Parameterized Intractability of Even Set and Shortest Vector Problem from Gap-ETH

Arnab Bhattacharyya[1]
Indian Institute of Science, Bangalore, India
arnabb@iisc.ac.in

Suprovat Ghoshal
Indian Institute of Science, Bangalore, India
suprovat@iisc.ac.in

Karthik C. S.[2]
Weizmann Institute of Science, Rehovot, Israel
karthik.srikanta@weizmann.ac.il

Pasin Manurangsi[3]
University of California, Berkeley, USA
pasin@berkeley.edu

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. Here, k is the parameter of the problem. The question of whether k-Even Set is fixed parameter tractable (FPT) has been repeatedly raised in literature and has earned its place in Downey and Fellows' book (2013) as one of the "most infamous" open problems in the field of Parameterized Complexity.

In this work, we show that k-Even Set does not admit FPT algorithms under the (randomized) Gap Exponential Time Hypothesis (Gap-ETH) [Dinur'16, Manurangsi-Raghavendra'16]. In fact, our result rules out not only exact FPT algorithms, but also any constant factor FPT approximation algorithms for the problem. Furthermore, our result holds even under the following weaker assumption, which is also known as the *Parameterized Inapproximability Hypothesis* (PIH) [Lokshtanov et al.'17]: no (randomized) FPT algorithm can distinguish a satisfiable 2CSP instance from one which is only 0.99-satisfiable (where the parameter is the number of variables).

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, this problem is also a long-standing open problem in the field of Parameterized Complexity. We show that, for any $p > 1$, k-SVP is hard to approximate (in FPT time) to some constant factor, assuming PIH. Furthermore, for the case of $p = 2$, the inapproximability factor can be amplified to any constant.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Parameterized Complexity, Inapproximability, Even Set, Minimum Distance Problem, Shortest Vector Problem, Gap-ETH

[1] This work was supported by Ramanujan Fellowship DSTO 1358.
[2] This work was supported by Irit Dinur's ERC-CoG grant 772839.
[3] Some part of this work was done while the author was visiting Indian Institute of Science and supported by the Indo-US Joint Center for Pseudorandomness in Computer Science.

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 17; pp. 17:1–17:15

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.17

Related Version A full version of the paper is available at http://arxiv.org/abs/1803.09717.

Acknowledgements We are grateful to Ishay Haviv for providing insights on how the gap amplification from [26] works. PM thanks Danupon Nanongkai for introducing him to k-Even Set.

1 Introduction

The study of error-correcting codes gives rise to many 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[4] 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 $\min_{\mathbf{0} \neq \mathbf{x} \in \mathbb{F}_2^m} \|\mathbf{Ax}\|_0$ where $\|\cdot\|_0$ denotes the 0-norm (aka the Hamming norm).

The study of MDP dates back to at least 1978 when Berlekamp et al. [8] conjectured that it is NP-hard. This conjecture remained open for almost two decades until it was positively resolved by Vardy [46, 47]. Later, Dumer et al. [22] strengthened this 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 [11] and further simplified by Austrin and Khot [6] and Micciancio [36].

While the aforementioned 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: *parameterization*.

In parameterized problems, part of the input is an integer 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[5] k-MDP has k as the parameter and the question is to decide if the code generated by $\mathbf{A}$ has distance at most k in time $T(k) \cdot \text{poly}(mn)$ where T can be any computable function that depends only on k.

The parameterized complexity of k-MDP was first posed as an open problem by Downey et al. [21][6,7] who showed that parameterized variants of several other coding-theoretic problems, including the Nearest Codeword Problem and the Nearest Vector Problem[8] 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:

[4] Note that MDP can be defined over larger fields as well; we discuss more about this in Section 3.
[5] 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.
[6] k-MDP is formulated differently in [21] where the input is the parity-check matrix instead of the generator matrix. Since we can efficiently compute one given the other, the two formulations are equivalent.
[7] k-MDP is commonly referred to as *k-Even Set* due to its graph theoretic interpretation (see [21]).
[8] The Nearest Vector Problem is also referred to in the literature as the Closest Vector Problem.

▶ **Open Question 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 book of Downey and Fellows [19] and has been reiterated numerous times over the years [15, 23, 25, 20, 12, 14, 9, 13, 31]. In fact, in their second book [20], Downey and Fellows even include this problem as one of the six "most infamous" open questions in the area of Parameterized Complexity.

Another question posted in [21] 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 $\min_{\mathbf{0} \neq \mathbf{x} \in \mathbb{Z}^m} \|\mathbf{Ax}\|_p \leqslant k$. Again, k is the parameter of the problem. Note that, similar to [21], we require the basis of the lattice to be integer-valued, which is sometimes not enforced in the literature (e.g. [45, 3]). This is because, if $\mathbf{A}$ is allowed to have rational entries, then parameterization is meaningless because we can simply scale $\mathbf{A}$ down by a large multiplicative factor.

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 [28] can be used to factor rational polynomials, and to solve integer programming (parameterized by the number of unknowns) [29] and many other computational number-theoretic problems (see e.g. [38]). Furthermore, the hardness of (approximating) SVP has been used as the basis of several cryptographic constructions [3, 4, 39, 40]. Since these topics are out of scope of our paper, we refer the interested readers to the following surveys for more details: [41, 37, 38, 42].

On the computational hardness side of the problem, van Emde-Boas [45] 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 [2] 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 [10]. Finally, Micciancio [33] managed to achieve a factor that is bounded away from one. Specifically, Micciancio [33] showed (again under randomized reductions) that SVP on the ℓ_p norm is NP-hard to approximate within a factor of $\sqrt[p]{2}$ for every $1 \leqslant p < \infty$. Khot [27] later improved the ratio to any constant, and even to $2^{\log^{1/2-\varepsilon}(nm)}$ under a stronger assumption. Haviv and Regev [26] 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 [35]); 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 is known [16]: 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 [7, 1]. The authors of [7, 1] 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[9] and the (randomized) *Gap Exponential Time Hypothesis (Gap-ETH)* [17, 32], which states that no randomized subexponential time algorithm can distinguish between a satisfiable 3CNF formula and one which is only 0.99-satisfiable.

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

As with MDP, Downey et al. [21] 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 [19, 20], albeit, in their second book, k-SVP was in the "tough customers" list instead of the "most infamous" list that k-MDP belonged to. And again, as with Open Question 1, this question remains unresolved to this day:

▶ **Open Question 2.** *Is k-SVP fixed parameter tractable?*

1.1 Our Results

The main result of this paper is a resolution to the previously mentioned Open Question 1 and 2: more specifically, we prove that k-MDP and k-SVP (on ℓ_p norm for any $p > 1$) do not admit any FPT algorithm, assuming the aforementioned (randomized) Gap-ETH. In fact, our result is slightly stronger than stated here in a couple of ways:

- We rule out not only exact FPT algorithms but also FPT approximation algorithms.
- Second, our result works even under the so-called *Parameterized Inapproximability Hypothesis (PIH)* [30], which asserts that no (randomized) FPT algorithm[10] can distinguish between a satisfiable 2CSP instance and one which is only 0.99-satisfiable, where the parameter is the number of variables. It is known that Gap-ETH implies PIH.

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 3.** *Assuming PIH, for any $\gamma \geqslant 1$ and any computable function T, no $T(k) \cdot poly(nm)$-time algorithm, on input $(\mathbf{A}, k) \in \mathbb{F}_2^{n \times m} \times \mathbb{N}$, can 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 $\gamma \cdot k$.*

While our above result rules out FPT approximation algorithms with *any* constant approximation ratio for k-MDP, we can only prove FPT inapproximability with *some* constant ratio for k-SVP in ℓ_p norm for $p > 1$, with the exception of $p = 2$ for which the ratio in our result can be amplified to any constant. These are stated more precisely below.

▶ **Theorem 4.** *For any $p > 1$, there exists a constant $\gamma_p > 1$ such that, assuming PIH, for any computable function T, no $T(k) \cdot poly(nm)$-time algorithm, on input $(\mathbf{A}, k) \in \mathbb{Z}^{n \times m} \times \mathbb{N}$, can 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 $> \gamma_p \cdot k$.*

▶ **Theorem 5.** *Assuming PIH, for any computable function T and constant $\gamma \geqslant 1$, no $T(k) \cdot poly(nm)$-time algorithm, on input $(\mathbf{A}, k) \in \mathbb{Z}^{n \times m} \times \mathbb{N}$, can distinguish between*

- *the ℓ_2 norm of the shortest vector of the lattice generated by $\mathbf{A}$ is $\leqslant k$, and,*
- *the ℓ_2 norm of the shortest vector of the lattice generated by $\mathbf{A}$ is $> \gamma \cdot k$.*

We remark that our results do not yield hardness for SVP in the ℓ_1 norm and this remains an interesting open question. Section 3 contains discussion on this problem. We also note that, for Theorem 4 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 $k = 1$ [45]!

[10] The original formulation from [30] is slightly different in that it states that the problem is W[1]-hard.

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.

As an intermediate step of our proof, we show that the k-NCP and k-NVP problems are hard to approximate[11]. This should be compared to [21], in which the authors show that both problems are W[1]-hard. The distinction here is that our result rules out not only exact algorithms but also approximation algorithms, at the expense of the stronger assumption than that of [21]. Indeed, if one could somehow show that k-NCP and k-NVP are W[1]-hard to approximate (to some constant strictly greater than one), then our reduction would imply W[1]-hardness of k-MDP and k-SVP (under randomized reductions). Unfortunately, no such W[1]-hardness of approximation of k-NCP and k-NVP is known yet.

We end this section by remarking that the computational complexity of both (non-parameterized) NCP and NVP are also thoroughly studied (see e.g. [34, 18, 44, 5, 24] 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.

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, via relatively simple reductions from PIH, that both k-NCP and k-NVP are hard to approximate. Then, we reduce k-NCP and k-NVP to k-MDP and k-SVP respectively. In this second step, we employ Dumer et al.'s reduction [22] for k-MDP and Khot's reduction [27] for k-SVP. While the latter works almost immediately in the parameterized regime, there are several technical challenges in adapting Dumer et al.'s reduction 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 settings.

The starting point of all the hardness results in this paper is Gap-ETH. As mentioned earlier, it is well-known that Gap-ETH implies PIH, i.e., PIH is weaker than Gap-ETH. Hence, for the rest of this section, we may start from PIH instead of Gap-ETH.

2.1 Parameterized Intractability of k-MDP from PIH

We start this subsection by describing the Dumer et al.'s (henceforth DMS) reduction [22]. The starting point of the DMS reduction is the NP-hardness of approximating NCP to any constant factor [5]. 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. [5] 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

[11] While our k-MDP result only applies for $\mathbb{F}_2$, our result for k-NCP holds for any finite field $\mathbb{F}_q$ too.

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[12] $r < d$ 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:

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

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 particular the dimensions of $\mathbf{L}$ and $\mathbf{T}$ such that all these computations are valid). Let $\mathbf{z} = \mathbf{z}' \circ 1$ (where $\circ$ is used to denote the concatenation operation on vectors), 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[13] $\gamma > 2$, 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 codes with all of the above described properties. Specifically, they gave an efficient deterministic construction of $\mathbf{L}$, but only gave a randomized algorithm that finds $\mathbf{T}$ and $\mathbf{s}$ 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 [11] and Micciancio [36] 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 does not depend on the input size). This is because

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

[13] 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.

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 constant inapproximability of k-NCP under PIH. Specifically, assuming PIH, we first show the parameterized inapproximability of k-NCP for some constant factor greater than 1, and then boost the gap using a composition operator (self-recursively). Note that in order to follow the DMS reduction, we need the inapproximability of k-NCP for some constant factor greater than 2; in other words, the gap amplification for k-NCP is necessary, even if we are not interested in showing the inapproximability of k-NCP for all constant factors.

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 only one way: in the YES case, we want to find $\mathbf{x} \in \mathcal{B}(\mathbf{0}, k)$ (rather than from the entire space $\mathbb{F}_2^m$), such that $\|\mathbf{Ax} - \mathbf{y}\|_0 \leqslant k$. In other words, we only allow sparse $\mathbf{x}$ as a solution. We show that k-NCP can be reduced to the sparse nearest codeword problem.

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

We note that overcoming the second and third obstacles are our main technical contributions. Specifically, our result on sparse covering codes might be of independent interest.

The full reduction goes through several intermediate steps, which we will describe in more detail in the coming paragraphs. 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. For every $\varepsilon \geqslant 0$, we denote by $\text{Gap2CSP}_\varepsilon$ the gap problem where we have to determine if a given 2CSP instance Γ, i.e., a graph $G = (V, E)$ and a set of constraints $\{C_{uv}\}_{(u,v)\in E}$ over an alphabet set Σ, has an assignment to its vertices that satisfies all the constraints or if every assignment violates more than ε fraction of the constraints. Here each C_{uv} is simply the set of all $(\sigma_u, \sigma_v) \in \Sigma \times \Sigma$ that satisfy the constraint. The parameter of the problem is $|V|$. PIH asserts that there exists some constant $\varepsilon > 0$ such that no randomized FPT algorithm can solve $\text{Gap2CSP}_\varepsilon$.

Reducing Gap2CSP$_\varepsilon$ to GapMLD$_\gamma$. We start by showing the parameterized inapproximability of k-NCP for some constant ratio. Instead of working with k-NCP, we work with its equivalent formulation (by converting the generator matrix given as input into a parity-check matrix) which in the literature is referred to as the *maximum likelihood decoding problem*[14]. We define the gap version of this problem (i.e., a promise problem), denoted by GapMLD_γ (for some constant $\gamma \geqslant 1$) as follows: on input $(\mathbf{A}, \mathbf{y}, k)$, distinguish between the YES case

[14] The two formulations are equivalent but we use different names for them to avoid confusion when we use *Sparse Nearest Codeword Problem* later on.

where there exists $\mathbf{x} \in \mathcal{B}(\mathbf{0}, k)$ such that $\mathbf{Ax} = \mathbf{y}$, and the NO case where for all $\mathbf{x} \in \mathcal{B}(\mathbf{0}, \gamma k)$ we have $\mathbf{Ax} \neq \mathbf{y}$. It is good to keep in mind that this is equivalent to asking whether there exist k columns of $\mathbf{A}$ whose sum is equal to $\mathbf{y}$ or whether any $\leqslant \gamma k$ columns of $\mathbf{A}$ do not sum up to $\mathbf{y}$.

Next, we sketch the reduction from an instance $(G = (V, E), \Sigma, \{C_{uv}\}_{(u,v)\in E})$ of Gap2CSP$_\varepsilon$ to an instance $(\mathbf{A}, \mathbf{y}, k)$ of GapMLD$_{1+\varepsilon/3}$. The matrix $\mathbf{A}$ has $|V||\Sigma| + \sum_{(u,v)\in E} |C_{uv}|$ columns and $|V| + |E| + 2|E||\Sigma|$ rows. The first $|V||\Sigma|$ columns of $\mathbf{A}$ are labelled with $(u, \sigma_u) \in V \times \Sigma$, and the remaining columns are labeled by (e, σ_u, σ_v) where $e = (u, v) \in E$ and $(\sigma_u, \sigma_v) \in C_{uv}$.

Before we continue with our description of $\mathbf{A}$, let us note that, in the YES case where there is a satisfying assignment $\phi : V \to \Sigma$, our intended solution for our GapMLD instance is to pick the $(u, \phi(u))$-column for every $u \in V$ and the $((u, v), \phi(u), \phi(v))$-column for every $(u, v) \in E$. Notice that $|V| + |E|$ columns are picked, and indeed we set $k = |V| + |E|$. Moreover, we set the first $|V| + |E|$ coordinates of $\mathbf{y}$ to be one and the rest to be zero.

We also identify the first $|V|$ rows of $\mathbf{A}$ with $u \in V$, the next $|E|$ rows with $e \in E$, and the remaining $2|E||\Sigma|$ rows with $(e, \sigma, b) \in E \times \Sigma \times \{0, 1\}$. Figure 1 provides an illustration of $\mathbf{A}$. The rows of $\mathbf{A}$ will be designed to serve the following purposes: the first $|V|$ rows will ensure that, for each $u \in V$, at least one column of the form $(u, \cdot)$ is picked, the next $|E|$ rows will ensure that, for each $e \in E$, at least one column of the form $(e, \cdot, \cdot)$ is picked, and finally the remaining $2|E||\Sigma|$ rows will "check" that the constraint is indeed satisfied.

Specifically, each u-row for $u \in V$ has only $|\Sigma|$ non-zero entries: those in column (u, σ_u) for all $\sigma_u \in \Sigma$. Since our target vector $\mathbf{y}$ has $\mathbf{y}_u = 1$, we indeed have that at least one column of the form $(u, \cdot)$ must be selected for every $u \in V$. Similarly, each e-row for $e = (u, v) \in E$ has $|C_{uv}|$ non-zero entries: those in column (e, σ_u, σ_v) for all $(\sigma_u, \sigma_v) \in C_{uv}$. Again, these make sure that at least one column of the form $(e, \cdot, \cdot)$ must be picked for every $e \in E$.

Finally, we will define the entries of the last $2|E||\Sigma|$ rows. To do so, let us recall that, in the YES case, we pick the columns $(u, \phi(u))$ for all $u \in V$ and $((u, v), \phi(u), \phi(v))$ for all $(u, v) \in E$. The goal of these remaining rows is to not only accept such a solution but also prevent any solution that picks columns $(u, \sigma_u), (v, \sigma_v)$ and $((u, v), \sigma'_u, \sigma'_v)$ where $\sigma_u \neq \sigma'_u$ or $\sigma_v \neq \sigma'_v$. In other words, these rows serve as a "consistency checker" of the solution. Specifically, the $|\Sigma|$ rows of the form $((u, v), \cdot, 0)$ will force σ_u and σ'_u to be equal whereas the $|\Sigma|$ rows of the form $((u, v), \cdot, 1)$ will force σ_v and σ'_v to be equal. For convenience, we will only define the entries for the $((u, v), \cdot, 0)$-rows; the $((u, v), \cdot, 1)$-rows can be defined similarly. Each $((u, v), \sigma, 0)$-row has only one non-zero entry within the first $|V||\Sigma|$ rows: the one in the (u, σ)-column. For the remaining columns, the entry in the $((u, v), \sigma, 0)$-row and the (e, σ_0, σ_1)-column is non-zero if and only if $e = (u, v)$ and $\sigma_0 = \sigma$.

It should be clear from the definition that our intended solution for the YES case is indeed a valid solution because, for each $((u, v), \phi(u), 0)$-row, the two non-zero entries from the columns $(u, \phi(u))$ and $((u, v), \phi(u), \phi(v))$ cancel each other out. On the other hand, for the NO case, the main observation is that, for each edge $(u, v) \in E$, if only one column of the form $(u, \cdot)$, one of the form $(v, \cdot)$ and one of the form $((u, v), \cdot, \cdot)$ are picked, then the assignment corresponding to the picked columns satisfy the constraint C_{uv}. In particular, it is easy to argue that, if we can pick $(1 + \varepsilon/3)(|V| + |E|)$ columns that sum up to $\mathbf{y}$, then all but ε fraction of all constraints fulfill the previous conditions, meaning that we can find an assignment that satisfies $1 - \varepsilon$ fraction of the constraints. Thus, we have also proved the soundness of the reduction.

Gap Amplification for GapMLD$_\gamma$. We have sketched the proof of the hardness of GapMLD$_\gamma$ for *some* constant $\gamma \geqslant 1$, assuming PIH. The next step is to amplify the gap and arrive at the

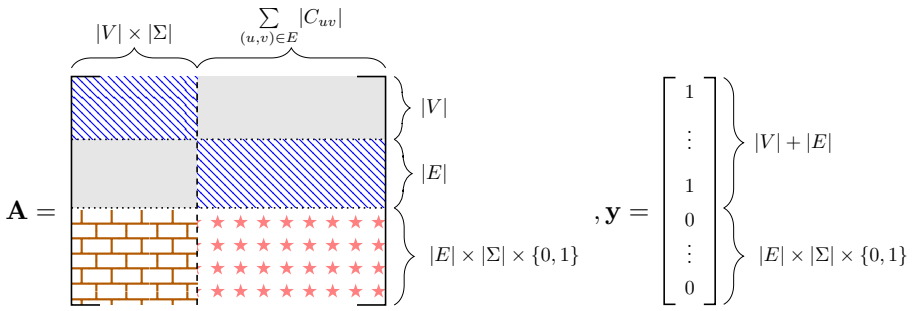

Figure 1 An illustration of $\mathbf{A}$ and $\mathbf{y}$. All entries in shaded areas are zero. Each *row* in the brick pattern area has one non-zero entry in that area, and each *column* in the star pattern area has two non-zero entries in the area. Finally, each column has one non-zero entry in the lines pattern areas.

hardness for GapMLD$_\gamma$ for *every* constant $\gamma \geqslant 1$. To do so, we define an operator $\oplus$ over every pair of instances of GapMLD$_\gamma$ with the following property: if two instances $(\mathbf{A}_1, \mathbf{y}_1, k_1)$ and $(\mathbf{A}_2, \mathbf{y}_2, k_2)$ are both YES instances, then $(\mathbf{A}, \mathbf{y}, k) := (\mathbf{A}_1, \mathbf{y}_1, k_1) \oplus (\mathbf{A}_2, \mathbf{y}_2, k_2)$ is a YES instance for GapMLD$_{\gamma'}$ where $\gamma' \approx \gamma^2$. On the other hand, if both $(\mathbf{A}_1, \mathbf{y}_1, k_1)$ and $(\mathbf{A}_2, \mathbf{y}_2, k_2)$ are NO instances, then $(\mathbf{A}, \mathbf{y}, k)$ is a NO instance for GapMLD$_{\gamma'}$. Hence, we can apply $\oplus$ repeatedly to the GapMLD$_\gamma$ instance from the previous step (with itself) and amplify the gap to be any arbitrarily large constant. The definition of $\oplus$, while simple, is slightly tedious to formalize and we defer it to the full version of this paper.

Reducing GapMLD to GapSNC. Now we introduce the *sparse nearest codeword problem* that we had briefly talked about. 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 $\mathbf{x} \in \mathcal{B}(\mathbf{0}, k)$ such that $\|\mathbf{A}'\mathbf{x} - \mathbf{y}'\|_0 \leqslant k$, and the NO case where for all $\mathbf{x}$ (in the entire space), we have $\|\mathbf{A}'\mathbf{x} - \mathbf{y}'\|_0 > \gamma k$. We highlight that the difference between k-NCP and GapSNC$_\gamma$ is that, in the YES case of the latter, we are promised that $\mathbf{x} \in \mathcal{B}(\mathbf{0}, k)$. 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

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} \\ \vdots \\ \mathbf{A} \\ \mathrm{Id} \end{bmatrix} \Big\} \gamma k + 1 \text{ copies}, \qquad \mathbf{y}' = \begin{bmatrix} \mathbf{y} \\ \vdots \\ \mathbf{y} \\ \mathbf{0} \end{bmatrix} \Big\} \gamma k + 1 \text{ copies}.$$

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.

Sparse Covering Codes. Before reducing GapSNC to GapMDP$_{1.99}$ we need to introduce in more detail the notion of *sparse covering codes* that we previously mentioned.

A sparse covering code (SCC) is a linear code $\mathbf{L}$ of block length h with minimum distance d admitting a ball $\mathcal{B}(\mathbf{s}, r)$ centered at $\mathbf{s}$ of radius $r < d$ and containing a large (i.e., about h^k, where $k = \Omega(d)$) number of codewords. Moreover, for our reduction to go through, we require not only $\mathbf{L}$ and $\mathbf{s}$, but also a linear transformation $\mathbf{T}$ used to index the codewords in $\mathcal{B}(\mathbf{s}, r)$, i.e., $\mathbf{T}(\mathcal{B}(\mathbf{s}, r) \cap \mathbf{L})$ needs to contains the ball of radius k centered at $\mathbf{0}$. Similar to

how Dumer et al. only managed to show the probabilistic existence of the center, we too cannot find an explicit $\mathbf{s}$ for the SCCs that we construct, but instead provide an efficiently samplable distribution such that, for any $\mathbf{x} \in \mathcal{B}(\mathbf{0}, k)$, the probability (over $\mathbf{s}$ sampled from the distribution) that $\mathbf{x} \in \mathbf{T}(\mathcal{B}(\mathbf{s}, r) \cap \mathbf{L})$ is non-negligible. This is what makes our reduction from GapSNC to $\text{GapMDP}_{1.99}$ 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 Hamming (aka sphere-packing) 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, because the balls of radius $\lceil \frac{d-1}{2} \rceil$ centered at the codewords do not intersect. Our main theorem regarding the existence of SCC is that any code that is "near" the Hamming bound is a sparse covering code with $r = \lceil \frac{d-1}{2} \rceil + k \approx 0.501d$. 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 do not sketch the proof of the theorem here, we note that the idea is to set $\mathbf{T}$ and the distribution over $\mathbf{s}$ in such a way that the probability that $\mathbf{x}$ lies in $\mathbf{T}(\mathcal{B}(\mathbf{s}, r) \cap \mathbf{L})$ is at least the probability that a random point in $\mathbb{F}_2^h$ is within distance $r - k = \lceil \frac{d-1}{2} \rceil$ of some codeword. The latter is non-negligible since $\mathbf{L}$ nears the Hamming 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 SCC, we want to cover a ball $\mathcal{B}(\mathbf{0}, k)$ rather than the whole space, and it is not hard to see that there are very large subsets Y such that no linear map $\mathbf{T}$ satisfies $\mathbf{T}(Y) \supseteq \mathcal{B}(\mathbf{0}, k)$. A simple example of this is when Y is a subspace of $\mathbb{F}_2^h$; in this case, even when Y is as large as $\exp(\text{poly}(h))$, no desired linear map $\mathbf{T}$ exists.

Reducing GapSNC_γ to $\text{GapMDP}_{1.99}$. Next, we prove the hardness of $\text{GapMDP}_{\gamma'}$ for all constant $\gamma' \in [1, 2)$, assuming PIH, using a gadget constructed from sparse covering codes.

Given an instance $(\mathbf{A}, \mathbf{y}, k)$ of GapSNC_γ for some $\gamma > 2$ and a SCC $(\mathbf{L}, \mathbf{T}, \mathbf{s})$ we build an instance $(\mathbf{B}, ak + br)$ of $\text{GapMDP}_{\gamma'}$ where $\gamma' < \frac{2\gamma}{2+\gamma}$, by following the DMS reduction (which was previously described; see (1)). If $\|\mathbf{Ax} - \mathbf{y}\|_0 \leqslant k$ for some $\mathbf{x} \in \mathcal{B}(\mathbf{0}, k)$, then consider $\mathbf{z}'$ such that $\mathbf{TLz}' = \mathbf{x}$; the existence of such a $\mathbf{z}'$ is guaranteed by the definition of SCC. Consider $\mathbf{z} = \mathbf{z}' \circ 1$, and note that $\|\mathbf{Bz}\|_0 = a\|\mathbf{Ax} - \mathbf{y}\|_0 + b\|\mathbf{Lz} - \mathbf{s}\|_0 \leqslant ak + br$. In other words, as in the DMS reduction, 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, similar to the DMS reduction, if we had that $\|\mathbf{Ax} - \mathbf{y}\|_0 > \gamma k$ for all $\mathbf{x}$, then $\|\mathbf{Bz}\|_0 > \gamma'(ak + br)$ for all $\mathbf{z}$. The parameterized intractability of $\text{GapMDP}_{1.99}$ is obtained by setting $\gamma = 400$ in the above reduction.

Gap Amplification for $\text{GapMDP}_{1.99}$. 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. We can use this proposition to reduce GapMDP_γ to GapMDP_{γ^2} for any $\gamma \geqslant 1$. In particular, we can obtain, for any constant γ, the intractability of GapMDP_γ starting from $\text{GapMDP}_{1.99}$ by just recursively tensoring the input code $\lceil \log_{1.99} \gamma \rceil$ times.

2.2 Parameterized Intractability of k-SVP from PIH

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 [5]. 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$\|\mathbf{Ax} - \mathbf{y}\|_p^p$ is at most k. The result of Arora et al. [5] 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 $\text{GapNVP}_{p,\eta}$ for any constant $\eta \geqslant 1$ the gap version of k-NVP in the ℓ_p norm. Recall that in $\text{GapNVP}_{p,\eta}$ 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 case when there exists $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{Ax} - \mathbf{y}\|_p^p \leqslant k$ from the case when for all $\mathbf{x} \in \mathbb{Z}^m$ we have that $\|\mathbf{Ax} - \mathbf{y}\|_p^p > \eta k$. As it turns out, our reduction from $\text{Gap2CSP}_\varepsilon$ to GapSNC (with arbitrary constant gap), having GapMLD_γ and GapMLD as intermediate steps, can be translated to show the constant inapproximability of GapNVP_p (under PIH) in a straightforward manner. We will not elaborate on this part of the proof any further here and defer the detailed proof to the full version of this paper.

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 $\text{GapSVP}_{p,\gamma}$ for some constant $\gamma(p) \geqslant 1$ 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 $\|\mathbf{Bx}\|_p^p \leqslant k$ from the case when for all $\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ we have that $\|\mathbf{Bx}\|_p^p > \gamma k$. Let $\gamma^* := \frac{2^p}{2^{p-1}+1}$. Following Khot's reduction, we obtain the inapproximability of $\text{GapSVP}_{p,\gamma^*}$ (under PIH). To obtain inapproximability of GapSVP_2 for all constant ratios, we use the tensor product of lattices; the argument needed here is slightly more subtle than the similar step in MDP because, unlike distances of codes, the ℓ_2 norm of the shortest vector of the tensor product of two lattices is not necessarily equal to the product of the ℓ_2 norm of the shortest vector of each lattice. Fortunately, Khot's construction is tailored so that the resulting lattice is "well-behaved" under tensoring [27, 26], and gap amplification is indeed possible for such instances.

We remark here that, for the (non-parameterized) inapproximability of SVP, the techniques of [27, 26] allow one to successfully amplify gaps for ℓ_p norm where $p \neq 2$ as well. Unfortunately, this does not work in our settings, as it requires the distance k to be dependent on nm which is not possible for us since k is the parameter of the problem.

3 Discussion and Open Questions

While our results give an evidence of intractability of k-MDP and k-SVP, there are still many questions that remain open. First and foremost, it is still open whether the hardness of both problems can be based on more standard assumptions, such as ETH or W[1] $\neq$ FPT. On this front, we would like to note that the only reason we need PIH is to arrive at the inapproximability of the non-homogeneous variants of the problems, which is needed for us even if we want to only rule out exact FPT algorithms for k-MDP and k-SVP. Hence, if one could prove the hardness of approximation for these problems under weaker assumptions, then the inapproximability of k-MDP and k-SVP would still follow.

Another obvious question is whether k-SVP in the ℓ_1 norm is in FPT. Khot's reduction unfortunately does not work for ℓ_1; indeed, in [26], the hardness of approximating SVP in the ℓ_1 norm is shown by embedding SVP instances in ℓ_2 to instances in ℓ_1 using an earlier result of Regev and Rosen [43]. This 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 [33], whose reduction produces irrational bases.

An additional question regarding k-SVP is whether we can prove inapproximability for *every* constant factor for $p \neq 2$. As described earlier, the gap amplification techniques of [27, 26] 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 be interesting to come up with different gap amplification techniques that also work for our settings.

Furthermore, k-MDP 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 $p > 2$. The issue here is that, in our proof of existence of Sparse Covering Codes, 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 from PIH to 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 DMS reduction [22] that we employ can be derandomized, as long as one has a deterministic construction for Locally Dense Codes [11, 36]. In our settings, if one can deterministically construct Sparse Covering Codes, we would also get a deterministic reduction for k-MDP.

Finally, another interesting research direction is to prove more concrete running time lower bounds for k-MDP and k-SVP. For instance, k-MDP can be trivially solved (exactly) in $N^{O(k)}$ time, where $N = nm$ is the input size. On the other hand, while not stated explicitly above, our proof implies that k-MDP cannot be solved (or even approximated) in time $N^{o(k^c)}$ for some small constant $c > 0$, assuming Gap-ETH. Would it be possible to improve this running time lower bound to the tight $N^{o(k)}$? Similar questions also apply to k-SVP.

References

1. Divesh Aggarwal and Noah Stephens-Davidowitz. (gap/s)eth hardness of SVP. *CoRR*, abs/1712.00942, 2017. arXiv:1712.00942.
2. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108, 1996. doi:10.1145/237814.237838.
3. Miklós Ajtai. The shortest vector problem in ℓ_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, pages 10–19, 1998. doi:10.1145/276698.276705.
4. Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293, 1997. doi:10.1145/258533.258604.
5. Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. doi:10.1006/jcss.1997.1472.
6. 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. doi:10.1109/TIT.2014.2340869.
7. Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. On the quantitative hardness of CVP. In *FOCS*, pages 13–24, 2017. doi:10.1109/FOCS.2017.11.
8. 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. doi:10.1109/TIT.1978.1055873.
9. Arnab Bhattacharyya, Ameet Gadekar, Suprovat Ghoshal, and Rishi Saket. On the hardness of learning sparse parities. In *ESA*, pages 11:1–11:17, 2016. doi:10.4230/LIPIcs.ESA.2016.11.
10. 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. doi:10.1006/jcss.1999.1649.
11. Qi Cheng and Daqing Wan. A deterministic reduction for the gap minimum distance problem. *IEEE Trans. Information Theory*, 58(11):6935–6941, 2012. doi:10.1109/TIT.2012.2209198.
12. 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.
13. Marek Cygan, Fedor V. Fomin, Danny Hermelin, and Magnus Wahlström. Randomization in parameterized complexity (dagstuhl seminar 17041). *Dagstuhl Reports*, 7(1):103–128, 2017. doi:10.4230/DagRep.7.1.103.
14. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
15. Erik D. Demaine, Gregory Gutin, Dániel Marx, and Ulrike Stege. 07281 open problems – structure theory and FPT algorithmcs for graphs, digraphs and hypergraphs. In Erik D. Demaine, Gregory Z. Gutin, Dániel Marx, and Ulrike Stege, editors, *Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs, 08.07. - 13.07.2007*, volume 07281 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2007. URL: http://drops.dagstuhl.de/opus/volltexte/2007/1254.
16. Irit Dinur. Approximating SVP_∞ to within almost-polynomial factors is NP-hard. *Theor. Comput. Sci.*, 285(1):55–71, 2002. doi:10.1016/S0304-3975(01)00290-0.
17. Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *ECCC*, 23:128, 2016. URL: http://eccc.hpi-web.de/report/2016/128.

18 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. doi:10.1007/s00493-003-0019-y.

19 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. doi:10.1007/978-1-4612-0515-9.

20 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

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

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

23 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. doi:10.4230/DagRep.2.6.26.

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

25 Petr A. Golovach, Jan Kratochvíl, and Ondrej Suchý. Parameterized complexity of generalized domination problems. *Discrete Applied Mathematics*, 160(6):780–792, 2012. doi:10.1016/j.dam.2010.11.012.

26 Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *STOC*, pages 469–477, 2007. doi:10.1145/1250790.1250859.

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

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

29 Hendrik Willem Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983. doi:10.1287/moor.8.4.538.

30 Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Parameterized complexity and approximability of directed odd cycle transversal. *CoRR*, abs/1704.04249, 2017. arXiv:1704.04249.

31 Ruhollah Majdoddin. Parameterized complexity of CSP for infinite constraint languages. *CoRR*, abs/1706.10153, 2017. arXiv:1706.10153.

32 Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense CSPs. *CoRR*, abs/1607.02986, 2016. URL: http://arxiv.org/abs/1607.02986, arXiv:1607.02986.

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

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

35 Daniele Micciancio. Inapproximability of the shortest vector problem: Toward a deterministic reduction. *Theory of Computing*, 8(1):487–512, 2012. doi:10.4086/toc.2012.v008a022.

36 Daniele Micciancio. Locally dense codes. In *CCC*, pages 90–97. IEEE Computer Society, 2014. doi:10.1109/CCC.2014.17.

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

38 Phong Q. Nguyen and Brigitte Vallée, editors. *The LLL Algorithm - Survey and Applications*. Information Security and Cryptography. Springer, 2010. doi:10.1007/978-3-642-02295-1.

39 Oded Regev. New lattice based cryptographic constructions. In Lawrence L. Larmore and Michel X. Goemans, editors, *STOC*, pages 407–416. ACM, 2003. doi:10.1145/780542.780603.

40 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005. doi:10.1145/1060590.1060603.

41 Oded Regev. Lattice-based cryptography. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 131–141. Springer, 2006. doi:10.1007/11818175_8.

42 Oded Regev. The learning with errors problem (invited survey). In *CCC*, pages 191–204, 2010. doi:10.1109/CCC.2010.26.

43 Oded Regev and Ricky Rosen. Lattice problems and norm embeddings. In *STOC*, pages 447–456, 2006. doi:10.1145/1132516.1132581.

44 Jacques Stern. Approximating the number of error locations within a constant ratio is NP-complete. In Gérard D. Cohen, Teo Mora, and Oscar Moreno, editors, *AAECC*, volume 673 of *Lecture Notes in Computer Science*, pages 325–331. Springer, 1993. doi:10.1007/3-540-56686-4_54.

45 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.

46 Alexander Vardy. Algorithmic complexity in coding theory and the minimum distance problem. In *STOC*, pages 92–109, 1997. doi:10.1145/258533.258559.

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