The Code Distortion Problem

Huck Bennett
University of Colorado Boulder, CO, USA

Matthew Fox
University of Colorado Boulder, CO, USA

Bryant Morrell
University of Colorado Boulder, CO, USA

Abstract

Two linear error-correcting codes $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$ are called *linearly equivalent* if there is a linear isometry mapping $\mathcal{C}_1$ to $\mathcal{C}_2$. In this work, we generalize the notion of linear equivalence and study the minimum distortion $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$ of a linear mapping between codes $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$, which *quantifies* how similar $\mathcal{C}_1$ and $\mathcal{C}_2$ are. We introduce and study the Code Distortion Problem (CDP), which asks to find a minimum distortion mapping between two input codes $\mathcal{C}_1$ and $\mathcal{C}_2$. CDP generalizes the Linear Code Equivalence Problem (LCE), which is essentially the special case of CDP where $\mathcal{D}(\mathcal{C}_1, C_2) = 1$ and which is well-studied because of its role in cryptography.

We prove that (decisional) CDP is NP-hard to approximate to within any constant factor, and that it is in Σ_2^{P}. We also give a single-exponential-time k^2-approximation algorithm for CDP, where k is the dimension of the input codes. Furthermore, we give a single-exponential-time $\left(\frac{2k+1}{3}\right)^2$-approximation algorithm for a natural special case of CDP, and we show that our analysis is tight in this case.

We use techniques from analogous work on the Lattice Distortion Problem (LDP) by Bennett, Dadush, and Stephens-Davidowitz (ESA, 2016). We also introduce or study a number of additional concepts that might be of independent interest. These include an adaptation of the celebrated reduction of Goldreich, Micciancio, Safra, and Seifert (IPL, 1999) from the Shortest Vector Problem (SVP) to the Closest Vector Problem (CVP) on lattices to the analogous problems on codes; successive minima bases for codes; and the matrix $0 \to 0$ "norm" on subspaces.

2012 ACM Subject Classification Theory of computation → Error-correcting codes; Theory of computation → Random projections and metric embeddings; Theory of computation → Problems, reductions and completeness; Theory of computation → Approximation algorithms analysis

Keywords and phrases Code Distortion, Code Equivalence, Error-Correcting Codes, Metric Embeddings, Approximation Algorithms, Hardness of Approximation

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.24

Category APPROX

Funding *Huck Bennett*: Supported by NSF Award CCF-2432132.
Bryant Morrell: Supported by NSF Award CCF-2432132.

Acknowledgements We thank Alexander Golovnev and Noah Stephens-Davidowitz for useful conversations [20], and in particular for allowing us to include Theorem 22 about the coNP-hardness of approximating the matrix $0 \to 0$ norm on subspaces in this work. We also thank the anonymous APPROX reviewers for helpful comments, and in particular for identifying an incorrect inequality (which is now fixed).

1 Introduction

A *linear error-correcting code* (or simply, *code*) is a linear subspace $\mathcal{C} \subseteq \mathbb{F}_q^n$. Although best known for their use in robust communication, codes also appear prominently in algorithms, computational complexity, and cryptography. Notably, several of the cryptosystems appearing

in late rounds or standardized by the National Institute of Standards and Technology's (NIST) post-quantum cryptography standardization process are based on the intractability of certain computational problems on codes [1, 12, 17].

One central problem on codes of cryptographic interest is the *Linear Code Equivalence Problem* (LCE), in which, given generator matrices of two codes $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$ as input, the goal is to decide whether $\mathcal{C}_1$ and $\mathcal{C}_2$ are *linearly equivalent*, i.e., whether one can convert $\mathcal{C}_1$ into $\mathcal{C}_2$ simply by permuting coordinates and scaling coordinates by non-zero values. In particular, the LESS cryptosystem [12, 5] crucially relies on the hardness of LCE. It is currently a second-round candidate in NIST's "Additional Digital Signature Schemes" standardization process [26]. Furthermore, a large body of work has studied LCE and its variants from an algorithmic and complexity-theoretic standpoint. See, e.g., [21, 28, 30, 4, 10, 6, 15, 9, 27, 7].

A linear isometry (i.e., distance-preserving map) on $\mathbb{F}_q^n$ corresponds to a matrix $M \in \mathbb{F}_q^{n\times n}$ such that for all $\boldsymbol{x} \in \mathbb{F}_q^n$, $\|M\boldsymbol{x}\|_0 = \|\boldsymbol{x}\|_0$, where $\|\cdot\|_0$ denotes the Hamming weight of a vector. It is not hard to check that a matrix M satisfying this property can be written as the product $M = DP$ of a non-singular diagonal matrix D (which corresponds to scaling) and a permutation matrix P (such matrices M are called *monomial matrices*). In other words, the two operations in the definition of LCE – permuting and scaling coordinates by non-zero values – exactly characterize the linear isometries on $\mathbb{F}_q^n$.

So, LCE asks whether there is a (perfectly) distance-preserving linear map between two input codes $\mathcal{C}_1, \mathcal{C}_2$. However, it is natural to ask a refined version of this question:

Is there a linear map T such that $T(\mathcal{C}_1) = \mathcal{C}_2$ and T *approximately* preserves distances?

We make this precise and quantitative by asking what the minimum *distortion* of such a map T from $\mathcal{C}_1$ to $\mathcal{C}_2$ is. We define the distortion of a map T with $T(\mathcal{C}_1) = \mathcal{C}_2$ as

$$\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) := \Big(\max_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\Big) \Big/ \Big(\min_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\Big) \,, \tag{1}$$

and we define $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) := \min_T \mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2)$, where the minimum is taken over all linear maps T such that $T(\mathcal{C}_1) = \mathcal{C}_2$. Informally, $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2)$ is the ratio of the most T blows up distances and the most it shrinks distances. We note that $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2)$ is simply the distortion of the map T in the standard sense of metric embeddings, where the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are the ambient spaces and each is equipped with the Hamming metric. (See, e.g., the lecture notes of Matoušek [22] for background on metric embeddings.) Moreover, $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$ is the minimum distortion over all such linear maps T, and it therefore quantifies how similar $\mathcal{C}_1$ and $\mathcal{C}_2$ are. Intuitively, linearly equivalent codes are essentially the same, and codes $\mathcal{C}_1, \mathcal{C}_2$ with $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \gg 1$ are quite different. One can show that $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \geq 1$ (see Lemma 15), and that equality holds if and only if there exist linearly equivalent "unary scalings" of $\mathcal{C}_1$ and $\mathcal{C}_2$ (see Proposition 19).

Recall that a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ of dimension k, called an $[n, k]_q$ code, is often represented by a generator matrix (basis) $G \in \mathbb{F}_q^{n\times k}$.[1] I.e., the code generated by G is $\mathcal{C}(G) := \{G\boldsymbol{x} : \boldsymbol{x} \in \mathbb{F}_q^k\}$. And, one can represent an arbitrary linear map between two $[n, k]_q$ codes in terms of generator matrices $G_1 := (\boldsymbol{g}_1^{(1)}, \ldots, \boldsymbol{g}_k^{(1)}) \in \mathbb{F}_q^{n\times k}$ and $G_2 := (\boldsymbol{g}_1^{(2)}, \ldots, \boldsymbol{g}_k^{(2)}) \in \mathbb{F}_q^{n\times k}$ of the respective codes by $\boldsymbol{g}_i^{(1)} \mapsto \boldsymbol{g}_i^{(2)}$.

[1] In this work we use *column* bases for codes, although row bases are often used in the literature.

For example, consider the generator matrices

$$G_1 := \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \in \mathbb{F}_2^{4\times 2}\ , \qquad G_2 := \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \in \mathbb{F}_2^{4\times 2}\ , \qquad G_2' := \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{4\times 2}\ .$$

Notice that $\mathcal{C}_1 := \mathcal{C}(G_1)$ and $\mathcal{C}_2 := \mathcal{C}(G_2)$ are not linearly equivalent, but that $\mathcal{C}_2 = \mathcal{C}(G_2')$, i.e., G_2 and G_2' are different generator matrices of the same code. Letting T and T' be linear maps such that $T(G_1) = G_2$ and $T'(G_1) = G_2'$, it is straightforward to check that

$$\begin{aligned}\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) &= \Big(\max_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\Big) \Big/ \Big(\min_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\Big) \\ &= \Big(\frac{\|(1,1,1,0)\|_0}{\|(1,0,0,0)\|_0}\Big) \Big/ \Big(\frac{\|(1,0,0,1)\|_0}{\|(1,1,1,1)\|_0}\Big) \\ &= \frac{3}{1} \cdot \Big(\frac{1}{2}\Big)^{-1} = 6\ ,\end{aligned}$$

and

$$\begin{aligned}\mathcal{D}_{T'}(\mathcal{C}_1, \mathcal{C}_2) &= \Big(\max_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}} \frac{\|T'\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\Big) \Big/ \Big(\min_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}} \frac{\|T'\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\Big) \\ &= \Big(\frac{\|(1,0,0,1)\|_0}{\|(1,0,0,0)\|_0}\Big) \Big/ \Big(\frac{\|(1,1,1,0)\|_0}{\|(1,1,1,1)\|_0}\Big) \\ &= \frac{2}{1} \cdot \Big(\frac{3}{4}\Big)^{-1} = \frac{8}{3}\ .\end{aligned}$$

In other words, T' is a substantially lower distortion mapping between $\mathcal{C}_1$ and $\mathcal{C}_2$ than T is. Part of the reason for this is that G_2' is a better basis of $\mathcal{C}_2$ than G_2 is in a sense that we will make precise in the sequel.

In this work, we study code distortion from a computational standpoint. Let $\gamma \geq 1$ be an approximation factor. Specifically, we study the γ-approximate Code Distortion Problem (γ-CDP), which asks to compute a linear mapping T between two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) \leq \gamma \cdot \mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$. We also study its decision version, γ-GapCDP, in which the goal is to decide whether $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \leq D$ or $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) > \gamma D$ for a given input distortion threshold value $D \geq 1$. When emphasizing the finite field $\mathbb{F}_q$ that the input codes are over, we add a subscript q to the problems.

1.1 Our Results

We study CDP from both an algorithmic and complexity-theoretic perspective. We first prove hardness of approximation for GapCDP with (fixed) $D > 1$.

▶ **Theorem 1.** *For any constant $\gamma \geq 1$, any constant $D > 1$, and any prime power $q \leq \mathrm{poly}(n)$, γ-GapCDP$_q$ with distortion threshold D is* NP*-hard under deterministic Cook reductions.*

As we formalize in Proposition 19, GapCDP with distortion threshold $D = 1$ is equivalent to LCE up to scaling. From this it follows that GapCDP with $D = 1$ is in NP, but, because of a result of Petrank and Roth [28] showing that LCE is in coAM, it is not NP-hard unless the polynomial hierarchy collapses. In contrast, Theorem 1 asserts that if D is just slightly larger than 1 then GapCDP is NP-hard even to approximate to within any constant factor γ.

For example, taking $D = 1.01$ and $\gamma = 1000$, Theorem 1 shows that it is NP-hard to decide whether $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \leq 1.01$ or $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) > 1010$ when one of these is promised to hold. The result holds for codes $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$ for any field $\mathbb{F}_q$ with $q \leq \text{poly}(n)$.

We further show in Theorem 24 that GapCDP is in the complexity class Σ_2^{P}. Interestingly, it is not clear that GapCDP is contained in NP. This stands in contrast both to LCE and the Lattice Distortion Problem (LDP), the analogous problem to GapCDP on lattices, which [8] noted is in NP.[2] The main difference between LDP and GapCDP is that the $2 \to 2$ norm $\|T\|$ of a real-valued matrix T is efficient to compute (indeed, $\|T\|$ is simply the largest singular value of T), whereas the problem of computing the $0 \to 0$ "norm" of a linear operator T mapping one code to another is coNP-hard even to approximate to within any constant factor! See Corollary 23.[3]

We next give algorithms for CDP. In Lemma 27, we give and analyze a brute force, exact algorithm for CDP_q on $[n,k]_q$ codes, which runs in $O^*(q^{k(k+1)}) \approx q^{k^2}$ time – essentially the amount of time it takes to enumerate all generator matrices of a given $[n,k]_q$ code.[4] It is unclear how to get a faster exact algorithm, and so we turn to giving approximation algorithms. We first give a single-exponential-time, k^2-approximation algorithm for CDP on $[n,k]_q$ codes.

▶ **Theorem 2.** *For any prime power q, there is an algorithm for k^2-CDP on $[n,k]_q$ codes that runs in $O^*(q^k)$ time and $\text{poly}(n, \log q)$ space.*

We contrast the approximation factor of k^2 in Theorem 2 with the trivial bound bound of $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) \leq n^2$, which holds for any linear map T such that $T(\mathcal{C}_1) = \mathcal{C}_2$ for $[n,k]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$; see Lemma 15. Since $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) \geq 1$ also always holds, this implies a trivial n^2-approximation algorithm for GapCDP. We do not know of a better bound (even a non-constructive one), although we show an upper bound of n when allowing for *non-linear* maps between the codes; see Lemma 17. However, even in this case it is not clear that *finding* such a map would be at all efficient, and we note that Theorem 2 holds for approximate *search* CDP.

For an $[n,k]_q$ code $\mathcal{C}$ and $i \in [k]$, we define the *ith successive minimum* $\lambda_i(\mathcal{C})$ of $\mathcal{C}$ to be the minimum value $r \in \{1, \ldots, n\}$ such that $\mathcal{C}$ contains i linearly independent vectors of Hamming weight at most r. In particular, $\lambda_1(\mathcal{C})$ is the minimum distance of $\mathcal{C}$. We get a strengthening of Theorem 2 when the input codes are binary and when all of the successive minima of both of the input codes are the same.

▶ **Theorem 3.** *There is an algorithm for $\left(\frac{2k+1}{3}\right)^2$-CDP on $[n,k]_2$ codes $\mathcal{C}_1, \mathcal{C}_2$ satisfying $\lambda_1(\mathcal{C}_1) = \cdots = \lambda_k(\mathcal{C}_1) = \lambda_1(\mathcal{C}_2) = \cdots = \lambda_k(\mathcal{C}_2)$ that runs in $O^*(2^k)$ time and $\text{poly}(n)$ space.*

In fact, the *algorithm* used to prove both Theorems 2 and 3 is the same, but all of the codes' successive minima being the same allows for a tighter analysis of its approximation factor. In fact, the analysis of Theorem 3 is tight. However, due to space constraints, we defer the proof to the full version of our paper. Additionally, we note that although we give Theorem 3 only for $\mathbb{F}_2$, one can show that it extends to $\mathbb{F}_q$ for arbitrary prime powers q with a more complicated proof; see [25].

---

[2] Recall that a *lattice* is the analog of a code over the integers or real numbers. Specifically, the lattice generated by a basis $B \in \mathbb{R}^{n \times k}$ with full column rank is $\mathcal{L}(B) := \{B\boldsymbol{z} : \boldsymbol{z} \in \mathbb{Z}^k\}$.

[3] The $0 \to 0$ norm of a linear operator acting on all of $\mathbb{F}_q^n$ (and not restricted restricted to a code/subspace) is also efficiently computable; see Proposition 20.

[4] In this paper, we use $O^*(\cdot)$ to suppress polynomial factors in n and $\log q$.

1.2 Overview of Techniques

At a high level, the proofs of Theorems 1 and 2 follow along similar lines to those used to show upper and lower bounds for the Lattice Distortion Problem (LDP) in [8]. However, they also require a number of additional concepts, which may be of independent interest. The proof of Theorem 3 goes along similar lines to the proof of Theorem 2, but it is substantially more nuanced.

Approximation algorithms

We prove a constructive upper bound using what we call *successive minima bases* for codes. We define a successive minima basis of an $[n,k]_q$ code $\mathcal{C}$ to be a generator matrix $G := (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_k) \in \mathbb{F}_q^{n\times k}$ such that $\|\boldsymbol{g}_i\|_0 = \lambda_i(\mathcal{C})$, i.e., the columns of G are linearly independent and their Hamming weights achieve the successive minima of $\mathcal{C}$. Every code has a successive minima basis (essentially because any k linearly independent vectors in $\mathcal{C}$ form a basis of $\mathcal{C}$) and they can be computed in roughly q^k time; see Lemma 6. In contrast, not every lattice $\mathcal{L}$ has a successive minima basis.[5] Although successive minima are widely used in the study of lattices, to the best of our knowledge they have not previously been explicitly used for codes.

Our approximation algorithm for CDP that leads to Theorems 2 and 3 is simply to compute successive minima bases $G_1 := (\boldsymbol{g}_1^{(1)}, \ldots, \boldsymbol{g}_k^{(1)}) \in \mathbb{F}_q^{n\times k}$ and $G_2 := (\boldsymbol{g}_1^{(2)}, \ldots, \boldsymbol{g}_k^{(2)}) \in \mathbb{F}_q^{n\times k}$ of the respective input $[n,k]_q$ codes $\mathcal{C}_1$ and $\mathcal{C}_2$, and to take the mapping $T : \boldsymbol{g}_i^{(1)} \mapsto \boldsymbol{g}_i^{(2)}$ that they induce.

We show that such a mapping T is a k^2-approximation, proving Theorem 2, as follows. For $[n,k]_q$ codes $\mathcal{C}_1$ and $\mathcal{C}_2$, define $M(\mathcal{C}_1,\mathcal{C}_2) := \max_{i\in[k]} \lambda_i(\mathcal{C}_1)/\lambda_i(\mathcal{C}_2)$. (An analogous quantity for lattices appeared in [8].) We show that the mapping T defined above in terms of successive minima bases satisfies $\mathcal{D}_T(\mathcal{C}_1,\mathcal{C}_2) \le k^2 \cdot M(\mathcal{C}_1,\mathcal{C}_2)M(\mathcal{C}_2,\mathcal{C}_1)$. Furthermore, we show that *any* linear map T' such that $T'\mathcal{C}_1 = \mathcal{C}_2$ must satisfy $\mathcal{D}_{T'}(\mathcal{C}_1,\mathcal{C}_2) \ge M(\mathcal{C}_1,\mathcal{C}_2)M(\mathcal{C}_2,\mathcal{C}_1)$, and therefore $\mathcal{D}(\mathcal{C}_1,\mathcal{C}_2) \ge M(\mathcal{C}_1,\mathcal{C}_2)M(\mathcal{C}_2,\mathcal{C}_1)$. Combining these two bounds, we have that

$$M(\mathcal{C}_1,\mathcal{C}_2)M(\mathcal{C}_2,\mathcal{C}_1) \le \mathcal{D}(\mathcal{C}_1,\mathcal{C}_2) \le \mathcal{D}_T(\mathcal{C}_1,\mathcal{C}_2) \le k^2 \cdot M(\mathcal{C}_1,\mathcal{C}_2)M(\mathcal{C}_2,\mathcal{C}_1)\ ,$$

and therefore that T is k^2-approximately optimal, i.e., that $\mathcal{D}_T(\mathcal{C}_1,\mathcal{C}_2) \le k^2 \cdot \mathcal{D}(\mathcal{C}_1,\mathcal{C}_2)$.

Showing the refined result in Theorem 3 for codes over $\mathbb{F}_2$ all of whose successive minima are the same requires more work. We assume that G_1 and G_2 are successive minima bases of the input $[n,k]_2$ codes (so that each vector in G_1 and G_2 has Hamming weight $\lambda := \lambda_1(\mathcal{C}_1) = \cdots = \lambda_k(\mathcal{C}_1) = \lambda_1(\mathcal{C}_2) = \cdots = \lambda_k(\mathcal{C}_2)$), and we let T be a mapping such that $TG_1 = G_2$. We then analyze

$$\mathcal{D}_T(\mathcal{C}_1,\mathcal{C}_2) := \max_{\boldsymbol{a}\in\mathbb{F}_2^k}\left(\frac{\|G_2\boldsymbol{a}\|_0}{\|G_1\boldsymbol{a}\|_0}\right)\max_{\boldsymbol{b}\in\mathbb{F}_2^k}\left(\frac{\|G_1\boldsymbol{b}\|_0}{\|G_2\boldsymbol{b}\|_0}\right) = \max_{\boldsymbol{a},\boldsymbol{b}\in\mathbb{F}_2^k}\left(\frac{\|G_1\boldsymbol{b}\|_0}{\|G_1\boldsymbol{a}\|_0}\frac{\|G_2\boldsymbol{a}\|_0}{\|G_2\boldsymbol{b}\|_0}\right)$$

[5] As an explicit example, consider the lattice $\mathcal{L} = \mathcal{L}(B) \subset \mathbb{R}^5$ generated by the basis

$$B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_5) := \begin{pmatrix} 1 & 0 & 0 & 0 & 1/2 \\ 0 & 1 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 \end{pmatrix}\ .$$

One can check that $\boldsymbol{e}_i \in \mathcal{L}$ for $i = 1, \ldots, 5$, and that every vector in $\mathcal{L}$ has norm at least 1. So, $\lambda_1(\mathcal{L}) = \cdots = \lambda_5(\mathcal{L}) = 1$. However, any basis of $\mathcal{L}$ must contain $\pm\boldsymbol{b}_5$, which has norm greater than 1.

We upper bound the quantity $\frac{\|G_2\boldsymbol{a}\|_0\|G_1\boldsymbol{b}\|_0}{\|G_1\boldsymbol{a}\|_0\|G_2\boldsymbol{b}\|_0}$ on the right-hand side for non-zero $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_q^k$ in cases. In each case, we make use of the fact that for non-zero $\boldsymbol{x}$, $\lambda \leq \|G_1\boldsymbol{x}\|_0 \leq \lambda\|\boldsymbol{x}\|_0$, where the upper bound follows from the triangle inequality. Similar bounds hold for G_2.

If $\|\boldsymbol{a}\|_0 + \|\boldsymbol{b}\|_0 \leq \left(\frac{2k+1}{3}\right)$, then by the AM-GM inequality $\|\boldsymbol{a}\|_0\|\boldsymbol{b}\|_0 \leq \left(\frac{2k+1}{3}\right)^2\lambda^2$. So,

$$\frac{\|G_2\boldsymbol{a}\|_0\|G_1\boldsymbol{b}\|_0}{\|G_1\boldsymbol{a}\|_0\|G_2\boldsymbol{b}\|_0} \leq \frac{\left(\frac{2k+1}{3}\right)^2\lambda^2}{\lambda^2} = \left(\frac{2k+1}{3}\right)^2 .$$

On the other hand, if $\|\boldsymbol{a}\|_0 + \|\boldsymbol{b}\|_0 > \left(\frac{2k+1}{3}\right)$ then the supports of $\boldsymbol{a}$ and $\boldsymbol{b}$ have large intersection, which leads to a similar upper bound on $\frac{\|G_2\boldsymbol{a}\|_0\|G_1\boldsymbol{b}\|_0}{\|G_1\boldsymbol{a}\|_0\|G_2\boldsymbol{b}\|_0}$.

Hardness of approximation

To show hardness of approximation for GapCDP, we roughly follow the approach used in [8] for showing hardness of approximation for LDP. We give a pair of reductions, the first of which reduces from the Minimum Distance Problem (GapMDP) to a promise variant of the Nearest Codeword Problem (GapNCP). This first reduction is an adaptation of the reduction from the Shortest Vector Problem (GapSVP) to the Closest Vector Problem (GapCVP) on lattices from [19], and it is likely of independent interest.[6]

The second reduction is from GapNCP to GapCDP. We recall that the input to GapNCP is a pair consisting of a generator matrix $G \in \mathbb{F}_q^{n\times k}$ of a code $\mathcal{C}$ and a target vector $\boldsymbol{t} \in \mathbb{F}_q^n$, and the goal is to decide whether the minimum distance between $\boldsymbol{t}$ and a codeword in $\mathcal{C}$ (i.e., $\min_{\boldsymbol{x}\in\mathcal{C}} \|\boldsymbol{t} - \boldsymbol{x}\|_0$) is at most some threshold. The idea behind our hardness reduction is to construct the following pair of generator matrices, corresponding to an instances of CDP:

$$G_1 := \begin{pmatrix} G & \mathbf{0} \\ 0 & \mathbf{1}_r \end{pmatrix} , \qquad G_2 := \begin{pmatrix} G & -\boldsymbol{t} \\ 0 & \mathbf{1}_r \end{pmatrix} . \tag{2}$$

Here $r \in \mathbb{Z}^+$. Let $\mathcal{C}_1 := \mathcal{C}(G_1)$ and $\mathcal{C}_2 := \mathcal{C}(G_2)$.

If $\min_{\boldsymbol{c}\in\mathcal{C}} \|\boldsymbol{c} - \boldsymbol{t}\|_0$ is small then $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$ is small. If on the other hand $\min_{\boldsymbol{c}\in\mathcal{C}} \|\boldsymbol{c} - \boldsymbol{t}\|_0$ is large, then we would *hope* is that $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$ is large. However, this does not clearly work directly, and we need to modify the reduction. (The attempted reduction sketched here in Equation (2) is the natural analog of the analogous hardness reduction in [8] for LDP; the modified reduction is slightly more involved.)

Because GapMDP is known to be NP-hard to approximate to within any constant [16, 13], this pair of reductions implies that CDP is also NP-hard to approximate to within any constant.

[6] We note that GapMDP and GapNCP are the coding problems analogous to the lattice problems GapSVP and GapCVP, respectively. Furthermore, we note that while [2] showed hardness of approximation for GapNCP, it did not show hardness of the variant that we use in the present work, which has an added promise in the NO case. Finally, we note that our reduction has the advantage of being directly from GapMDP, and of being dimension- and approximation-preserving.

The complexity of GapCDP and its relationship to the $0 \to 0$ norm

We note that the complexity of GapCDP seems closely related to the complexity of computing what we call the matrix $0 \to 0$ *norm restricted to a subspace*. The matrix $0 \to 0$ norm[7]

$$\|T\|_{0\to 0} := \max_{\boldsymbol{x}\in\mathbb{F}_q^n\setminus\{\mathbf{0}\}} \frac{\|M\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} .$$

of a matrix $T \in \mathbb{F}_q^{m\times n}$ is also efficiently computable; see Proposition 20. However, the matrix $0 \to 0$ norm restricted to a subspace $\mathcal{C} \subseteq \mathbb{F}_q^n$ (i.e., code $\mathcal{C}$), defined as,

$$\|T_{|\mathcal{C}}\|_{0\to 0} := \max_{\boldsymbol{x}\in\mathcal{C}\setminus\{\mathbf{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} .$$

is coNP-hard even to approximate; see Corollary 23. We note that for a linear map T with $T\mathcal{C}_1 = \mathcal{C}_2$ for codes $\mathcal{C}_1, \mathcal{C}_2$, $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) = \|T_{|\mathcal{C}_1}\|_{0\to 0} \cdot \|(T^{-1})_{|\mathcal{C}_2}\|_{0\to 0}$. The most obvious approach to showing that GapCDP is in NP would be to use a low-distortion map T between the input codes as a witness, and to have the verifier check that $\|T_{|\mathcal{C}_1}\|_{0\to 0} \cdot \|(T^{-1})_{|\mathcal{C}_2}\|_{0\to 0} \le D$. However, because computing the matrix $0 \to 0$ norm restricted to a subspace is coNP-hard, this approach does not work. (We note that this coNP-hardness does not obviously translate to GapCDP.) As a result, the best upper bound that we get on GapCDP is to show containment in Σ_2^{P}. (As we also note elsewhere, it is a tantalizing question whether one can improve our NP-hardness result for GapCDP to Σ_2^{P}-hardness, and therefore show that GapCDP is Σ_2^{P}-complete.)

1.3 An Approach for a Faster Approximation Algorithm

A natural direction for making our approximation algorithm in Theorem 2 more efficient would be to use reduced (but not necessarily optimal) bases in place of successive minima bases. Indeed, this is how [8] achieves a time-approximation tradeoff in its algorithm for LDP, which includes a polynomial-time approximation algorithm in one regime. (On the other hand, our approximation algorithm in Theorem 2 only runs in polynomial time for an extreme parameter regime – when $k = O(\log_q n)$.) And, a recent line of work has studied basis reduction algorithms for codes [14, 18], including analogs of basis reduction algorithms for lattices. However, it is not clear how to leverage bases reduced in any of the senses they consider to get an approximation algorithm running in a reasonable amount of time and with a reasonable approximation factor.

1.4 Open Questions

Our work leaves open several interesting questions. On the complexity side, our results showing that GapCDP is both NP-hard and in Σ_2^{P} begs the question of whether GapCDP is also Σ_2^{P}-hard, and thus Σ_2^{P}-complete. Moreover, as part of the inspiration for this work comes from the analogous Lattice Distortion Problem (LDP) [8], it is natural to ask whether (approximate) CDP reduces to LDP. We note that a fairly simple reduction from LCE to the

[7] In fact, $\|\cdot\|_{0\to 0}$ is not a matrix norm just as Hamming weight is not a vector norm (even over the real numbers). It does not satisfy the absolute homogeneity property $\|\alpha M\|_{0\to 0} = |\alpha| \, \|M\|_{0\to 0}$ for real-valued matrices M and scalars α, and the magnitude of a finite field element is not even defined. However, the $0 \to 0$ norm satisfies the other axioms of a matrix norm, including the triangle inequality ($\|A + B\|_{0\to 0} \le \|A\|_{0\to 0} + \|B\|_{0\to 0}$), and it is sub-multiplicative (i.e., $\|AB\|_{0\to 0} \le \|A\|_{0\to 0}\|B\|_{0\to 0}$). So, we abuse notation slightly and refer to $\|\cdot\|_{0\to 0}$ as a matrix norm.

Lattice Isomorphism Problem (LIP) – problems that essentially correspond to the distortion $D = 1$ cases of CDP and LDP, respectively – was given in [29, 9]. On the algorithmic side, it is natural to ask whether there is a better exact algorithm than the brute force, roughly q^{k^2}-time algorithm that we give, and whether there is a more efficient approximation algorithm for CDP than our roughly q^k. Finally, we ask given whether it is possible to build cryptography whose security guarantee rests on the hardness of approximate CDP. This seems particularly natural given the role of LCE both as security assumption for the LESS cryptosystem [12], and as a closely related problem to the McEliece cryptosystem [23, 1].[8]

2 Preliminaries

We define a *monomial matrix* M to be the product $M = DP$ of a non-singular diagonal matrix D and a permutation matrix P. We denote the set of $n \times n$ monomial matrices over a field $\mathbb{F}$ as $\mathcal{M}_n(\mathbb{F})$.

2.1 Codes

A *linear code* is a linear subspace $\mathcal{C} \subseteq \mathbb{F}^n$, where $\mathbb{F}$ is field. Generally, $\mathbb{F}$ is the finite field $\mathbb{F}_q$ for some prime power q, and when $q = 2$ (as is common in practice) the code is a *binary code*. A linear code $\mathcal{C}$ is often characterized as an $[n, k]$ or $[n, k, d]_q$ code, where q indicates the order of the field $\mathbb{F}_q$ that the code is taken over, k is the dimension of the code, and d is the minimum Hamming distance of $\mathcal{C}$, i.e.

$$d := \min_{\substack{\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}\\ \boldsymbol{x}\neq\boldsymbol{y}}} \|\boldsymbol{x}-\boldsymbol{y}\|_0 = \min_{\boldsymbol{x}\in\mathcal{C}\setminus\{\boldsymbol{0}\}} \|\boldsymbol{x}\|_0.$$

We also call n the *block length* of the code.

An $[n, k]$ code $\mathcal{C} \subseteq \mathbb{F}^n$ is usually specified by a *generator matrix* $G \in \mathbb{F}^{n\times k}$ with linearly independent columns. The code is then the span of the columns of G, and any *codeword* $\boldsymbol{x} \in \mathcal{C}$ can be expressed as $G\boldsymbol{m}$ for some *message* $\boldsymbol{m} \in \mathbb{F}^k$. Likewise, each message $\boldsymbol{m} \in \mathbb{F}^k$ corresponds to a unique codeword $G\boldsymbol{m} \in \mathcal{C}$.

We will also find it useful to refer to the *support* of a vector $\boldsymbol{x} \in \mathbb{F}^k$. Denoted $\mathrm{supp}(\boldsymbol{x})$, the support of $\boldsymbol{x}$ is the set of indices i for which $\boldsymbol{x}_i \neq 0$. The Hamming weight can then be expressed as $|\mathrm{supp}(\boldsymbol{x})|$, the size of the support.

▶ **Definition 4** (Successive Minima for Codes)**.** *Let $n, k \in \mathbb{Z}^+, k \leq n$, let q be a prime power, and let $\mathcal{C}$ be an $[n, k]_q$ code. For $1 \leq i \leq k$, the* ith successive minimum *of $\mathcal{C}$, denoted $\lambda_i(\mathcal{C})$, is defined as the minimum value $r \in \mathbb{Z}^+$ such that $\mathcal{C}$ contains at least i linearly independent codewords with Hamming weight at most r.*

We note that $\lambda_1(\mathcal{C})$ is the minimum distance d of $\mathcal{C}$. We emphasize again that successive minima are well-studied for lattices, but do not seem to have been studied much for codes. We will make use of the following definition, which formalizes one notion of optimal bases for codes.

▶ **Definition 5** (Successive Minima Basis)**.** *Let $n, k \in \mathbb{Z}^+, k \leq n$, let q be a prime power, and let $\mathcal{C}$ be an $[n, k]_q$ code. A generator matrix $G = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k)$ of $\mathcal{C}$ is a* successive minima basis *of $\mathcal{C}$ if $\|\boldsymbol{v}_i\|_0 = \lambda_i(\mathcal{C})$ for all $1 \leq i \leq k$.*

[8] One can check that the public and private generator matrices in McEliece generate linearly equivalent codes. However, it is not clear that breaking McEliece reduces to solving LCE or vice-versa.

We next give an algorithm for computing successive minima bases, which in particular implies that such bases always exist.

▶ **Lemma 6.** *Let $n, k \in \mathbb{Z}^+, k \leq n$, let q be a prime power, and let $\mathcal{C}$ be an $[n,k]_q$ code. Then there exists a $O^*(q^k)$-time, polynomial space algorithm for computing a sucessive minima basis of $\mathcal{C}$. In particular, such a basis of $\mathcal{C}$ always exists.*

Proof. Let $\boldsymbol{v}_1 \in \mathcal{C}$ be a codeword that satisfies $\|\boldsymbol{v}_1\| = \lambda_1(\mathcal{C})$, and define $\boldsymbol{v}_i$ recursively for $1 < i \leq k$ so that

$$\boldsymbol{v}_i \in \operatorname*{arg\,min}_{\boldsymbol{x} \in (\mathcal{C} \setminus \operatorname{span}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{i-1}))} \|\boldsymbol{x}\|_0 \ .$$

We prove inductively that $\|\boldsymbol{v}_i\|_0 = \lambda_i(\mathcal{C})$. This is clearly true in the base case of $i = 1$ since $\boldsymbol{v}_1$ is a shortest non-zero codeword by definition. For the inductive step with $i \geq 2$, assume that we can find $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{i-1} \in \mathcal{C}$ satisfying $\|\boldsymbol{v}_j\|_0 = \lambda_j(\mathcal{C})$ for $1 \leq j \leq i-1$. Additionally, observe that for any $1 \leq i \leq k$, the span of the set $S_i = \{\boldsymbol{v} \in \mathcal{C} : \|\boldsymbol{v}\|_0 \leq \lambda_i(\mathcal{C})\}$ has dimension at least i as a result of there being at least i linearly independent codewords in S_i. On the other hand, the set $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{i-1}\} \subset S_i$ has dimension only $i-1$, so there must exist some $\boldsymbol{v}_i \in \mathcal{C}$ such that $\|\boldsymbol{v}_i\|_0 \leq \lambda_i(\mathcal{C})$ and $\boldsymbol{v}_i \notin \operatorname{span}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{i-1})$.

We next show that $\|\boldsymbol{v}_i\|_0 \geq \lambda_i(\mathcal{C})$ (and therefore $\|\boldsymbol{v}_i\|_0 = \lambda_i(\mathcal{C})$) for $i \geq 2$. Suppose not. Then $\|\boldsymbol{v}_i\|_0 < \lambda_i(\mathcal{C})$. We have by definition that $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_i$ are linearly independent, and by the induction hypothesis, $\lambda_j(\mathcal{C}) = \|\boldsymbol{v}_j\|_0$ for all $1 \leq j \leq i-1$. By definition, $\|\boldsymbol{v}_1\|_0 \leq \cdots \leq \|\boldsymbol{v}_i\|_0$, and it follows that $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_i \in \mathcal{C}$ are linearly independent vectors all of Hamming weight strictly less than $\lambda_i(\mathcal{C})$, which is a contradiction.

The recursive process above yields k linearly independent codewords $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ with $\|\boldsymbol{v}_i\|_0 = \lambda_i(\mathcal{C})$ for all i. Since $\mathcal{C}$ is a k-dimensional subspace of $\mathbb{F}_q^n$, any k linearly independent vectors in $\mathcal{C}$ (and particularly our $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$) must span $\mathcal{C}$ and thus form a basis for it.

Computing each $\boldsymbol{v}_i$ involves a search over the $O(q^k)$ codewords in $\mathcal{C}$ with polynomial time required to check the linear independence and Hamming weight of each, so it takes $O^*(q^k)$ time to compute. This enumeration can be done in polynomial space. Therefore, by repeating this k times, we find the entire successive minima basis in $O^*(q^k)$ time and polynomial space. ◀

2.2 Coding Problems

Given a code, it is natural to ask what its minimum distance is. The Minimum Distance Problem formalizes this, and we give here the approximation version of the problem.

▶ **Definition 7.** *For $\gamma = \gamma(k) \geq 1$ and a prime power q, the* decisional γ-approximate Minimum Distance Problem over $\mathbb{F}_q$ *(γ-GapMDP$_q$) is the decision problem defined as follows. An instance consists of (a generator matrix $G \in \mathbb{F}_q^{n \times k}$ of) a code $\mathcal{C}$ and an integer distance $0 \leq d \leq m$. It is a:*

- *YES instance if $\lambda_1(\mathcal{C}) \leq d$.*
- *NO instance if $\lambda_1(\mathcal{C}) > \gamma d$.*

We note that GapMDP is NP-hard to approximate to within any constant factor. This was first established by Cheng and Wan [13]. (Earlier work showed NP-hardness under randomized reductions [16], and subsequent work showed simplified deterministic reductions [3, 24].)

▶ **Theorem 8** ([13])**.** *For all constants $\gamma \geq 1$ and all prime powers q, γ-GapMDP$_q$ is* NP*-hard.*

We now define a variant of the Nearest Codeword Problem with a stronger promise in the NO case.

▶ **Definition 9.** *For $\gamma = \gamma(k) \geq 1$, $\alpha = \alpha(k) > 0$, and a prime power q, the* decisional γ-approximate Nearest Codeword Problem over $\mathbb{F}_q$ with distance promise α *(γ-GapNCP$_q^\alpha$) is the decision problem defined as follows. An instance consists of (a generator matrix $G \in \mathbb{F}_q^{n\times k}$ of) a code $\mathcal{C}$, a target vector $\boldsymbol{t} \in \mathbb{F}_q^n$, and a distance parameter $d \in \mathbb{Z}^+$. It is a:*

- *YES instance if* $\operatorname{dist}(\boldsymbol{t}, \mathcal{C}) \leq d$.
- *NO instance if* $\operatorname{dist}(\boldsymbol{t}, \mathcal{C}) > \gamma d$ *and* $d < \alpha\lambda_1(\mathcal{C})$.

We additionally define γ-GapNCP$_q$ to be γ-GapNCP$_q^\infty$. I.e., in γ-GapNCP$_q$ there is no promised upper bound on d in the NO case. We note that γ-GapNCP$_q^\alpha$ trivially reduces to γ-GapNCP$_q^{\alpha'}$ for $\alpha' \geq \alpha$, and in particular γ-GapNCP$_q^\alpha$ trivially reduces to "plain" γ-GapNCP$_q$ for any $\alpha > 0$.

2.3 Matrix Norms

While there are many ways to define matrix norms, the most useful for us is the matrix $p \to q$ norm, where $p, q \in [1, \infty]$. We note in passing that matrix $p \to q$ norms over the real numbers have been studied from a computational standpoint; see, e.g., [11].

We can extend the definition of the matrix $p \to q$ norm to arbitrary (finite) fields $\mathbb{F}$ and to include $p = 0$ and $q = 0$, with $\|\boldsymbol{x}\|_0$ being the Hamming weight as is standard. Note, however, that the matrix $0 \to 0$ "norm" is not in fact a matrix norm as it is not scale-invariant. However, it does satisfy the triangle inequality and it is sub-multiplicative. Ultimately, throughout this article, we abuse notation and nevertheless refer to it as a norm. Additionally, we employ a "restricted" version of the matrix $0 \to 0$ norm, where one only maximizes over those vectors $\boldsymbol{x} \in \mathcal{C}$ for some subspace $\mathcal{C} \subseteq \mathbb{F}^n$. More formally:

▶ **Definition 10.** *Let $n \in \mathbb{Z}^+$, let $\mathbb{F}$ be a field, let $T \in \mathrm{GL}_n(\mathbb{F})$ be a matrix, and let $\mathcal{C} \subseteq \mathbb{F}^n$ be a subspace. The* matrix $0 \to 0$ norm of T restricted to the subspace $\mathcal{C}$ *is the quantity*

$$\|T|_{\mathcal{C}}\|_{0\to 0} \coloneqq \max_{\boldsymbol{x}\in\mathcal{C}\setminus\{\boldsymbol{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}.$$

Note that the restricted matrix $0 \to 0$ norm is also sub-multiplicative.

2.4 Distortion

To compare pairs of general codes, we introduce a quantity that captures how similar they are. We formalize this with *distortion*, similar to the formulation for the Lattice Distortion Problem [8] and distortion on metric embeddings. We start by defining the distortion of a particular transformation between two codes.

▶ **Definition 11.** *Let $n, k \in \mathbb{Z}^+$, $k \leq n$, let q be a prime power, let $\mathcal{C}_1, \mathcal{C}_2$ be $[n, k]_q$ codes, and let T be a linear map such that $T(\mathcal{C}_1) = \mathcal{C}_2$. The* distortion $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2)$ of T *the minimum value D such that for all $\boldsymbol{x} \in \mathcal{C}_1$,*

$$D_1 \cdot \|\boldsymbol{x}\|_0 \leq \|T\boldsymbol{x}\|_0 \leq D_2 \cdot \|\boldsymbol{x}\|_0$$

for some values $D_1, D_2 > 0$ satisfying $D_2/D_1 \leq D$. Equivalently,

$$\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) := \Big(\max_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\boldsymbol{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\Big) \Big/ \Big(\min_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\boldsymbol{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\Big)\ ,$$

which is the definition we gave in Section 1. Moreover, when $T \in \mathrm{GL}_n(\mathbb{F}_q)$ *and thus is invertible, this is equivalent to*

$$\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) = \|T|_{\mathcal{C}_1}\|_{0\to 0} \cdot \|T^{-1}|_{\mathcal{C}_2}\|_{0\to 0}. \tag{3}$$

Consequently, at least in the case when T is invertible, the distortion between two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ is inextricably tied to computing two matrix $0 \to 0$ norms, where one is restricted to $\mathcal{C}_1$ and the other is restricted to $\mathcal{C}_2$. Interestingly, as we show in Lemma 18, the assumption that T is invertible can be made without loss of generality, so we may take Equation (3) as an equivalent definition of distortion. Furthermore, while we require the codes in Definition 11 to have the same block length n, the definition can be extended to codes with different block lengths by simply padding the shorter code with zeros (or, one could allow for non-square matrices T).

We can now define the distortion between two codes generically as the lowest possible distortion between them using any transformation.

▶ **Definition 12.** *Let* $n, k \in \mathbb{Z}^+$, $k \le n$ *and let* q *be a prime power. The* distortion $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$ *between* $[n,k]_q$ *codes* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *is defined as the minimum value* D *such that there exists* $T \in \mathrm{GL}_n(\mathbb{F}_q)$ *with* $T(\mathcal{C}_1) = \mathcal{C}_2$ *and* $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) = D$*. In other words,*

$$\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \coloneqq \min_{\substack{T \in \mathrm{GL}_n(\mathbb{F}_q),\\ T(\mathcal{C}_1)=\mathcal{C}_2}} \mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2)\ . \tag{4}$$

2.4.1 The Code Distortion Problem

Using this definition of code distortion, we now define both the search and decision versions of the (γ-approximate) Code Distortion Problem (CDP).

▶ **Definition 13.** *For* $\gamma = \gamma(n) \ge 1$ *and a prime power* q*, the* γ-approximate Code Distortion Problem over $\mathbb{F}_q$ *(γ-*CDP_q*) is the search problem defined as follows. An instance consists of (generator matrices* $G_1, G_2 \in \mathbb{F}_q^{m\times n}$ *of) codes* $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$*, and the goal is to output* D *such that* $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \le D \le \gamma \cdot \mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$*.*

▶ **Definition 14.** *For* $\gamma = \gamma(n) \ge 1$ *and a prime power* q*, the* decisional γ-approximate Code Distortion Problem over $\mathbb{F}_q$ *(γ-*GapCDP_q*) is the decision problem defined as follows. An instance consists of (generator matrices* $G_1, G_2 \in \mathbb{F}_q^{m\times n}$ *of) codes* $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$ *and* $D \ge 1$*. It is a:*

- *YES instance if* $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \le D$*.*
- *NO instance if* $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) > \gamma D$*.*

The exact versions of these problems, CDP_q and GapCDP_q, are then defined to be 1-CDP_q and 1-GapCDP_q, respectively. Furthermore, the Linear Code Equivalence Problem is essentially the special case of CDP with $D = 1$. We formalize this relationship in Proposition 19 in the next section.

2.5 Basic Facts about Distortion

We next present several basic facts about the distortion between codes. We start with the following bounds on $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$.

▶ **Lemma 15.** *Let* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *be* $[n,k]_q$ *codes. Then, for any linear map* T *such that* $T(\mathcal{C}_1) = \mathcal{C}_2$*,* $1 \le \mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) \le n^2$*. As a consequence,* $1 \le \mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \le n^2$

Proof. For the lower bound, simply note that by the sub-multiplicativity of the matrix $0 \to 0$ norm on subspaces, it holds for all $T \in \mathrm{GL}_n(\mathbb{F})$ such that $T(\mathcal{C}_1) = \mathcal{C}_2$ that

$$1 = \|(T^{-1}T)|_{\mathcal{C}_1}\|_{0\to 0} \leq \|T^{-1}|_{\mathcal{C}_2}\|_{0\to 0} \cdot \|T|_{\mathcal{C}_1}\|_{0\to 0} = \mathcal{D}_T(C_1, \mathcal{C}_2).$$

For the upper bound, note that for all $T \in \mathrm{GL}_n(\mathbb{F})$ such that $T(\mathcal{C}_1) = \mathcal{C}_2$, it holds that

$$\|T|_{\mathcal{C}_1}\|_{0\to 0} = \max_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} \leq \max_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}} \|T\boldsymbol{x}\|_0 \leq n \ ,$$

where the last inequality follows from the fact that $\mathcal{C}_2$ is an $[n,k]_q$ code and for all $x \in \mathcal{C}_1\setminus\{\mathbf{0}\}$, $T\boldsymbol{x} \in \mathcal{C}_2\setminus\{\mathbf{0}\}$. A similar argument establishes that $\|T^{-1}|_{\mathcal{C}_2}\|_{0\to 0} \leq n$, and consequently

$$\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) = \|T|_{\mathcal{C}_1}\|_{0\to 0} \cdot \|T^{-1}|_{\mathcal{C}_2}\|_{0\to 0} \leq n^2 \ ,$$

as desired. The fact that $1 \leq \mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \leq n^2$ then follows from the definition of $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$. ◀

It is an open question if the upper bound in Lemma 15 can be reduced below n^2. Indeed, we do not know an example of a code with distortion n^2, but we do know an example of a code with distortion n, which we detail in the following.

▶ **Lemma 16.** *For every $n \in \mathbb{Z}^+$, there exist $[n,2]_2$ codes $\mathcal{C}_1, \mathcal{C}_2$ such that $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) = n$.*

Proof. Let $\mathcal{C}_1, \mathcal{C}_2$ be the codes defined as follows:

$$\mathcal{C}_1 = \left\{\mathbf{0}_n, \begin{pmatrix}1\\ \mathbf{0}_{n-1}\end{pmatrix}, \begin{pmatrix}0\\ \mathbf{1}_{n-1}\end{pmatrix}, \mathbf{1}_n\right\}, \quad \mathcal{C}_2 = \left\{\mathbf{0}_n, \begin{pmatrix}1\\1\\0\\ \mathbf{0}_{n-3}\end{pmatrix}, \begin{pmatrix}0\\1\\1\\ \mathbf{0}_{n-3}\end{pmatrix}, \begin{pmatrix}1\\0\\1\\ \mathbf{0}_{n-3}\end{pmatrix}\right\}.$$

It is easy to check that $\mathcal{C}_1$ and $\mathcal{C}_2$ are linear codes. Each non-zero codeword in $\mathcal{C}_2$ has Hamming weight 2 (the constant weight of non-zero codewords in $\mathcal{C}_2$ is the only property of it that we are using) while the non-zero codewords in $\mathcal{C}_1$ have minimum weight 1 and maximum weight n. As such, the distortion for any transformation T from $\mathcal{C}_1$ to $\mathcal{C}_2$ is

$$\max_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} \max_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}} \frac{\|\boldsymbol{x}\|_0}{\|T\boldsymbol{x}\|_0} = \frac{2}{1} \cdot \frac{n}{2} = n \ .$$ ◀

Ultimately, while we are unable to prove a matching upper bound of n on $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$, we are nevertheless able to prove such a bound when including *non-linear transformations* T.

▶ **Lemma 17.** *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be $[n,k]_q$ codes. Then, $\min_T \mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) \leq n$, where the minimum is over all maps T (including non-linear maps) such that $T(\mathcal{C}_1) = \mathcal{C}_2$.*

Proof. List all non-zero codewords in $\mathcal{C}_1$ in monotonically increasing order of Hamming weight, and prepare a similar list for $\mathcal{C}_2$. Let T be the map that takes codeword i in the first list to codeword i in the second list. Let $N := q^k - 1$, and let $1 \leq a_1 \leq \cdots \leq a_N \leq n$ and $1 \leq b_1 \leq \cdots \leq b_N \leq n$ be the Hamming weights of the non-zero codewords in the respective lists. Then, the distortion of T is

$$\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) = \left(\max_{i\in[N]} \frac{a_i}{b_i}\right) \cdot \left(\max_{j\in[N]} \frac{b_j}{a_j}\right) \ .$$

Now put $i^* = \arg\max_{i\in[N]} a_i/b_i$, and let $M = a_{i^*}/b_{i^*}$. We assume without loss of generality that $M \geq 1$ (otherwise, repeat the argument with the two codes flipped). Then, for all $j > i^*$,

$b_j/a_j \le n/a_{i^*} \le nb_{i^*}/a_{i^*} = n/M$. And, for all $j \le i^*$, $b_j/a_j \le b_{i^*}/1 = a_{i^*}/M \le n/M$. Consequently, for all $j \in [N]$, $b_j/a_j \le n/M$. Therefore,

$$\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) = \left(\max_{i\in[N]} \frac{a_i}{b_i}\right) \cdot \left(\max_{j\in[N]} \frac{b_j}{a_j}\right) \le M \cdot \frac{n}{M} = n \ ,$$

as desired. ◀

We next show that if there exists a matrix T such that $T\mathcal{C}_1 = \mathcal{C}_2$, then then exists an efficiently computable, *invertible* matrix T' such that $T'\mathcal{C}_1 = \mathcal{C}_2$ and $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) = \mathcal{D}_{T'}(\mathcal{C}_1, \mathcal{C}_2)$.

▶ **Lemma 18.** *Let T be a (not necessarily invertible) matrix such that $T(\mathcal{C}_1) = \mathcal{C}_2$ for $[n,k]_q$ codes $\mathcal{C}_1$, $\mathcal{C}_2$ such that $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) \le D$. Then there exists an invertible matrix T' such that $T'(\mathcal{C}_1) = \mathcal{C}_2$ and $\mathcal{D}_{T'}(\mathcal{C}_1, \mathcal{C}_2)$.*

Proof. Now consider a generator matrix $G_1 \in \mathbb{F}_q^{n\times k}$ of $\mathcal{C}_1$ and the corresponding generator matrix of $\mathcal{C}_2$, $G_2 := T(G_1) \in \mathbb{F}_q^{n\times k}$. Now, let $H_1, H_2 \in \mathbb{F}_q^{n\times(n-k)}$ be such that $A := (G_1|H_1), B := (G_2|H_2) \in \mathbb{F}_q^{n\times n}$ are full-rank. (In particular, it suffices to take H_1 and H_2 to be (transposed) parity-check matrices for $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively.) Finally, let $T' := BA^{-1}$. Note that T' is invertible, and that $T\boldsymbol{c} = T'\boldsymbol{c}$ for every $\boldsymbol{c} \in \mathcal{C}$. It follows that $T'(\mathcal{C}_1) = \mathcal{C}_2$ and $\mathcal{D}_{T'}(\mathcal{C}_1, \mathcal{C}_2) = \mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) = D$, as needed. ◀

We next show that two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ being linearly equivalent is equivalent to $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) = 1$ up to "scaling."

▶ **Proposition 19.** *Let $\mathcal{C}_1, \mathcal{C}_2$ be $[n,k]_q$ codes for some $n, k \in \mathbb{Z}^+$ and prime power q. Then $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) = 1$ if and only if $\mathcal{C}_1 \otimes \mathbf{1}_{r_1}$ (padded with some number of zeros to reach block length $r_2 n$) is linearly equivalent to $\mathcal{C}_2 \otimes \mathbf{1}_{r_2}$ for some $r_1, r_2 \in \mathbb{Z}^+$, $r_1 \le r_2$.*

Proof. We start with the backward direction. Let $\mathcal{C}_1'$ be the result of padding $\mathcal{C}_1 \otimes \mathbf{1}_{r_1}$ with $r_2 n - r_1 n$ zeros and let $\mathcal{C}_2' = \mathcal{C}_2 \otimes \mathbf{1}_{r_2}$. Suppose we have $r_1, r_2 \in \mathbb{Z}^+$, $r_1 \le r_2$ such that $\mathcal{C}_1'$ is linearly equivalent to $\mathcal{C}_2'$. Then, there must exist a monomial matrix T such that $T(\mathcal{C}_1') = \mathcal{C}_2'$ and $\|T(\boldsymbol{x} \otimes \mathbf{1}_{r_1})\|_0 = \|\boldsymbol{x} \otimes \mathbf{1}_{r_1}\|_0$ for all $\boldsymbol{x} \in \mathcal{C}_1$. Additionally, we can define $S_1 = I_n \otimes \mathbf{1}_{r_1}$, which maps $\mathcal{C}_1$ to $\mathcal{C}_1'$ with uniform scaling $\|S_1\boldsymbol{x}\|_0 = r_1\|\boldsymbol{x}\|_0$ for any $\boldsymbol{x} \in \mathcal{C}_1$, and $S_2 = I_n \otimes \boldsymbol{e}_1^\top$,[9] which maps $\mathcal{C}_2'$ to $\mathcal{C}_2$ with uniform scaling of $\|S_2\boldsymbol{x}\|_0 = \frac{1}{r_2}\|\boldsymbol{x}\|_0$ for any $\boldsymbol{x} \in \mathcal{C}_2'$. We now compose these to define a transformation $T' = S_2 T S_1$ that maps $\mathcal{C}_1$ to $\mathcal{C}_2$ and note that for any $\boldsymbol{x} \in \mathcal{C}_1$,

$$\frac{\|T'\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} = \frac{\|S_2 T S_1 \boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} = \frac{1}{r_2} \cdot \frac{\|TS_1\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} = \frac{1}{r_2} \cdot \frac{\|S_1\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} = \frac{r_1}{r_2} \cdot \frac{\|\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} = \frac{r_1}{r_2}.$$

As this is true for all $\boldsymbol{x} \in \mathcal{C}_1$, we can see that $\|T'|_{\mathcal{C}_1}\|_{0\to 0} = \frac{r_1}{r_2}$ and $\|(T')^{-1}|_{\mathcal{C}_2}\|_{0\to 0} = \frac{r_2}{r_1}$. Thus, $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) = \mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) = 1$.

For the forward direction, we are now given that $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) = 1$, i.e., there exists some T such that $T(\mathcal{C}_1) = \mathcal{C}_2$ and $\|T\boldsymbol{x}\|_0 = c\|\boldsymbol{x}\|_0$ for all $\boldsymbol{x} \in \mathcal{C}_1$ and some $c \in \mathbb{Q}^+$ (we assume without loss of generality that $c \le 1$). Let $c = \frac{r_1}{r_2}$ for some $r_1, r_2 \in \mathbb{Z}^+$, and once again take $\mathcal{C}_1' = \mathcal{C}_1 \otimes \mathbf{1}_{r_1}$ (padded with $r_2 n - r_1 n$ zeros) and $\mathcal{C}_2' = \mathcal{C}_2 \otimes \mathbf{1}_{r_2}$. Finally, we redefine $S_1 = \left(I_n \otimes \begin{pmatrix}1 & \mathbf{0}_{r_1-1}\end{pmatrix}\right)$, $S_2 = I_n \otimes \mathbf{1}_{r_2}$. As before, note that $S_1(\mathcal{C}_1') = \mathcal{C}_1$, $S_2(\mathcal{C}_2) = \mathcal{C}_2'$, $\|S_1\boldsymbol{x}\|_0 = \frac{1}{r_1}\|\boldsymbol{x}\|_0$ for all $\boldsymbol{x} \in \mathcal{C}_1'$, and $\|S_2\boldsymbol{x}\|_0 = r_2\|\boldsymbol{x}\|_0$ for all $\boldsymbol{x} \in \mathcal{C}_2$. We define $T' = S_2 T S_1$ so that T' maps $\mathcal{C}_1'$ to $\mathcal{C}_2'$, and again compute that

$$\|T'\boldsymbol{x}\|_0 = \|S_2 T S_1 \boldsymbol{x}\|_0 = r_2\|TS_1\boldsymbol{x}\|_0 = cr_2\|S_1\boldsymbol{x}\|_0 = \frac{cr_2}{r_1}\|\boldsymbol{x}\|_0 = \|\boldsymbol{x}\|_0 \ .$$

This is true for all $\boldsymbol{x} \in \mathcal{C}_1'$, so $\mathcal{C}_1' = \mathcal{C}_1 \otimes \mathbf{1}_{r_1}$ is equivalent to $\mathcal{C}_2' = \mathcal{C}_2 \otimes \mathbf{1}_{r_2}$ as required to complete the proof. ◀

[9] Here $\boldsymbol{e}_1 \in \mathbb{F}_q^{r_2}$ is the first standard normal basis vector in r_2 dimensions.

2.6 Reductions

In this paper, we employ three different types of reductions. The first and most general notion is that of a *Turing reduction*. Formally, there is a Turing reduction from a decision problem A to a decision problem B if and only if there exists an algorithm for A, given oracle access to B. In other words, A Turing-reduces to B if, given a subroutine for B, one can decide A. The second type of reduction is a *Cook reduction*. Formally, a Cook reduction is a polynomial-time Turing reduction, i.e., A Cook-reduces to B if and only if A Turing-reduces to B and the overall algorithm runs in polynomial time. The third and final type of reduction is that of a *Karp reduction* (a.k.a., a *polynomial-time many-one reduction*). Formally, a Karp reduction is a deterministic and polynomial-time algorithm that maps YES and NO instances of A to YES and NO instances of B, respectively.

3 The Complexity of the Code Distortion Problem

We now study some basic facts about the complexity of both the exact and approximate versions of GapCDP. Given the relationship of GapCDP to the matrix $0 \to 0$ norm (Section 2.3), we start by discussing the complexity of computing both the unrestricted and restricted versions of the matrix $0 \to 0$ norm.

3.1 The Complexity of the Matrix $0 \to 0$ Norm

Interestingly, the (unrestricted) matrix $0 \to 0$ norm is easy to compute.

▶ **Proposition 20.** *Let $\mathbb{F}$ be a field, let $m, n \in \mathbb{Z}^+$, and let $T = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n) \in \mathbb{F}^{m \times n}$. Then,*

$$\|T\|_{0\to 0} = \max_{i\in[n]} \|\boldsymbol{t}_i\|_0 \ .$$

Proof. By the triangle inequality, it holds for all $\boldsymbol{x} \in \mathbb{F}^n \setminus \{\boldsymbol{0}\}$ that

$$\begin{aligned}
\frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} &= \frac{\|\sum_{i\in\mathrm{supp}(\boldsymbol{x})} x_i \boldsymbol{t}_i\|_0}{|\mathrm{supp}(\boldsymbol{x})|} \\
&\leq \frac{\sum_{i\in\mathrm{supp}(\boldsymbol{x})} \|x_i \boldsymbol{t}_i\|_0}{|\mathrm{supp}(\boldsymbol{x})|} \\
&\leq \frac{|\mathrm{supp}(\boldsymbol{x})| \cdot \max_{i\in\mathrm{supp}(\boldsymbol{x})} \|\boldsymbol{t}_i\|_0}{|\mathrm{supp}(\boldsymbol{x})|} \\
&= \max_{i\in\mathrm{supp}(\boldsymbol{x})} \|\boldsymbol{t}_i\|_0 \ .
\end{aligned}$$

Furthermore, $\max_{i\in\mathrm{supp}(x)} \|\boldsymbol{t}_i\|_0 \leq \max_{i\in[n]} \|\boldsymbol{t}_i\|_0$, so $\|T\|_{0\to 0} \leq \max_{i\in[n]} \|\boldsymbol{t}_i\|_0$. On the other hand,

$$\|T\|_{0\to 0} \geq \max_{i\in[n]} \frac{\|T\boldsymbol{e}_i\|_0}{\|\boldsymbol{e}_i\|_0} = \max_{i\in[n]} \|\boldsymbol{t}_i\|_0 \ . \qquad \blacktriangleleft$$

We note that an immediate consequence of Proposition 20 is that there is a $\mathrm{poly}(m, n)$-time algorithm for computing $\|T\|_{0\to 0}$ for a matrix $T \in \mathbb{F}^{m\times n}$: simply compute the maximum Hamming weight of a column of T. That said, we show that matrix $0 \to 0$ norm *restricted to a subspace* is hard to compute, even approximately. We formalize this by defining a decisional version of the problem.

▶ **Definition 21.** *Let $\mathbb{F}$ be a field, let $m, n, k, q \in \mathbb{Z}^+$, and let $\gamma = \gamma(m,n) \geq 1$. The* decisional γ-approximate matrix $0 \to 0$ norm problem over a subspace of $\mathbb{F}_q$ *(γ-GapNorm$_q^{0\to 0}$) is the promise problem defined as follows. An instance consists of a basis $G \subseteq \mathbb{F}_q^{n\times k}$ of a subspace $\mathcal{C} \subseteq \mathbb{F}_q^n$, a matrix $T \in \mathbb{F}_q^{m\times n}$, and threshold $r \geq 0$. It is a:*

- *YES instance if $\|T|_{\mathcal{C}}\|_{0\to 0} \leq r$.*
- *NO instance if $\|T|_{\mathcal{C}}\|_{0\to 0} > \gamma r$.*

We will now give a reduction from γ'-GapMDP$_2$ to γ-GapNorm$_2^{0\to 0}$ for which it suffices to take $\gamma' = O(\gamma)$. This result is due to Golovnev and Stephens-Davidowitz [20].

▶ **Theorem 22** ([20])**.** *Let $\gamma \geq 1$ and let $\epsilon > 0$ be a constant. Then, for all $\gamma' > \left(\frac{1+\epsilon}{1-\epsilon}\right) \cdot \gamma$, there is a Karp reduction from γ'-GapMDP$_2$ to γ-GapNorm$_2^{0\to 0}$.*

Proof. Let $(G \in \mathbb{F}_2^{m\times n}, r')$ be an instance of γ'-GapMDP$_2$, where $m, n \in \mathbb{Z}^+$. Moreover, let T be a generator matrix of an $[m,n]_2$ code that is ε-balanced, i.e., all non-zero codewords in $\mathcal{C}(T)$ have Hamming weight in the interval $[(1-\varepsilon)\frac{m}{2}, (1+\varepsilon)\frac{m}{2}]$. By a result of Ta-Shma [31], such a T exists and, moreover, can be constructed in deterministic polynomial time.

Now, if (G, r') is a YES instance of γ'-MDP$_2$, then for all $\boldsymbol{x} \in \mathcal{C}\backslash\{\mathbf{0}\}$ with $\|\boldsymbol{x}\|_0 = \lambda_1(\mathcal{C})$,

$$\|T|_{\mathcal{C}}\|_{0\to 0} \geq \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} \geq \frac{\|T\boldsymbol{x}\|_0}{r'} \geq \frac{(1-\epsilon)m/2}{r'}.$$

Here, the first inequality uses the definition of the restricted matrix $0 \to 0$ norm, the second uses the fact that (G, r') is a YES instance, and the third uses the fact that T generates an ε-balanced code.

On the other hand, if (G, r') is a NO instance of γ'-MDP$_2$, then for all $\boldsymbol{x} \in \mathcal{C}\backslash\{\mathbf{0}\}$,

$$\frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} < \frac{(1+\epsilon)m/2}{\gamma' r'},$$

which implies

$$\|T|_{\mathcal{C}}\|_{0\to 0} = \max_{\boldsymbol{x}\in\mathcal{C}\backslash\{\mathbf{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} < \frac{(1+\epsilon)m/2}{\gamma' r'}.$$

Altogether, with $\gamma = \frac{1-\varepsilon}{1+\varepsilon}(\gamma' + \delta)$ and $r = \frac{(1+\epsilon)m/2}{(\gamma'+\delta)r'}$ for any $\delta > 0$, it holds that $\|T|_{\mathcal{C}}\|_{0\to 0} > \gamma r$ (a YES instance of co-γ-GapNorm$_2^{0\to 0}$) implies a YES instance of γ'-MDP$_2$, and $\|T|_{\mathcal{C}}\|_{0\to 0} \leq r$ (a NO instance of co-γ-GapNorm$_2^{0\to 0}$) implies a NO instance of γ'-MDP$_2$. ◀

Consequently, by Theorem 8, γ-GapNorm$_2^{0\to 0}$ is coNP-hard. In fact, it is coNP-complete.

▶ **Corollary 23.** *For all $\gamma \geq 1$, γ-GapNorm$_2^{0\to 0}$ is coNP-complete.*

Proof. Since γ-GapNorm$_2^{0\to 0}$ is coNP-hard, it suffices to prove that γ-GapNorm$_2^{0\to 0} \in$ coNP. But this is plain, since if (G, T, r) is a YES instance, then for all $\boldsymbol{x} \in \mathcal{C}\backslash\{\mathbf{0}\}$, $\|T\boldsymbol{x}\|_0/\|\boldsymbol{x}\|_0 \leq r$, and if (G, T, r) is a NO instance, then there exists $\boldsymbol{x} \in \mathcal{C}\backslash\{\mathbf{0}\}$ for which $\|T\boldsymbol{x}\|_0/\|\boldsymbol{x}\|_0 \geq \gamma r$. ◀

3.2 GapCDP is in Σ_2^{P}

We now show that the code distortion problem is in Σ_2^{P}, the second level of the polynomial hierarchy. Recall that a language $L \in \Sigma_2^{\mathsf{P}}$ if and only if there exists two polynomials p_1 and p_2 as well as a polynomial-time deterministic Turing machine M such that for all inputs $x \in \{0,1\}^*$, $x \in L$ if and only if $(\exists y \in \{0,1\}^{p_1(|x|)})(\forall z \in \{0,1\}^{p_2(|x|)}) : M(x,y,z) = 1$.

▶ **Theorem 24.** *For all $q \in \mathbb{Z}^+$, $\mathrm{GapCDP}_q \in \Sigma_2^{\mathsf{P}}$.*

Proof. Let (G_1, G_2, D) be an instance of GapCDP_q, where G_1 and G_2 generate the codes $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. By definition, (G_1, G_2, D) is a YES instance if and only if $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \leq D$. This holds if and only if there exists $T \in \mathrm{GL}_m(\mathbb{F}_q)$ with $T(\mathcal{C}_1) = \mathcal{C}_2$ such that for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}_1 \backslash \{\mathbf{0}\}$,

$$\left(\frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\right) \cdot \left(\frac{\|T\boldsymbol{y}\|_0}{\|\boldsymbol{y}\|_0}\right)^{-1} \leq D.$$

For all such T, $\boldsymbol{x}$, and $\boldsymbol{y}$, the above inequality is checkable in deterministic polynomial time. Moreover, the size of all such T, $\boldsymbol{x}$, and $\boldsymbol{y}$ is a polynomial in the size of the input (G_1, G_2, D). Consequently, as the order of the $\exists$ and $\forall$ quantifiers in the above reformulation of GapCDP_q is consistent with the class Σ_2^{P}, it holds that $\mathrm{GapCDP}_q \in \Sigma_2^{\mathsf{P}}$, as desired. ◀

As mentioned in Section 1.4, we suspect GapCDP_q is also Σ_2^{P}-hard, thus making it Σ_2^{P}-complete. However, proving this remains an interesting open question.

3.3 GMSS for Codes

We will now give a reduction from γ-GapMDP to γ-GapNCP that is analogous to the seminal reduction from the Shortest Vector Problem to the Closest Vector Problem on lattices due to [19]. We will use this result in our proof of NP-hardness of approximate CDP.

▶ **Theorem 25** (GMSS for Codes)**.** *Let $m, n \in \mathbb{Z}^+$ with $n \leq m$, let $\gamma = \gamma(m, n) \geq 1$, and let q be a prime power. Then there is a $\mathrm{poly}(m, q)$-time Turing reduction from γ-GapMDP_q on $[m, n]_q$ codes to γ-$\mathrm{GapNCP}_q^{1/\gamma}$ (and hence also γ-GapNCP_q) on $[m, n-1]_q$ codes.*

Proof. Let $(G = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n) \in \mathbb{F}_q^{m \times n}, d)$ be the input instance of γ-GapMDP_q. The reduction does the following. It constructs $(q-1)n$ many instances $(G_i, \boldsymbol{t}_{i,j} := -j \cdot \boldsymbol{g}_i, d)$ of GapNCP_q for $1 \leq i \leq n$ and $j \in \mathbb{F}_q^*$, where $G_i := (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{i-1}, \boldsymbol{g}_{i+1}, \ldots, \boldsymbol{g}_n)$ is "G with its ith column removed." It then calls its γ-$\mathrm{GapNCP}_q^{1/\gamma}$ oracle on each instance $(G_i, \boldsymbol{t}_{i,j}, d)$. The reduction outputs YES if the oracle responds with YES on some input, and otherwise it outputs NO.

It is clear that the reduction runs in the stated amount of time, and it remains to show its correctness. Suppose that the input is a YES instance. Then there exists $\boldsymbol{a} \in \mathbb{F}_q^n \backslash \{\mathbf{0}\}$ such that $\|G\boldsymbol{a}\|_0 \leq d$. Let a_i be a non-zero coordinate of $\boldsymbol{a}$, and let $\boldsymbol{a}' := (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n) \in \mathbb{F}_q^{n-1}$ be "$\boldsymbol{a}$ with its ith coordinate deleted." Then for $j = a_i$, we have that

$$\|G_i\boldsymbol{a}' - \boldsymbol{t}_{i,j}\|_0 = \|G_i\boldsymbol{a}' + j\boldsymbol{g}_i\|_0 = \Big\| \sum_{\ell \neq i} a_\ell \boldsymbol{g}_\ell + a_i \boldsymbol{g}_i \Big\|_0 = \|G\boldsymbol{a}\|_0 \leq d \ .$$

So, the γ-$\mathrm{GapNCP}_q^{1/\gamma}$ oracle outputs YES on input $(G_i, \boldsymbol{t}_{i,j}, d)$, as needed.

Now, suppose that the input is a NO instance. Let $\mathcal{C} := \mathcal{C}(G)$ and let $\mathcal{C}_i := \mathcal{C}(G_i)$. We note that for every $1 \leq i \leq n$ and $j \in \mathbb{F}_q^*$,

$$\mathcal{C}_i \subset \mathcal{C} \text{ and } \mathcal{C}_i - \boldsymbol{t}_{i,j} \subset \mathcal{C} \setminus \{\mathbf{0}\} \ .$$

It follows that $\lambda_1(\mathcal{C}_i) \geq \lambda_1(\mathcal{C})$ and that $\mathrm{dist}(\boldsymbol{t}_{i,j}, \mathcal{C}_i) \geq \lambda_1(\mathcal{C})$ for all $1 \leq i \leq n$ and $j \in \mathbb{F}_q^*$. Because the input is a NO instance, $\lambda_1(\mathcal{C}) > \gamma d$, and so we have that for every $1 \leq i \leq n$, $\lambda_1(\mathcal{C}_i) \geq \lambda_1(\mathcal{C}) > \gamma d$, and therefore $d < \lambda_1(\mathcal{C}_i)/\gamma$. Moreover, for every $1 \leq i \leq n$ and $j \in \mathbb{F}_q^*$, $\mathrm{dist}(\boldsymbol{t}_{i,j}, \mathcal{C}_i) \geq \lambda_1(\mathcal{C}) > \gamma d$. Therefore, each instance $(G_i, \boldsymbol{t}_{i,j}, d)$ is a NO instance of γ-$\mathrm{GapNCP}_q^{1/\gamma}$, as needed. ◀

3.4 NP-hardness of the Code Distortion Problem

We next give a reduction from γ'-GapNCP$_q^\alpha$ to γ-CDP$_q$ for which it suffices to take $\gamma' = O(\gamma)$.

▶ **Theorem 26.** *Let $\gamma \geq 1$, let q be a prime power, and let $D > 1$ be a constant. Then for any $\gamma' > \lceil \frac{D+1}{D-1} \rceil D \cdot \gamma$, there is a Karp reduction from γ'-GapNCP$_q^{1/\gamma'}$ to γ-CDP with distortion D.*

Proof. Let $(G \in \mathbb{F}_q^{n \times k}, \boldsymbol{t} \in \mathbb{F}_q^n, d \in \mathbb{Z}^+)$ be an instance of γ'-GapNCP$_q^{1/\gamma'}$. Define

$$G_1 := \begin{pmatrix} G & 0 & \mathbf{0} \\ 0 & G & \mathbf{0} \\ 0 & 0 & \mathbf{1}_r \end{pmatrix}, \qquad G_2 := \begin{pmatrix} G & 0 & -\boldsymbol{t} \\ 0 & G & \mathbf{0} \\ 0 & 0 & \mathbf{1}_r \end{pmatrix}$$

with $r := \lceil \frac{D+1}{D-1} \rceil \cdot d$. The reduction outputs the γ-CDP$_q$ instance (G_1, G_2, D).

It is clear that the reduction is efficient, and it remains to shows its correctness. Let $\mathcal{C} := \mathcal{C}(G)$, $\mathcal{C}_1 := \mathcal{C}(G_1)$, and $\mathcal{C}_2 := \mathcal{C}(G_2)$. Suppose that the input is a YES instance of γ'-GapNCP$_q^{1/\gamma'}$. Let $\boldsymbol{t}' \in \mathcal{C} - \boldsymbol{t}$ be such that $\|\boldsymbol{t}'\|_0 = \mathrm{dist}(\boldsymbol{t}, \mathcal{C})$, i.e., $\boldsymbol{t}'$ is a shortest codeword in the coset $\mathcal{C} - \boldsymbol{t}$. Define

$$G_2' := \begin{pmatrix} G & 0 & \boldsymbol{t}' \\ 0 & G & \mathbf{0} \\ 0 & 0 & \mathbf{1}_r \end{pmatrix}$$

Notice that $\mathcal{C}(G_2) = \mathcal{C}(G_2')$, and, because the input is a YES instance, that $\|\boldsymbol{t}'\|_0 \leq d$.[10] Let $G_1 = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{2n+1})$, let $G_2' = (\boldsymbol{g}_1', \ldots, \boldsymbol{g}_{2n+1}')$, and let M be an invertible linear map such that $M : \boldsymbol{g}_i \mapsto \boldsymbol{g}_i'$ for every i. Such a full-rank map M exists because G_1 and G_2' are full-rank, and it is clear that M maps $\mathcal{C}_1$ to $\mathcal{C}_2$.

Let $\boldsymbol{x} \in \mathcal{C}_1$ be an arbitrary non-zero codeword. We can write $\boldsymbol{x} = (\boldsymbol{x}', \boldsymbol{x}'', a\mathbf{1}_r)$ for some $\boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{C}$ and $a \in \mathbb{F}_q$. If $a = 0$, then $\|M\boldsymbol{x}\|_0 = \|\boldsymbol{x}\|_0$, and so $\|M\boldsymbol{x}\|_0 / \|\boldsymbol{x}\|_0 = 1$. On the other hand, assume that $a \neq 0$. Then $M\boldsymbol{x} = \boldsymbol{x} + (a\boldsymbol{t}', \mathbf{0})$, and so by triangle inequality and the fact that $\|\boldsymbol{t}'\|_0 \leq d$, $\|M\boldsymbol{x}\|_0 \leq \|\boldsymbol{x}\|_0 + d$. Therefore,

$$\frac{\|M\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} \leq \frac{\|\boldsymbol{x}\|_0 + d}{\|\boldsymbol{x}\|_0} \leq \frac{\|\boldsymbol{x}\|_0 + d}{\|a\mathbf{1}_r\|_0} = 1 + d/r \ . \tag{5}$$

Furthermore, $\boldsymbol{x} = M\boldsymbol{x} - (a\boldsymbol{t}', \mathbf{0})$, and so $\|\boldsymbol{x}\|_0 = \|M\boldsymbol{x} - (a\boldsymbol{t}', \mathbf{0})\|_0 \leq \|M\boldsymbol{x}\|_0 + \|\boldsymbol{t}'\|_0 \leq \|M\boldsymbol{x}\|_0 + d$. It follows that $\|M\boldsymbol{x}\|_0 \geq \|\boldsymbol{x}\|_0 - d$, and therefore,

$$\frac{\|M\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} \geq \frac{\|\boldsymbol{x}\|_0 - d}{\|\boldsymbol{x}\|_0} = 1 - \frac{d}{\|\boldsymbol{x}\|_0} \geq 1 - \frac{d}{\|a\mathbf{1}_r\|_0} = 1 - d/r \ . \tag{6}$$

Recalling that $r = \lceil \frac{D+1}{D-1} \rceil d \geq \frac{D+1}{D-1} d$, we get that $d/r \leq \frac{D-1}{D+1}$. By combining Equations (5)

[10] Actually computing $\boldsymbol{t}'$ amounts to solving γ'-GapNCP$_q^{1/\gamma'}$, which is a hard problem. However, here we are only using $\boldsymbol{t}'$ and G_2' for analysis, and we are not computing them.

and (6) we then have that

$$\begin{aligned}\mathcal{D}(\mathcal{C}_1,\mathcal{C}_2) &\leq \max_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}}\left(\frac{\|M\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\right)\Big/\min_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}}\left(\frac{\|M\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\right)\\ &\leq (1+d/r)\,/\,(1-d/r)\\ &\leq \left(1+\frac{D-1}{D+1}\right)/\left(1-\frac{D-1}{D+1}\right)\\ &= \left(\frac{2D}{D+1}\right)/\left(\frac{2}{D+1}\right)\\ &= D,\end{aligned}$$

as needed.

Now, suppose that the input is a NO instance, and let M be an invertible linear map such that $M(\mathcal{C}_1)=\mathcal{C}_2$. Let $\boldsymbol{y}:=M\boldsymbol{g}_{2n+1}=M\cdot(\mathbf{0},\mathbf{1}_r)$, and note that $\boldsymbol{y}=(\boldsymbol{y}'-a\boldsymbol{t},\boldsymbol{y}'',a\mathbf{1}_r)$ for some $\boldsymbol{y}',\boldsymbol{y}''\in\mathcal{C}$ and $a\in\mathbb{F}_q$. If $a\neq 0$, then

$$\|\boldsymbol{y}\|_0=\|\boldsymbol{y}'-a\boldsymbol{t}\|_0+\|\boldsymbol{y}''\|_0+\|a\mathbf{1}_r\|_0\geq \operatorname{dist}(-a\boldsymbol{t},\mathcal{C})+r>\gamma' d+r\ ,$$

where we have used that $\operatorname{dist}(-a\boldsymbol{t},\mathcal{C})=\operatorname{dist}(\boldsymbol{t},\mathcal{C})$ for $a\neq 0$. On the other hand, if $a=0$, then

$$\|\boldsymbol{y}\|_0=\|\boldsymbol{y}'\|_0+\|\boldsymbol{y}''\|_0\geq\lambda_1(\mathcal{C})>\gamma' d\ ,$$

where the first inequality holds because at least one of $\boldsymbol{y}'$ and $\boldsymbol{y}''$ is a non-zero codeword in $\mathcal{C}$ (since $\boldsymbol{y}\neq\mathbf{0}$), and the second inequality holds because of the assumption that $d<\lambda_1(\mathcal{C})/\gamma'$, which follows from the definition of NO instances of γ'-GapNCP$_q^{1/\gamma'}$. In either case, we have that

$$\max_{\boldsymbol{x}\in\mathcal{C}_1\setminus\{\mathbf{0}\}}\frac{\|M\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}\geq\frac{\|M\boldsymbol{g}_{2n+1}\|_0}{\|\boldsymbol{g}_{2n+1}\|_0}=\frac{\|\boldsymbol{y}\|_0}{r}>\gamma' d/r\ . \tag{7}$$

Now, let $\boldsymbol{x}'\in\mathcal{C}$ be such that $\|\boldsymbol{x}'\|_0=\lambda_1(\mathcal{C})$, and define $\boldsymbol{x}_1:=(\boldsymbol{x}',\mathbf{0},\mathbf{0}),\boldsymbol{x}_2:=(\mathbf{0},\boldsymbol{x}',\mathbf{0})$. Note that $\boldsymbol{x}_1,\boldsymbol{x}_2\in\mathcal{C}_2$, and that $\boldsymbol{y}_1:=M^{-1}\boldsymbol{x}_1=(\boldsymbol{y}_1',\boldsymbol{y}_1'',a_1\mathbf{1}_r)$ and $\boldsymbol{y}_2:=M^{-1}\boldsymbol{x}_2=(\boldsymbol{y}_2',\boldsymbol{y}_2'',a_2\mathbf{1}_r)$ for some $\boldsymbol{y}_1',\boldsymbol{y}_1'',\boldsymbol{y}_2',\boldsymbol{y}_2''\in\mathcal{C}$ and $a_1,a_2\in\mathbb{F}_q$. Furthermore, because $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are linearly independent and M is invertible, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ must also be linearly independent. So, at least one of $\boldsymbol{y}_1',\boldsymbol{y}_1'',\boldsymbol{y}_2',\boldsymbol{y}_2''$ is non-zero, and therefore $\max\{\|M^{-1}\boldsymbol{x}_1\|_0,\|M^{-1}\boldsymbol{x}_2\|_0\}\geq\lambda_1(\mathcal{C})$. It follows that

$$\begin{aligned}\max_{\boldsymbol{x}\in\mathcal{C}_2\setminus\{\mathbf{0}\}}\frac{\|M^{-1}\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0}&\geq\max\left\{\frac{\|M^{-1}\boldsymbol{x}_1\|_0}{\|\boldsymbol{x}_1\|_0},\frac{\|M^{-1}\boldsymbol{x}_2\|_0}{\|\boldsymbol{x}_2\|_0}\right\}\\ &=\max\{\|M^{-1}\boldsymbol{x}_1\|_0,\|M^{-1}\boldsymbol{x}_2\|_0\}/\lambda_1(\mathcal{C})\geq 1\ .\end{aligned} \tag{8}$$

Combining Equations (7) and (8) then gives

$$\mathcal{D}(\mathcal{C}_1,\mathcal{C}_2)>\gamma' d/r=\frac{\gamma'}{\lceil\frac{D+1}{D-1}\rceil}>\frac{\lceil\frac{D+1}{D-1}\rceil\cdot\gamma D}{\lceil\frac{D+1}{D-1}\rceil}=\gamma D\ ,$$

as needed. ◀

From Theorems 25 and 26 and the fact that γ-GapMDP is NP-hard for any constant $\gamma\geq 1$, we conclude that the CDP is NP-hard to approximate to within any constant factor.

▶ **Theorem 1.** *For any constant $\gamma \geq 1$, any constant $D > 1$, and any prime power $q \leq \text{poly}(n)$, γ-GapCDP$_q$ with distortion threshold D is* NP*-hard under deterministic Cook reductions.*

Proof. Combine the NP-hardness result in Theorem 8 with the reductions in Theorems 25 and 26. ◀

We note that restriction on the field size q in Theorem 1 is due the running time of the reduction in Theorem 25.

4 Algorithms for Code Distortion

We now turn to giving algorithms for code distortion.

4.1 An Exact Algorithm

We first analyze the running time of a brute force exact algorithm for CDP, which is the best exact algorithm we know.

▶ **Lemma 27.** *Let $n, k \in \mathbb{Z}^+$, $k \leq n$, let q be a prime power, and let $\mathcal{C}_1, \mathcal{C}_2$ be $[n,k]_q$ codes. The distortion $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$ can be computed in $O^*(q^{k(k+1)})$ time.*

Proof. A linear transformation T with $T\mathcal{C}_1 = \mathcal{C}_2$ must map a generator matrix of $\mathcal{C}_1$ to a generator matrix of $\mathcal{C}_2$. So, it suffices to fix a generator matrix of $G_1 \in \mathbb{F}_q^{n\times k}$, enumerate generator matrices $G_2 \in \mathbb{F}_q^{n\times k}$ of $\mathcal{C}_2$, and compute $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2)$ for the induced map T such that $TG_1 = G_2$.

To enumerate generator matrices G_2 of $\mathcal{C}_2$, it suffices to compute $G_2 := G_2'U$ for a fixed generator matrix G_2' of $\mathcal{C}_2$ and each $U \in \text{GL}_k(\mathbb{F}_q)$. In order to enumerate elements of $\text{GL}_k(\mathbb{F}_q)$, it suffices to enumerate all matrices $U \in \mathbb{F}_q^{k\times k}$, and discard them if they are singular. This takes $O^*(q^{k^2})$ time.

To compute $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2)$ for a given map T, it suffices to compute $\|T\boldsymbol{x}\|_0/\|\boldsymbol{x}\|_0$ for all $\boldsymbol{x} \in \mathcal{C}_1 \setminus \{\mathbf{0}\}$ and $\|T^{-1}\boldsymbol{y}\|_0/\|\boldsymbol{y}\|_0$ for all $\boldsymbol{y} \in \mathcal{C}_2 \setminus \{\mathbf{0}\}$. This can be done in $O^*(q^k)$ time since $|\mathcal{C}_1| = |\mathcal{C}_2| = q^k$. So, overall the algorithm runs in $O^*(q^{k(k+1)})$, as needed. ◀

We then get the following corollary.

▶ **Corollary 28.** *Let $n, k \in \mathbb{Z}^+$, $k \leq n$, let q be a prime power, and let $\mathcal{C}_1, \mathcal{C}_2$ be $[n,k]_q$ codes. The distortion $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$ can be computed in* poly(n) *time if $k = O\left(\sqrt{\log_q n}\right)$.*

Notably, Lemma 27 runs in roughly q^{k^2} time, and at a minimum it would be desirable to find a single-exponential, roughly q^k-time algorithm. Although we do not give an exact such q^k-time algorithm, we do give a roughly q^k-time *approximation* algorithm for CDP$_q$.

4.2 A General Approximation Algorithm

Our approximation algorithm uses the following fact.

▶ **Lemma 29.** *Let $n, k \in \mathbb{Z}^+, k \leq n$, let q be a prime power, and let $\mathcal{C}_1, \mathcal{C}_2$ be $[n,k]_q$ codes with successive minima bases $G_1 = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k), G_2 = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k)$, respectively, and let T be a linear transformation such that $TG_1 = G_2$. Furthermore, let $\boldsymbol{x} = G_1\boldsymbol{a}$ for $\boldsymbol{a} \in \mathbb{F}_q^k$ be a non-zero codeword in $\mathcal{C}_1$. Suppose that $j \in [k]$ is the maximum index such that $a_j \neq 0$. Then*

$$\frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} \leq \frac{\sum_{i=1}^{j} \lambda_i(\mathcal{C}_2)}{\lambda_j(\mathcal{C}_1)} .$$

Proof. Since $TG_1 = G_2$, we know that $T\boldsymbol{v}_i = \boldsymbol{w}_i$ for $1 \leq i \leq k$ and that $T\boldsymbol{x} = \sum_{i=1}^{j} T(a_i\boldsymbol{v}_i) = \sum_{i=1}^{j} a_i\boldsymbol{w}_i$. Using the triangle inequality, we get the upper bound $\|T\boldsymbol{x}\|_0 \leq \sum_{i=1}^{j}\|a_i\boldsymbol{w}_i\|_0 \leq \sum_{i=1}^{j}\|\boldsymbol{w}_i\|_0 = \sum_{i=1}^{j}\lambda_i(\mathcal{C}_2)$.

On the other hand, the j vectors $\boldsymbol{x}$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{j-1}$ must be linearly independent because $a_j \neq 0$ and $\boldsymbol{v}_j \notin \operatorname{span}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{j-1})$. By the definition of successive minima, we then have that $\|\boldsymbol{x}\|_0 \geq \lambda_j(\mathcal{C}_1)$. Combining this lower bound on $\|\boldsymbol{x}\|_0$ with the upper bound on $\|T\boldsymbol{x}\|_0$ above implies the claim. ◀

We define the following quantity relating the successive minima of $[n,k]_q$ codes $\mathcal{C}_1$ and $\mathcal{C}_2$:

$$M(\mathcal{C}_1, \mathcal{C}_2) \coloneqq \max_{i\in[k]} \frac{\lambda_i(\mathcal{C}_2)}{\lambda_i(\mathcal{C}_1)} \ . \tag{9}$$

We will use this quantity to give lower and upper bounds on the distortion $\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$.

▶ **Theorem 30.** *Let $n, k \in \mathbb{Z}^+, k \leq n$, let q be a prime power, and let $\mathcal{C}_1, \mathcal{C}_2$ be $[n,k]_q$ codes. Let $G_1 = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k)$ and $G_2 = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k)$ be successive minima bases of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, and let T be a linear transformation such that $TG_1 = G_2$ Then*

$$M(\mathcal{C}_1, \mathcal{C}_2)M(\mathcal{C}_2, \mathcal{C}_1) \leq \mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \leq \mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) \leq k^2 \cdot M(\mathcal{C}_1, \mathcal{C}_2)M(\mathcal{C}_2, \mathcal{C}_1) \ .$$

Proof. We first prove the upper bound. For any $\boldsymbol{x} \in \mathcal{C}_1 \backslash \{\boldsymbol{0}\}$, we can write $\boldsymbol{x} = G\boldsymbol{a}$ for some $\boldsymbol{a} \in \mathbb{F}_q^k \setminus \{\boldsymbol{0}\}$. Let $j \in [k]$ be the maximum index such that $a_j \neq 0$. By Lemma 29, we then have that

$$\frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} \leq \frac{\sum_{i=1}^{j}\lambda_i(\mathcal{C}_2)}{\lambda_j(\mathcal{C}_1)} \leq \frac{\sum_{i=1}^{j}\lambda_j(\mathcal{C}_2)}{\lambda_j(\mathcal{C}_1)} \leq k \cdot \frac{\lambda_j(\mathcal{C}_2)}{\lambda_j(\mathcal{C}_1)} \leq k \cdot \max_{i\in[k]} \frac{\lambda_i(\mathcal{C}_2)}{\lambda_i(\mathcal{C}_1)} \ .$$

As this is true for all $x \in \mathcal{C}_1 \backslash \{\boldsymbol{0}\}$, it holds that

$$\max_{\boldsymbol{x}\in\mathcal{C}_1\backslash\{\boldsymbol{0}\}} \frac{\|T\boldsymbol{x}\|_0}{\|\boldsymbol{x}\|_0} \leq k \cdot \max_{i\in[k]} \frac{\lambda_i(\mathcal{C}_2)}{\lambda_i(\mathcal{C}_1)} = k \cdot M(\mathcal{C}_1, \mathcal{C}_2) \ .$$

A symmetric argument applied to T^{-1} yields

$$\max_{\boldsymbol{y}\in\mathcal{C}_2\backslash\{\boldsymbol{0}\}} \frac{\|T^{-1}\boldsymbol{y}\|_0}{\|\boldsymbol{y}\|_0} \leq k \cdot M(\mathcal{C}_2, \mathcal{C}_1) \ ,$$

and plugging both into the definition of distortion yields the upper bound

$$\mathcal{D}(\mathcal{C}_1, \mathcal{C}_2) \leq k^2 \cdot M(\mathcal{C}_1, \mathcal{C}_2)M(\mathcal{C}_2, \mathcal{C}_1) \ .$$

We next prove the lower bound using an analogous proof to one in [8]. Let U be an arbitrary full-rank linear map such that $U\mathcal{C}_1 = \mathcal{C}_2$. Then, because $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_i$ are linearly independent for every $i \in [k]$, $U\boldsymbol{v}_1, \ldots, U\boldsymbol{v}_i$ are also linearly independent. It therefore follows that

$$\lambda_i(\mathcal{C}_2) \leq \max_{j\in[i]}\|U\boldsymbol{v}_j\|_0 \leq \|U|_{\mathcal{C}_1}\|_{0\to 0} \cdot \max_{j\in[i]}\|\boldsymbol{v}_j\|_0 = \|U|_{\mathcal{C}_1}\|_{0\to 0}\lambda_i(\mathcal{C}_1) \ .$$

This yields the bound $\frac{\lambda_i(\mathcal{C}_2)}{\lambda_i(\mathcal{C}_1)} \leq \|U|_{\mathcal{C}_1}\|_{0\to 0}$ for all i, which implies $\max_{i\in[k]} \frac{\lambda_i(\mathcal{C}_2)}{\lambda_i(\mathcal{C}_1)} \leq \|U|_{\mathcal{C}_1}\|_{0\to 0}$. A symmetrical argument with $(U^{-1})|_{\mathcal{C}_2}$ mapping successive minima vectors of $\mathcal{C}_2$ to $\mathcal{C}_1$ yields $\max_{i\in[n]} \frac{\lambda_i(\mathcal{C}_1)}{\lambda_i(\mathcal{C}_2)} \leq \|U^{-1}|_{\mathcal{C}_2}\|_{0\to 0}$, and multiplying the bounds together shows that

$$\left(\max_{i\in[n]} \frac{\lambda_i(\mathcal{C}_2)}{\lambda_i(\mathcal{C}_1)}\right)\left(\max_{i\in[n]} \frac{\lambda_i(\mathcal{C}_1)}{\lambda_i(\mathcal{C}_2)}\right) \leq \|U|_{\mathcal{C}_1}\|_{0\to 0}\|U^{-1}|_{\mathcal{C}_2}\|_{0\to 0}.$$

Because U was chosen arbitrarily, we then have that $M(\mathcal{C}_1, \mathcal{C}_2)M(\mathcal{C}_2, \mathcal{C}_1) \leq \mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$. ◀

We now restate and prove Theorem 2, which gives our main approximation algorithm for CDP_q.

▶ **Theorem 2.** *For any prime power q, there is an algorithm for k^2-*CDP *on $[n,k]_q$ codes that runs in $O^*(q^k)$ time and $\mathrm{poly}(n, \log q)$ space.*

Proof. The algorithm works by computing successive minima bases G_1 and G_2 for $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, using Lemma 6, and then computing and outputting a linear transformation T such that $TG_1 = G_2$. By Lemma 6, computing G_1 and G_2 takes $O^*(q^k)$ time and polynomial space, and using these bases it is efficient to compute T. Furthermore, $\mathcal{D}_T(\mathcal{C}_1, \mathcal{C}_2) \leq k^2 \cdot \mathcal{D}(\mathcal{C}_1, \mathcal{C}_2)$ by Theorem 30. The theorem follows. ◀

References

1. Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, , and Wen Wang. Classic McEliece, 2022. NIST Post-Quantum Cryptography Standardization Project submission.
2. 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. Preliminary version in FOCS 1993. doi:10.1006/JCSS.1997.1472.
3. Per Austrin and Subhash Khot. A simple deterministic reduction for the gap minimum distance of code problem. *IEEE Trans. Inf. Theory*, 2014. Preliminary verison in ICALP 2011. doi:10.1109/TIT.2014.2340869.
4. László Babai, Paolo Codenotti, Joshua A. Grochow, and Youming Qiao. Code equivalence and group isomorphism. In *SODA*, 2011.
5. Alessandro Barenghi, Jean-François Biasse, Edoardo Persichetti, and Paolo Santini. LESS-FM: fine-tuning signatures from the code equivalence problem. In *PQCrypto*, 2021.
6. Alessandro Barenghi, Jean-François Biasse, Edoardo Persichetti, and Paolo Santini. On the computational hardness of the code equivalence problem in cryptography. *Adv. Math. Commun.*, 17(1):23–55, 2023. doi:10.3934/AMC.2022064.
7. Huck Bennett, Drisana Bhatia, Jean-François Biasse, Medha Durisheti, Lucas LaBuff, Vincenzo Pallozzi Lavorante, and Philip Waitkevich. Asymptotic improvements to provable algorithms for the code equivalence problem. *IEEE Transactions on Information Theory*, 72(2):1093–1108, 2026. doi:10.1109/TIT.2025.3637683.
8. Huck Bennett, Daniel Dadush, and Noah Stephens-Davidowitz. On the lattice distortion problem. In *ESA*, 2016.
9. Huck Bennett and Kaung Myat Htay Win. Relating code equivalence to other isomorphism problems. *Designs, Codes and Cryptography*, 93(3):701–723, December 2024. doi:10.1007/S10623-024-01542-3.
10. Ward Beullens. Not enough LESS: an improved algorithm for solving code equivalence problems over $\mathbb{F}_q$. In *SAC*, 2020.
11. Vijay Bhattiprolu, Mrinalkanti Ghosh, Venkatesan Guruswami, Euiwoong Lee, and Madhur Tulsiani. Approximability of $p \to q$ matrix norms: Generalized krivine rounding and hypercontractive hardness. In *SODA*, 2019.
12. Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. LESS is more: Code-based signatures without syndromes. In *AFRICACRYPT*, volume 12174, pages 45–65. Springer, 2020. doi:10.1007/978-3-030-51938-4_3.
13. Qi Cheng and Daqing Wan. A deterministic reduction for the gap minimum distance problem. *IEEE Trans. Inf. Theory*, 58(11):6935–6941, 2012. Preliminary version in STOC 2009. doi:10.1109/TIT.2012.2209198.

14 Thomas Debris-Alazard, Léo Ducas, and Wessel P. J. van Woerden. An algorithmic reduction theory for binary codes: LLL and more. *IEEE Trans. Inf. Theory*, 68(5):3426–3444, 2022. doi:10.1109/TIT.2022.3143620.

15 Léo Ducas and Shane Gibbons. Hull attacks on the lattice isomorphism problem. In *PKC*, 2023.

16 Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Inf. Theory*, 49(1):22–37, 2003. Preliminary version in FOCS 1999. doi:10.1109/TIT.2002.806118.

17 Philippe Gaborit, Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Edoardo Persichetti, Gilles Zémor, Jurjen Bos, Arnaud Dion, Jérôme Lacan, Jean-Marc Robert, Pascal Véron, Paulo L. Barreto, Santosh Ghosh, Shay Gueron, Tim Güneysu, Rafael Misoczki, Jan Richter-Brokmann, Nicolas Sendrier, Jean-Pierre Tillich, and Valentin Vasseur. HQC cryptosystem specification. https://pqc-hqc.org/doc/hqc_specifications_2025_08_22.pdf, 2025. Version dated 2025-08-22.

18 Surendra Ghentiyala and Noah Stephens-Davidowitz. More basis reduction for linear codes: Backward reduction, BKZ, slide reduction, and more. In *APPROX*, 2024.

19 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.

20 Alexander Golovnev and Noah Stephens-Davidowitz. Personal communication, 2026.

21 J. Leon. Computing automorphism groups of error-correcting codes. *IEEE Transactions on Information Theory*, 28(3):496–511, 1982. doi:10.1109/TIT.1982.1056498.

22 Jiří Matoušek. Lecture notes on metric embeddings. Lecture notes, Charles University, 2013. Available at https://kam.mff.cuni.cz/~matousek/ba-a4.pdf.

23 Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory, 1978. DSN Progress Report.

24 Daniele Micciancio. Locally dense codes. In *CCC*, 2014.

25 Bryant Morrell. Adventures with code distortion: Hardness and approximation. Master's thesis, University of Colorado Boulder, 2026.

26 National Institute of Standards and Technology (NIST). Post-Quantum Cryptography: Additional Digital Signature Schemes — Round 2 Additional Signatures. https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures, 2025.

27 Julian Nowakowski. An improved algorithm for code equivalence. In *PQCrypto*, pages 71–103, 2025. doi:10.1007/978-3-031-86599-2_3.

28 E. Petrank and R.M. Roth. Is code equivalence easy to decide? *IEEE Transactions on Information Theory*, 43(5):1602–1604, 1997. doi:10.1109/18.623157.

29 Oded Regev, 2014. Personal communication.

30 Nicolas Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Trans. Inf. Theory*, 46(4):1193–1203, 2000. doi:10.1109/18.850662.

31 Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 238–251, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3055399.3055408.