient to view the inapproximability result of ONE-SIDED BICLIQUE from [Lin18] as a hardness of the following problem which we call BIPARTITE SUBGRAPH WITH MINIMUM DEGREE (BSMD): given a bipartite graph G and positive integers s, h with $s \leqslant h$, find smallest (in terms of edges) non-empty subgraph of G such that every left vertex of the subgraph has degree at least h and every right vertex has degree at least s. Here the parameter is $s + h$. The gap version of BSMD, called GAPBSMD$_\gamma$, is to distinguish between (i) the YES case in which G contains a complete bipartite graph with s vertices on the left and h on the right (which satisfies the property with hs edges) and (ii) the NO case in which every desired subgraph must have at least $\gamma \cdot hs$ edges.

It is not hard to see that Lin's reduction, with appropriate parameter setting, gives W[1]-hardness of GAPBSMD$_\gamma$ for any constant γ. In what follows we sketch the reduction from GAPBSMD to GAPLDS. Given an instance $(G = (L, R, E), s, h)$ of GAPBSMD, we choose a

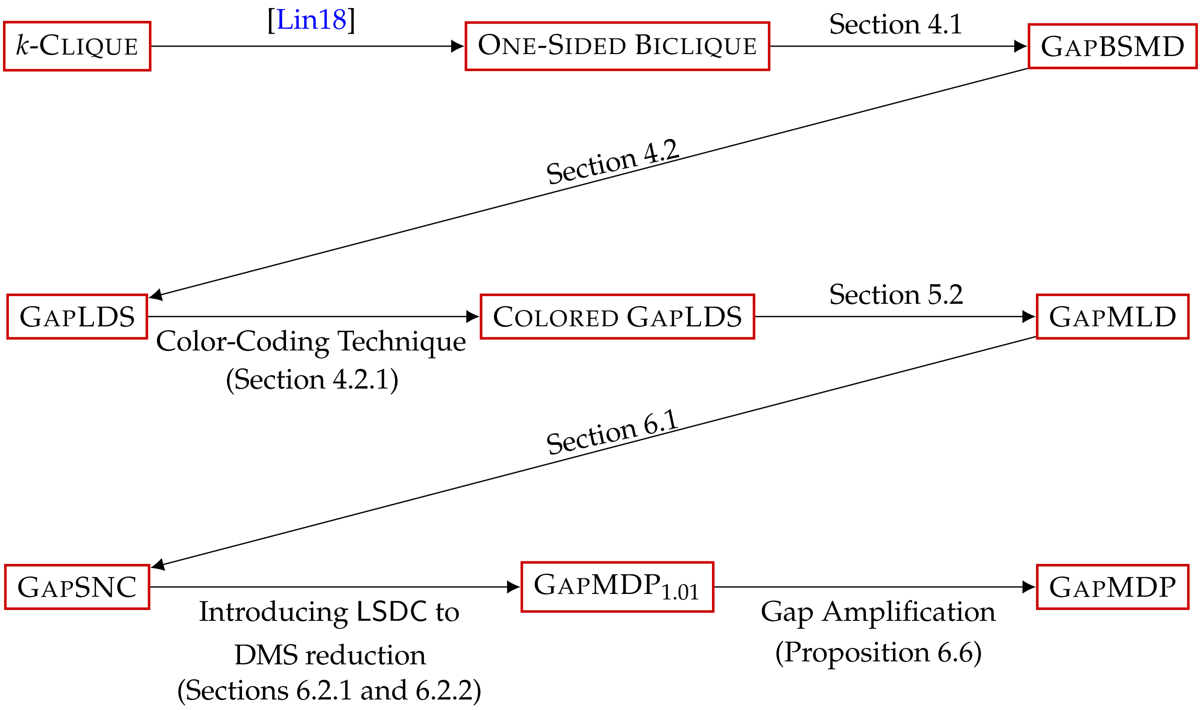

Figure 1: The figure provides an overview of our reduction from the canonical W[1]-complete k-Clique problem to the parameterized Minimum Distance problem. Our starting point is the gap one-sided biclique problem which is now known to be W[1]-hard from Lin's work [Lin18]. Based on the hardness of approximating the one-sided biclique problem, we obtain the constant inapproximability of a different graph problem, namely the bipartite subgraph with minimum degree problem (GAPBSMD); see Section 4.1 for details. Next, we reduce GAPBSMD to the gap linear dependent set problem (GAPLDS) in Section 4.2, and then use standard color-coding techniques in Section 4.2.1 to obtain the constant inapproximability of a colored version of GAPLDS over fields of non-constant size. In Section 5.2, we reduce the aforementioned colored version of GAPLDS to the GAPMLD problem over $\mathbb{F}_2$, and thus rule out constant approximation parameterized algorithms for NCP. Via a simple reduction from GAPMLD, in Section 6.1 we obtain the constant parameterized inapproximability of GAPSNC. In Section 6.2.1, we formally introduce locally suffix dense codes and show how to efficiently (but probabilistically) construct them. These codes are then used in Section 6.2.2 to obtain the parameterized innapproximability of GAPMDP$_{1.01}$. The final step is a known gap amplification by tensoring (Proposition 6.6).

large finite field $\mathbb{F}_q$ so that the vertices of G can be treated as elements of $\mathbb{F}_q$. Then we construct a function $\iota : L \cup R \to \mathbb{F}_q^{h-1}$ such that:

(L1) the images of any $s-1$ vertices in L under ι are linearly independent;

(L2) the images of any s vertices in L under ι are linearly dependent.

Similarly,

(R1) the images of any $h-1$ vertices in R under ι are linearly independent;

(R2) while the images of any h vertices in R under ι are linearly dependent.

We point out that one can construct functions satisfying the above properties by mapping the vertices (now identified with field elements) to the columns of a Vandermonde matrix of appropriate dimensions padded with zeros. Finally we construct a vector $\mathbf{w}_e \in \mathbb{F}_q^{q(h-1)}$ for every edge in $e \in E$ and then let $\{\mathbf{w}_e : e \in E\}$ be our target instance of GAPLDS. To define w_e, firstly, we partition each vector in $\mathbb{F}_q^{q(h-1)}$ into q blocks. Each vertex in G has its unique corresponding block. Each block has $h-1$ elements. Suppose $e = \{u, v\}$ where $u \in L$ and $v \in R$. We set the u-th block of the vector $\mathbf{w}_e$ equal to $\iota(v)$, the the v-th block of the vector $\mathbf{w}_e$ equal to $\iota(u)$ and all the other blocks of $\mathbf{w}_e$ equal to $\mathbf{0}$. Note that the u-th block is equal to $\iota(v)$ (not to $\iota(u)$!) and the v-th block is equal to $\iota(u)$.

Suppose that $u_1, \dots, u_s \in L$ and $v_1, \dots, v_h \in R$ form a complete bipartite subgraph in G. We will show that the sh-sized set $W = \{\mathbf{w}_{u_i, v_j} : i \in [s], j \in [h]\}$ is linearly dependent. It is not hard to see that for all $i \in [s]$, the restriction of W to the u_i-th block is a set of h vectors $\{\iota(v_1), \dots, \iota(v_h)\}$. By the property (R2) of ι, these vectors are linearly dependent, i.e., there are $b_1, \dots, b_h \in \mathbb{F}_q$ such that $\sum_{j \in [h]} b_j \iota(v_j) = \mathbf{0}$. Similarly, we can see that for all $j \in [h]$, the restriction of W to the v_j-th block is a set of s linearly dependent vectors $\{\iota(u_1), \dots, \iota(u_s)\}$ and $\sum_{i \in [s]} a_i \iota(u_i) = \mathbf{0}$ for some $a_1, \dots, a_s \in \mathbb{F}_q$. It is easy to check that $\sum_{i \in [s], j \in [h]} a_i b_j w_{u_i, v_j} = \mathbf{0}$ and $a_i b_j$ ($i \in [s], j \in [h]$) are not all zero.

On the other hand, if G is a NO instance of GAPBSMD$_\gamma$, we will show that any linearly dependent set must have at least $\gamma \cdot hs$ vectors. Observe that every vector in the GAPLDS instance is corresponding to an edge in the graph G. Suppose W is a set of linearly dependent vectors. We consider the graph H_W in G induced by the edges corresponding to vectors in W. We can argue that every vertex on the left side of H_W must have at least h neighbors and every vertex on the right side of H_W must have at least s neighbors, using properties (R1) and (L1) respectively. From the definition of the NO instance of GAPBSMD$_\gamma$, we can immediately conclude that $|W| \geqslant \gamma \cdot hs$.

2.3 Parameterized Inapproximability of k-NCP

In the second step, we prove the inapproximability of k-NCP using the hardness of GAPLDS. Note that this is the step in which we reduce the field size to two, i.e., the hardness for GAPLDS described above is for a large field ($\mathbb{F}_q$ where $q = \Theta(n)$) but the k-NCP problem is for $\mathbb{F}_2$.

The reduction is simpler to state if we use the dual (equivalent) formulation of k-NCP called Maximum Likelihood Decoding (k-MLD). The gap version of the problem, denoted by GAPMLD$_\gamma$, can be formulated as follows: Given a matrix $\mathbf{A} \in \mathbb{F}_2^{n \times m}$, a vector $\mathbf{y} \in \mathbb{F}_2^n$ and a positive integer $k \in \mathbb{N}$, the goal of GAPMLD$_\gamma$ problem is to distinguish the case when there exists a nonzero vector $\mathbf{x} \in \mathbb{F}_2^m$ with Hamming weight at most k such that $\mathbf{Ax} = \mathbf{y}$ from the case when for all $\mathbf{x} \in \mathbb{F}_2^m$ with Hamming weight at most γk, $\mathbf{Ax} \neq \mathbf{y}$.

Reducing GAPLDS to GAPMLD. We present a reduction from GAPLDS to GAPMLD. For ease of presentation, we think of the input of GAPMLD as a set $\mathcal{W}$ of vectors (i.e. column vectors of $\mathbf{A}$) in $\mathbb{F}_2^n$, the goal is to distinguish the case when there exist k vectors whose sum is $\mathbf{y}$ from the case when the sum of any nonempty subset of vectors in $\mathcal{W}$ of size at most γk is not equal to $\mathbf{y}$.

We start with the hardness of $\textsc{GapLDS}_\gamma$ where the input vectors are $\mathbb{F}_{2^d}$-vectors, for $d = \Theta(\log n)$. To reduce the field size, we transform vectors from $\mathbb{F}_{2^d}^m$ into $\mathbb{F}_2^{dm}$ using a linear bijection f between $\mathbb{F}_{2^d}^m$ and $\mathbb{F}_2^{dm}$. Observe that, even if $\mathbf{w}_1, \ldots, \mathbf{w}_k$ are linearly dependent vectors in $\mathbb{F}_{2^d}^m$, the sum of their images under f is not necessarily zero. This is because we need coefficients $a_1, \ldots, a_k \in \mathbb{F}_{2^d} \setminus \{0\}$ so that $\sum_{i \in [k]} a_i \mathbf{w}_i = \mathbf{0}$ and hence $\sum_{i \in [k]} f(a_i \mathbf{w}_i) = \mathbf{0}$, while $\sum_{i \in [k]} f(\mathbf{w}_i) = \mathbf{0}$ may not hold.

With these in mind, we will try to construct an instance $\mathcal{W}'$ of $\textsc{GapMLD}$ such that for all $a \in \mathbb{F}_{2^d}$ and $\mathbf{w} \in \mathcal{W}$, $f(a\mathbf{w})$ has a corresponding vector in $\mathcal{W}'$. And if $\mathcal{W}$ has k linearly dependent vectors $\sum_{i \in [k]} a_i \mathbf{w}_i = \mathbf{0}$, then the sum of vectors corresponding to $f(a_1\mathbf{w}_1), \ldots, f(a_k\mathbf{w}_k)$ is equal to $\mathbf{y}$.

We need some mechanism to force the solution of $\textsc{GapMLD}$ to select vectors corresponding to at least k distinct vectors $f(a_1\mathbf{w}_1), \ldots, f(a_k\mathbf{w}_k)$. To that end, we use the color-coding technique to reduce $\textsc{GapLDS}$ to its colored version (see Section 4.2.1 for details). Thus, we can assume that the instance $\mathcal{W}$ of $\textsc{GapLDS}$ comes with a coloring $c : \mathcal{W} \to [k]$ such that if $\mathcal{W}$ is a YES instance, then there are exactly k vectors in $\mathcal{W}$ with distinct colors under c that are linearly dependent.

For $i \in [k]$, let $\mathbf{e}_i \in \mathbb{F}_2^k$ be the vector whose i-th coordinate is 1 and the other coordinates are equal to 0. It is natural to construct a reduction as follows: given an instance $\mathcal{W}$ of $\textsc{GapLDS}$ over $\mathbb{F}_{2^d}$ and a coloring function $c : \mathcal{W} \to [k]$, output

$$\mathcal{W}' = \{\mathbf{e}_{c(\mathbf{w})} \circ f(a\mathbf{w}) : \mathbf{w} \in \mathcal{W}, a \in \mathbb{F}_{2^d} \setminus \{0\}\} \text{ and } \mathbf{y} = \mathbf{1}_k \circ \mathbf{0}_{md}$$

as the target instance of $\textsc{GapMLD}$, where $\circ$ stands for the concatenation of vectors.

It is easy to see that if $\mathcal{W}$ contains k linearly dependent vectors $\sum_{i \in [k]} a_i \mathbf{w}_i = \mathbf{0}$, then the sum of the vectors $\mathbf{e}_{c(\mathbf{w}_1)} \circ f(a_1\mathbf{w}_1), \ldots, \mathbf{e}_{c(\mathbf{w}_k)} \circ f(a_k\mathbf{w}_k)$ is equal to $\mathbf{1}_{dm} \circ \mathbf{0}_k$.

On the other hand, if any $3k$ vectors of $\mathcal{W}$ are linearly independent, we will show that for any $W \subseteq \mathcal{W}'$ such that $\sum_{\mathbf{x} \in W} \mathbf{x} = \mathbf{y}$, we have $|W| \geqslant 3k$. Let the elements of W be $\mathbf{e}_{c(\mathbf{w}_1)} \circ f(a_1\mathbf{w}_1), \ldots, \mathbf{e}_{c(\mathbf{w}_{k'})} \circ f(a_{k'}\mathbf{w}_{k'})$, and suppose for the sake of contradiction that $k' < 3k$. By restricting the equation $\sum_{\mathbf{x} \in W} \mathbf{x} = \mathbf{y}$ onto the last md coordinates, it follows that

$$\sum_{i \in [k']} f(a_i \mathbf{w}_i) = \mathbf{0},$$

which implies

$$\sum_{i \in [k']} a_i \mathbf{w}_i = \mathbf{0}.$$

At this moment, we cannot yet say that the set $\{\mathbf{w}_1, \ldots, \mathbf{w}_{k'}\}$ is linearly dependent (and therefore contradicts $k' < 3k$), because $\mathbf{w}_1, \ldots, \mathbf{w}_{k'}$ may contain duplicated elements. For example, it is possible that $k' = 3$, $a_1 + a_2 + a_3 = 0$ and $\mathbf{w}_1 = \mathbf{w}_2 = \mathbf{w}_3$ could be any nonzero vector. To get a contradiction by this way, we need to show that there is a vector $\mathbf{w}$ which appears exactly once in $\mathbf{w}_1, \ldots, \mathbf{w}_{k'}$.

To see this, first observe that, since $k' < 3k$, there must be a color $j \in [k]$ that corresponds to at most two vectors from $\mathbf{w}_1, \ldots, \mathbf{w}_{k'}$ (duplicated counted). However, if we restrict the equation $\sum_{\mathbf{x} \in W} \mathbf{x} = \mathbf{y}$ to only the j-th coordinate, we can see that the left hand side equals to the number of occurrences of color j modulo 2, whereas the right hand side is one. This means that there is only a unique vector among $\mathbf{w}_1, \ldots, \mathbf{w}_{k'}$ that is of the j-th color; this immediately implies that this vector occurs only once in $\mathbf{w}_1, \ldots, \mathbf{w}_{k'}$. This in turns means that $\{\mathbf{w}_1, \ldots, \mathbf{w}_{k'}\}$ is linearly dependent and therefore $k' > 3k$, a contradiction.

Note that our argument only gives hardness of approximation with factor $3 - \varepsilon$ for any $\varepsilon > 0$. Nonetheless, this factor suffices for the subsequent steps. We can in fact also prove hardness for every constant factor, using a slight tweak of the above idea. Please see Section 5.2 for more details

Reducing GAPMLD to GAPSNC. Now we introduce the *sparse nearest codeword problem* that we will use to prove the parameterized inapproximability of k-MDP. We define the gap version of this problem, denoted by GAPSNC$_\gamma$ (for some constant $\gamma \geqslant 1$) as follows: on input $(\mathbf{A}', \mathbf{y}', k)$, distinguish between the YES case where there exists $\mathbf{x} \in \mathbb{F}_2^m$ such that $\|\mathbf{A}'\mathbf{x} - \mathbf{y}'\|_0 + \|\mathbf{x}\|_0 \leqslant k$, and the NO case where for all $\mathbf{x}$ (in the entire space), we have $\|\mathbf{A}'\mathbf{x} - \mathbf{y}'\|_0 + \|\mathbf{x}\|_0 > \gamma k$. We highlight that the difference between k-NCP and GAPSNC$_\gamma$ is that the objective also depends on the Hamming weight of the coefficient vector $\mathbf{x}$. We sketch below the reduction from an instance $(\mathbf{A}, \mathbf{y}, k)$ of GAPMLD$_\gamma$ to an instance $(\mathbf{A}', \mathbf{y}', k)$ of GAPSNC$_\gamma$. Given $\mathbf{A}, \mathbf{y}$, let

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} \\ \vdots \\ \mathbf{A} \\ \mathrm{Id} \end{bmatrix} \left.\vphantom{\begin{matrix} \mathbf{A} \\ \vdots \\ \mathbf{A} \end{matrix}}\right\} \gamma k + 1 \text{ copies}, \quad \mathbf{y}' = \begin{bmatrix} \mathbf{y} \\ \vdots \\ \mathbf{y} \\ \mathbf{0} \end{bmatrix} \left.\vphantom{\begin{matrix} \mathbf{y} \\ \vdots \\ \mathbf{y} \end{matrix}}\right\} \gamma k + 1 \text{ copies} .$$

Notice that for any $\mathbf{x}$ (in the entire space), we have

$$\|\mathbf{A}'\mathbf{x} - \mathbf{y}'\|_0 = (\gamma k + 1)\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0,$$

and thus both the completeness and soundness of the reduction easily follow.

2.4 Parameterized Inapproximability of k-MDP

Let us recall that in the NCP we are given a matrix $\mathbf{A} \in \mathbb{F}_2^{n \times q}$, an integer k, and a target vector $\mathbf{y} \in \mathbb{F}_2^n$, and the goal is to determine whether there is exists a vector $\mathbf{x} \in \mathbb{F}_2^m$ such that $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_0$ is at most k. A natural first idea for reducing an NCP instance $(\mathbf{A} \in \mathbb{F}_2^{n \times m}, \mathbf{y} \in \mathbb{F}_2^n)$ to MDP would be to introduce the $n \times (m+1)$ matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{A} & -\mathbf{y} \end{bmatrix}; \tag{2}$$

then any solution $\mathbf{x} \in \mathbb{F}_2^m$ of the NCP instance with $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_0 \leqslant k$ would give a solution $\mathbf{x}' = \mathbf{x} \circ 1 \in \mathbb{F}_2^{m+1}$ of the MDP instance with $|\mathbf{B}\mathbf{x}'| \leqslant k$. However, the problem is that if the MDP instance has a solution $\mathbf{x}' = \mathbf{x} \circ 0$ (i.e., the last coordinate is zero), then $\|\mathbf{B}\mathbf{x}'\|_0 \leqslant k$ implies only $\|\mathbf{A}\mathbf{x}\|_0 \leqslant k$, but does not imply $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_0 \leqslant k$. Thus we need a way to force the last coordinate to 1 in the solution of the MDP instance. We can try to use error correcting codes for this purpose. Let $\mathbf{L} \in \mathbb{F}_2^{h \times m}$ be the generator matrix of an error correcting code with minimum distance d. Let us consider now the matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{A} & -\mathbf{y} \\ \mathbf{L} & -\mathbf{s} \end{bmatrix}; \tag{3}$$

for some arbitrarily chosen vector $\mathbf{s} \in \mathbb{F}_2^h$. Now for any nonzero $\mathbf{x}' = \mathbf{x} \circ 0$, we have $\|\mathbf{B}\mathbf{x}'\|_0 = \|\mathbf{A}\mathbf{x}\|_0 + \|\mathbf{L}\mathbf{x}\|_0 \geqslant \|\mathbf{A}\mathbf{x}\|_0 + d$, since $\mathbf{x}$ is a nonzero vector and the code generated by $\mathbf{L}$ has minimum distance d. Thus the second term gives a penalty of d if the last coordinate of $\mathbf{x}'$ is 0. However, the problem now is that if $\mathbf{x}$ is a solution of the NCP instance with $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_0 \leqslant k$, then defining $\mathbf{x}' = \mathbf{x} \circ 1$ gives $\|\mathbf{B}\mathbf{x}'\|_0 = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_0 + \|\mathbf{L}\mathbf{x} - \mathbf{s}\|_0 = k + \|\mathbf{L}\mathbf{x} - \mathbf{s}\|_0$. We would need to argue that this second term $\|\mathbf{L}\mathbf{x} - \mathbf{s}\|_0$ is small, much smaller than the penalty d in the previous case. While in general, there is no reason why the chosen vector $\mathbf{s}$ would be close to $\mathbf{L}\mathbf{x}$ for the hypothetical solution $\mathbf{x}$. However, we can hope to increase the chances of finding such an $\mathbf{s}$, if we could somehow enforce that there are many distinct choices of $\mathbf{x}$ for which we would have $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_0 \leqslant k$. This can indeed by achieved by padding the matrix $\mathbf{A}$ with additional dummy zero columns, and padding the corresponding solution $\mathbf{x}$ with additional dummy coordinates. In particular, this ensures that for even a random choice of $\mathbf{s}$ (sampled from an appropriate distribution) is close to $\mathbf{L}\mathbf{x}$ for at least one of the choices of $\mathbf{x}$ with non-negligible probability. We formalize this intuition in the form of *Locally Suffix Dense Codes* described below.

Locally Suffix Dense Codes. A locally suffix dense code (LSDC) is a linear code $\mathbf{L} \in \mathbb{F}_2^{h \times m}$ of block length h with minimum distance d such that the following holds. For any choice of prefix $\mathbf{x} \in \mathbb{F}_2^q$ and a randomly drawn suffix vector $\mathbf{s} \overset{\text{u.a.r}}{\sim} \mathbb{F}_2^{h-q}$ the vector $\mathbf{x} \circ \mathbf{s}$ is r close to the code $\mathbf{L}$ with non-negligible probability. In other words, for every choice of prefix vector $\mathbf{x}$, the restriction of the code $\mathbf{L}$ to the affine subspace $V_{\mathbf{x}} := \{\mathbf{x}\} \times \mathbb{F}_2^{h-q}$ is *dense*. While one can think of the suffix vector $\mathbf{s}$ as being analogous to the center in LDC, note that $\mathbf{s}$ is merely a suffix which is used to extend the vector $\mathbf{x}$. Therefore, due to systematicity of the code, the distance of the vector $\mathbf{x} \circ \mathbf{s}$ to the code $\mathbf{L}$ depends only on the choice of $\mathbf{s}$, which allows us to ensure that the parameters r and d can be chosen to functions of k, without explicitly depending on the block length h.

As in the case of Dumer et al. we too cannot find an explicit suffix $\mathbf{s}$ for the LSDCs that we construct, but instead provide an efficiently samplable distribution such that, for any $\mathbf{x} \in \mathbb{F}_2^q$, the probability (over $\mathbf{s}$ sampled from the distribution) that $\mathcal{B}(\mathbf{x} \circ \mathbf{s}, r) \cap \mathbf{L} \neq \emptyset$ is non-negligible. This is what makes our reduction from GAPSNC$_{2.5}$ to GAPMDP$_{1.01}$ randomized. We will not elaborate more on this issue here, but focus on the (probabilistic) construction of such codes. For convenience, we will assume throughout this overview that k is much smaller than d, i.e., $k = 0.001d$.

Recall that the sphere-packing bound (aka Hamming bound) states that a binary code of block length h and distance d can have at most $2^h / |\mathcal{B}(\mathbf{0}, \lceil \frac{d-1}{2} \rceil)|$ codewords; this is simply because the balls of radius $\lceil \frac{d-1}{2} \rceil$ at the codewords do not intersect. Our main theorem regarding the existence of locally dense suffix code is that any systematic code that is "near" the sphere-packing bound is a locally dense suffix code with $r = \lceil \frac{d-1}{2} \rceil$. Here "near" means that the number of codewords must be at least $2^h / |\mathcal{B}(\mathbf{0}, \lceil \frac{d-1}{2} \rceil)|$ divided by $f(d) \cdot \text{poly}(h)$ for some function f that depends only on d. (Equivalently, this means that the message length must be at least $h - (d/2 + O(1)) \log h$.) The BCH code over binary alphabet is an example of a code satisfying such a condition.

While we will not sketch the proof of the existence theorem here, we note that the general idea is as follows. We choose $\mathbf{L}$ in such a way that for every choice of $\mathbf{x} \in \mathbb{F}_2^q$, the restriction of $\mathbf{L}$ to the affine subspace $V_\mathbf{x}$ is near the sphere packing bound. Then from the above discussion, it follows that for $\mathbf{s}$ sampled uniformly from $\mathbb{F}_2^{h-q}$, the probability that $\mathcal{B}(\mathbf{x} \circ \mathbf{s}, r) \cap \mathbf{L} \neq \emptyset$ is at least the probability that a random point in $\mathbb{F}_2^{h-q}$ is within distance $r = \lceil \frac{d-1}{2} \rceil$ of some codeword from $V_\mathbf{x} \cap \mathbf{L}$. The latter is non-negligible from our choice of $\mathbf{L}$ which ensures that the restriction of the code to any affine subspace $V_\mathbf{x}$ nears the sphere-packing bound.

Finally, we remark that our proof here is completely different from the DMS proof of existence of locally dense codes. Specifically, DMS uses a group-theoretic argument to show that, when a code exceeds the Gilbert-Varshamov bound, there must be a center $\mathbf{s}$ such that $\mathcal{B}(\mathbf{s}, r)$ contains many codewords. Then, they pick a random linear map $\mathbf{T}$ and show that w.h.p. $\mathbf{T}(\mathcal{B}(\mathbf{s}, r) \cap \mathbf{L})$ is the entire space. Note that this second step does not use any structure of $\mathcal{B}(\mathbf{s}, r) \cap \mathbf{L}$; their argument is simply that, for any sufficiently large subset Y, a random linear map $\mathbf{T}$ maps Y to an entire space w.h.p. However, such an argument fails for us, due to the fact that, in LSDC, we want to ensure that $\mathbf{L}$ is dense (up to Hamming distance $r = O(k)$) in all the affine subspaces $\{V_\mathbf{x} : \mathbf{x} \in \mathbb{F}_2^q\}$, instead of exactly covering the whole space $\mathbb{F}_2^h$. Now if we insist on exactly covering all the affine subspaces using a linear map $\mathbf{T}$, as in the DMS construction, we will then have $\mathbf{T}(\mathcal{B}(\mathbf{s}, r)) \supseteq \mathbb{F}_2^h$. This would instead require r to depend on h, whereas in our setting we want r to depend only on the parameter k.

Reducing GAPSNC$_{2.5}$ to GAPMDP$_{1.01}$. Equipped with the notion of locally suffix dense codes defined above, we now prove the hardness of GAPMDP$_{1.01}$.

We begin with an instance $(\mathbf{A}, \mathbf{y}, k)$ with $\mathbf{A} \in \mathbb{F}_2^{n \times q}$ of GAPSNC$_{2.5}$. Let $\mathbf{L} \in \mathbb{F}_2^{h \times m}$ be a locally suffix dense code with distance $d \approx 2.5k$, where we can choose $h, m \leqslant \text{poly}(q, d)$. We also choose a vector $\mathbf{s} \in \mathbb{F}_2^h$ uniformly at random with the first q coordinates equal to zero and construct the matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{n \times (m-q)} & -\mathbf{y} \\ \mathbf{L} & & -\mathbf{s} \end{bmatrix}.$$

We shall show that with probability at least $p = p(k)$ [9], we have that $(\mathbf{B}, k + d/2)$ is an instance of GAPMDP$_{1.01}$.

If $(\mathbf{A}, \mathbf{y}, k)$ is a YES instance of GAPSNC$_{2.5}$, then there exists $\mathbf{x} \in \mathcal{B}(\mathbf{0}, k)$ such that $\|\mathbf{Ax} - \mathbf{y}\|_0 \leqslant k$. Furthermore, from the guarantees of the locally suffix dense codes, with probability at least p (over the choice of the vector s), we have $\|\mathbf{Lx} - \mathbf{s}\|_0 \leqslant (d-1)/2$. Therefore, setting $\mathbf{z} = \mathbf{x}' \circ 1$, we get that

$$\|\mathbf{Bz}\|_0 = \|\mathbf{A}'\mathbf{x}' - \mathbf{y}\|_0 + \|\mathbf{Lx}' - \mathbf{s}\|_0 \leqslant k + (d-1)/2.$$

In other words, if $(\mathbf{A}, \mathbf{y}, k)$ is a YES instance of NCP, then $(\mathbf{B}, k + d/2)$ is a YES instance of MDP$_{1.01}$.

[9] Here the probability $p = p(k)$ depends only on the parameter k

On the other hand, if we had that $\|\mathbf{Ax}-\mathbf{y}\|_0 + \|\mathbf{x}\|_0 > 2.5k$ for all $\mathbf{x}$, then for all non-zero $\mathbf{z} \in \mathbb{F}_2^m$,

$$\|\mathbf{B}(\mathbf{z}\circ 0)\|_0 = \|\mathbf{A}'\mathbf{z}\|_0 + \|\mathbf{Lz}\|_0 \geqslant d,$$

and

$$\|\mathbf{B}(\mathbf{z}\circ 1)\|_0 = \|\mathbf{A}'\mathbf{z}-\mathbf{y}\|_0 + \|\mathbf{Lz}-\mathbf{s}\|_0 \geqslant 2.5k.$$

Since from our choice of parameters, we have $d \approx 2.5k \geqslant 1.01(k+d/2)$, which implies that $(\mathbf{B}, k+d/2)$ is a NO instance of $\mathsf{MDP}_{1.01}$.

Gap Amplification for GapMDP$_{1.01}$. It is well known that the distance of the tensor product of two linear codes is the product of the distances of the individual codes (see Proposition 6.6 for a formal statement). We can use this proposition to reduce GapMDP$_\gamma$ to GapMDP$_{\gamma^2}$ for any $\gamma \geqslant 1$. In particular, we can obtain, for any constant γ, the intractability of GapMDP$_\gamma$ starting from GapMDP$_{1.01}$ by just recursively tensoring the input code $\lceil \log_{1.01}\gamma \rceil$ times.

2.5 Parameterized Intractability of k-SVP

We begin this subsection by briefly describing Khot's reduction. The starting point of Khot's reduction is the NP-hardness of approximating NVP in every ℓ_p norm to any constant factor [ABSS97]. Let us recall that in NVP in the ℓ_p norm, we are given a matrix $\mathbf{A} \in \mathbb{Z}^{n\times m}$, an integer k, and a target vector $\mathbf{y} \in \mathbb{Z}^n$, and the goal is to determine whether there is any $\mathbf{x} \in \mathbb{Z}^m$ such that[10] $\|\mathbf{Ax}-\mathbf{y}\|_p^p$ is at most k. The result of Arora et al. [ABSS97] states that for any constant $\gamma \geqslant 1$, it is NP-hard to distinguish the case when there exists $\mathbf{x}$ such that $\|\mathbf{Ax}-\mathbf{y}\|_p^p \leqslant k$ from the case when for all (integral) $\mathbf{x}$ we have that $\|\mathbf{Ax}-\mathbf{y}\|_p^p > \gamma k$. Khot's reduction proceeds in four steps. First, he constructs a gadget lattice called the "BCH Lattice" using BCH Codes. Next, he reduces NVP in the ℓ_p norm (where $p \in (1,\infty)$) to an instance of SVP on an intermediate lattice by using the BCH Lattice. This intermediate lattice has the following property. For any YES instance of NVP the intermediate lattice contains multiple copies of the witness of the YES instance; For any NO instance of NVP there are also many "annoying vectors" (but far less than the total number of YES instance witnesses) which look like witnesses of a YES instance. However, since the annoying vectors are outnumbered, Khot reduces this intermediate lattice to a proper SVP instance, by randomly picking a sub-lattice via a random homogeneous linear constraint on the coordinates of the lattice vectors (this annihilates all the annoying vectors while retaining at least one witness for the YES instance). Thus he obtains some constant factor hardness for SVP. Finally, the gap is amplified via "Augmented Tensor Product". It is important to note that Khot's reduction is randomized, and thus his result of inapproximability of SVP is based on NP $\neq$ RP.

Trying to follow Khot's reduction, in order to show the parameterized intractability of k-SVP, we face only one obstacle: there is no known parameterized inapproximability of k-NVP for any constant factor greater than 1. Let us denote by GapNVP$_{p,\eta}$ for any constant $\eta \geqslant 1$ the gap version of k-NVP in the ℓ_p norm. Recall that in GapNVP$_{p,\eta}$ we are given a matrix $\mathbf{A} \in \mathbb{Z}^{n\times m}$, a target vector $\mathbf{y} \in \mathbb{Z}^n$, and a parameter k, and we would like to distinguish the

[10] Previously, we use $\|\mathbf{Ax}-\mathbf{y}\|_p$ instead of $\|\mathbf{Ax}-\mathbf{y}\|_p^p$. However, from the fixed parameter perspective, these two versions are equivalent since the parameter k is only raised to the p-th power, and p is a constant in our setting.

case when there exists $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_p^p \leqslant k$ from the case when for all $\mathbf{x} \in \mathbb{Z}^m$ we have that $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_p^p > \eta k$. As it turns out, our reduction from k-LDS to GAPMLD, can be translated to show the inapproximability of GAPMLD over any larger (but still constant) field in a straightforward manner. We then provide a simple for GAPMLD over large field to GAPNVP$_p$ that establishes W[1]-hardness of the latter.

Once we have established the constant parameterized inapproximability of GAPNVP$_p$, we follow Khot's reduction, and everything goes through as it is to establish the inapproximability for some factor of the gap version of k-SVP in the ℓ_p norm (where $p \in (1, \infty)$). We denote by GAPSVP$_{p,\gamma}$ for some constant $\gamma(p) \geqslant 1$ the the gap version of k-SVP (in the ℓ_p norm) where we are given a matrix $\mathbf{B} \in \mathbb{Z}^{n \times m}$ and a parameter $k \in \mathbb{N}$, and we would like to distinguish the case when there exists a non-zero $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{B}\mathbf{x}\|_p^p \leqslant k$ from the case when for all $\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ we have that $\|\mathbf{B}\mathbf{x}\|_p^p > \gamma k$. Let $\gamma^* := \frac{2^p}{2^{p-1}+1}$. Following Khot's reduction, we obtain the inapproximability of GAPSVP$_{p,\gamma^*}$.

Summarizing, in Figure 2, we provide the proof outline of our W[1]-hardness result of GAPSVP$_p$ to some constant approximation factor, for every $p \in (1, \infty)$.

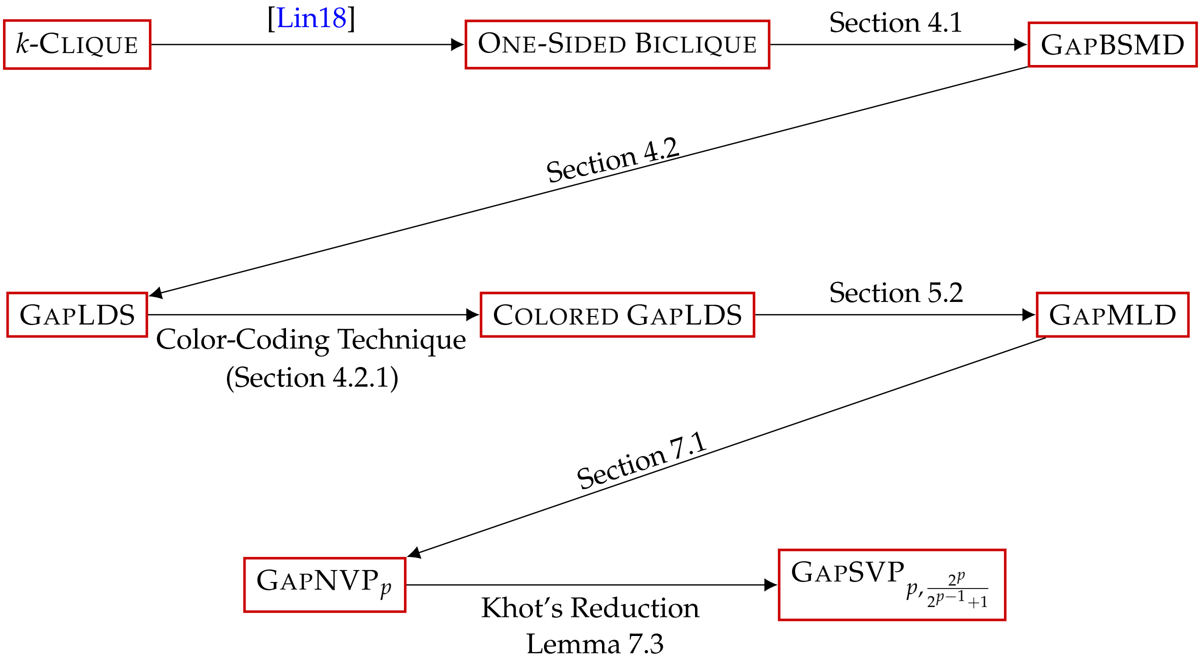

Figure 2: The figure provides an overview of our reduction from the canonical W[1]-complete k-Clique problem to the parameterized Shortest Vector problem in the ℓ_p norm, where $p \in (1, \infty)$. The proof outline of the reduction from k-Clique to GAPMLD (to rule out constant approximation parameterized algorithms for NCP) is reiterated in the above figure. In Section 7.1, we reduce GAPMLD to GAPNVP and obtain the constant inapproximability of NVP. Then, applying Lemma 7.3 (i.e., Khot's reduction) implies the parameterized inapproximability of GAPSVP$_{p,\frac{2^p}{2^{p-1}+1}}$.

3 Preliminaries

We use the following notation throughout the paper.

Notation. We use boldface (e.g. $\mathbf{x}, \mathbf{A}$ or $\mathbf{0}$) to stress that the objects are vectors or matrices. When we refer to a vector $\mathbf{x}$, we assume that it is a column vector. Moreover, since subscripts will often be used for other purposes, we instead use the notation $\mathbf{x}[i]$ for $i \in \mathbb{N}$ to denote the value of the i-th coordinate of the vector. For matrices, we use $\mathbf{A}[i]$ to denote its i-th column vector.

For $p \in \mathbb{N}$, we use $\mathbf{1}_p$ (respectively, $\mathbf{0}_p$) to denote the all ones (respectively, all zeros) vector of length p. We sometimes drop the subscript if the dimension is clear from the context. For $p, q \in \mathbb{N}$, we use $\mathbf{0}_{p \times q}$ to denote the all zeroes matrix of p rows and q columns. We use Id_q to denote the identity matrix of q rows and q columns.

For any vector $\mathbf{x} \in \mathbb{R}^d$, the ℓ_p norm of $\mathbf{x}$ is defined as $\ell_p(\mathbf{x}) = \|\mathbf{x}\|_p = \left(\sum_{i=1}^d |\mathbf{x}[i]|^p\right)^{1/p}$. Thus, $\ell_\infty(\mathbf{x}) = \|\mathbf{x}\|_\infty = \max_{i \in [d]}\{|\mathbf{x}_i|\}$. The ℓ_0 norm of $\mathbf{x}$ is defined as $\ell_0(\mathbf{x}) = \|\mathbf{x}\|_0 = |\{\mathbf{x}[i] \neq 0 : i \in [d]\}|$, i.e., the number of non-zero entries of $\mathbf{x}$. We note that the ℓ_0 norm is also referred to as the Hamming norm. For $a \in \mathbb{N}$, $t \in \mathbb{N} \cup \{0\}$, and $\mathbf{s} \in \{0,1\}^a$, we use $\mathcal{B}_a(\mathbf{s}, t)$ to denote the Hamming ball of radius t centered at $\mathbf{s}$, i.e., $\mathcal{B}_a(\mathbf{s}, t) = \{\mathbf{x} \in \{0,1\}^a \mid \|\mathbf{s} - \mathbf{x}\|_0 \leqslant t\}$. Finally, given two vectors $\mathbf{x}$ and $\mathbf{y}$, we use $\mathbf{x} \circ \mathbf{y}$ to denote the concatenation of vectors $\mathbf{x}$ and $\mathbf{y}$.

We sometimes use $\dot\cup$ to emphasize that the sets are disjoint; for instance, we may write $G = (L \,\dot\cup\, R, E)$ for bipartite graphs to indicate that L, R are disjoint.

3.1 Parameterized Promise Problems and (Randomized) FPT Reductions

In this subsection, we briefly describe the various kinds of fixed-parameter reductions that are used in this paper. We start by defining the notion of promise problems in the fixed-parameter world, which is naturally analogues to promise problems in the NP world (see e.g. [Gol06]).

Definition 3.1. *A parameterized promise problem Π is a pair of parameterized languages (Π_{YES}, Π_{NO}) such that $\Pi_{YES} \cap \Pi_{NO} = \emptyset$.*

Next, we formalize the notion of algorithms for these parameterized promise problems:

Definition 3.2. *A deterministic algorithm $\mathcal{A}$ is said to be an* FPT algorithm for Π *if the following holds:*

- *On any input (x, k), $\mathcal{A}$ runs in time $f(k)|x|^c$ for some computable function f and constant c.*
- *(YES) For all $(x, k) \in \Pi_{YES}$, $\mathcal{A}(x, k) = 1$.*
- *(NO) For all $(x, k) \in \Pi_{NO}$, $\mathcal{A}(x, k) = 0$.*

Definition 3.3. *A Monte Carlo algorithm $\mathcal{A}$ is said to be a* randomized FPT algorithm for Π *if the following holds:*

- *$\mathcal{A}$ runs in time $f(k)|x|^c$ for some computable function f and constant c (on every randomness).*

- *(YES) For all* $(x,k) \in \Pi_{YES}$*,* $\Pr[\mathcal{A}(x,k) = 1] \geqslant 2/3$*.*
- *(NO) For all* $(x,k) \in \Pi_{NO}$*,* $\Pr[\mathcal{A}(x,k) = 0] \geqslant 2/3$*.*

Finally, we define deterministic and randomized reductions between these problems.

Definition 3.4. *A (deterministic) FPT reduction from a parameterized promise problem* Π *to a parameterized promise problem* Π' *is a (deterministic) procedure that transforms* (x,k) *to* (x',k') *that satisfies the following:*

- *The procedure runs in* $f(k)|x|^c$ *for some computable function* f *and constant* c*.*
- *There exists a computable function* g *such that* $k' \leqslant g(k)$ *for every input* (x,k)*.*
- *For all* $(x,k) \in \Pi_{YES}$*,* $(x',k') \in \Pi'_{YES}$*.*
- *For all* $(x,k) \in \Pi_{NO}$*,* $(x',k') \in \Pi'_{NO}$*.*

Definition 3.5. *A randomized (one sided error) FPT reduction from a parameterized promise problem* Π *to a parameterized promise problem* Π' *is a randomized procedure that transforms* (x,k) *to* (x',k') *that satisfies the following:*

- *The procedure runs in* $f(k)|x|^c$ *for some computable function* f *and constant* c *(on every randomness).*
- *There exists a computable function* g *such that* $k' \leqslant g(k)$ *for every input* (x,k)*.*
- *For all* $(x,k) \in \Pi_{YES}$*,* $\Pr[(x',k') \in \Pi'_{YES}] \geqslant 1/(f'(k)|x|^{c'})$ *for some computable function* f' *and constant* c'*.*
- *For all* $(x,k) \in \Pi_{NO}$*,* $\Pr[(x',k') \in \Pi'_{NO}] = 1$*.*

Note that the above definition corresponds to the notion of *Reverse Unfaithful Random (RUR) reductions* in the classical world [Joh90]. The only difference (besides the allowed FPT running time) is that the above definition allows the probability that the YES case gets map to the YES case to be as small as $1/(f'(k)\text{poly}(|x|))$, whereas in the RUR reductions this can only be $1/\text{poly}(|x|)$. The reason is that, as we will see in Lemma 3.7 below, FPT algorithms can afford to repeat the reduction $f'(k)\text{poly}(|x|)$ times, whereas polynomial time algorithms can only repeat $\text{poly}(|x|)$ times.

We also consider randomized two-sided error FPT reductions, which are defined as follows.

Definition 3.6. *A randomized* two sided error *FPT reduction from a parameterized promise problem* Π *to a parameterized promise problem* Π' *is a randomized procedure that transforms* (x,k) *to* (x',k') *that satisfies the following:*

- *The procedure runs in* $f(k)|x|^c$ *for some computable function* f *and constant* c *(on every randomness).*
- *There exists a computable function* g *such that* $k' \leqslant g(k)$ *for every input* (x,k)*.*

- *For all* $(x,k) \in \Pi_{YES}$*,* $\Pr[(x',k') \in \Pi'_{YES}] \geqslant 2/3$*.*
- *For all* $(x,k) \in \Pi_{NO}$*,* $\Pr[(x',k') \in \Pi'_{NO}] \geqslant 2/3$*.*

Note that this is not a generalization of the standard randomized FPT reduction (as defined in Definition 3.5), since the definition requires the success probabilities for the YES and NO cases to be constants independent of the parameter. In both cases, using standard techniques randomized FPT reductions, can be used to transform randomized FPT algorithms for Π' to randomized FPT algorithm for Π, as stated by the following lemma:

Lemma 3.7. *Suppose there exists a randomized (one sided/ two sided) error FPT reduction from a parameterized promise problem* Π *to a parameterized promise problem* Π'*. If there exists a randomized FPT algorithm* $\mathcal{A}$ *for* Π'*, there there also exists a randomized FPT algorithm for* Π*.*

Proof. We prove this for one sided error reductions, the other case follows using similar arguments. Suppose there exists a randomized one sided error reduction from Π to Π'. Let $f'(\cdot), c'$ be as in Definition 3.5. We consider the following subroutine. Given instance (x,k) of promise problem Π, we apply the randomized reduction on (x,k) to get instance (x',k') of promise problem Π'. We run $\mathcal{A}$ on (x',k') repeatedly $100\log(f'(k)|x|^c)$ times, and output the majority of the outcomes.

If (x,k) is a YES instance, then with probability at least $1/(f'(k)|x|^{c'})$, (x',k') is also a YES instance for Π'. Using Chernoff bound, conditioned on (x',k') being a YES instance, the majority of the outcomes is YES with probability at least $1 - e^{-10\log(f'(k)|x|^{c'})}$. Therefore using union bound, the output of the above algorithm is YES with probability at least $1/(f'(k)|x|^{c'}) - e^{-10\log(f'(k)|x|^{c'})} \geqslant 1/2(f'(k)|x|^{c'})$. Similarly, if (x,k) is a NO instance, then the subroutine outputs YES with probability at most $e^{-10\log(f'(k)|x|^{c'})}$.

Equipped with the above subroutine, our algorithm is simply the following: given (x,k), it runs the subroutine $10f'(k)|x|^{c'}$ times. If at least one of the outcomes is YES, then the algorithm outputs YES, otherwise it outputs NO. Again we can analyze this using elementary probability. If (x,k) is a YES instance, then the algorithm outputs NO only if outcomes of all the trials is NO. Therefore, the algorithm outputs YES with probability at least $1 - (1 - 1/2(f'(k)|x|^{c'}))^{10f'(k)|x|^{c'}} \geqslant 0.9$. Conversely, if (x,k) is a NO instance, then by union bound, the algorithm outputs NO with probability at least $1 - 10f'(k)|x|^{c'}e^{-10\log(f'(k)|x|^{c'})} \geqslant 0.9$. Finally, if $\mathcal{A}$ is FPT, then the running time of the proposed algorithm is also FPT. Hence the claim follows[11]. □

Since the conclusion of the above proposition holds for both types of randomized reductions, we will not be distinguishing between the two types in the rest of the paper.

3.2 Bipartite Subgraph with Minimum Degrees

As stated in the proof overview, it will be convenient to view Lin's hardness of BICLIQUE in terms of hardness of approximating Bipartite Subgraph with Minimum Degree, where the

[11] For the case of 2-sided error, we change the final step of the algorithm as follows; we invoke the subroutine $O(\log 1/\delta)$-times (where δ is a constant) and again output the majority of the outcomes. The guarantees again follow by a Chernoff bound argument.

goal, given a bipartite graph G, is to find a non-empty subgraph H of G such that every left vertex in H has degree at least h and every right vertex of H has degree at least s. The parameter here is $s + h$.

The gap version that we will use is to distinguish between the YES case where there is such a subgraph with hs edges, i.e., a complete bipartite subgraph with s left vertices and h right vertices, and the NO case where every such subgraph H must contains more than $\gamma \cdot hs$ edges (for $\gamma \geqslant 1$). This is defined more precisely below.

γ-Gap Bipartite Subgraph with Minimum Degree Problem (GapBSMD$_\gamma$)

Input: A bipartite graph $G = (L \dot\cup R, E)$ with n vertices, $s, h \in \mathbb{N}$

Parameter: $s + h$

Question: Distinguish between the following two cases:

- (YES) There is a complete bipartite subgraph of G with s vertices in L and h vertices in R.
- (NO) For any non-empty subgraph H of G such that every left vertex of H has degree at least h and every right vertex of H has degree at least s, H contains at least $\gamma \cdot (sh)$ edges.

3.3 Linear Dependent Set Problems

We next introduce the parameterized Linear Dependent Problem. In this problem, we are given $\mathbb{F}_q$-vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$ and the goal is to find a smallest number of vectors that are linearly dependent. It should be stressed here that the field $\mathbb{F}_q$ is part of the input (i.e. q will be of the order of n in our proofs); this is indeed the main difference between this problem and the Minimum Distance Problem which is in fact equivalent to the Linear Dependent Problem for a fixed $q = 2$.

γ-Gap Linear Dependent Set Problem (GapLDS$_\gamma$)

Input: A field $\mathbb{F}_q$, a set $\mathcal{W} \subseteq \mathbb{F}_q^m$ and a positive integer $k \in \mathbb{N}$.

Parameter: k

Question: Distinguish between the following two cases:

- (YES) there exist k distinct vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k \in \mathcal{W}$ and $a_1, \ldots, a_k \in \mathbb{F}_q \setminus \{0\}$ such that $\sum_{i \in [k]} a_i \mathbf{w}_i = \mathbf{0}$ (which implies that $\mathbf{w}_1, \ldots, \mathbf{w}_k$ are linearly dependent)
- (NO) there are no $\gamma \cdot k$ vectors in $\mathcal{W}$ that are linearly dependent

Notice here that the guarantee in the YES case is slightly stronger than "there exist k vectors that are linearly dependent", as we also require the coefficients to be non-zero. (This would be automatically true if, for instance, any $k-1$ vectors are linearly dependent.) We remark that this does not significantly change the complexity of the problem, as our hardness applies to both versions; however, it will be more convenient in subsequent steps to have such an additional guarantee.

It will also be convenient to work with a colored version of GapLDS which we introduce below.

γ-Gap Colored Linear Dependent Set Problem (GapLDS$_\gamma^{\mathsf{col}}$)

Input: A field $\mathbb{F}_q$, a set $\mathcal{W} \subseteq \mathbb{F}_q^m$, a positive integer $k \in \mathbb{N}$ and a coloring $c : \mathcal{W} \to [k]$

Parameter: k

Question: Distinguish between the following two cases:

- (YES) there exist k vectors $\mathbf{w}_1, \dots, \mathbf{w}_k \in \mathcal{W}$ of distinct colors (i.e. $c(\{\mathbf{w}_1, \dots, \mathbf{w}_k\}) = [k]$) and $a_1, \dots, a_k \in \mathbb{F}_q \setminus \{0\}$ such that $\sum_{i \in [k]} a_i \mathbf{w}_i = \mathbf{0}$
- (NO) there are no $\gamma \cdot k$ vectors in $\mathcal{W}$ that are linearly dependent

We point out that in we require the vectors to have distinct colors only in the YES case; in the NO case, we assume that there are no $\gamma \cdot k$ linearly dependent vectors of arbitrary colors.

3.4 Minimum Distance Problem

In this subsection, we define the fixed-parameter variant of the minimum distance problem and other relevant parameterized problems. We actually define them as gap problems – as later in the paper, we show the constant inapproximability of these problems.

For every $\gamma \geqslant 1$, we define the γ-gap minimum distance problem[12] as follows:

γ-Gap Minimum Distance Problem (GapMDP$_\gamma$)

Input: A matrix $\mathbf{A} \in \mathbb{F}_2^{n \times m}$ and a positive integer $k \in \mathbb{N}$

Parameter: k

Question: Distinguish between the following two cases:

- (YES) there exists $\mathbf{x} \in \mathbb{F}_2^m \setminus \{\mathbf{0}\}$ such that $\|\mathbf{A}\mathbf{x}\|_0 \leqslant k$
- (NO) for all $\mathbf{x} \in \mathbb{F}_2^m \setminus \{\mathbf{0}\}$, $\|\mathbf{A}\mathbf{x}\|_0 > \gamma \cdot k$

[12] In the parameterized complexity literature, this problem is referred to as the k-Even set problem [DFVW99] and the input to the problem is (equivalently) given through the parity-check matrix, instead of the generator matrix as described in this paper.

Next, for every $\gamma \geqslant 1$, we define the γ-gap maximum likelihood decoding problem[13] as follows:

γ-Gap Maximum Likelihood Decoding Problem ($\textsc{GapMLD}_\gamma$)

Input: A matrix $\mathbf{A} \in \mathbb{F}_2^{n \times m}$, a vector $\mathbf{y} \in \mathbb{F}_2^n$ and a positive integer $k \in \mathbb{N}$

Parameter: k

Question: Distinguish between the following two cases:

- (YES) there exists $\mathbf{x} \in \mathcal{B}_m(\mathbf{0}, k)$ such that $\mathbf{A}\mathbf{x} = \mathbf{y}$
- (NO) for all $\mathbf{x} \in \mathcal{B}_m(\mathbf{0}, \gamma k)$, $\mathbf{A}\mathbf{x} \neq \mathbf{y}$

For brevity, we shall denote the exact version (i.e., $\textsc{GapMLD}_1$) of the problem as MLD.

It should be noted that the Odd Set problem discussed in the introduction is closely related to GAPMLD; in particular, the only different is that, in ODDSET, $\mathbf{y}$ is not part of the input but is always fixed as $\mathbf{1}$, the all-ones vector. Indeed, it is not hard to see that our parameterized hardness of approximation for GAPMLD also transfers to that of GAPODDSET. This is formulated in Appendix A.

We also define the GAPMLD problem over larger (constant) field $\mathbb{F}_p$ below; this version of the problem will be used in proving hardness of Nearest Vector Problem. In this version, we have an additional requirement that, in the YES case, the solution $\mathbf{x}$ must be a $\{0,1\}$-vector. (Note that this is automatically the case for GAPMLD over $\mathbb{F}_2$.)

γ-Gap Maximum Likelihood Decoding Problem over $\mathbb{F}_p$ ($\textsc{GapMLD}_{\gamma,p}$)

Input: A matrix $\mathbf{A} \in \mathbb{F}_p^{n \times m}$, a vector $\mathbf{y} \in \mathbb{F}_p^n$ and a positive integer $k \in \mathbb{N}$

Parameter: k

Question: Distinguish between the following two cases:

- (YES) there exists $\mathbf{x} \in \{0,1\}^m$ with $\|\mathbf{x}\|_0 \leqslant k$ such that $\mathbf{A}\mathbf{x} = \mathbf{y}$
- (NO) for all $\mathbf{x} \in \mathbb{F}_q^m$ such that $\|\mathbf{x}\|_0 \leqslant \gamma k$, $\mathbf{A}\mathbf{x} \neq \mathbf{y}$

Finally, we introduce a "sparse" version of the GAPMLD problem called the sparse nearest codeword problem, and later in the paper, we show a reduction from GAPMLD to this problem, followed by a reduction from this problem to GAPMDP. As its name suggest, the sparse nearest codeword problem priorities not only the Hamming distance of the codeword $\mathbf{A}\mathbf{x}$ to the target vector $\mathbf{y}$ but also the "sparsity" (i.e. Hamming weight) of $\mathbf{x}$. Formally, for every $\gamma \geqslant 1$, we define the γ-gap sparsest nearest codeword problem as follows:

[13] The maximum likelihood decoding problem is also equivalently known in the literature as the nearest codeword problem.

γ-Gap Sparse Nearest Codeword Problem (GAPSNC$_\gamma$)

Input: A matrix $\mathbf{A} \in \mathbb{F}_2^{n\times m}$, a vector $\mathbf{y} \in \mathbb{F}_2^n$ and a positive integer $k \in \mathbb{N}$

Parameter: k

Question: Distinguish between the following two cases:

- (YES) there exists $\mathbf{x} \in \mathbb{F}_2^m$ such that $\|\mathbf{Ax} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0 \leqslant k$
- (NO) for all $\mathbf{x} \in \mathbb{F}_2^m$, $\|\mathbf{Ax} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0 > \gamma \cdot k$

3.5 Shortest Vector Problem and Nearest Vector Problem

In this subsection, we define the fixed-parameter variants of the shortest vector and nearest vector problems. As in the previous subsection, we define them as gap problems, for the same reason that later in the paper, we show the constant inapproximability of these two problems.

Fix $p \in \mathbb{R}_{\geqslant 1}$. For every $\gamma \geqslant 1$, we define the γ-gap shortest vector problem in the ℓ_p-norm[14] as follows:

γ-Gap Shortest Vector Problem (GAPSVP$_{p,\gamma}$)

Input: A matrix $\mathbf{A} \in \mathbb{Z}^{n\times m}$ and a positive integer $k \in \mathbb{N}$

Parameter: k

Question: Distinguish between the following two cases:

- (YES) there exists $\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ such that $\|\mathbf{Ax}\|_p^p \leqslant k$
- (NO) for all $\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$, $\|\mathbf{Ax}\|_p^p > \gamma \cdot k$

For every $\gamma \geqslant 1$, we define the γ-gap nearest vector problem in the ℓ_p-norm as follows:

γ-Gap Nearest Vector Problem (GAPNVP$_{p,\gamma}$)

Input: A matrix $\mathbf{A} \in \mathbb{Z}^{n\times m}$, vector $\mathbf{y} \in \mathbb{Z}^n$ and a positive integer $k \in \mathbb{N}$

Parameter: k

Question: Distinguish between the following two cases:

- (YES) there exists $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{Ax} - \mathbf{y}\|_p^p \leqslant k$

[14] Note that we define GAPNVP and GAPSVP problems in terms of ℓ_p^p, whereas traditionally, it is defined in terms of ℓ_p. However, it is sufficient for us to work with the ℓ_p^p variant, since an α-factor inapproximability in ℓ_p^p translates to an $\alpha^{1/p}$-factor inapproximabillity in the ℓ_p norm, for any $\alpha \geqslant 1$

- (NO) for all $\mathbf{x} \in \mathbb{Z}^m$, $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_p^p > \gamma \cdot k$

3.6 Error-Correcting Codes

An error correcting code C over alphabet Σ is a function $C : \Sigma^m \to \Sigma^h$ where m and h are positive integers which are referred to as the *message length* (aka *dimension*) and *block length* of C respectively. Intuitively, C encodes an original message of length m to an encoded message of length h. The *distance* of a code, denoted by $d(C)$, is defined as $\min_{x \neq y \in \Sigma^m} \|C(x) - C(y)\|_0$, i.e., the number of coordinates on which $C(x)$ and $C(y)$ disagree. We also define the systematicity of a code as follows: Given $s \in \mathbb{N}$, a code $C : \Sigma^m \to \Sigma^h$ is *s-systematic* if there exists a size-s subset of $[h]$, which for convenience we identify with $[s]$, such that for every $x \in \Sigma^s$ there exists $w \in \Sigma^m$ in which $x = C(w) \mid_{[s]}$. We use the shorthand $[h, m, d]_{|\Sigma|}$ to denote a code of message length m, block length h, and distance d.

Additionally, we will need the following existence and efficient construction of BCH codes for every message length and distance parameter.

Theorem 3.8 (BCH Code [Hoc59, BR60]). *For any choice of $h, d \in \mathbb{N}$ such that $h + 1$ is a power of two and that $d \leqslant h$, there exists a linear code over $\mathbb{F}_2$ with block length h, message length $h - \left\lceil \frac{d-1}{2} \right\rceil \cdot \log(h+1)$ and distance d. Moreover, the generator matrix of this code can be computed in poly(h) time.*

Finally, we define the tensor product of codes which will be used later in the paper. Consider two linear codes $C_1 \subseteq \mathbb{F}_2^m$ (generated by $\mathbf{G}_1 \in \mathbb{F}_2^{m \times m'}$) and $C_2 \subseteq \mathbb{F}_2^n$ (generated by $\mathbf{G}_2 \in \mathbb{F}_2^{n \times n'}$). Then the tensor product of the two codes $C_1 \otimes C_2 \subseteq \mathbb{F}_2^{m \times n}$ is defined as

$$C_1 \otimes C_2 = \{\mathbf{G}_1 \mathbf{X} \mathbf{G}_2^\top | \mathbf{X} \in \mathbb{F}_2^{m' \times n'}\}.$$

We will only need two properties of tensor product codes. First, the generator matrix of the tensor products of two linear codes C_1, C_2 with generator matrices $\mathbf{G}_1, \mathbf{G}_2$ can be computed in polynomial time in the size of $\mathbf{G}_1, \mathbf{G}_2$. Second, the distance of $C_1 \otimes C_2$ is exactly the product of the distances of the two codes, i.e.,

$$d(C_1 \otimes C_2) = d(C_1) d(C_2).$$

4 Parameterized Inapproximability of Linear Dependent Set

In this section, we show that the Linear Dependent Set problem has no constant factor FPT approximation algorithm unless $\mathsf{W}[1] = \mathsf{FPT}$. More formally, we prove the following:

Theorem 4.1. *For every $\gamma \geqslant 1$, $\textsc{GapLDS}_\gamma$ and $\textsc{GapLDS}_\gamma^{col}$ are W[1]-hard.*

The proof consists of two steps. First, we will reformulate Lin's reduction for the Biclique problem in terms of hardness of GAPBSMD. Then, we reduce GAPBSMDto our target problem GAPLDS.

4.1 Translating ONE-SIDED BICLIQUE to GAPBSMD

In the first step of our proof, we will show that GAPBSMD is W[1]-hard to approximate to within any constant factor, as stated more precisely below.

Theorem 4.2. *For every $\gamma \geqslant 1$, GAPBSMD$_\gamma$ is W[1]-hard.*

Our result relies crucially on the recent W[1]-hardness of approximation result for the ONE-SIDED BICLIQUE problem by Lin [Lin18]. Recall that, in ONE-SIDED BICLIQUE, we are given a bipartite graph G and an integer s and the goal is to find s left vertices with maximum number of common neighbors. The following theorem is the main result of Lin [Lin18] for ONE-SIDED BICLIQUE.

Theorem 4.3 ([Lin18, Theorem 1.3]). *There is a polynomial time algorithm $\mathbb{A}$ such that, given a graph G with n vertices and $k \in \mathbb{N}$ with $\lceil n^{\frac{6}{k+6}} \rceil > (k+6)!$ and $6 \mid k+1$, it outputs a bipartite graph $G' = (A \dot\cup B, E)$ and $s = \binom{k}{2}$ satisfying:*

1. *(YES) If G contains a k-clique, then there are s vertices in A with at least $\lceil n^{\frac{6}{k+1}} \rceil$ common neighbors in B;*
2. *(NO) If G does not contain a k-clique, any s vertices in A have at most $(k+1)!$ common neighbors in B.*

Another ingredient of our reduction is a simple observation regarding the size of bipartite graphs with prescribed minimum degrees, conditioned on the fact that any small subset of left vertices have small number of neighbors. This is stated below.

Claim 4.4. *For any $s, \ell, h \in \mathbb{N}$, let $(X \cup Y, E_W)$ be a non-empty bipartite graph such that*

(i) every vertex in X has at least h neighbors,

(ii) every vertex in Y has at least s neighbors, and,

(iii) every s-vertex set of X has at most ℓ common neighbors.

Furthermore, the parameters h, ℓ and s satisfy $h/\ell \geqslant \gamma^s s^s$. Then, $|E_W| \geqslant (h/\ell)^{1/s} \geqslant \gamma \cdot hs$.

Proof of Claim 4.4. Consider any vertex $u \in X$. By (i), u has at least h neighbors in Y, so $|Y| \geqslant h$. By (ii), for every $v \in Y$, v has at least s neighbors in X. If $\binom{|X|}{s}\ell < |Y|$, then there must exist a s-vertex set in X which has more than ℓ common neighbors in Y. Thus, we must have

$$|X|^s \geqslant \binom{|X|}{s} \geqslant \frac{|Y|}{\ell} \geqslant \frac{h}{\ell}.$$

By (i) and our choice of parameters h, ℓ, s, we can conclude that $|E_W| \geqslant h|X| \geqslant (h/\ell)^{1/s} \cdot h \geqslant \gamma \cdot hs$, as desired. □

With Theorem 4.3 and Claim 4.4 in place, we can prove Theorem 4.2 simply by using the reduction from Theorem 4.3 and choosing an appropriate value of h; the guarantee in the NO case would then follow from Claim 4.4.

Proof of Theorem 4.2. We reduce from the k-CLIQUE problem which is well-known to be W[1]-complete. Let (G, k) be an instance of k-CLIQUE and n be the number of vertices in G. Without loss of generality, we can assume that $6 \mid k+1$ and $\lceil n^{\frac{6}{k+6}} \rceil > (k+6)! \cdot (\gamma \cdot k^2)^{k^2}$. Using the reduction in Theorem 4.3, we can produce $(G', s = \binom{k}{2})$ in polynomial time with the guarantees as in the theorem. We then set $h = (k+6)! \cdot (\gamma \cdot k^2)^{k^2}$ and let (H, s, h) be our instance of GAPBSMD$_\gamma$. We will next show that this is indeed a valid reduction from k-Clique to GAPBSMD$_\gamma$.

(YES Case) Suppose that G contains a k-clique. Then, Theorem 4.3 guarantees that G' contains a complete bipartite subgraph with s left vertices and h right vertices as desired.

(NO Case) Suppose that G does not contain a k-clique. Now, consider any non-empty subgraph H of G' such that every left vertex of H has at least h neighbors and every right vertex of H contains at least s neighbors, i.e., H satisfies condition (i) and (ii) in Claim 4.4. Furthermore, since G does not contain a k-clique, guarantees that every s vertices in A contains at most $\ell = (k+1)!$ common neighbors. It can be easily verified that our setting of parameters h, ℓ and s satisfies the inequality $h/\ell \geqslant \gamma^s s^s$. Hence, by applying Claim 4.4 on H, the number of edges in H must be at least $\gamma \cdot (hs)$. This means that (H, s, h) is a NO instance of GAPBSMD$_\gamma$ as desired. □

4.2 Reducing GAPBSMD to GAPLDS

We now move on to the next step of our proof, which is the reduction from GAPBSMD to GAPLDS.

Since the reduction itself will be used in the subsequent proofs (with different parameter selections), we also state it separately below. We remark that the reduction as stated below goes from GAPBSMD$_\gamma$ to the uncolored version of the problem (GAPLDS$_\gamma$); we will state how to go from here to the colored version later on.

Theorem 4.5. *Let $\gamma \geqslant 1$ be any constant. There is a polynomial time algorithm that, given an instance (G, s, h) of* GAPBSMD$_\gamma$ *where G contains n vertices and any prime power $q > n$, produces an instance $(\mathcal{W} \subseteq \mathbb{F}_q^m, k = hs)$ of* GAPLDS$_\gamma$ *such that*

- *(YES) If (G, s, h) is a YES instance of* GAPBSMD$_\gamma$*, then $(\mathcal{W}, k)$ is a YES instance of* GAPLDS$_\gamma$*.*
- *(NO) If (G, s, h) is a NO instance of* GAPBSMD$_\gamma$*, then $(\mathcal{W}, k)$ is a NO instance of* GAPLDS$_\gamma$*.*

Proof. Assume that an instance $(G = (L \dot\cup R, E), s, h)$ of GAPBSMD$_\gamma$ and a prime power $q > |L| + |R|$ are given. Before we construct $\mathcal{W}$, let us first define additional notation. We identify vertices in $L \dot\cup R$ with distinct elements of $\mathbb{F}_q$. Let $B := s + h$ and let $\iota : L \cup R \to \mathbb{F}_q^B$ be defined as follows.

- for each $v \in R$, $\iota(v) := (1, v, \ldots, v^{h-2}) \circ \mathbf{0}_{B-h+1}$,
- for each $u \in L$, $\iota(u) := (1, u, \ldots, u^{s-2}) \circ \mathbf{0}_{B-s+1}$.

By a well-known property of Vandermonde matrices, any $h-1$ vectors in $\iota(R)$ are linearly independent and any h vectors from $\iota(R)$ are linearly dependent. To summarize, we have

(R1) For all $I \in \binom{R}{h}$, the vectors $\{\iota(v) : v \in I\}$ are linearly dependent.

(R2) For all $I \in \binom{R}{h-1}$, the vectors $\{\iota(v) : v \in I\}$ are linearly independent.

Similarly, we also have

(L1) For all $I \in \binom{L}{s}$, the vectors $\{\iota(u) : u \in I\}$ are linearly dependent.

(L2) For all $I \in \binom{L}{s-1}$, the vectors $\{\iota(u) : u \in I\}$ are linearly independent.

Let $m = qB$ and consider vectors from $\mathbb{F}_q^m = \mathbb{F}_q^{qB}$, which can be seen as the concatenation of q blocks, each of B coordinates. For $x \in \mathbb{F}_q^m$, we use the notation $\mathbf{x}^{(i)}$ to refer to the i-block, i.e. the B-dimensional vector given by coordinates $(i-1)B+1, (i-1)B+2, \ldots, iB$.

Construction of $(\mathcal{W}, k)$**.** First, we let $k = hs$. Then, for each $(u, v) = e \in E$ (where $u \in L, v \in R$), we introduce a vector $\mathbf{w}_e \in \mathbb{F}_q^{qB}$ such that

(W1) for all $i \in [q] \setminus \{v, u\}$, $\mathbf{w}_e^{(i)} = \mathbf{0}_B$,

(W2) $\mathbf{w}_e^{(v)} = \iota(u)$,

(W3) $\mathbf{w}_e^{(u)} = \iota(v)$.

That is, we can imagine w_e as being partitioned q blocks of B coordinates, with the representation of u appearing in the v-th block and the representation of v appearing in the u-th block. Note the use of u and v in the definition: the v-th block on its own describes both v (by its position) and u (by its content), and similarly the u-th block also describes both endpoints of e. We then let

$$\mathcal{W} := \{w_e : e \in E\}.$$

Obviously, $(\mathcal{W}, k)$ can be computed in polynomial time. We next argue its correctness.

(YES case) Suppose (G, s, h) is a YES instance of GAPBSMD$_\gamma$. There exist a set $X \in \binom{L}{s}$ and a set $Y \in \binom{R}{h}$ such that for all $u \in X$ and $v \in Y$, $(u, v) \in E$. By (R1) and (L1), there exists $b_u \in \mathbb{F}_q$ for each $u \in X$ and $b_v \in \mathbb{F}_q$ for each $v \in Y$ such that

$$\sum_{u \in X} b_u \iota(u) = \mathbf{0}_B \text{ and } \sum_{v \in Y} b_v \iota(v) = \mathbf{0}_B.$$

By (R2) and (L2), we deduce that, for all $u \in X$ and $v \in Y$, $b_u \neq 0$ and $b_v \neq 0$. We now claim that $\{\mathbf{w}_{(u,v)}\}_{u \in X, v \in Y}$ is the set of desired vectors, with the coefficient of $\mathbf{w}_{(u,v)}$ being $b_u b_v \neq 0$.

In other words, we are left to show that

$$\sum_{u\in X, v\in Y} b_u b_v \mathbf{w}_{(u,v)} = \mathbf{0}_m.$$

To see that this is true, let $\mathbf{w} = \sum_{i\in[s], j\in[h]} b_u b_v \mathbf{w}_{\{u_i,v_j\}}$. It is easy to check that

- by (W1), for every $z \in [q] \setminus (X \cup Y)$, $w^{(z)} = \mathbf{0}_B$,
- by (W2), for every $v \in Y$, $w^{(v)} = \sum_{u\in X} b_u b_v \iota(u) = b_v \sum_{u\in X} b_u \iota(u) = \mathbf{0}_B$,
- by (W3), for every $u \in X$, $w^{(u_i)} = \sum_{j\in[h]} a_i b_j \iota(v_j) = a_i \sum_{j\in[h]} b_j \iota(v_j) = \mathbf{0}_B$.

Hence, we have completed the proof for the YES case.

(NO case) Suppose (G, s, h) is a NO instance of GapBSMD$_\gamma$. Let $W \subseteq \mathcal{W}$ be a set of vectors that are linearly dependent. We define two vertex sets and their edge set as follows. Let

$$X := \{u \in L : \text{there exists } v \in R \text{ such that } w_{(u,v)} \in W\},$$
$$Y := \{v \in R : \text{there exists } u \in L \text{ such that } w_{(u,v)} \in W\},$$

and

$$E_W := \{e \in E : w_e \in W\}.$$

Note that X and Y are not empty because W is non-empty. By (R2) and (W3), for every $u \in X$, there exist at least h vertices in Y that are adjacent to u, i.e. $|N(u) \cap Y| \geqslant h$. Similarly, by (L2) and (W2), for every $v \in Y$, we have $|N(v) \cap X| \geqslant s$. Hence, by the guarantee in the NO case of GapBSMD$_\gamma$, we can conclude that $\gamma \cdot sh \leqslant |E_W| = |W|$ as desired. □

4.2.1 Reducing Uncolored LDS to Colored LDS

In this section, we show a simple reduction from the uncolored version of LDS to the colored version of LDS. As is usual in such a reduction, we will need the definition of perfect hash families and an efficient construction stated below.

Definition 4.6. *An (n, k)-perfect hash family is a collection $\mathcal{F}$ of functions from $[n]$ to $[k]$ such that, for every subset $S \subseteq [n]$ of size k, there exists $f \in \mathcal{F}$ that maps every S to distinct elements in $[k]$, i.e., $f(S) = [k]$.*

Theorem 4.7 ([NSS95])**.** *There exists an algorithm that, for any $n, k \in \mathbb{N}$, constructs an (n, k)-perfect hash family in time $2^{O(k)}$poly(n).*

If we use perfect hash families to reduce GapLDS to GapLDS$^{\text{col}}$ in a straightforwad manner, we will end up with a Turing reduction, i.e., we will produce multiple instances of GapLDS$^{\text{col}}$. Our observation here is that these instances can be "merged" into a single instance, i.e., by shifting the vectors appropriately so that the coordinates of vectors from different instances are not overlap:

Lemma 4.8. *There exists an algorithm reduction that takes in $\mathcal{W} \subseteq \mathbb{F}_q^m$ and an integer k, runs in $2^{O(k)}$poly$(m, |\mathcal{W}|)$ time, and outputs $\mathcal{W}' \subseteq \mathbb{F}_q^{m'}$ and a coloring $c : \mathcal{W}' \to [k]$ such that*

- *(YES) if $(\mathcal{W}, k)$ is a YES instace of* GAPLDS$_\gamma$*, then $(\mathcal{W}', k, c)$ is a YES instace of* GAPLDS$_\gamma^{col}$*;*
- *(NO) if $(\mathcal{W}, k)$ is a NO instace of* GAPLDS$_\gamma$*, then $(\mathcal{W}', k, c)$ is a NO instace of* GAPLDS$_\gamma^{col}$*;*

Proof. Let $(\mathcal{W}, k)$ be any instance of GAPLDS$_\gamma$, and let n denote $|\mathcal{W}|$. We use Theorem 4.7 to construct an (n, k)-perfect hash family $\mathcal{F} = \{f_1, \dots, f_R\}$ where $R = 2^{O(k)} poly(n)$. For every $\mathbf{w} \in \mathcal{W}$ and $j \in [R]$, we add a vector $\mathbf{0}_{m(j-1)} \circ \mathbf{w} \circ \mathbf{0}_{m(R-j)} \in \mathbb{F}_2^{mR}$ to $\mathcal{W}'$ and color this vector by $f_j(\mathbf{w})$. Finally, k remains the same as before.

It is obvious that the reduction runs in $2^{O(k)} poly(n)$ time. We now argue its correctness.

(YES Case) Suppose that $(\mathcal{W}, k)$ is a YES instance of GAPLDS$_\gamma$, i.e., there exist $a_1, \dots, a_k \in \mathbb{F}_q \setminus \{0\}$ such that $a_1\mathbf{w}_1 + \cdots + a_k\mathbf{w}_k = 0$. Since $\mathcal{F}$ is a perfect hash family, there exists $j \in [R]$ such that $f_j(\{\mathbf{w}_1, \dots, \mathbf{w}_k\}) = [k]$. In this case, we have $\sum_{i \in [k]} a_i(\mathbf{0}_{m(j-1)} \circ \mathbf{w}_i \circ \mathbf{0}_{m(R-j)}) = \mathbf{0}$ and that the vectors $\mathbf{0}_{m(j-1)} \circ \mathbf{w}_1 \circ \mathbf{0}_{m(R-j)}, \dots, \mathbf{0}_{m(j-1)} \circ \mathbf{w}_k \circ \mathbf{0}_{m(R-j)}$ are of different colors. Hence, $(\mathcal{W}', k, c)$ is a YES instance of GAPLDS$_\gamma^{\text{col}}$.

(NO Case) Suppose that $(\mathcal{W}, k)$ is a NO instance of GAPLDS$_\gamma$. Consider any $W' \subseteq \mathcal{W}'$ such that the vectors in W' are linearly dependent; we may pick such a set that is minimum, i.e., for every $\mathbf{w}' \in W'$, there exists a coefficient $a_{\mathbf{w}'}$ so that $\sum_{\mathbf{w}' \in W'} a_{\mathbf{w}'}\mathbf{w}' = \mathbf{0}$.

Consider any element of W'; suppose that it is of the form $\mathbf{0}_{m(j-1)} \circ \mathbf{w}^* \circ \mathbf{0}_{m(R-j)}$ for some $j \in [R]$. Let W be $\{\mathbf{w} \in \mathcal{W} : \mathbf{0}_{m(j-1)} \circ \mathbf{w} \circ \mathbf{0}_{m(R-j)} \in W'\}$. By restricting the equation $\sum_{\mathbf{w}' \in W'} a_{\mathbf{w}'}\mathbf{w}' = \mathbf{0}$ only to the coordinates $m(j-1)+1, \dots, mj$, we can conclude that the vectors in W are linearly dependent. Hence, we must have $|W'| \geqslant |W| > \gamma k$; that is, $(\mathcal{W}, k, c)$ is a NO instance of GAPLDS$_\gamma^{\text{col}}$ as desired. □

Combining Theorem 4.5 and Lemma 4.8, we can get the following theorem, which implies the W[1]-hardness of GAPLDS$_\gamma^{\text{col}}$.

Theorem 4.9. *Let $\gamma \geqslant 1$ be any constant. There is a polynomial time algorithm that, given an instance (G, s, h) of* GAPBSMD$_\gamma$ *where G contains n vertices and any prime power $q > n$, produces an instance $(\mathcal{W} \subseteq \mathbb{F}_q^m, k = hs, c)$ of* GAPLDS$_\gamma$ *such that*

- *(YES) If (G, s, h) is a YES instance of* GAPBSMD$_\gamma$*, then $(\mathcal{W}, k, c)$ is a YES instance of* GAPLDS$_\gamma^{col}$*.*
- *(NO) If (G, s, h) is a NO instance of* GAPBSMD$_\gamma$*, then $(\mathcal{W}, k, c)$ is a NO instance of* GAPLDS$_\gamma^{col}$*.*

5 Parameterized Inapproximability of Maximum Likelihood Decoding

In this section, we will show the parameterized intractability of GAPMLD as stated below.

Theorem 5.1. *For every $\gamma \geqslant 1$ and any prime number p,* GAPMLD$_{\gamma,p}$ *is W[1]-hard.*

We will divide the section into two parts. In the first part, we will give a simpler proof that only yields a hardness of approximation with factor $3 - \varepsilon$ for any $\varepsilon > 0$, and we only focus on the case $p = 2$ for simplicity. We note that this already suffices for proving hardness for Even Set problem. (In fact, any inapproximability result with factor greater than two suffices; see Lemma 6.5.)

Next, in the second part, we add an additional step in the proof that allows us to prove hardness of approximation with any constant factor and every prime field. We note here that, while this additional step is not used in proving hardness of Even Set, the technique not only gives the better inapproximability factor for GAPMLD but is also crucial in proving hardness of the Shortest Vector Problem (see Appendix 7.1).

5.1 $(3 - \varepsilon)$ Factor Inapproximability of Maximum Likelihood Decoding

In this subsection, we will show the inapproximability of MLD over $\mathbb{F}_2$ for any constant factor less than three. More formally, we show the following:

Theorem 5.2. *For any constant $\varepsilon > 0$, $\textsc{GapMLD}_{3-\varepsilon}$ is W[1]-hard.*

Proof. We will reduce from $\textsc{GapBSMD}_3$, which is W[1]-hard due to Theorem 4.2. Let $(G = (L \dot\cup R, E), s, h)$ be an instance of $\textsc{GapBSMD}_3$. We first run the reduction in Theorem 4.9 with $q = 2^{\lceil \log(|L|+|R|) \rceil}$. This gives us an instance $(\mathcal{W} \subseteq \mathbb{F}_{2^d}^m, k, c)$ of $\textsc{GapLDS}_3^{\text{col}}$. We use n to denote $|\mathcal{W}|$.

We now describe how we construct the instance $(\mathbf{A} \in \mathbb{F}_2^{m' \times n'}, \mathbf{y} \in \mathbb{F}_2^{m'}, k)$ of $\textsc{GapMLD}_{3-\varepsilon}$ where $m' = md + k$ and $n' = (2^d - 1)n$. First, the parameter k remains the same from the $\textsc{GapBSMD}_3^{\text{col}}$. Second, $\mathbf{y}$ is the m'-dimensional vector whose first k coordinates are ones and the remaining coordinates are zeros, i.e., $\mathbf{y} = \mathbf{1}_k \circ \mathbf{0}_{md}$.

To define $\mathbf{A}$, we need to introduce some notation. First, recall that the elements of the field $\mathbb{F}_{2^d}$ can be viewed as d-dimensional $\mathbb{F}_2$-vectors. In other words, there is a map $f : \mathbb{F}_{2^d} \to \mathbb{F}_2^d$ such that $f(x + y) = f(x) + f(y)$ for all $x, y \in \mathbb{F}_{2^d}$, and $f(x) = \mathbf{0}_d$ iff $x = 0$. We additionally define $F : \mathbb{F}_{2^d}^m \to \mathbb{F}_2^{md}$ by $F(\mathbf{v}) = f(\mathbf{v}[1]) \circ \cdots \circ f(\mathbf{v}[m])$. Again, we have $F(\mathbf{u} + \mathbf{v}) = F(\mathbf{u}) + F(\mathbf{v})$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{F}_{2^d}^m$, and $F(\mathbf{v}) = \mathbf{0}_{md}$ iff $\mathbf{v} = \mathbf{0}_m$.

Moreover, for every $i \in [k]$, let $\mathbf{e}_i$ be the k-dimensional vector with one at the i-th coordinate and zero elsewhere. We identify the column indices of $\mathbf{A}$ by $\mathcal{W} \times (\mathbb{F}_{2^d} \setminus \{0\})$. Then, we construct $\mathbf{A}$ by letting its $(\mathbf{w}, a)$-column be

$$\mathbf{A}[(\mathbf{w}, a)] := \mathbf{e}_{c(\mathbf{w})} \circ F(a \cdot \mathbf{w}).$$

This completes our reduction description. It is simple to verify that the reduction runs in polynomial time. We now move on to prove the correctness of the reduction.

(YES Case) Suppose that (G, s, h) is a YES instance of $\textsc{GapBSMD}_3$. From Theorem 4.9, there exist $\mathbf{w}_1, \dots, \mathbf{w}_k \in \mathcal{W}$ all of different colors and non-zero $a_1, \dots, a_k \in \mathbb{F}_{2^d} \setminus \{0\}$ such that $\sum_{i \in [k]} a_i \cdot \mathbf{w}_i = \mathbf{0}$. Let $\mathbf{x} \in \mathbb{F}_2^{n'}$ such that $\mathbf{x}[(\mathbf{w}_i, a_i)] = 1$ for all $i \in [k]$ and all other coordinates of $\mathbf{x}$ are zero. Clearly, $\|\mathbf{x}\|_0 = k$ and

$$\mathbf{A}\mathbf{x} = \sum_{i \in [k]} \mathbf{A}[(\mathbf{w}_i, a_i)] = \sum_{i \in [k]} \mathbf{e}_{c(\mathbf{w})} \circ F(a_i \cdot \mathbf{w}_i) = \mathbf{1}_k \circ F\left(\sum_{i \in [k]} a_i \cdot \mathbf{w}_i\right) = \mathbf{1}_k \circ F(\mathbf{0}) = \mathbf{1}_k \circ \mathbf{0}_{md} = \mathbf{y},$$

which means that $(\mathbf{A}, \mathbf{y}, k)$ is indeed a YES instance.

(NO Case) Suppose that (G, s, h) is a NO instance of GAPBSMD$_3$. From Theorem 4.9, $(\mathcal{W}, k, c)$ is a NO instance of GAPLDS$_3^{\text{col}}$. Suppose for the sake of contradiction that $(\mathbf{A}, \mathbf{y}, k)$ is not a NO instance of GAPMLD$_{3-\varepsilon}$. That is, there exists $\mathbf{x} \in \mathbb{F}_2^{n'}$ such that $\mathbf{A}\mathbf{x} = \mathbf{y}$ and $\|\mathbf{x}\|_0 \leqslant (3-\varepsilon)k < 3k$.

For every $i \in [k]$, let us define X_i as

$$X_i := \{(\mathbf{w}, a) \in \mathcal{W} \times (\mathbb{F}_{2^d} \setminus \{0\}) : \mathbf{x}[(\mathbf{w}, a)] = 1\}.$$

We can write $\mathbf{A}\mathbf{x}$ as

$$\mathbf{A}\mathbf{x} = \sum_{i \in [k]} \sum_{(\mathbf{w}, a) \in X_i} \mathbf{e}_i \circ F(a \cdot \mathbf{w}) = \left(\sum_{i \in [k]} |X_i| \mathbf{e}_i\right) \circ F\left(\sum_{i \in [k]} \sum_{(\mathbf{w}, a) \in X_i} a \cdot \mathbf{w}\right).$$

Since $\mathbf{A}\mathbf{x} = \mathbf{y}$, we must have $|X_i| \equiv 1 \pmod 2$ for all $i \in [k]$ and

$$\sum_{i \in [k]} \sum_{(\mathbf{w}, a) \in X_i} a \cdot \mathbf{w} = \mathbf{0}_m. \tag{4}$$

Moreover, observe that $\|\mathbf{x}\|_0 = \sum_{i \in [k]} |X_i|$. Since $\|\mathbf{x}\|_0 < 3k$ and $|X_i| \equiv 1 \pmod 2$ for all $i \in [k]$, there must be $i^* \in [k]$ such that $|X_{i^*}| = 1$. Let $(\mathbf{w}^*, a^*)$ be the unique element of X_{i^*}. Notice that $\mathbf{w}^*$ appears only once in the left hand side of (4) with coefficient $a^* \neq 0$; as a result, this is a non-empty linear combination of less than $3k$ vectors in $\mathcal{W}$. Hence, there are less than $3k$ vectors in $\mathcal{W}$ that are linearly dependent, which contradicts the fact that $(\mathcal{W}, k, c)$ is a NO instance of GAPLDS$_3^{\text{col}}$.

Thus, $(\mathcal{W}, k, c)$ must be a NO instance of GAPMLD$_{3-\varepsilon}$ as desired. □

5.2 Every Constant Factor Inapproximability of Maximum Likelihood Decoding

In this section, we will prove our main result of this section, i.e., Theorem 5.1.

To demonstrate the main additional idea, let us recall why the proof in the previous section fails to give us the hardness of factor three. The reason is as follows: when $d > 2$, we can pick three non-zero elements $a, b, c \in \mathbb{F}_{2^d}$ whose sum is zero. We can then select any $\mathbf{w}_1, \dots, \mathbf{w}_k$ of different colors, and set $\mathbf{x}[(\mathbf{w}_i, a)], \mathbf{x}[(\mathbf{w}_i, b)], \mathbf{x}[(\mathbf{w}_i, c)]$ to be ones for all $i \in [k]$, and set the rest of coordinates of $\mathbf{x}$ to be zero. Clearly, $\|\mathbf{x}\|_0 = 3k$ and this gives

$$\mathbf{A}\mathbf{x} = \mathbf{1}_k \circ F\left(\sum_{i \in [k]} (a + b + c)\mathbf{w}_i\right) = \mathbf{1}_k \circ \mathbf{0}_{md} = \mathbf{y}.$$

That is, the fact that $a + b + c = 0$ allows us to zero out the coefficient of each $\mathbf{w}_i$. Our fix to overcome this issue is rather straightforward. First, observe that we can write $\mathbb{F}_2^d \setminus \{0\} = C_1 \cup \cdots \cup C_d$ such that no such "problematic" tuples (a, b, c) appears in C_i, where C_i is defined as $\{a \in \mathbb{F}_q \setminus \{0\} : f(a)[i] = 1\}$ (where f is as defined in Theorem 5.2). In fact, this guarantees not only that any triplet in C_i sums to non-zero, but also that any odd number of elements in

C_i sums to non-zero.

Now, the modification is very simple: instead of creating columns for $(\mathbf{w}, a)$ for all $a \in \mathbb{F}_{2^d} \setminus \{0\}$, we will only create columns for $(\mathbf{w}, a)$ for $a \in C_{g[c(\mathbf{w})]}$ where $g \in [d]^k$, i.e., we restrict the coefficients to only $C_{g[j]}$ for each color j. This helps us avoid "problematic" coefficients as described above. In particular, we can construct a instance $\mathbf{A}_g$ for every choice of c. As in the reduction to GAPLDS$^{\text{col}}$, we can merge the various instances corresponding to different choices of g into a single instance using the shifting trick employed in the proof of Lemma 4.8.

For a general prime p, we can write $\mathbb{F}_{p^d}$ similarly as above into a union of subsets, such that each subset does not contain "problematic" tuples of elements, as stated below. Note that the definition of "problematic" is slightly more complicated for general p. Now, the tuple $(a_1, \ldots, a_t) \in \mathbb{F}_{p^d}^t$ is "problematic" if we can find $b_{a_1}, \ldots, b_{a_t} \in \mathbb{F}_p$ such that $b_{a_1} + \cdots + b_{a_t} \neq 0$ (over $\mathbb{F}_p$) but $b_{a_1} \cdot a_1 + \cdots b_{a_t} \cdot a_t = 0$ (over $\mathbb{F}_{p^d}$).

Definition 5.3. *For $q = p^d$ where $d \in \mathbb{N}$ and p is a prime, let $f : \mathbb{F}_q \to \mathbb{F}_p^d$ be the isomorphism between $\mathbb{F}_q^+$ and the $\mathbb{F}_p$-vector space $\mathbb{F}_p^d$. For every $i \in [d]$ and $\alpha \in \mathbb{F}_p \setminus \{0\}$, we define $C_{(i,\alpha)} := \{a \in \mathbb{F}_q \setminus \{0\} : f(a)[i] = 1\}$. Observe that*

(i) $\mathbb{F}_q \setminus \{0\} = \bigcup_{i \in [d], \alpha \in \mathbb{F}_p \setminus \{0\}} C_{(i,\alpha)}$

(ii) for any $i \in [d], \alpha \in \mathbb{F}_p \setminus \{0\}$ and any $(b_a)_{a \in C_{(i,\alpha)}} \in (\mathbb{F}_p)^{C_{(i,\alpha)}}$ such that $\sum_{a \in C_{(i,\alpha)}} b_a \neq 0$, we have $\sum_{a \in C_{(i,\alpha)}} b_a \cdot a \neq 0$.

With this definition, we can easily generalize the (sketched) reduction from $\mathbb{F}_2$ to $\mathbb{F}_p$. The properties of the reduction are summarized and proved below.

Theorem 5.4. *Given an instance $(\mathcal{W} \subseteq \mathbb{F}_{p^d}^m, k, c)$ of* GAPLDS$_\gamma$ *where p is a prime, we can create an instance of* GAPMLD$_{\gamma,p}$ *(with the same parameter k) in $O((dp)^k \cdot poly(|\mathcal{W}|, m, p^d))$ time such that*

- *(YES) If $(\mathcal{W}, k, c)$ is a YES instance of* GAPLDS$_\gamma$*, then the* GAPMLD$_{\gamma,p}$ *is a YES instance.*
- *(NO) If $(\mathcal{W}, k, c)$ is a NO instance of* GAPLDS$_\gamma$*, then the* GAPMLD$_{\gamma,p}$ *is a NO instance.*

Proof. Let $(\mathcal{W} \subseteq \mathbb{F}_{p^d}^m, k, c)$ be an instance of GAPLDS$_\gamma$. Let $n = |\mathcal{W}|$, and $f : \mathbb{F}_q \to \mathbb{F}_p^d$ be the isomorphism between $\mathbb{F}_q^+$ and the $\mathbb{F}_p$-vector space $\mathbb{F}_p^d$. Furthermore, let $F : \mathbb{F}_{p^d}^m \to \mathbb{F}_p^{md}$ be defined by $F(\mathbf{v}) = f(\mathbf{v}[1]) \circ \cdots \circ f(\mathbf{v}[m])$. We will also find it convenient to define $\ell = d^k(p-1)^k$, which is the total number of distinct choices of g.

For every $g \in [\ell]$, we construct a matrix $\mathbf{A}_g \in \mathbb{F}_2^{m' \times md}$ where $m' = k + \ell md$. As before, we index the columns of $\mathbf{A}_g$ with the set $I_g = \cup_{\mathbf{w} \in \mathcal{W}} \{\mathbf{w}\} \times C_{g[c(\mathbf{w})]}$. Here, for any $[\mathbf{w}, a] \in I_g$, we let the corresponding column be

$$\mathbf{A}_g[\mathbf{w}, a] := \mathbf{e}_{c^{-1}(\mathbf{w})} \circ \Big(\mathbf{0}_{md(i-1)} \circ F(a \cdot \mathbf{w}) \circ \mathbf{0}_{md(\ell - i)}\Big) \tag{5}$$

Finally, we define the matrix $\mathbf{A} = [\mathbf{A}_g]_{g \in [\ell]} \in \mathbb{F}_p^{m' \times n'}$ to be the concatenation of all the $\mathbf{A}_g$ matrices, where $n' = \ell n$. Note that the above construction ensures that for any distinct pair of $g, g' \in [\ell]$, the column supports of the sub-matrices $\mathbf{A}_g$ and $\mathbf{A}_{g'}$ do not intersect in the coordinates $[k+1, n']$. We also define the target vector $\mathbf{y} = \mathbf{1}_k \circ \mathbf{0}_{md\ell}$. We set $(\mathbf{A}, \mathbf{y}, k)$ to be the GAPMLD$_{\gamma,p}$ instance output by the reduction. Clearly, the reduction runs in $O((dp)^k \cdot poly(n, m, p^d))$ time. We next argue its correctness.

(YES Case) Suppose there exist $\mathbf{w}_1, \dots, \mathbf{w}_k \in \mathcal{W}$ all of different colors and non-zero $a_1, \dots, a_k \in \mathbb{F}_{p^d} \setminus \{0\}$ such that $\sum_{i \in [k]} a_i \cdot \mathbf{w}_i = \mathbf{0}$. We claim that $(\mathbf{A}, \mathbf{y}, k)$ is a YES instance of GAPMLD$_{\gamma,p}$. To see this, consider g^* where a_i belongs to $C_{g^*[i]}$ for all $i \in [k]$. . Also let $\mathbf{x} \in \mathbb{F}_p^{n'}$ be such that $\mathbf{x}_{g^*}[(\mathbf{w}_i, a_i)] = 1$ for all $i \in [k]$, where $\mathbf{x}_{g^*}$ is the vector $\mathbf{x}$ restricted to coordinates corresponding to the sub-matrix $\mathbf{A}_{g^*}$. We set all other coordinates of $\mathbf{x}$ to zero. (Note that the column $(\mathbf{w}_i, a_i)$ exists in $\mathbf{A}_{g^*}$ because $a_i \in C_{g^*[i]}$.) Clearly, $\mathbf{x}$ is a $\{0,1\}$-vector with $\|\mathbf{x}\|_0 = k$ and

$$\mathbf{A}\mathbf{x} = \mathbf{A}_{g^*}\mathbf{x}_{g^*} = \sum_{i \in [k]} \mathbf{A}_{g^*}[(\mathbf{w}_i, a_i)] = \sum_{i \in [k]} \mathbf{e}_{c(\mathbf{w}_i)} \circ F\left(a_i \cdot \mathbf{w}_i\right) = \mathbf{1}_k \circ F\left(\sum_{i \in [k]} a_i \cdot \mathbf{w}_i\right) = \mathbf{1}_k \circ F(\mathbf{0}) = \mathbf{y},$$

which means that $(\mathbf{A}, \mathbf{y}, k)$ is indeed a YES instance of GAPMLD$_{\gamma,p}$.

(NO Case) Suppose that $(\mathcal{W}, k, c)$ is a NO instance of GAPLDS$_\gamma$. Consider any $\mathbf{x} \in \mathbb{F}_p^{n'}$ such that $\mathbf{A}\mathbf{x} = \mathbf{y}$. Recall that for any $g \in [\ell]$, $\mathbf{x}_g$ is the sub-vector of $\mathbf{x}$ which acts on the sub-matrix $\mathbf{A}_g$. Let us rewrite $\mathbf{A}\mathbf{x}$ as follows:

$$\mathbf{A}\mathbf{x} = \left(\sum_{i \in [k]} \left(\sum_{g \in [\ell]} \sum_{\mathbf{w} \in c^{-1}(i), a \in C_{g[i]}} \mathbf{x}_g[(\mathbf{w}, a)]\right) \mathbf{e}_i\right) \circ \mathbf{v}_1 \circ \mathbf{v}_2 \circ \cdots \circ \mathbf{v}_\ell$$

where for any $g \in [\ell]$, the vector $\mathbf{v}_g$ is the sub-vector of $\mathbf{A}_g\mathbf{x}_g$ which can be formally expressed as

$$\mathbf{v}_g = F\left(\sum_{\mathbf{w} \in \mathcal{W}} \left(\sum_{a \in C_{g[c(\mathbf{w})]}} \mathbf{x}_g[(\mathbf{w}, a)] \cdot a\right) \cdot \mathbf{w}\right)$$

In other words, it is the block resulting from $\mathbf{A}_g\mathbf{x}_g$ in the coordinates $k+1, k+2, \dots, m'$. Since $\mathbf{A}\mathbf{x} = \mathbf{y}$, we must have

$$\sum_{g \in [\ell]} \sum_{\mathbf{w} \in c^{-1}(i), a \in C_{g[i]}} \mathbf{x}_g[(\mathbf{w}, a)] = 1 \qquad \forall i \in [d] \tag{6}$$

and for every $g \in [\ell]$,

$$\sum_{\mathbf{w} \in \mathcal{W}} \left(\sum_{a \in C_{g[c(\mathbf{w})]}} \mathbf{x}_g[(\mathbf{w}, a)] \cdot a\right) \cdot \mathbf{w} = \mathbf{0}_m. \tag{7}$$

From (6) with $i = 1$, there must be $g^* \in [\ell]$ such that

$$\sum_{\mathbf{w} \in c^{-1}(i), a \in C_{g^*[1]}} \mathbf{x}_{g^*}[(\mathbf{w}, a)] \neq 0$$

which in turn implies that there exists $\mathbf{w}^*$ of color 1 such that $\sum_{a \in C_{g^*[1]}} \mathbf{x}_{g^*}[(\mathbf{w}^*, a)] \neq 0$. From this and observation (ii) in Definition 5.3, we have $\sum_{a \in C_{g^*[1]}} \mathbf{x}_{g^*}[(\mathbf{w}^*, a)] \cdot a \neq 0$. This means that the left hand side of (7) instantiated with g^* is a non-zero linear combination of at most $\|\mathbf{x}_{g^*}\|_0$ vectors from $\mathcal{W}$. Since $(\mathcal{W}, k, c)$ is a NO instance of GAPLDS$_\gamma$, we can conclude that $\|\mathbf{x}_{g^*}\|_0$ (and consequently $\|\mathbf{x}\|_0$) must be larger than $\gamma \cdot k$. Hence, $(\mathbf{A}, \mathbf{y}, k)$ is a NO instance for GAPMLD$_{\gamma,p}$. □

Finally, we note that the above theorem together with Theorems 4.2 and 4.9 imply the main result of this section (Theorem 5.1). In particular, by selecting $d = \lceil \log_p(|L| + |R|) \rceil$ in Theorem 4.9 and applying Theorem 5.4 afterwards, we get a (Turing) reduction from GAPBSMD$_\gamma$ to GAPMLD$_{\gamma,p}$ that runs in time

$$O((pd)^k \cdot \mathrm{poly}(|L|+|R|)) \leqslant O\left(\left((pd)^{\sqrt{pd}} + (k^2)^k\right) \cdot \mathrm{poly}(|L|+|R|)\right) = k^{O(k)} \cdot \mathrm{poly}(|L|+|R|),$$

which is FPT. From this and from W[1]-hardness of GAPBSMD$_\gamma$ (Theorem 4.2), we arrive at Theorem 5.1.

6 Parameterized Intractability of Minimum Distance Problem

Next, we will prove our main theorem regarding parameterized intractability of GAPMDP:

Theorem 6.1. GAPMDP$_\gamma$ *for any* $\gamma \geqslant 1$ *is W[1]-hard under randomized reductions.*

This again proceeds in two steps. First, we give a simple reduction from GAPMLD to GAPSNC in Section 6.1. Then, we reduce the latter to GAPMDP in Section 6.2.

6.1 Parameterized Inapproximability of Sparse Nearest Codeword Problem

We start with a simple approximation-preserving reduction from GAPMLD to GAPSNC.

Theorem 6.2. GAPSNC$_\gamma$ *for any* $\gamma \geqslant 1$ *is W[1]-hard under randomized reductions.*

Proof. We reduce from GAPMLD$_\gamma$, which is W[1]-hard from Theorem 5.1. Let $(\mathbf{B}, \mathbf{z}, k)$ be the input for GAPMLD$_\gamma$ where $\mathbf{B} \in \mathbb{F}_2^{n \times m}$, $\mathbf{y} \in \mathbb{F}_2^n$, and t is the parameter. Let $a = \lceil \gamma k + 1 \rceil$. We produce an instance $(\mathbf{A}, \mathbf{y}, k)$ for GAPSNC$_\gamma$ by letting

$$\mathbf{A} = \left.\begin{bmatrix} \mathbf{B} \\ \vdots \\ \mathbf{B} \end{bmatrix}\right\} a \text{ copies}, \mathbf{y} = \left.\begin{bmatrix} \mathbf{z} \\ \vdots \\ \mathbf{z} \end{bmatrix}\right\} a \text{ copies}$$

The reduction clearly runs in polynomial time, we are only left to argue that it appropriately maps YES and NO cases from GAPMLD$_\gamma$ to those in GAPSNC$_\gamma$.

(YES Case) Suppose that $(\mathbf{B}, \mathbf{z}, k)$ is a YES instance of GAPMLD$_\gamma$, i.e., there exists $\mathbf{x} \in \mathcal{B}_q(\mathbf{0}, k)$ such that $\mathbf{Bx} = \mathbf{z}$. This implies that $\|\mathbf{Ax} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0 = \|\mathbf{x}\|_0 \leqslant k$ as desired.

(NO Case) Suppose that $(\mathbf{B}, \mathbf{z}, k)$ is a NO instance of GAPMLD$_\gamma$, i.e., for all $\mathbf{x} \in \mathcal{B}_m(\mathbf{0}, \gamma k)$, we have $\mathbf{Bx} \neq \mathbf{z}$. Now, let us consider two cases, based on whether $\mathbf{x} \in \mathcal{B}_m(\mathbf{0}, \gamma k)$. First, if

$\mathbf{x} \in \mathcal{B}(\mathbf{0}, \gamma k)$, then we have $\|\mathbf{Ax} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0 \geqslant a\|\mathbf{Bx} - \mathbf{z}\|_0 \geqslant a > \gamma k$. On the other hand, if $\mathbf{x} \notin \mathcal{B}_m(\mathbf{0}, \gamma k)$, then $\|\mathbf{Ax} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0 \geqslant \|\mathbf{x}\|_0 > \gamma k$.

Thus, in both cases, $\|\mathbf{Ax} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0 > \gamma k$ and $(\mathbf{A}, \mathbf{y}, k)$ is a NO instance of GAPSNC$_\gamma$. □

6.2 Reducing GAPSNC to GAPMDP

In order to reduce GAPSNC to GAPMDP, we need to formalize the definition of Locally Suffix Dense Codes (LSDC) and prove their existence; these are done in in Section 6.2.1. Finally, we show how to use them in the reduction in Section 6.2.2.

6.2.1 Locally Suffix Dense Codes

Before we formalize the notion of *Locally Suffix Dense Codes* (LSDC), let us give an intuitive explanation of LSDC: informally, LSDC is a linear code $\mathcal{C} \subseteq \mathbb{F}_2^h$ where, given any short prefix $\mathbf{x} \in \mathbb{F}_2^q$ where $q \ll h$ and a random suffix $\mathbf{s} \in \mathbb{F}_2^{h-q}$, we can, with non-negligible probability, find a codeword that shares the prefix $\mathbf{x}$ and has a suffix that is "close" in Hamming distance to $\mathbf{s}$ (i.e. one should think of r below as roughly $d/2$). More formally, LSDC can be defined as follows.

Definition 6.3. *A Locally Suffix Dense Code (LSDC) over $\mathbb{F}_2$ with parameters*[15] (m, q, d, r, δ) *an m-dimensional systematic linear code with minimum distance (at least) d given by its generator matrix $\mathbf{L} \in \mathbb{F}_2^{h \times m}$ such that for any $\mathbf{x} \in \mathbb{F}_2^q$, the following holds:*

$$\Pr_{\mathbf{s} \sim \mathbb{F}_2^{h-q}} \left[\exists \mathbf{z} \in \mathcal{B}_{h-q}(\mathbf{s}, r) : (\mathbf{x} \circ \mathbf{z}) \in \mathbf{L}(\mathbb{F}_2^m) \right] \geqslant \delta. \tag{8}$$

We note that our notion of Locally Suffix Dense Codes is closely related and inspired by the notion of Locally Dense Codes (LDC) of Dumer et al. [DMS03]. Essentially speaking, the key differences in the two definitions are that (i) Locally Dense Codes are for the case of $q = 0$, i.e., there is no prefix involved, and (ii) $\mathbf{s}$ in LDC is not chosen at random from $\mathbb{F}_2^q$ but rather from $\mathcal{B}_q(\mathbf{0}, r)$. Note that, apart from these, there are other subtle additional requirements in Locally Dense Codes that we do not need in our reduction, such as the requirements that the "center" $\mathbf{s}$ is close to not just one but many codewords; however, these are not important and we will not discuss them further.

Unfortunately, the proof of Dumer et al. does not directly give us the desired LSDC; the main issue is that, when there is no prefix, the set of codewords is a linear subspace, and their proof relies heavily on the linear structure of the set (which is also why $\mathbf{s}$ is randomly chosen from $\mathcal{B}_q(\mathbf{0}, r)$ instead of $\mathbb{F}_2^q$). However, the set of our interest is $\left\{\mathbf{z} \in \mathbb{F}^{h-q} \middle| \mathbf{x} \circ \mathbf{z} \in \mathbf{L}(\mathbb{F}_2^m)\right\}$, which is not a linear subspace but rather an affine subspace; Dumer et al.'s argument (specifically Lemma 13 in [DMS03]) does not apply in the affine subspace case.

[15] We remark that the parameter h is implicit in specifying LSDC.

Below, we provide a different proof than Dumer et al. for the construction of LSDC. Our bound is more related to the *Sphere Packing* (aka Hamming) bound for codes. In particular, we show below that BCH codes, which "near" the Sphere Packing bound gives us LSDC with certain parameters. It should be noted however that the probability guarantee δ that we have is quite poor, i.e. $\delta \geqslant d^{-\Theta(d)}$, but this works for us since d is bounded by a function of the parameter of our problem. On the other hand, this would not work in NP-hardness reductions of [DMS03] (and, on top of this, our codes may not satisfy other additional properties required in LDC).

Lemma 6.4. *For any $q, d \in \mathbb{N}$ such that d is an odd number larger than one, there exist $h, m \in \mathbb{N}$ and $\mathbf{L} \in \mathbb{F}_2^{h \times m}$ which is a LSDCwith parameters $\left(m, q, d, \frac{d-1}{2}, \frac{1}{d^{d/2}}\right)$. Additionally, the following holds:*

- *$h, m \leqslant poly(q, d)$ and $m \geqslant q$,*
- *$\mathbf{L}$ can be computed in $poly(q, d)$ time.*

Proof. Let h be the smallest integer such that $h + 1$ is a power of two and that $h \geqslant \max\{2q, 10d \log d\}$, and let $m = h - \left(\frac{d-1}{2}\right) \log(h + 1)$. Clearly, h and m satisfy the first condition.

Let $\mathbf{L}$ be the generator matrix of the $[h, m, d]_2$ linear code as given by Theorem 3.8. Without loss of generality, we assume that the code is systematic on the first m coordinates. From Theorem 3.8, $\mathbf{L}$ can be computed in $\text{poly}(h) = \text{poly}(q, d)$ time.

It remains to show that for our choice of $\mathbf{L}$, (8) holds for any fixed choice of $\mathbf{x} \in \mathbb{F}_2^q$. Fix a vector $\mathbf{x} \in \mathbb{F}_2^q$ and define the set $\mathcal{C} = \left\{\mathbf{z} \in \mathbb{F}_2^{h-q} \middle| \mathbf{x} \circ \mathbf{z} \in \mathbf{L}(\mathbb{F}_2^m)\right\}$. Since the code generated by $\mathbf{L}$ is systematic on the first $m \geqslant q$ coordinates, we have that $|\mathcal{C}| \geqslant 2^{m-q}$.

Moreover, since the code generated by $\mathbf{L}$ has distance d, every distinct $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{C}$ are at least d-far from each other (i.e. $\|\mathbf{z}_1 - \mathbf{z}_2\|_0 \geqslant d$). Therefore, for any distinct pair of vectors $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{C}$, the sets $\mathcal{B}_{h-q}(\mathbf{z}_1, \frac{d-1}{2})$ and $\mathcal{B}_{h-q}(\mathbf{z}_2, \frac{d-1}{2})$ are disjoint. Hence the number of vectors in the union of $\left(\frac{d-1}{2}\right)$-radius Hamming balls around every $\mathbf{z} \in \mathcal{C}$ is at least

$$2^{m-q} \left|\mathcal{B}_{h-q}\left(\mathbf{0}, \frac{d-1}{2}\right)\right| \geqslant 2^{m-q} \binom{h-q}{\frac{d-1}{2}} \geqslant 2^{m-q} \binom{h/2}{\frac{d-1}{2}} \geqslant 2^{m-q} \left(\frac{h}{d-1}\right)^{\frac{d-1}{2}}$$

On the other hand, $|\mathbb{F}_2^{h-q}| = 2^{h-q} = 2^{m-q}(h+1)^{\frac{d-1}{2}}$. Hence, with probability at least $\left(\frac{h}{(d-1)(h+1)}\right)^{\frac{d-1}{2}} \geqslant \frac{1}{d^{d/2}}$, a vector $\mathbf{s}$ sampled uniformly from $\mathbb{F}_2^{h-q}$ lies in $\mathcal{B}_{h-q}\left(\mathbf{z}, \frac{d-1}{2}\right)$ for some vector $\mathbf{z} \in \mathcal{C}$. This is indeed the desired condition in (8), which completes our proof. □

6.2.2 The Reduction

In this subsection, we state and prove the FPT reduction from the GAPSNC problem to the GAPMDP problem. It is inspired by the reduction from [DMS03], which is then modified (and simplified) to work in combination with LSDC instead of LDC.

Lemma 6.5. *There is a randomized FPT reduction from* GAPSNC$_{2.5}$ *to* GAPMDP$_{1.01}$.

Proof. Let $(\mathbf{B}, \mathbf{y}, t)$ be the input for $\textsc{GapSNC}_{\gamma'}$ where $\mathbf{B} \in \mathbb{F}_2^{n \times q}$, $\mathbf{y} \in \mathbb{F}_2^n$, and t is the parameter. We may assume without loss of generality that $t \geqslant 1000$. Let d be the smallest odd integer greater than $2.5t$. Let $h, m \in \mathbb{N}, \mathbf{L} \in \mathbb{F}_2^{h \times m}$ be as in Lemma 6.4.

We produce an instance $(\mathbf{A}, k)$ for $\textsc{GapMDP}_\gamma$ by first sampling a random $\mathbf{s} \sim \mathbb{F}_2^{h-q}$. Then, we set $k = t + (d-1)/2$, $\mathbf{s}' = \mathbf{0}_q \circ -\mathbf{s}$ and

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} & \mathbf{0}_{n \times (m-q)} & \mathbf{y} \\ \mathbf{L} & & \mathbf{s}' \end{bmatrix} \in \mathbb{F}_2^{(n+h) \times (m+1)}.$$

Notice that the zeros are padded onto the right of $\mathbf{B}$ so that the number of rows is the same as that of $\mathbf{L}$.

Since $k = t + (d-1)/2 = O_{\gamma'}(t)$ and the reduction clearly runs in polynomial time, we are only left to argue that it appropriately maps YES and NO cases from $\textsc{GapSNC}_{\gamma'}$ to those in $\textsc{GapMDP}_\gamma$.

(YES Case) Suppose that $(\mathbf{B}, \mathbf{y}, t)$ is a YES instance of $\textsc{GapSNC}_{\gamma'}$, i.e., there exists $\mathbf{x} \in \mathbb{F}_2^q$ such that $\|\mathbf{Bx} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0 \leqslant t$. From Lemma 6.4, with probability at least $1/d^{d/2}$, there exists $\mathbf{u} \in \mathcal{B}_{h-q}\left(\mathbf{s}, \frac{d-1}{2}\right)$ such that $\mathbf{x} \circ \mathbf{u} \in \mathbf{L}(\mathbb{F}_2^m)$. From this and from systematicity of $\mathbf{L}$, there exists $\mathbf{z}' \in \mathbb{F}_2^{m-q}$ such that $\mathbf{L}(\mathbf{x} \circ \mathbf{z}') = \mathbf{x} \circ \mathbf{u}$. Conditioned on this, we can pick $\mathbf{z} = \mathbf{x} \circ \mathbf{z}' \circ 1 \in \mathbb{F}_2^{m+1}$, which yields

$$\|\mathbf{Az}\|_0 = \|\mathbf{Bx} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0 + \|\mathbf{u} - \mathbf{s}\|_0 \leqslant t + \frac{d-1}{2} = k.$$

In other words, with probability at least $1/d^{d/2}$, $(\mathbf{A}, k)$ is a YES instance of $\textsc{GapMDP}_\gamma$ as desired.

(NO Case) Suppose that $(\mathbf{B}, \mathbf{y}, t)$ is a NO instance of $\textsc{GapSNC}_{\gamma'}$. We will show that, for all non-zero $\mathbf{z} \in \mathbb{F}_2^{m+1}$, $\|\mathbf{Az}\|_0 > 2.5t$; with our choice of parameters and our assumption on t, it is simple to check that $2.5t > 1.01k$. Hence, this implies that $(\mathbf{A}, k)$ is a NO instance of $\textsc{GapMDP}_\gamma$.

To show that $\|\mathbf{Az}\|_0 > \gamma' t$ for all $\mathbf{z} \in \mathbb{F}_2^{m+1} \setminus \{\mathbf{0}\}$, let us consider two cases, based on the last coordinate $\mathbf{z}[m+1]$ of $\mathbf{z}$. For convenience, we write $\mathbf{z}$ as $\mathbf{x} \circ \mathbf{z}' \circ \mathbf{z}[m+1]$, where $\mathbf{x} \in \mathbb{F}_2^q$ and $\mathbf{z}' \in \mathbb{F}_2^{m-q}$.

If $\mathbf{z}[m+1] = 0$, then $\|\mathbf{Az}\|_0 = \|\mathbf{Bx}\|_0 + \|\mathbf{L}(\mathbf{x} \circ \mathbf{z}')\|_0 \geqslant \|\mathbf{L}(\mathbf{x} \circ \mathbf{z}')\|_0 \geqslant d$, where the last inequality comes from the fact that $\mathbf{L}$ is a generator matrix of a code of distance d (and that $\mathbf{z} \neq \mathbf{0}$). Finally, recall that we select $d > 2.5t$, which yields the desired result for this case.

On the other hand, if $\mathbf{z}_{m+1} = 1$, then $\|\mathbf{Az}\|_0 \geqslant \|\mathbf{Bx} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0 \geqslant 2.5t$, where the second inequality comes from the assumption that $(\mathbf{B}, \mathbf{y}, t)$ is a NO instance of $\textsc{GapSNC}_{2.5}$.

In conclusion, $\|\mathbf{Az}\|_0 > 2.5t$ in all cases considered, which completes our proof. □

Gap Amplification. Finally, the above gap hardness result can be boosted to any constant gap using the now standard technique of tensoring the code (c.f. [DMS03],[AK14]) which is stated formally in the following proposition:

Proposition 6.6 (E.g. [DMS03]). *Given two linear codes $C_1 \subseteq \mathbb{F}_2^m$ and $C_2 \subseteq \mathbb{F}_2^n$, let $C_1 \otimes C_2 \subseteq \mathbb{F}^{m \times n}$ be the tensor product of C_1 and C_2. Then $d(C_1 \otimes C_2) = d(C_1)d(C_2)$.*

We briefly show how the above proposition can be used to amplify the gap. Consider a GapMDP$_\gamma$ instance $(\mathbf{A}, k)$ where $\mathbf{A} \in \mathbb{F}_2^{m \times n}$. Let $C \subseteq \mathbb{F}_2^m$ be the linear code generated by it. Let $C^{\otimes 2} = C \otimes C$ be the tensor product of the code with itself, and let $\mathbf{A}^{\otimes 2}$ be its generator matrix. By the above proposition, if $(\mathbf{A}, k)$ is a YES instance, then $d(C^{\otimes 2}) \leqslant k^2$. Conversely, if $(\mathbf{A}, k)$ is a NO instance, then $d(C^{\otimes 2}) \geqslant \gamma^2 k^2$. Therefore $(\mathbf{A}^{\otimes 2}, k^2)$ is a GapMDP$_{\gamma^2}$ instance. Hence, for any $\alpha \in \mathbb{R}_+$, repeating this argument $\lceil \log_\gamma \alpha \rceil$-number of times gives us an FPT reduction from k-GapMDP$_\gamma$ to $k^{2\lceil \log_\gamma \alpha \rceil}$-GapMDP$_\alpha$. We have thereby completed our proof of Theorem 6.1.

7 Parameterized Intractability of Shortest Vector Problem

The main result of this section is the parameterized inapproximability of GapSVP, as stated below.

Theorem 7.1 (FPT Inapproximability of GapSVP). *For any $p > 1$, there exists constant $\gamma_p > 1$ (where γ_p depends on p), such that there GapSVP$_{p,\gamma_p}$ is W[1]-hard (under randomized reductions).*

Similar to the Minimum Distance Problem, the proof of Theorem 7.1 goes through two steps. First, we show that the non-homogeneous variant, the Nearest Vector Problem. Then, in the second step, we reduce it to the Shortest Vector Problem.

7.1 FPT Inapproximability of Nearest Vector Problem

In this section, we prove the inapproximability of Nearest Vector Problem, as stated more formally below. The proof is via a simple reduction from Maximum Likelihood Decoding over a large field.

Theorem 7.2 (FPT Inapproximability of GapNVP). *For any $\eta, p \geqslant 1$, GapNVP$_{\eta,p}$ is W[1]-hard.*

Proof. Let q be the smallest prime number such that $q > 2\eta$. We will reduce from GapMLD$_{2\eta,q}$, which is W[1]-hard from Theorem 5.1. Let $(\mathbf{A} \in \mathbb{F}_q^{n \times m}, \mathbf{y} \in \mathbb{F}_q^n, k)$ be an instance of GapMLD$_{2\eta,q}$. We create an instance of $(\mathbf{A}', \mathbf{y}', k')$ of GapNVP$_{\eta,p}$ as follows. First, we set $k' = 2k$ and let

$$\mathbf{A}' = \begin{bmatrix} \mathbf{1}_a \otimes \mathbf{A} & \mathbf{1}_a \otimes (q \cdot \mathrm{Id}_n) \\ \mathrm{Id}_n & \mathbf{0}_{n \times n} \\ \mathbf{0}_{k \times k} & \mathbf{0}_{k \times k} \end{bmatrix} \in \mathbb{Z}^{n' \times m'}, \text{and, } \mathbf{y}' = \begin{bmatrix} \mathbf{1}_a \otimes \mathbf{y} \\ \mathbf{0}_n \\ \mathbf{1}_k \end{bmatrix} \in \mathbb{Z}^{n'},$$

where $a = \lceil 2\eta k + 2 \rceil$, $n' = an + k$ and $k' = m + n$. Clearly, the reduction runs in polynomial time. We next argue its correctness.

(YES Case) Suppose that $(\mathbf{A}, \mathbf{y}, k)$ is a YES instance of GapMLD$_{2\eta,q}$, i.e., that there exists $\mathbf{x} \in \{0,1\}^n$ with $\|\mathbf{x}\|_0 \leqslant k$ such that $\mathbf{A}\mathbf{x} = \mathbf{y}$ when operations are over $\mathbb{F}_q$. This means that, when view operations over $\mathbb{Z}$, we have $\mathbf{A}\mathbf{x} = \mathbf{y} + q \cdot \mathbf{z}$ for some $\mathbf{z} \in \mathbb{Z}^n$. Let $\mathbf{x}' = \mathbf{x} \circ (-\mathbf{z}) \in \mathbb{Z}^{m'}$. Then, we have (over $\mathbb{Z}$)

$$\|\mathbf{A}'\mathbf{x}'\|_p^p = \|\mathbf{0}_a \circ \mathbf{x} \circ \mathbf{1}_k\|_p^p \leqslant 2k = k'.$$

In other words, $(\mathbf{A}', \mathbf{y}', k')$ is a YES instance of $\textsc{GapNVP}_{\eta,p}$ as desired.

(NO Case) Suppose that $(\mathbf{A}, \mathbf{y}, k)$ is a NO instance of $\textsc{GapMLD}_{2\eta,q}$. Consider any $\mathbf{x}' \in \mathbb{Z}^{m'}$ and any $w \in \mathbb{Z} \setminus \{0\}$. We would like to show that $\|\mathbf{A}'\mathbf{x}' - w \cdot \mathbf{y}'\|_p^p > \eta \cdot k' = 2\eta k$. To do so, let us write $\mathbf{x}'$ as $\mathbf{x} \circ \mathbf{z}$ where $\mathbf{x} \in \mathbb{Z}^m$ and $\mathbf{z} \in \mathbb{Z}^n$. We can now rearrange $\|\mathbf{A}'\mathbf{x}' - w \cdot \mathbf{y}'\|_p^p$ as

$$\|\mathbf{A}'\mathbf{x}' - w \cdot \mathbf{y}'\|_p^p = a\|\mathbf{A}\mathbf{x} + q\mathbf{z} - \mathbf{y}\|_p^p + \|\mathbf{x}\|_p^p + |w|^p k.$$

As a result, if $\mathbf{A}\mathbf{x} + q\mathbf{z} \neq \mathbf{y}$, then $\|\mathbf{A}'\mathbf{x}' - w \cdot \mathbf{y}'\|_p^p \geqslant a > 2\eta k$. Furthermore, if $|w| \geqslant q$, then we also have $\|\mathbf{A}'\mathbf{x}' - w \cdot \mathbf{y}'\|_p^p \geqslant |w|^p k \geqslant qk > 2\eta k$. Hence, we may henceforth assume that $|w| < q$ and $\mathbf{A}'\mathbf{x}' + q\mathbf{z} = w \cdot \mathbf{y}'$. Since $|w| < q$, it has an inverse modulo q, i.e., there exists $u \in [q-1]$ such that $uw \equiv 1 \pmod{q}$. Now, let us consider $\tilde{\mathbf{x}} \in \mathbb{F}_q^m$ where $\tilde{\mathbf{x}}[i]$ is defined as the remainder of $u \cdot \mathbf{x}[i]$ modulo q. From $\mathbf{A}\mathbf{x} + q\mathbf{z} \neq \mathbf{y}$, we have (over $\mathbb{F}_q$)

$$\mathbf{A}\tilde{\mathbf{x}} = (uw) \cdot \mathbf{y} = \mathbf{y}.$$

Since $(\mathbf{A}, \mathbf{y}, k)$ is a NO instance of $\textsc{GapMLD}_{2\eta,q}$, we must have $\|\tilde{\mathbf{x}}\|_0 > 2\eta k$. Observe that $\|\mathbf{x}\|_0 \geqslant \|\tilde{\mathbf{x}}\|_0$. Thus, we have $\|\mathbf{A}'\mathbf{x}' - w \cdot \mathbf{y}'\|_p^p \geqslant \|\mathbf{x}\|_p^p \geqslant \|\mathbf{x}\|_0 > 2\eta k$. In other words, we can conclude that $(\mathbf{A}', \mathbf{y}', k')$ is a NO instance of $\textsc{GapNVP}_{\eta,p}$. □

7.2 Following Khot's Reduction from NVP to SVP

We will now reduce from GapNVP to GapSVP. This step is almost the same as that of Khot [Kho05], with small changes in parameter selection. Despite this, we repeat the whole argument here (with appropriate adjustments) for completeness.

The main properties of the (randomized) FPT reduction from $\textsc{GapNVP}_{p,\eta}$ to $\textsc{GapSVP}_{p,\gamma}$ are summarized below. For succinctness, we define a couple of additional notation: let $\mathcal{L}(\mathbf{A})$ denote the lattice generated by the matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$, i.e., $\mathcal{L}(\mathbf{A}) = \{\mathbf{A}\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^m\}$, and let $\lambda_p(\mathcal{L})$ denote the length (in the ℓ_p norm) of the shortest vector of the lattice $\mathcal{L}$, i.e., $\lambda_p(\mathcal{L}) = \min\limits_{\mathbf{0} \neq \mathbf{y} \in \mathcal{L}} \|\mathbf{y}\|_p$.

Lemma 7.3. *Fix $p > 1$, and let $\eta \geqslant 1$ be such that $\frac{1}{2} + \frac{1}{2^p} + \frac{(2^p+1)}{\eta} < 1$. Let $(\mathbf{B}, \mathbf{y}, t)$ be a $\textsc{GapNVP}_{p,\eta}$ instance, as given by Theorem 7.2. Then, there is a randomized FPT reduction from $\textsc{GapNVP}_{p,\eta}$ instance $(\mathbf{B}, \mathbf{y}, t)$ to $\textsc{GapSVP}_{p,\gamma}$ instance $(\mathbf{B}_{\mathrm{svp}}, \gamma_p^{-1} l)$ with $l = \eta \cdot t$ such that*

- *(YES) If $(\mathbf{B}, \mathbf{y}, t)$ is a YES instance, then with probability at least 0.8, $\lambda_p(\mathcal{L}(\mathbf{B}_{\mathrm{svp}}))^p \leqslant \gamma_p^{-1} l$.*
- *(NO) If $(\mathbf{B}, \mathbf{y}, t)$ is a NO instance, then with probability at least 0.9, $\lambda_p(\mathcal{L}(\mathbf{B}_{\mathrm{svp}}))^p > l$.*

Here $\gamma_p := \frac{1}{\frac{1}{2} + (2^p+1)/\eta + 1/2^p}$ is strictly greater than 1 by our choice of η.

Combining the above lemma with Theorem 7.2 gives us Theorem 7.1.

We devote the rest of this subsection to describing the reduction (which is similar to that from [Kho05]) and proving Lemma 7.3. In Section 7.2.1, we define the BCH lattice, which is the key gadget used in the reduction. Using the BCH lattice and the GAPNVP$_{p,\eta}$ instance, we construct the intermediate lattice $\mathbf{B}_{\text{int}}$ in Section 7.2.2. The intermediate lattice serves to blow up the number of "good vectors" for the YES case, while controlling the number of "bad vectors" for the NO case. In particular, this step ensures that the number of good vectors in the YES case (Lemma 7.5) far outnumber the number of bad vectors in the NO case (Lemma 7.6). Finally, in Section 7.2.3 we compose the intermediate lattice with a random homogeneous constraint (sampled from an appropriate distribution), to give the final GAPSVP$_{p,\gamma}$ instance. The additional random constraint is used to annihilate all bad vectors in the NO case, while retaining at least one good vector in the YES case.

For the rest of the section, we fix $(\mathbf{B}, \mathbf{y}, t)$ to be a GAPNVP$_{p,\eta}$ instance (as given by Theorem 7.2), and set $l := \eta \cdot t$ and $r := \left(\frac{1}{2} + \frac{1}{2^p} + \frac{1}{\eta}\right) l$. For simplicity of calculations, we will assume that both l and r are integers, and that l is even. Furthermore, we say that a vector $\mathbf{u}$ is *good* (for the YES case) if $\|\mathbf{u}\|_p^p \leqslant \gamma_p^{-1} l$, and we say that $\mathbf{u}$ is *bad* (for the NO case) if $\|\mathbf{u}\|_p^p \leqslant l$.

7.2.1 The BCH Lattice gadget

We begin by defining the BCH lattices which is the key gadget used in the reduction. Given parameters $l, h \in \mathbb{N}$ where $h+1$ is a power of 2 and $l < h$. Let $g = (l/2) \cdot \log(h+1)$. Theorem 3.8 guarantees that there exists a BCH code with block length h, message length $h - g$ and distance $l + 1$. Let $\mathbf{P}_{\text{BCH}} \in \{0,1\}^{g \times h}$ be the parity check matrix of such code. The BCH lattice is defined by

$$\mathbf{B}_{\text{BCH}} = \begin{bmatrix} \mathrm{Id}_h & \mathbf{0}_{h \times g} \\ l \cdot \mathbf{P}_{\text{BCH}} & 2l \cdot \mathrm{Id}_g \end{bmatrix} \in \mathbb{Z}^{(h+g)\times(h+g)}.$$

The following lemma, which is simply a restatement[16] of Lemma 4.3 in [Kho05], summarizes the key properties of BCH lattices, as defined above.

Lemma 7.4 ([Kho05]). *Let $\mathbf{B}_{\text{BCH}} \in \mathbb{Z}^{(h+g)\times(h+g)}$ be as above. There exists a randomized polynomial time algorithm that, with probability at least 0.99, returns a vector $\mathbf{s} \in \mathbb{Z}^{h+g}$ such that the following holds: there are at least $\frac{1}{100} 2^{-g} \binom{h}{r}$ distinct vectors $\mathbf{z} \in \mathbb{Z}^{h+g}$ such that $\|\mathbf{B}_{\text{BCH}}\mathbf{z} - \mathbf{s}\|_p^p = r$.*

7.2.2 The Intermediate Lattice

We now define the intermediate lattice. Let $(\mathbf{B}, \mathbf{y}, t)$ be an instance of GAPNVP$_{p,\eta}$, where $\mathbf{B} \in \mathbb{Z}^{n \times q}$. The intermediate lattice $\mathbf{B}_{\text{int}}$ is constructed as follows. Let $l = \eta t$. Let h be the smallest power of 2 such that $h \geqslant \max\{2n, (10^{10} l)^{2\eta}\}$, and let $\mathbf{B}_{\text{BCH}}$ be constructed as above. Then

$$\mathbf{B}_{\text{int}} = \begin{bmatrix} 2\mathbf{B} & \mathbf{0}_{n \times (h+g)} & 2\mathbf{y} \\ \mathbf{0}_{(h+g)\times q} & \mathbf{B}_{\text{BCH}} & \mathbf{s} \end{bmatrix} \in \mathbb{Z}^{(n+h+g)\times(q+h+g+1)}.$$

where $\mathbf{s} \in \mathbb{Z}^{h+g}$ is the vector given by Lemma 7.4.

[16] In fact, Lemma 7.4 is even weaker than Khot's lemma, since we do not impose a bound on $\|\mathbf{z}\|_p$.

Bounding Good Vectors in YES Case. We now prove a lower bound on the number of good vectors in the YES case.

Lemma 7.5. *Let $(\mathbf{B}, \mathbf{y}, t)$ be a YES instance, and let $\mathbf{B}_{\text{int}}$ be the corresponding intermediate lattice. With probability at least 0.99, there are at least $h^r \left(200h^{l/2}l^l\right)^{-1}$ good non-zero vectors in $\mathcal{L}(\mathbf{B}_{\text{int}})$.*

Proof. Since $(\mathbf{B}, \mathbf{y}, t)$ is a YES instance, there exists $\tilde{\mathbf{x}} \in \mathbb{Z}^q$ such that $\|\mathbf{B}\tilde{\mathbf{x}} - \mathbf{y}\|_p^p \leqslant t$. From Lemma 7.4, with probability at least 0.99, there exist at least $2^{-g}\binom{h}{r}/100$ distinct vectors $\mathbf{z} \in \mathbb{Z}^{h+g}$ such that $\|\mathbf{B}_{\text{BCH}}\mathbf{z} - \mathbf{s}\|^p = r$. For each such $\mathbf{z}$, consider the vector $\mathbf{x} = \tilde{\mathbf{x}} \circ \mathbf{z} \circ -1$. It follows that $\mathbf{B}_{\text{int}}\mathbf{x} = (2\mathbf{B}\tilde{\mathbf{x}} - 2\mathbf{y}) \circ (\mathbf{B}_{\text{BCH}}\mathbf{z} - \mathbf{s})$ is a non-zero vector and $\|\mathbf{B}_{\text{int}}\mathbf{x}\|_p^p = 2^p\|\mathbf{B}\tilde{\mathbf{x}} - \mathbf{y}\|_p^p + \|\mathbf{B}_{\text{BCH}}\mathbf{z} - \mathbf{s}\|_p^p \leqslant 2^p t + r = \gamma_p^{-1} l$. Since the number of such vectors $\mathbf{x}$ is at least the number of distinct coefficient vectors $\mathbf{z}$, it can be lower bounded by

$$\frac{1}{100} \cdot 2^{-g}\binom{h}{r} \geqslant \frac{1}{100} \cdot 2^{-\frac{l}{2}\log(h+1)}\binom{h}{r} \geqslant \frac{1}{100} \cdot \frac{h^r}{r^r(h+1)^{l/2}} \geqslant \frac{1}{200} \cdot \frac{h^r}{l^l h^{l/2}},$$

where the last inequality follows from $r \leqslant l$ and $l < h$. Finally, observe that each $\mathbf{z}$ produces different $\mathbf{B}_{\text{BCH}}\mathbf{z}$ and hence all $\mathbf{B}_{\text{int}}\mathbf{x}$'s are distinct. □

Bounding Bad Vectors in NO Case. We next bound the number of bad vectors in the NO case:

Lemma 7.6. *Let $(\mathbf{B}, \mathbf{y}, t)$ be a NO instance, and let $\mathbf{B}_{\text{int}}$ be the corresponding intermediate lattice. Then the number of bad vectors in $\mathcal{L}(\mathbf{B}_{\text{int}})$ is at most $10^{-5}h^r \left(200h^{l/2}l^l\right)^{-1}$.*

At the heart of the proof is the claim that every bad vector must have even coordinates:

Claim 7.7. *Let $(\mathbf{B}, \mathbf{y}, t)$ be a NO instance, and let $\mathbf{B}_{\text{int}}$ be the corresponding intermediate lattice. Then, for every bad $\mathbf{u} \in \mathcal{L}(\mathbf{B}_{\text{int}})$, all coordinates of $\mathbf{u}$ must be even.*

Proof. Let $\mathbf{u}$ be any bad vector in $\mathcal{L}(\mathbf{B}_{\text{int}})$ and let $\mathbf{x} \in \mathbb{Z}^{q+h+g+1}$ be such that $\mathbf{B}_{\text{int}}\mathbf{x} = \mathbf{u}$. We write $\mathbf{x}$ as $\mathbf{x}_1 \circ \mathbf{x}_2 \circ x$ where $\mathbf{x}_1 \in \mathbb{Z}^q$, $\mathbf{x}_2 \in \mathbb{Z}^{m+h}$ and $x \in \mathbb{Z}$. Using this, we can express $\mathbf{u}$ as $\mathbf{B}_{\text{int}}\mathbf{x} = (2\mathbf{B}\mathbf{x}_1 - 2x \cdot \mathbf{y}) \circ (\mathbf{B}_{\text{BCH}}\mathbf{x}_2 - x \cdot \mathbf{s})$. Recall that $\mathbf{u}$ is bad means that $\|\mathbf{u}\|_p^p \leqslant l$, which implies that $\|\mathbf{B}\mathbf{x}_1 - x \cdot \mathbf{y}\| \leqslant l = \eta \cdot t$. Since $(\mathbf{B}, \mathbf{y}, t)$ is a NO instance, it must be that $x = 0$.

Note that we now have $\mathbf{u} = (2\mathbf{B}\mathbf{x}_1) \circ (\mathbf{B}_{\text{BCH}}\mathbf{x}_2)$. Let us assume for the sake of contradiction that $\mathbf{u}$ has at least one odd coordinate; it must be that $(\mathbf{B}_{\text{BCH}}\mathbf{x}_2)$ has at least one odd coordinate. Let us further write $\mathbf{x}_2$ as $\mathbf{x}_2 = \mathbf{w}_1 \circ \mathbf{w}_2$ where $\mathbf{w}_1 \in \mathbb{Z}^m$ and $\mathbf{w}_2 \in \mathbb{Z}^h$. Notice that $\mathbf{B}_{\text{BCH}}\mathbf{x}_2 = \mathbf{w}_1 \circ (l(\mathbf{P}_{\text{BCH}}\mathbf{w}_1 - 2\mathbf{w}_2))$. Since every coordinate of $\mathbf{B}_{\text{BCH}}\mathbf{x}_2$ must be less than l in magnitude, it must be the case that $\mathbf{P}_{\text{BCH}}\mathbf{w}_1 - 2\mathbf{w}_2 = \mathbf{0}$. In other words, $(\mathbf{w}_1 \mod 2)$ is a codeword of the BCH code. However, since the code has distance $l+1$, this means that, if $\mathbf{w}_1$ has at least one odd coordinate, it must have at least $l+1$ odd (non-zero) coordinates, which contradicts $\|\mathbf{u}\|_p^p \leqslant l$. □

Having proved Claim 7.7, we can now prove Lemma 7.6 by a simple counting argument.

Proof of Lemma 7.6. From Claim 7.7, all coordinates of $\mathbf{u}$ must be even. Therefore, $\mathbf{u}$ must have at most $l/2^p$ non-zero coordinates, all of which have magnitude at most $\lfloor l^{1/p} \rfloor \leqslant l-1$. Hence, we can upper bound the total number of such vectors by

$$(2(l-1)+1)^{l/2^p} \binom{n+h+g}{\lfloor \frac{l}{2^p} \rfloor} \leqslant (2l)^l (n+h+g)^{l/2^p} \leqslant (2l)^l (2lh)^{l/2^p} \leqslant (2l)^{2l} h^{l/2^p}$$

where the second-to-last step holds since $g \leqslant \frac{l}{2}\log(h+1) \leqslant lh/2$ and $n \leqslant h/2$. On the other hand,

$$\frac{h^r}{h^{l/2}l^l} = \frac{h^{\left(\frac{1}{2}+\frac{1}{\eta}+\frac{1}{2^p}\right)l}}{h^{l/2}l^l} = h^{l/2^p}(h/l^\eta)^{l/\eta} \geqslant 10^8\left((2l)^{2l}h^{l/2^p}\right),$$

which follows from $h \geqslant (10^{10}l)^{2\eta}$. Combining the two bounds completes the proof. □

7.2.3 The GapSVP$_{p,\gamma}$ Instance and Proof of The Main Lemma

Finally, we construct $\mathbf{B}_{\text{svp}}$ from $\mathbf{B}_{\text{int}}$ by adding a random homogeneous constraint similar to [Kho05]. For ease of notation, let N_g denote the lower bound on the number of distinct coefficient vectors guaranteed by Lemma 7.5 in the YES case. Similarly, let N_a denote the upper bound on the number of annoying vectors as given in Lemma 7.6. Combining the two Lemmas we have $N_g \geqslant 10^5 N_a$, which will be used crucially in the construction and analysis of the final lattice.

Construction of the Final Lattice. Let ρ be any prime number in[17] $\left[10^{-4}N_g, 10^{-2}N_g\right]$. Furthermore, let $\mathbf{r} \overset{\text{unif}}{\sim} [0, \rho-1]^{n+h+g}$ be a uniformly sampled lattice point. We construct $\mathbf{B}_{\text{svp}}$ as

$$\mathbf{B}_{\text{svp}} = \begin{bmatrix} \mathbf{B}_{\text{int}} & 0 \\ l \cdot \mathbf{r}^T \mathbf{B}_{\text{int}} & l \cdot \rho \end{bmatrix} \in \mathbb{Z}^{(n+h+g+1)\times(q+h+g+2)}.$$

This can be thought of as adding a random linear constraint to the intermediate lattice. The choice of parameters ensures that with good probability, in the YES case, at least one of the good vectors $\mathbf{x} \in \mathbb{Z}^{q+h+g+1}$ evaluates to 0 modulo ρ on the random constraint, and therefore we can pick $u \in \mathbb{Z}$ such that $\mathbf{B}_{\text{svp}}(\mathbf{x} \circ u) = (\mathbf{B}_{\text{int}}\mathbf{x}) \circ 0$ still has small ℓ_p norm. On the other hand, since $N_a \ll N_g$, with good probability, all of bad vectors evaluate to non-zeros, and hence will contribute a coordinate of magnitude l. This intuition is formalized below.

Proof of Lemma 7.3. Let $\mathbf{B}_{\text{svp}}$ be the corresponding final lattice of $(\mathbf{B}, \mathbf{y}, t)$ as described above. Observe that given the GapNVP$_{p,\eta}$-instance $(\mathbf{B}, \mathbf{y}, t)$, we can construct $\mathbf{B}_{\text{svp}}$ in poly(n, q, t)-time.

Moreover, observe that $\mathcal{L}(\mathbf{B}_{\text{svp}})$ is exactly equal to $\{\mathbf{u} \circ (l \cdot w) \mid \mathbf{u} \in \mathcal{L}(\mathbf{B}_{\text{int}}), w \equiv \mathbf{r}^T\mathbf{u} \pmod{\rho}\}$.

[17] Note that the density of primes in this range is at least $1/\log N_g = 1/r\log h$. Therefore, a random sample of size $O(r \log h)$ in this range contains a prime with high probability. Since we can test primality for any $\rho \in \left[10^{-4}N_g, 10^{-2}N_g\right]$ in FPT time, this gives us an FPT algorithm to sample such a prime number efficiently .

Suppose that $(\mathbf{B}, \mathbf{y}, t)$ is a NO instance. Consider any $\mathbf{u} \circ (l \cdot w) \in \mathcal{L}(\mathbf{B}_{\text{svp}})$. If $\|\mathbf{u} \circ (l \cdot w)\|_p^p \leqslant l$, it must be that $\|\mathbf{u}\|_p^p \leqslant l$ and $w = 0$; the latter is equivalent to $\mathbf{r}^T\mathbf{u} \equiv 0 \pmod{\rho}$. However, from Lemma 7.6, there are only N_a bad vectors $\mathbf{u}$ in $\mathcal{L}(\mathbf{B}_{\text{int}})$. For each such non-zero $\mathbf{u}$, the probability that $\mathbf{r}^T\mathbf{u} \equiv 0 \pmod{\rho}$ is exactly $1/\rho$. As a result, by taking union bound over all such $\mathbf{u} \neq \mathbf{0}$, we can conclude that, with probability at least $1 - N_a/\rho \geqslant 0.9$, we have $\lambda_p(\mathcal{L}(\mathbf{B}_{\text{svp}}))^p > l$.

Next, suppose that $(\mathbf{B}, \mathbf{y}, t)$ is a YES instance. We will show that, with probability at least 0.8, $\lambda_p(\mathcal{L}(\mathbf{B}_{\text{svp}}))^p \leqslant \gamma_p^{-1} l$. To do this, we first condition on the event that there exists at least N_g good vectors as guaranteed by Lemma 7.5. Consider any two good vectors $\mathbf{u}_1 \neq \mathbf{u}_2$. Since each entry of $\mathbf{u}_1$ and $\mathbf{u}_2$ is of magnitude at most $(\gamma_p^{-1} l)^{1/p}$, they are pairwise independent modulo $\rho > 2l$. Therefore, instantiating Lemma 5.8 from [Kho05] with the lower bound on the number of good vectors N_g, and our choice of ρ, it follows that with probability at least 0.9, there exists a good vector $\mathbf{u}$ such that $\mathbf{r}^T\mathbf{u} \equiv 0 \pmod{\rho}$, i.e., $\mathbf{u} \circ 0$ belongs to $\mathcal{L}(\mathbf{B}_{\text{svp}})$. Therefore, by union bound, with probability at least 0.8 (over the randomness of Lemma 7.5 and the choice of $\mathbf{r}$), there exists a good $\mathbf{u} \in \mathcal{L}(\mathbf{B}_{\text{int}})$ such that $\mathbf{u} \circ 0$ remains in $\mathcal{L}(\mathbf{B}_{\text{svp}})$, which concludes the proof. □

8 Conclusion and Open Questions

In this work, we have shown the parameterized inapproximability of k-Minimum Distance Problem (k-MDP) and k-Shortest Vector Problem (k-SVP) in the ℓ_p norm for every $p > 1$ assuming W[1] $\neq$ FPT (and under randomized reductions).

An immediate open question stemming from our work is whether k-SVP in the ℓ_1 norm is in FPT. Khot's reduction unfortunately does not work for ℓ_1; indeed, in the work of Haviv and Regev [HR07], they arrive at the hardness of approximating SVP in the ℓ_1 norm by embedding SVP instances in ℓ_2 to instances in ℓ_1 using an earlier result of Regev and Rosen [RR06]. The Regev-Rosen embedding inherently does not work in the FPT regime either, as it produces non-integral lattices. Similar issue applies to an earlier hardness result for SVP on ℓ_1 of [Mic00], whose reduction produces irrational bases.

An additional question regarding k-SVP is whether we can prove hardness of approximation for *every* constant factor for $p \neq 2$. We note here that for $p = 2$, we can use the tensor product of lattices to amplify the gap, as Khot's construction is tailored so that the resulting lattice is "well-behaved" under tensoring, and gap amplification is indeed possible for such instances. However, if $p \neq 2$ then the gap amplification techniques of [Kho05, HR07] require the distance k to be dependent on the input size nm, and hence are not applicable for us. To the best of our knowledge, it is unknown whether this dependency is necessary. If they are indeed required, it would also be interesting to see whether other different techniques that work for our settings can be utilized for gap amplification instead of those from [Kho05, HR07].

Furthermore, the Minimum Distance Problem can be defined for linear codes in $\mathbb{F}_p$ for any larger field of size $p > 2$ as well. It turns out that our result does not rule out FPT algorithms for k-MDP over $\mathbb{F}_p$ with $p > 2$, when p is fixed and is not part on the input. The issue here is that, in our proof of existence of Locally Suffix Dense Codes (Lemma 6.4), we need the co-dimension of the code to be small compared to its distance. In particular, the co-dimension

$h - m$ has to be at most $(d/2 + O(1)) \log_p h$ where d is the distance. While the BCH code over binary alphabet satisfies this property, we are not aware of any linear codes that satisfy this for larger fields. It is an intriguing open question to determine whether such codes exist, or whether the reduction can be made to work without existence of such codes.

Since the current reductions for both k-MDP and k-SVP are randomized, it is still an intriguing open question whether we can find deterministic reductions for these problems. As stated in the introduction, even in the non-parameterized setting, NP-hardness of SVP through deterministic reductions is not known. On the other hand, MDP is known to be NP-hard even to approximate under deterministic reductions; in fact, even the Dumer et al.'s reduction [DMS03] that we employ can be derandomized, as long as one has a deterministic construction for Locally Dense Codes [CW12, Mic14]. In our settings, if one can deterministically construct Locally Suffix Dense Codes (i.e. derandomize Lemma 6.4), then we would also get a deterministic reduction for k-MDP.

Acknowledgements

We are grateful to Ishay Haviv for providing insights on how the gap amplification for $p \neq 2$ from [HR07] works. Pasin would like to thank Danupon Nanongkai for introducing him to the k-Even Set problem and for subsequent useful discussions.

Arnab Bhattacharyya contributed to this work while he was at the Indian Institute of Science and was supported by Ramanujan Fellowship DSTO 1358 and the Indo-US Joint Center for Pseudorandomness in Computer Science. Édouard Bonnet, László Egri, Bingkai Lin, and Dániel Marx are supported by the European Research Council (ERC) consolidator grant No. 725978 SYSTEMATICGRAPH. László Egri is also supported by NSERC. Karthik C. S. is supported by Irit Dinur's ERC-CoG grant 772839. Bingkai Lin is also supported by JSPS KAKENHI Grant (JP16H07409) and the JST ERATO Grant (JPMJER1201) of Japan. Pasin Manurangsi is supported by the Indo-US Joint Center for Pseudorandomness in Computer Science.

References

[ABSS97] 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.

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293, 1997.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.

[Ajt98] Miklós Ajtai. The shortest vector problem in ℓ_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, pages 10–19, 1998.

[AK14] Per Austrin and Subhash Khot. A simple deterministic reduction for the gap minimum distance of code problem. *IEEE Trans. Information Theory*, 60(10):6636–6645, 2014.

[AS18] Divesh Aggarwal and Noah Stephens-Davidowitz. (gap/s)eth hardness of SVP. In *STOC*, pages 228–238, 2018.

[BGGS16] Arnab Bhattacharyya, Ameet Gadekar, Suprovat Ghoshal, and Rishi Saket. On the hardness of learning sparse parities. In *ESA*, pages 11:1–11:17, 2016.

[BGS17] Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. On the quantitative hardness of CVP. In *FOCS*, pages 13–24, 2017.

[BMvT78] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Trans. Information Theory*, 24(3):384–386, 1978.

[BR60] R. C. Bose and Dwijendra K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1):68–79, 1960.

[CFHW17] Marek Cygan, Fedor V. Fomin, Danny Hermelin, and Magnus Wahlström. Randomization in parameterized complexity (dagstuhl seminar 17041). *Dagstuhl Reports*, 7(1):103–128, 2017.

[CFJ$^+$14] Marek Cygan, Fedor Fomin, Bart MP Jansen, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, and Saket Saurabh. Open problems for fpt school 2014. 2014.

[CFK$^+$15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[CN99] Jin-yi Cai and Ajay Nerurkar. Approximating the SVP to within a factor $(1 + 1/\dim^{\xi})$ is NP-hard under randomized reductions. *J. Comput. Syst. Sci.*, 59(2):221–239, 1999.

[CW12] Qi Cheng and Daqing Wan. A deterministic reduction for the gap minimum distance problem. *IEEE Trans. Information Theory*, 58(11):6935–6941, 2012.

[DF99] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

[DF13] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[DFVW99] 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.

[DGMS07] Erik D. Demaine, Gregory Gutin, Dániel Marx, and Ulrike Stege. 07281 open problems – structure theory and FPT algorithmcs for graphs, digraphs and hypergraphs. In *Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs, 08.07. - 13.07.2007*, 2007.

[Din02] Irit Dinur. Approximating SVP$_\infty$ to within almost-polynomial factors is NP-hard. *Theor. Comput. Sci.*, 285(1):55–71, 2002.

[Din16] Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *ECCC*, 23:128, 2016.

[DKRS03] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003.

[DMS03] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Information Theory*, 49(1):22–37, 2003.

[FGMS12] Michael R. Fellows, Jiong Guo, Dániel Marx, and Saket Saurabh. Data reduction and problem kernels (dagstuhl seminar 12241). *Dagstuhl Reports*, 2(6):26–50, 2012.

[FM12] Fedor V. Fomin and Dániel Marx. FPT suspects and tough customers: Open problems of downey and fellows. In Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors, *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, pages 457–468. Springer, 2012.

[GKS12] Petr A. Golovach, Jan Kratochvíl, and Ondrej Suchý. Parameterized complexity of generalized domination problems. *Discrete Applied Mathematics*, 160(6):780–792, 2012.

[GMSS99] 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.

[Gol06] Oded Goldreich. On promise problems: A survey. In *Theoretical Computer Science, Essays in Memory of Shimon Even*, pages 254–290, 2006.

[Hoc59] Alexis Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, 2:147–156, September 1959.

[HR07] Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *STOC*, pages 469–477, 2007.

[Joh90] D. S. Johnson. Handbook of theoretical computer science. volume A (Algorithms and Complexity), chapter 2, A catalog of complexity classes, pages 67–161. Elseveir, 1990.

[Kho05] Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *J. ACM*, 52(5):789–808, 2005.

[KLM19] Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *J. ACM*, 66(5):33:1–33:38, 2019.

[Len83] Hendrik Willem Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983.

[Lin18] Bingkai Lin. The parameterized complexity of the k-biclique problem. *J. ACM*, 65(5):34:1–34:23, 2018.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[Maj17] Ruhollah Majdoddin. Parameterized complexity of CSP for infinite constraint languages. *CoRR*, abs/1706.10153, 2017.

[MG12] Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems: a cryptographic perspective*, volume 671. Springer Science & Business Media, 2012.

[Mic00] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput.*, 30(6):2008–2035, 2000.

[Mic01] Daniele Micciancio. The hardness of the closest vector problem with preprocessing. *IEEE Trans. Information Theory*, 47(3):1212–1215, 2001.

[Mic12] Daniele Micciancio. Inapproximability of the shortest vector problem: Toward a deterministic reduction. *Theory of Computing*, 8(1):487–512, 2012.

[Mic14] Daniele Micciancio. Locally dense codes. In *CCC*, pages 90–97, 2014.

[MR09] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

[MR16] Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense CSPs. *CoRR*, abs/1607.02986, 2016.

[NSS95] Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 182–191, 1995.

[NV10] Phong Q. Nguyen and Brigitte Vallée, editors. *The LLL Algorithm - Survey and Applications*. Information Security and Cryptography. Springer, 2010.

[Reg03] Oded Regev. New lattice based cryptographic constructions. In *STOC*, pages 407–416, 2003.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[Reg06] Oded Regev. Lattice-based cryptography. In *CRYPTO*, pages 131–141, 2006.

[Reg10] Oded Regev. The learning with errors problem (invited survey). In *CCC*, pages 191–204, 2010.

[RR06] Oded Regev and Ricky Rosen. Lattice problems and norm embeddings. In *STOC*, pages 447–456, 2006.

[Ste93] Jacques Stern. Approximating the number of error locations within a constant ratio is NP-complete. In *AAECC*, pages 325–331, 1993.

[Var97a] Alexander Vardy. Algorithmic complexity in coding theory and the minimum distance problem. In *STOC*, pages 92–109, 1997.

[Var97b] Alexander Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. Information Theory*, 43(6):1757–1766, 1997.

[vEB81] Peter van Emde-Boas. *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*. Report. Department of Mathematics. University of Amsterdam. Department, Univ., 1981.

A Inapproximability of Odd Set

In this section, we show how the hardness for the more general GapMLD, also implies hardness for the Odd Set problem, which can be defined in a similar manner as GapMLD except that $\mathbf{y}$ is always fixed as the all-ones vector instead of being part of the input. More formally, we define the gap version of Odd Set below.

γ-Gap Odd Set Problem (GapOddSet$_\gamma$)

Input: A matrix $\mathbf{A} \in \mathbb{F}_2^{n \times m}$ and a positive integer $k \in \mathbb{N}$

Parameter: k

Question: Distinguish between the following two cases:

- (YES) there exists $\mathbf{x} \in \mathcal{B}_m(\mathbf{0}, k)$ such that $\mathbf{A}\mathbf{x} = \mathbf{1}$
- (NO) for all $\mathbf{x} \in \mathcal{B}_m(\mathbf{0}, \gamma k)$, $\mathbf{A}\mathbf{x} \neq \mathbf{1}$

It is obvious that hardness for GapOddSet gives the hardness for GapMLD, by simply setting $\mathbf{y} = \mathbf{1}$. Below we show that the opposite implication is also true; note that, together with Theorem 5.1, it implies that GapOddSet$_\gamma$ is W[1]-hard for every $\gamma \geqslant 1$.

Proposition A.1. *For every $\gamma' > \gamma \geqslant 1$, there is an FPT reduction from* GapMLD$_{\gamma'}$ *to* GapOddSet$_\gamma$

Proof. Let $(\mathbf{A}, \mathbf{y}, k)$ be an instance of GapMLD$_{\gamma'}$ where $\mathbf{A} \in \mathbb{F}_2^{n \times m}$ and $\mathbf{y} \in \mathbb{F}_2^n$. We may assume without loss of generality that $k > \frac{\gamma}{\gamma' - \gamma}$. The instance $(\mathbf{A}' \in \mathbb{F}_2^{(n+1)\times(m+1)}, k')$ of GapOddSet$_\gamma$ is defined as follows. First, we let $k' = k + 1$. Then, for $i \in [m]$, we let the i-th column of $\mathbf{A}'$ be the i-th column concatenated with zero, and we let the $(m+1)$-th column be $(\mathbf{1}_n + \mathbf{y}) \circ 1$. That is,

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} & \mathbf{1}_n + \mathbf{y} \\ \mathbf{0}_{1\times m} & 1 \end{bmatrix}.$$

Clearly, the reduction runs in polynomial time. We next argue its correctness.

(YES Case) Suppose that $(\mathbf{A}, \mathbf{y}, k)$ is a YES instance of GapMLD$_{\gamma'}$, i.e., there exists $\mathbf{x} \in \mathbb{F}_2^n$ with $\|\mathbf{x}\|_0 \leqslant k$ such that $\mathbf{A}\mathbf{x} = \mathbf{y}$. Let $\mathbf{x}' = \mathbf{x} \circ 1$; it is simple to see that $\mathbf{A}'\mathbf{x}' = \mathbf{1}_{m+1}$ and that $\|\mathbf{x}'\|_0 \leqslant k + 1 = k'$. Hence, $(\mathbf{A}', k')$ is a YES instance of GapOddSet$_\gamma$.

(NO Case) Suppose that $(\mathbf{A}, \mathbf{y}, k)$ is NO instance of GapMLD$_{\gamma'}$. Now, let us consider any $\mathbf{x}' \in \mathbb{F}_2^{n+1}$ such that $\mathbf{A}'\mathbf{x}' = \mathbf{1}$. Notice that $(\mathbf{A}'\mathbf{x}')[m+1] = \mathbf{x}'[n+1]$, which implies that $\mathbf{x}'[n+1] = 1$.

Now, consider the vector $\mathbf{x} = (\mathbf{x}[1], \dots, \mathbf{x}[n])$; it is easy to verify that $\mathbf{A}\mathbf{x} = \mathbf{y}$. Since $(\mathbf{A}, \mathbf{y}, k)$ is NO instance of GapMLD$_{\gamma'}$, we have $\|\mathbf{x}\|_0 > \gamma' k$. As a result, $\|\mathbf{x}'\|_0 = 1 + \|\mathbf{x}\|_0 > 1 + \gamma' k > \gamma \cdot k'$, where the last inequality follows from $k \geqslant \frac{\gamma}{\gamma' - \gamma}$. Thus, $(\mathbf{A}', k')$ is indeed a NO instance of GapOddSet$_\gamma$. □