On the Hardness of Approximating Distances of Quantum Codes

Elena Grigorescu
David R. Cheriton School of Computer Science, University of Waterloo, Canada

Vatsal Jha
Department of Computer Science, Purdue University, West Lafayette, IN, USA

Eric Samperton
Departments of Mathematics and Computer Science, Purdue Quantum Science and Engineering Institute, Purdue University, West Lafayette, IN, USA

Abstract

The problem of computing distances of error-correcting codes is fundamental in both the classical and quantum settings. While hardness for the classical version of these problems has been known for some time (in both the exact and approximate settings), it was only recently that Kapshikar and Kundu showed these problems are also hard in the quantum setting. As our first main result, we reprove this using arguably simpler arguments based on hypergraph product codes. In particular, we get a direct reduction to CSS codes, the most commonly used type of quantum code, from the minimum distance problem for classical linear codes.

Our second set of results considers the distance of a graph state, which is a key parameter for quantum codes obtained via the codeword stabilized formalism. We show that it is NP-hard to compute/approximate the distance of a graph state when the adjacency matrix of the graph is the input. In fact, we show this is true even if we only consider X-type errors of a graph state. Our techniques moreover imply an interesting classical consequence: the hardness of computing or approximating the distance of classical codes with rate equal to 1/2.

One of the main motivations of the present work is a question raised by Kapshikar and Kundu concerning the NP-hardness of approximation when there is an additive error proportional to a quantum code's length. We show that no such hardness can hold for hypergraph product codes. These observations suggest the possibility of a new kind of square root barrier.

2012 ACM Subject Classification Theory of computation → Error-correcting codes

Keywords and phrases quantum codes, minimum distance problem, NP-hardness, graph state distance

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2025.34

Related Version *Full Version*: https://arxiv.org/abs/2509.21469

Funding *Elena Grigorescu*: Supported in part by NSF CCF-2228814 while at Purdue University.
Vatsal Jha: Supported in part by NSF CCF-2228814, NSF CCF-2330130, NSF CCF-2127806 and ONR Award N00014-24-1-2695.
Eric Samperton: Supported in part by NSF CCF-2330130.

Acknowledgements We thank Xuandi Ren for helpful correspondence, and some anonymous reviewers for valuable feedback.

1 Introduction

1.1 Quantum and classical distance problems

Scalable, fault-tolerant quantum computation is expected to require families of quantum error-correcting codes with larger and larger distance parameters, and CSS codes – introduced in [9] and [25] – provide a powerful and well-studied framework for building such families.

In both the classical and quantum settings, it is a problem of fundamental importance to understand how best to compute distances of codes. Recently, Kapshikar and Kundu [19] showed that it is NP-hard to compute the distance of CSS codes, both in the exact sense and in various approximate senses (with various notions of polynomial-time reduction). Their proof relies heavily on the codeword-stabilized (CWS) framework, where a quantum code on n-qubits is defined using a graph G on n-vertices and a binary classical code C of length n. The goal of the present paper is to study the hardness of other related metrics for CWS codes, namely the minimum graph state distance associated with the graph G. To do this, we consider a problem in classical coding theory, which we refer to as the MinDistDualDist problem, and then prove its hardness. The latter problem seems to be of independent interest as it also provides the hardness of computing the distance of classical codes with rate equal to 1/2. Along with it, we provide simpler proofs of the results in [19] and try to shed light on the hardness of approximating distances with an additive error that is proportional to a code's length. In particular, the constructions used in the reduction of [19] hint at the possibility of a new kind of "square-root barrier" that we aim to better understand.

Before stating our results, let us first recall some of the basic notions of quantum and classical codes (if only to set notation), as well as formulate the precise distance problems that interest us. Despite being examples of quantum codes, CSS codes are defined using certain pairs of *classical* codes, so we review classical codes first. (Perhaps the most important insight of the early efforts on quantum error correction was that one can reduce the problem of their construction to certain slightly unusual constructions with classical codes.)

A *(classical, binary) code* C of *length* n is any subset $C \subseteq \mathbb{F}_2^n$, where $\mathbb{F}_2 = \{0, 1\}$ is the binary field and $\mathbb{F}_2^n$ the n-dimensional $\mathbb{F}_2$-vector space consisting of all row vectors of length n. If $C \leq \mathbb{F}_2^n$ is a linear subspace, then we call C a *linear* code. All classical codes considered in this paper will be linear, and so we will often not use the word "linear" when we should. The two most important parameters of a classical code are its *dimension* $k \stackrel{\text{def}}{=} \dim_{\mathbb{F}_2} C$ and its *distance*

$$d \stackrel{\text{def}}{=} \min\{d_H(x, y) \mid x, y \in C, x \neq y\} = \min\{d_H(x, 0) \mid x \in C, x \neq 0\},$$

where $d_H(x, y) \stackrel{\text{def}}{=} |\{i \in [n] \mid x_i \neq y_i\}|$ is the Hamming distance between vectors in $\mathbb{F}_2^n$. A classical linear code with such parameters is called an $[n, k, d]$ code.

There are two standard ways to present a classical linear code: with a *generator matrix* G, or with a *parity-check matrix* H. The former is any binary matrix whose row space equals C, while the latter is any matrix whose kernel equals C – or, more precisely (due to the difference between row and column vectors),

$$C = \{x \in \mathbb{F}_2^n \mid Hx^T = 0\}.$$

For a given parity-check matrix H, we let $C(H)$ denote the code corresponding to H. Parity-check matrices can be converted to generator matrices, and *vice versa*, in polynomial time. Moreover, when convenient, we may assume without loss of generality that the parity check matrix H of a $[n, k, d]$ code is a $(n - k) \times n$ matrix (in particular, has full rank $n - k$). A natural way of specifying a parity-check matrix H is via the systematic form where $H = [I_{n-k} : P_{(n-k)\times k}]$.[1] Given the parity-check matrix in systematic form, $G = [P^T_{k\times(n-k)} : I_k]$ will be a generator matrix for C.

The problem of calculating the distance of a classical code from its parity-check matrix has the following standard decision variant.

[1] For matrices A and B with equal number of rows, [A:B] refers to their augmented matrix

Classical Minimum Distance Decision Problem (MinDist)
Instance: A binary matrix $H \in \mathbb{F}_2^{(n-k)\times n}$ and a non-negative integer t.
Output: YES if $d \leq t$. NO otherwise.

There are also important approximation variants of MinDist, either with a multiplicative error or an additive error. The decision variants of these are most conveniently formulated as promise problems with a gap. Let $\gamma \geq 1$ and $\tau > 0$.

Classical Minimum Distance Decision Problem with Multiplicative Gap γ ($\text{MultGapDist}_\gamma$)
Instance: A binary matrix $H \in \mathbb{F}_2^{n-k\times n}$ and a non-negative integer t.
Promise: Either $d \leq t$ or $d > \gamma t$.
Output: YES if $d \leq t$. NO if $d > \gamma t$.

Classical Minimum Distance Decision Problem with Additive Gap τn (AddGapDist_τ)
Instance: A binary matrix $H \in \mathbb{F}_2^{n-k\times n}$ and a non-negative integer t.
Promise: Either $d \leq t$ or $d > t + \tau n$.
Output: YES if $d \leq t$. NO if $d > t + \tau n$.

All three of these problems have been studied extensively, but the first two have received the most attention. The question of the complexity of MinDist was first raised in 1978 [2], and Vardy eventually showed that MinDist is NP-hard via deterministic Karp reduction [27]. Dumer, Miccancio, and Sudan later showed that $\text{MultGapDist}_\gamma$ and AddGapDist_τ are both NP-hard under RUR reduction for all $\gamma \geq 1$ and some $\tau > 0$ [14]. In the meantime, there have been various improvements [10, 1, 23, 4, 3], and it is now known that $\text{MultGapDist}_\gamma$ is NP-hard under deterministic Karp reduction. Remark 15 of [3] indicates that there is now a deterministic proof of the relatively near codeword problem in a specific parameter regime, but this does not appear to give the necessary result to derandomize the RUR reduction of [14] in the case of additive gaps (which goes through RNC). Nevertheless, Xuandi Ren kindly explained to us that the methods of [3] are in fact sufficient to show that AddGapDist_τ is NP-hard under deterministic Karp reduction for some $\tau > 0$ [24].

We need one more notion from classical codes before we can properly introduce CSS codes. Given a classical code C of length n, the *dual code* $C^\perp$ is another length n code defined as follows:

$$C^\perp \stackrel{\text{def}}{=} \{c' \in \mathbb{F}_2^n | \langle c', c\rangle = 0, \text{ for all } c \in C\}$$

where $\langle c', c\rangle$ is the usual $\mathbb{F}_2$ dot product. It is easy to see that a generator matrix for C is a parity-check matrix for $C^\perp$. In particular, if H is any parity-check matrix of a code C and G is a generator matrix for C, then $HG^T = 0$.

We are finally ready to briefly recall the description of quantum CSS codes.

To start, let us note that, in general, a *quantum error correcting code of length* n is any $\mathbb{C}$-linear subspace $\mathcal{C}$ of the Hilbert space of n qubits $(\mathbb{C}^2)^{\otimes n}$.[2] Similar to the classical setting, the *minimum distance* d_Q *of a quantum error-correcting code* is defined to be the minimum

[2] Thus, naively, every quantum error correcting code is "linear." However, the more important distinction in the quantum setting is between "additive" and "non-additive" codes. Additive quantum codes are understood as the proper quantum analog of linear classical codes, and non-additive quantum codes as the analogs of non-linear classical codes. We will not get into the details of this distinction here, except to note that all Pauli stabilizer codes are additive, and CSS codes are an important special case of Pauli stabilizer codes.

number of qubits where non-trivial errors must occur in order to effect a non-trivial logical error on the code-space. We call a quantum code of length n, with $\dim_{\mathbb{C}} \mathcal{C} = K$ and distance d a $((n, K, d))$ quantum code. If $K = 2^k$ happens to be a power of 2, then we call k the number of *logical qubits* in the code, and call the code a $[[n, k, d]]$ quantum code.

The seminal works [9] and [25] showed how to build certain quantum error-correcting codes – now called *CSS codes* – using any pair of classical binary linear codes (C_1, C_2) with $C_2^{\perp} \leq C_1$. If C_1 and C_2 have parameters $[n, k_1, d_1]$ and $[n, k_2, d_2]$, then the CSS code $CSS(C_1, C_2)$ has parameters $[[n, k_1 + k_2 - n, d_Q]]$ where

$$d_Q \stackrel{\text{def}}{=} \min\{wt_H(a) : a \in (C_1 \setminus C_2^{\perp}) \cup (C_2 \setminus C_1^{\perp})\}$$

is the (quantum) distance of $CSS(C_1, C_2)$. If the parity check matrices of C_1 and C_2 are H_1 and H_2, then the condition that $C_2^{\perp} \leq C_1$ is equivalent to to the condition $H_1 H_2^T = O$, and we will write $CSS(H_1, H_2)$ to mean $CSS(C_1, C_2)$, and may refer to H_1 and H_2 as the *quantum parity-check matrices* of the CSS code.

The quantum distance problems of primary interest in this work are the CSS analogs of the classical problems MinDist, MultGapDist and AddGapDist. Generalizing the first two is completely straightforward. As before, let $\gamma \geq 1$.

CSS Minimum Distance Decision Problem (CSSMinDist)
Instance: $H_X \in \mathbb{F}_2^{n-k_1 \times n}$ and $H_Z \in \mathbb{F}_2^{n-k_2 \times n}$ with $H_X H_Z^T = 0$, and a non-negative integer t.
Output: YES if $d_Q \leq t$. NO otherwise.

CSS Minimum Distance Decision Problem with Multiplicative Gap γ (MultGapCSSDist$_\gamma$)
Instance: $H_X \in \mathbb{F}_2^{n-k_1 \times n}$ and $H_Z \in \mathbb{F}_2^{n-k_2 \times n}$ with $H_X H_Z^T = 0$, and a non-negative integer t.
Promise: Either $d_Q \leq t$ or $d_Q > \gamma t$.
Output: YES if $d_Q \leq t$. NO if $d_Q > \gamma t$.

There is an important subtlety in generalizing AddGapDist to the CSS setting. We need *two* parameters for the problem: $\tau > 0$ as before and a new $\epsilon > 0$.

CSS Minimum Distance Decision Problem with Additive Gap τn^ϵ (AddGapCSSDist$_{\tau,\epsilon}$)
Instance: $H_X \in \mathbb{F}_2^{n-k_1 \times n}$ and $H_Z \in \mathbb{F}_2^{n-k_2 \times n}$ with $H_X H_Z^T = 0$, and a non-negative integer t.
Promise: Either $d_Q \leq t$ or $d_Q > t + \tau n^\epsilon$.
Output: YES if $d_Q \leq t$. NO if $d_Q > t + \tau n^\epsilon$.

As far as we are aware, the work of Kapshikar and Kundu [19] is the first to study these three problems explicitly, and they showed each is NP-hard. More precisely, they showed that CSSMinDist is NP-hard under Karp reduction, MultGapCSSDist$_\gamma$ is NP-hard under polynomial-time RUR reduction for every $\gamma \geq 1$, and there exists a $\tau > 0$ such that AddGapCSSDist$_{\tau,\epsilon}$ is NP-hard under polynomial time RUR reduction for every $\epsilon < \frac{1}{2}$. In fact, thanks to [10, 1, 3], their work is sufficient to establish that MultGapCSSDist$_\gamma$ is NP-hard under deterministic Karp reduction. Similarly, [3, 24] imply that AddGapCSSDist$_{\tau,\epsilon}$ (for some τ and all $\epsilon \leq \frac{1}{2}$) is hard under deterministic Karp reduction. We summarize these results:

▶ **Theorem 1** ([19]). *Each of the following is* NP*-hard under Karp reduction:*

- CSSMinDist
- MultGapCSSDist$_\gamma$ *for all* $\gamma \geq 1$
- AddGapCSSDist$_{\tau,\epsilon}$*, for each* $0 < \epsilon \leq \frac{1}{2}$ *and some* $\tau > 0$ *(depending on* ϵ*)*

Intriguingly, there are as yet no known hardness results for AddGapCSSDist$_{\tau,\epsilon}$ when $\frac{1}{2} < \epsilon \leq 1$. We initiated the present work as a step towards attempting to resolve this matter, and include extensive discussion in Section 5. The reader is particularly encouraged to consider Proposition 22 in Subsection 5.1 after reading the next subsection.

1.2 Our results

Our first main result, presented in Section 3, is a new proof of Theorem 1. Where Kapshikar and Kundu employ the codeword stabilized (CWS) formalism introduced in [12], we employ the hypergraph product (HGP) construction of Tillich and Zémor [26]. We quote directly from Kapshikar and Kundu in order to identify the basic difficulty that must be overcome in order to reduce the classical MinDist problem to the quantum CSSMinDist problem:

> Note that, due to the orthogonality condition on CSS codes, one can not use an arbitrary pair C_1, C_2. If we want to reduce the classical minimum distance problem, starting with C_1, we need to find a code C_2, such that, C_2 satisfies the orthogonality condition and has minimum distance not less than C_1. One way to get around this is to use self-dual (or weakly self-dual) classical codes in the CSS construction. But it is not clear whether the hardness result for classical codes still holds under the restriction that the code is self-dual (or weakly self-dual) [19].

We now compare and contrast the two constructions.

On one hand, the great insight of [19] is that one can go around the above obstacle by using the CWS formalism – rather than CSS codes – to build a (non-CSS) Pauli stabilizer code $CWS(C, G)$ on n qubits in a way that combines a classical linear code C with a graph G, with no restrictions on C and G other than that the number of vertices in G equal the length of C. With some additional deterministic polynomial time overhead, this Pauli stabilizer code can then be converted to a CSS code with the same parameters [5]. The delicate part when using this construction to reduce MinDist to CSSMinDist is then the choice of the graph G. For the CWS code based on C and G, the quantum distance d_Q is known to satisfy the inequality $d_Q \leq \min\{d, d_G\}$ where d is the classical distance of C and d_G is the *minimum graph state distance*

$$d_G \stackrel{\text{def}}{=} \min\{wt_H(x \vee z) : A_G x^T = z^T\}$$

(here A_G is the adjacency matrix of G). The reduction from MinDist to CSSMinDist in [19] ultimately succeeds by combining an input code C to MinDist with a graph G from a very special family of C_4-free graphs introduced in [15] that appear to be close to "optimal" [16]. With such a carefully chosen G, one gets that the quantum distance d_Q of $CWS(C, G)$ equals the classical distance d of C, which implies Theorem 1.

On the other hand, rather than combine a graph with a classical code, the *hypergraph product code construction* can combine *any* two classical linear codes (of *any* lengths) in a way that directly yields a CSS code [26]. Specifically, if H_1 and H_2 are the parity-check matrices for C_1 and C_2, then the *hypergraph product code* $HGP(H_1, H_2)$ is the CSS code with quantum parity check matrices H_1' and H_2' defined as follows:

$$H_1' \stackrel{\text{def}}{=} [H_1 \otimes I : I \otimes H_2^T], \qquad H_2' \stackrel{\text{def}}{=} [I \otimes H_2 : H_1^T \otimes I].$$

It is straightforward to verify that $H_1'(H_2')^T = 0$, and hence, this defines a valid CSS code. Moreover, if H_1 and H_2 have full rank and C_1 and C_2 have parameters $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$, then [26] show that $HGP(H_1, H_2)$ is a code with parameters

$$[[n_1 n_2 + (n_1 - k_1)(n_2 - k_2), k_1 k_2, \min\{d_1, d_2\}]].$$

Put succinctly, when given full rank classical parity-check matrices, we have

$$HGP([n_1, k_1, d_1], [n_2, k_2, d_2]) = [[n_1 n_2 + (n_1 - k_1)(n_2 - k_2), k_1 k_2, \min\{d_1, d_2\}]] \quad (1)$$

Our proof of Theorem 1 is then hardly delicate: given a classical code C_1 with parity check matrix H_1, we choose H_2 simply to be the parity-check matrix of a repetition code of an appropriate length. The details are in Section 3. The basic idea is similar in spirit to the tensor-based techniques of [1] used to establish a deterministic reduction for the classical MultGapDist problem.

Before moving on to our other results, we note that the HGP construction is an important technique, as it was the first to break the square root barrier of distance at a constant rate. This construction was thus an important precursor to a long line of works that culminated in the proof of existence of "good" quantum LDPC codes [6], [20] and [13], and so it is desirable to have direct methods for exhibiting the intrinsic hardness of problems involving HGP codes.

Our second main set of results are related to graph state distance itself, which, as we already saw, is an important primitive in the CWS code construction. More generally, graph states have applications in quantum secret-sharing, quantum metrology and even quantum error-correction. They are extensively studied in [22], [18], [7], for example. While their precise definition is not so important for the present work, we note here that a graph state is a kind of Pauli stabilizer state, and thus can be understood as a quantum code that encodes 0 logical qubits (in other words, a single fault-tolerant quantum state). For interested readers, we include some details in Section 2.

A trivial upper bound on the minimum distance of the graph state corresponding to a graph $G = (V, E)$ is $\delta_G \stackrel{\text{def}}{=} \min_{v \in V} deg(v) + 1$. However, it can happen that $d_G \ll \delta_G$. Indeed, consider the graph $G = (V, E)$ one gets by removing a single edge – say, $\{v_1, v_n\}$ – from the complete graph K_n on n vertices. Then $\delta_G = n - 1$, but $d_G = 2$.

With these observations in hand, it is natural to consider the following distance problems for graph states.

Graph State Minimum Distance Decision Problem (GraphMinDist)
Instance: The adjacency matrix A_G of a simple graph and a non-negative integer t.
Output: YES if $d_G \leq t$. NO otherwise.

Graph State Minimum Distance Decision Problem with Multiplicative Gap γ ($\text{MultGapGraphDist}_\gamma$)
Instance: The adjacency matrix A_G of a simple graph and a non-negative integer t.
Promise: Either $d_G \leq t$ or $d_G > \gamma t$.
Output: YES if $d_G \leq t$ and NO if $d_G > \gamma t$.

Graph State Minimum Distance Decision Problem with Additive Gap τn^ϵ ($\text{AddGapGraphDist}_{\tau,\epsilon}$)
Instance: The adjacency matrix A_G of a simple graph and a non-negative integer t.
Promise: Either $d_G \leq t$ or $d_G > t + \tau n^\epsilon$.
Output: YES if $d_G \leq t$ and NO if $d_G > t + \tau n^\epsilon$.

▶ **Theorem 2.** *Each of the following is* NP*-hard under Karp reduction:*

- GraphMinDist
- MultGapGraphDist$_\gamma$ *for all* $\gamma \geq 1$
- AddGapGraphDist$_{\tau,\epsilon}$ *for each* $0 < \epsilon \leq \frac{1}{3}$ *and some* $\tau > 0$ *(depending on* ϵ*)*

Interestingly, even computing the promise version of GraphMinDist where $z = 0$ is NP-hard. More precisely, define the *graph state* X*-distance* $d_{X,G}$ by

$$d_{X,G} \stackrel{\text{def}}{=} \min\{wt_H(x) : x \in \mathbb{F}_2^n, A_G x^T = 0_{n\times 1}\}.$$

We can then define decision problems GraphMinDist^X, MultGapGraphDist$^X_\gamma$ and AddGapGraphDist$^X_{\tau,\epsilon}$ exactly as before, but with d_G replaced by $d_{X,G}$.

▶ **Theorem 3.** *Each of the following is* NP*-hard under Karp reduction:*

- GraphMinDist^X
- MultGapGraphDist$^X_\gamma$ *for all* $\gamma \geq 1$
- AddGapGraphDist$^X_{\tau,\epsilon}$ *for each* $0 < \epsilon \leq \frac{1}{3}$ *and some* $\tau > 0$ *(depending on* ϵ*)*

Unlike our proof of Theorem 1, our proofs of Theorems 2 and 3 do not use the hypergraph product construction. Instead, we use the tensor product of codes, together with the fact that simultaneously minimizing the distance of a (classical) code and its dual – a problem we call MinDistDualDist and define formally below – is hard. To prove this latter fact, we use an interesting recent construction of codes whose distance and dual distance are both "large" [21]. See Lemmas 17 and 18 for details. The hardness of MinDistDualDist furthermore implies another result in classical coding theory: the problem of bounding the distance of binary codes with rate equal to 1/2 is NP-Complete. To the best of our knowledge this does not appear in any earlier literature.

1.3 Organization

Section 2 briefly reviews the the Pauli stabilizer formalism and more carefully defines graph states. This section is not really needed for our proofs, but is included for interested readers. We prove Theorem 1 in Section 3. Theorems 2 and 3 are proved in Subsections 4.1 and 4.2, respectively. The final Section 5 includes extensive discussion, especially concerning the matter of hardness for AddGapCSSDist$_{\tau,\epsilon}$ when $\epsilon > \frac{1}{2}$.

2 Preliminaries

As we have already discussed the formalism for classical error-correcting codes in the introduction, we directly discuss the formalism for quantum error-correcting codes along with stabilizer formalism for quantum codes introduced in [17] and [8].

The state space of a single qubit is the two-dimensional Hilbert space, $\mathbb{C}^2$ and the n-qubit state space is the n-fold tensor product $(\mathbb{C}^2)^{\otimes n}$. As the state space of a single qubit is a 2-dimensional Hilbert space, hence any state $|\psi\rangle$ of a single qubit can be denoted by:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \text{ where } |\alpha|^2 + |\beta|^2 = 1 \text{ and } |0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}.$$

As the state space of n-qubits is the n-fold tensor product of $\mathbb{C}^2$ i.e. $(\mathbb{C}^2)^{\otimes n}$. Hence, a state $|\psi\rangle$ of n$-$*qubits* can be represented by:

$$|\psi\rangle = \sum_{x\in\mathbb{F}_2^n} \lambda_x\,|x\rangle\,,$$

such that $\sum_{x\in\mathbb{F}_2^n} |\lambda_x|^2 = 1$.

▶ **Definition 4.** *The Pauli Group on n-qubits is defined as:*

$$\mathcal{P}_n = \{i^\lambda M_1 \otimes M_2 \otimes ... \otimes M_n : \lambda \in \{0,1,2,3\},\ M_i \in \{I, X, Y, Z\} \text{ for all } i \in [n]\},$$

where $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ *and* $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$.

As $Y = iXZ$, hence, for every $g \in \mathcal{P}_n$ there exist $a, b \in \mathbb{F}_2^n$ such that $g = i^\lambda \bigotimes_{i=1}^n X^{a_i} Z^{b_i}$. A compact way to write this is $g = i^\lambda X(a)Z(b)$.

▶ **Definition 5.** *A stabilizer subgroup, $\mathcal{S}$, of $\mathcal{P}_n$ is an abelian subgroup not containing $-I$.*

▶ **Definition 6.** *For a stabilizer subgroup S of the Pauli group $\mathcal{P}_n$, the stabilizer code $C(\mathcal{S})$ associated with it is the joint $+1$-eigenspace of the operators in S:*

$$C(\mathcal{S}) = \{|\psi\rangle :\ g\,|\psi\rangle = |\psi\rangle \ \text{ for all } g \in S\}.$$

Moreover, a stabilizer code $\mathcal{S}$ is said to encode k-logical qubits if $dim(C(\mathcal{S})) = 2^k$

▶ **Fact 7.** *For a stabilizer group of order 2^k, the corresponding stabilizer code has dimension 2^{n-k}.*

▶ **Fact 8.** *For any operator $A \in M_{n\times n}(\mathbb{C})$ acting on $n-$qubits, we have:*

$$A = \sum_{a,b \in \mathbb{F}_2^n} \lambda_{a,b} X(a)Z(b),$$

where $\lambda_{a,b} \in \mathbb{C}$. Furthermore, the above representation of A as a sum of $X(a)Z(b)$ is unique.

▶ **Definition 9.** *The weight of an operator, $E \in M(\mathbb{C}_{n\times n})$, with representation:*

$$E = \sum_{a,b \in \mathbb{F}_2^n} \lambda_{a,b} X(a)Z(b),$$

is defined as:

$$wt(E) \stackrel{def}{=} \max\{wt_H(a \vee b) : \lambda_{a,b} \neq 0\}.$$

We can now define the distance of a quantum code.

▶ **Definition 10.** *For a quantum code on n qubits, it is said to have a distance d, if it can correct all errors E of weight $\leq \lfloor \frac{d-1}{2} \rfloor$.*

We now define the graph state, which we touched on in the introduction.

▶ **Definition 11.** *For a graph G on n-vertices, specified by an adjacency matrix A_G, the graph state is defined as the stabilizer code corresponding to the stabilizer group:*

$$\langle X(e_i)Z(u_i) : i \in [n]\rangle.$$

where e_i is the i^{th} standard basis vector and u_i is the i^{th} row of A_G.

We now define the graph state, which we touched on in the introduction.

▶ **Definition 12.** *For a graph G on n-vertices, specified by an adjacency matrix A_G, the graph state is defined as the stabilizer code corresponding to the stabilizer group:*

$$\langle X(e_i)Z(u_i) : i \in [n]\rangle.$$

where e_i is the i^{th} standard basis vector and u_i is the i^{th} row of A_G.

3 Proof of Theorem 1

Theorem 1 says that each of the three different problems CSSMinDist, MultGapCSSDist and AddGapCSSDist is NP-hard, and so we must provide three reductions. Each will start from the classical analogs MinDist, MultGapDist and AddGapDist, respectively.

To see that CSSMinDist is NP-hard, consider an input instance (H_1, t) of the MinDist problem, where H_1 is the parity-check matrix of a classical $[n, k, d]$ code C. We may assume H_1 is full rank (if it is not, then we may, in polynomial time, replace it with a new, smaller parity-check matrix that is). Now let H_2 be the (full rank) parity-check matrix for the classical repetition code of length n, which has parameters $[n, 1, n]$. We reduce to the hypergraph product code $HGP(H_1, H_2)$. By Equation 1, this hypergraph product code has parameters $[[n^2 + (n-k)(n-1), k, d]]$. This implies the hardness of CSSMinDist.

The reduction just shown builds a quantum code whose minimum distance equals the distance of the original classical code. Hence, even finding a constant multiplicative approximation to CSSMinDist is NP-hard. That is, MultGapCSSDist is NP-hard.

Finally, to prove the hardness of $\text{AddGapCSSDist}_{\tau,\epsilon}$, we reduce from AddGapDist_{τ}, which was shown to be NP-hard under RUR reduction for some τ by [14], although this argument can now be derandomized [3, 24].

To make things clearer, we define a variant of the classical AddGapDist_{τ} problem with an ϵ parameter.

Classical Minimum Distance Decision Problem with Additive Gap τn^{ϵ} ($\text{AddGapDist}_{\tau,\epsilon}$)
Instance : Binary matrix $H \in \mathbb{F}_2^{n-k\times n}$ and a positive integer t
Promise: Either $d \leq t$ or $d > t + \tau n^{\epsilon}$.
Output: YES if $d \leq t$ and NO if $d > t + \tau n^{\epsilon}$

▶ **Lemma 13.** *For every $\epsilon \in (0, 1]$, there exists $\tau \in (0, 1)$ for which $\text{AddGapDist}_{\tau,\epsilon}$ is NP-hard.*

Proof. To prove the lemma we will be using tensor codes, which are defined as follows:

▶ **Definition 14.** *The tensor product of two matrices*

$$A_{n\times m} := \begin{bmatrix} a_{11} & a_{12} & . & . & . & a_{1m} \\ a_{21} & a_{22} & . & . & . & a_{2m} \\ & & . & & & \\ & & . & & & \\ & & . & & & \\ a_{n1} & a_{n2} & . & . & . & a_{nm} \end{bmatrix} \quad \text{and} \quad B_{p\times q} := \begin{bmatrix} b_{11} & b_{12} & . & . & . & b_{1q} \\ b_{21} & b_{22} & . & . & . & b_{2q} \\ & & . & & & \\ & & . & & & \\ & & . & & & \\ b_{p1} & b_{p2} & . & . & . & b_{pq} \end{bmatrix}$$

is defined as:

$$A \otimes B := \begin{bmatrix} a_{11}B & a_{12}B & . & . & . & a_{1m}B \\ a_{21}B & a_{22}B & . & . & . & a_{2m}B \\ & & . & & & \\ & & . & & & \\ & & . & & & \\ a_{n1}B & a_{n2}B & . & . & . & a_{nm}B \end{bmatrix}$$

▶ **Definition 15.** *For classical codes C_1, C_2 with parameters $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$ and generator matrices G_1, G_2 respectively, the tensor product code, $C_1 \otimes C_2$, is defined as the code corresponding to the generator matrix $G_1 \otimes G_2$.*

We will use the following fact about tensor product codes:

▶ **Fact 16.** *The tensor product of classical codes, C_1 and C_2 with parameters $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$ (respectively) is a classical code with parameters $[n_1 n_2, k_1 k_2, d_1 d_2]$. Moreover, $(C_1 \otimes C_2)^{\perp} = C_1^{\perp} \otimes \mathbb{F}_2^{n_2} + \mathbb{F}_2^{n_1} \otimes C_2^{\perp}$ and has parameters $[n_1 n_2, n_1 n_2 - k_1 k_2, \min\{d_2, d_2\}]$ (See also Lemma 3.3 in [11]).*

We reduce from AddGapDist$_\tau$. Consider an instance (H, t) for the AddGapDist$_\tau$ problem, where τ is as defined in [14] (Theorem 32) such that AddGapDist$_\tau$ is NP-hard. Let C be the code of length n corresponding to the parity-check matrix H and let d be its minimum distance. Now, consider the tensor code $C^{'} = C \otimes \mathbb{F}_2^{\lceil n^{1/\epsilon - 1} \rceil}$. By Fact 16, it follows that the distance of $C^{'}$ is d and its length is $N = n \cdot \lceil n^{1/\epsilon - 1} \rceil$. We claim that $\min\{d, N^{\epsilon}\} = d$. Since $\min\{d, N^{\epsilon}\} \leq d$, it suffices to show that $\min\{d, N^{\epsilon}\} \geq d$, which follows from:

$$\begin{aligned} \min\{d, N^{\epsilon}\} &= \min\{d, (n \cdot \lceil n^{1/\epsilon - 1} \rceil)^{\epsilon}\} \\ &\geq \min\{d, (n \cdot n^{1/\epsilon - 1})^{\epsilon}\} \\ &= \min\{d, n\} \\ &= d. \end{aligned}$$

By definition $N = n \cdot \lceil n^{1/\epsilon - 1} \rceil$. Note that as $\lceil n^{1/\epsilon} \rceil \leq 2n^{1/\epsilon}$, we have $n^{1/\epsilon} \leq N \leq 2n^{1/\epsilon}$. If (H, t) is a YES instance for AddGapDist$_\tau$, then $d \leq t$, and as above it follows that $\min\{d, N^{\epsilon}\} \leq t$. If (H, t) is a NO instance for AddGapDist$_\tau$, then $\min\{d, N^{\epsilon}\} = d > t + \tau n \geq t + \tau^{'} N^{\epsilon}$, where $\tau' = \tau / 2^{\epsilon}$. This establishes the lemma. ◀

We now reduce the AddGapDist$_{\tau,\epsilon}$ problem to the AddGapCSSDist$_{\tau,\epsilon}$ problem, hence showing the hardness of the latter problem. For a given $\alpha \in (0, 1]$, consider the repetition code of length n^{α}. Let (H, t) be the input instance for AddGapDist$_{\tau,\epsilon}$ and let C be the code corresponding to parity-check matrix H having length n and distance d. The hypergraph product code obtained from the code C and the repetition code will be of length $n \cdot n^{\alpha} + (n - k) \cdot (n^{\alpha} - 1) = n^{'}$ and distance $d^{'} = \min\{d, n^{\alpha}\}$.

We claim that the above conversion of a classical code C to a hypergraph product code is a deterministic reduction from the AddGapDist$_{\tau,\epsilon}$ problem to the AddGapCSSDist$_{\tau,\epsilon}$ problem.

If (H, t) is a YES instance of AddGapDist$_{\tau,\epsilon}$, then $\min\{d, n^{\alpha}\} = d' \leq t$. This implies that the corresponding hypergraph product code has distance $d' \leq t$. If (H, t) is a NO instance, then $d' > t + \tau n^{\alpha} \geq t + \tau'(n')^{\frac{\alpha}{1+\alpha}}$, where $\tau' = \tau / 2^{\frac{\alpha}{1+\alpha}}$. This follows from the observation $n^{1+\alpha} \leq n' \leq 2n^{1+\alpha}$. Clearly, the gap between the YES instances and NO instances of AddGapCSSDist$_{\tau,\epsilon}$ is maximized for $\alpha = 1$, which is the case when the NO instance is $d' \geq t + \tau n^{1/2}$. ◀

4 Hardness for distances of graph states

4.1 Proof of Theorem 2

We now prove the hardness of computing the minimum graph state distance. We reduce MinDist to GraphDist via intermediate variants of MinDistDualDist. Similarly, to prove hardness for the gap version of GraphMinDist we will need the corresponding gap versions MultGapMinDistDualDist and AddGapDistDualDist as defined next.

Minimum Distance Dual Distance (MinDistDualDist)
Instance: Binary matrix $H \in \mathbb{F}_2^{n-k\times n}$ and positive integer t.
Output: If $\min\{d(C),\ d(C^\perp)\} \leq t$ then YES else NO.

Minimum Distance Dual Distance with Multiplicative Gap γ ($\text{MultGapMinDistDualDist}_\gamma$)
Instance: Binary matrix $H \in \mathbb{F}_2^{n-k\times n}$ and positive integer t.
Output: If $\min\{d(C),\ d(C^\perp)\} \leq t$ then YES and NO if $> \gamma t$.

Minimum Distance Dual Distance with Additive Gap τn (AddGapMinDistDualDist)
Instance: Binary matrix $H \in \mathbb{F}_2^{n-k\times n}$ and positive integer t.
Output: If $\min\{d(C),\ d(C^\perp)\} \leq t$ then YES and NO if $\min\{d(C),\ d(C^\perp)\} > t + \tau n^{1/3}$.

▶ **Lemma 17.** MinDistDualDist *is* NP*-hard. Furthermore, for every constant* $\gamma \geq 1$*, computing a* MultGapDistDualDist *approximation is* NP*-hard while there exists a* $\tau \in (0,1)$ *such that finding an additive approximation with error* $\tau n^{1/3}$ *is* NP*-hard.*

Proof. For a given parity-check matrix H, let C be the associated code to it with length n and minimum distance denoted by $d(C)$. Now, consider a code $C^{'}$, that comes from a family of codes $\{C_N^{'}\}$ where $C_N^{'}$ is a binary code of length N such that $\min\{d(C^{'}),\ d(C^{'})\} \geq \sqrt{N}$.

We consider the tensor code $\tilde{C} := C^\perp \otimes C^{'}$ and its dual $\tilde{C}^\perp := C \otimes \mathbb{F}_2^N + \mathbb{F}_2^n \otimes {C^{'}}^\perp$

One such family of codes was introduced in [21] (Theorem 2.9, Theorem 2.15).

▶ **Lemma 18** ([21])**.** *Let* $m \geq 5$ *be an odd integer with* $m \equiv 5 \mod 6$*. For every such* m*, there exists binary classical code* C_m *with parameters* $[2^m - 1, \frac{2^{m+1}-1}{3}, d \geq 2^{\frac{m-1}{2}} + 1]$*. Moreover,* $C_m^\perp$ *has parameters* $[2^m - 1, \frac{2^m-2}{3}, d^\perp \geq 2^{\frac{m-1}{2}} + 4]$*.*

While [21] does not provide any analysis of the time required to build their codes' parity-check matrices, one can check that this can be done efficiently. (To do so, one must use the fact that the finite field $\mathbb{F}_{2^m}$ of order 2^m with $m = O(\log n)$ can be constructed in time $O(poly(n))$.) We also use the observation that for every integer $n \geq 5$ there exists $m \equiv 5 \mod 6$ for which $2^m \leq n \leq 2^{m+6}$. The aforementioned observation along with the following fact allows us to increase the length of the code while preserving the distance.

▶ **Fact 19.** *For a linear code* C *of parameters* $[n,k,d]$ *and parity-check matrix* H*, the code corresponding to the parity-check matrix*

$$H^{'}_{(n-k)\times n'} = H \oplus I_{n'-n} = \begin{bmatrix} H_{(n-k)\times n} & O_{(n-k)\times(n'-n)} \\ O_{(n'-n)\times n} & I_{(n'-n)\times(n'-n)} \end{bmatrix},$$

has parameters $[n^{'}, k, d]$*.*

Using Fact 16 about the tensor-product of codes, we have:

$$\min\{d(\tilde{C}), d(\tilde{C}^\perp)\} = \min\{d(C^\perp)d(C'), d(C), d({C'}^\perp)\} \leq d(C).$$

Given that the length of code C is n, and $d(C')$, $d({C'}^\perp)$ are both greater than or equal to $\sqrt{N}$, we get

$$\min\{d(\tilde{C}), d(\tilde{C}^\perp)\} \geq \min\{d(C^\perp)\cdot n, d(C), n\} \geq d(C),$$

for $N \geq n^2$. Hence, $\min\{d(\tilde{C}), d(\tilde{C}^{\perp})\} = d(C)$. For the error in additive approximation, as we are embedding code of length n in a space of length n^3, hence we get the additive approximation term with cubic error. ◀

We now reduce this intermediate problem to GraphDist.

Proof of Theorem 2. For an input instance (H, t) of MinDistDualDist consider the systematic form representation of H i.e., $H = [I : P]$. Now, consider the following symmetric matrix A_P corresponding to H, i.e.:

$$A_P = \begin{bmatrix} O_{n-k\times n-k} & P \\ P^T & O_{k\times n-k} \end{bmatrix}.$$

It can be verified that as A_P is a symmetric matrix hence it can be treated as the adjacency matrix of a graph (infact a simple graph as the diagonal entries are all zero.)

As mentioned in the definition of minimum graph-state distance, the minimum graph state distance of A_P is obtained by finding $x := (a_1, ..., a_n | b_1, ..., b_n) \neq 0$ with minimum $wt_H(a \vee b)$ such that

$$[I : A_P]x^T = 0 \tag{2}$$

Now, the system of equations given in the above equation can be broken down into the following system of equations.

$$\begin{bmatrix} I : P \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-k} \\ b_{n-k+1} \\ \vdots \\ b_n \end{bmatrix} = O_{n-k\times 1} \tag{3}$$

$$\begin{bmatrix} P^T : I \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_{n-k} \\ a_{n-k+1} \\ \vdots \\ a_n \end{bmatrix} = O_{k\times 1} \tag{4}$$

Let S_1, S_2 be the set of variables that correspond to the above two system of equations respectively. As the sets S_1 and S_2 are disjoint, the minimum graph state distance for A_P, is obtained by assigning one set of variables the value 0, and for the remaining equation, we can find the non-zero solution with the least Hamming weight. This implies that the minimum graph-state distance corresponding to the graph whose adjacency matrix is A_P is:

$$d_G = \min\{d(C), d(C^{\perp})\},$$

and completes the proof of the hardness of GraphMinDist. Moreover, the above reduction also shows hardness for MultGapGraphDist and AddGapGraphDist. ◀

As a corollary to Lemma 17, we obtain that computing the distance of classical linear codes having rate 1/2 is NP-complete.

▶ **Corollary 20.** *It is* NP*-hard under Karp reductions to compute the distance of constant rate linear codes with rate* $\in (0, 1/2)$*. Furthermore, it is* NP*-hard under Karp reduction to compute both constant factor approximations and* $n^{1/3}$ *additive approximations to the distance of constant rate linear codes with rate* $\in (0, 1/2)$*.*

Proof. Consider a code C with parameters $[n, k, d]$ and parity-check matrix $[I_{n-k} : P_{n-k\times k}]$. Now consider the matrix:

$$\begin{bmatrix} I_k & 0_{k\times n-k} & : & 0_{n-k\times k} & P_{n-k\times k} \\ 0_{n-k\times k} & I_{n-k} & : & P^T_{k\times n-k} & 0_{k\times n-k} \end{bmatrix}.$$

The above matrix can be treated as a parity-check matrix corresponding to a rate 1/2 code. Computing the distance of the above code is computing a non-zero vector $(x, y) \in \mathbb{F}_2^n \oplus \mathbb{F}_2^n$ of least Hamming weight for which:

$$\begin{bmatrix} I_k & 0_{k\times n-k} & : & 0_{n-k\times k} & P_{n-k\times k} \\ 0_{n-k\times k} & I_{n-k} & : & P^T_{k\times n-k} & 0_{k\times n-k} \end{bmatrix} \begin{bmatrix} x^T \\ y^T \end{bmatrix} = 0_{n\times 1}.$$

As in the proof of Theorem 2, one sees the distance of this code is $\min\{d(C), d(C^\perp)\}$. To prove the hardness of codes with having rate $\in (0, 1/2)$, we take a code of rate equal to 1/2 and pad it with appropriate number of zeros. By Fact 19, this preserves the distance but decreases the rate. ◀

4.2 Proof of Theorem 3

Having showed that GraphMinDist is NP-hard, one might wonder whether it is at least possible to compute in polynomial time the minimum number of X-errors that can be detected by the graph state. We show that this problem and its gap versions are also NP-hard under Karp reduction.

Proof of Theorem 3. We reduce the problem of MinDist to X-GraphMinDist. In our reduction, we construct a self-orthogonal code C' from the given input code C while ensuring that $d(C')$ is proportional to $d(C)$. We explain this conversion of an arbitrary code C to a self-orthogonal C' in the following claim.

▷ Claim 21. Let C be a $[n, k, d]$ be a binary linear code. Consider the map $\pi : \mathbb{F}_2^n \to (\mathbb{F}_2^n)^2$ defined as:

$$\pi(a) = (a, a).$$

Then the image of C, defined as:

$$\pi(C) = \{\pi(c) : c \in C\},$$

is a self-orthogonal linear code with parameters $[2n, k, 2d]$.

Proof. We first prove that the map $\pi(C)$ is a linear code of length $2n$. For any two vectors $\pi(c), \pi(c') \in \pi(C)$ and $\lambda\ \mu \in \mathbb{F}_2$ we have that:

$$\lambda\pi(c) + \mu\pi(c') = \lambda(c, c) + \mu(c', c') = (\lambda c + \mu c', \lambda c + \mu c').$$

As C is a linear subspace hence $\lambda c + \mu c^{'} \in C$, hence $(\lambda c + \mu c^{'}, \lambda c + \mu c^{'}) \in \pi(C)$.

We now prove that $dim(\pi(C)) = dim(C) = k$. This follows from the observation that the map π is an injective map. Indeed, $\pi(c) = (0,0)$ implies $(c,c) = (0,0)$, *i.e.* $c = 0$.

The distance of the linear code $\pi(C)$ is calculated by observing that

$$\begin{aligned}\min_{\pi(c)\in\pi(C):\pi(c)\neq(0,0)} wt_H(\pi(c)) &= \min_{c\in C:c\neq 0} \pi(c) \\ &= \min_{c\in C:c\neq 0} 2wt_H(c) = 2\min_{c\in C:c\neq 0} wt_H(c) = 2d_H(C).\end{aligned}$$

Now, to prove that $\pi(C)$ is self-orthogonal we need to show that any pair of codewords $\pi(c),\ \pi(c^{'}) \in \pi(C)$ is orthogonal to each other.

Consider the inner-product:

$$\langle \pi(c), \pi(c^{'})\rangle = \langle (c,c), (c^{'}, c^{'})\rangle = \langle c, c'\rangle + \langle c, c'\rangle = 0,$$

where the last equality is due to the fact that we are in the field, $\mathbb{F}_2$ which has characteristic 2. This proves the self-orthogonality of $\pi(C)$. ◁

Now we describe our reduction from MinDist to X-MinDist.

Let (H,t) be the instance for MinDist and let C be the corresponding linear code with parameters $[n,k,d]$. By claim 21, $\pi(C) \in (\mathbb{F}_2^n)^2$ is a self-orthogonal linear code with parameters $[2n,k,2d]$. The systematic form of the parity-check matrix for $\pi(C)$:

$$[I_{2n-k\times 2n-k} : P_{2n-k\times k}].$$

along with its generator matrix:

$$[P^T_{k\times 2n-k} : I_k],$$

can be obtained in polynomial time from H.

Now, consider the matrix:

$$A_P = \begin{bmatrix} I_k & P_{k\times 2n-k} \\ P^T_{2n-k\times k} & I_{2n-k}\end{bmatrix}$$

The matrix A_P is symmetric and hence can be as a graph (though not a simple graph). Now, consider vectors $(x,z) \in \mathbb{F}_2^{2n} \oplus \mathbb{F}_2^{2n}$ with $z = 0$ such that

$$[I : A_P](z,x)^T = 0_{2n\times 1}$$

or equivalently,

$$A_P x^T = 0_{n\times 1}.$$

Solving the above equation is equivalent to solving the following system of equations:

$$[I_k : P_{2n-k\times k}]x^T = 0_{2n\times 1}$$

and,

$$[P^T_{2n-k\times k} : I_k]x^T = 0_{2n\times 1}$$

Thus, we are finding those vectors $x \in \mathbb{F}_2^{2n}$ for which both $Hx^T = 0_{2n-k\times 1}$ and $Gx^T = 0_{k\times 1}$. In other words, we are finding vectors $x \in \mathbb{F}_2^{2n}$ for which $x \in \pi(C) \cap \pi(C)^{\perp}$. Since $\pi(C)$ is self-orthogonal, it follows that $x \in \pi(C)$.

This implies that we want to find a non-zero vector $x \in \mathbb{F}_2^n$ with the least Hamming weight with $x \in \pi(C)$. This is precisely the minimum Hamming distance of the classical code $\pi(C)$ which is $2d_H(C)$. This proves the hardness of X-GraphMinDist as well as X-MultGapGraphDist$_\gamma$ under Karp reduction for $\gamma > 1$. Further, as the length of $\pi(C)$ is $2n$, X-AddGapGraphDist$_\tau$ is NP-hard for some $\tau \in (0, 1)$. ◀

5 Discussion and outlook

5.1 Controlling ϵ in AddGapHGPDist$_{\tau,\epsilon}$

Let AddGapHGPDist$_{\tau,\epsilon}$ denote the version of the problem AddGapCSSDist$_{\tau,\epsilon}$ where the input CSS code is a hypergraph product code.

▶ **Proposition 22.** *For any fixed $\epsilon \in (1/2, 1]$ and $\tau > 0$, there does not exist a Karp reduction from* MinDist *to* AddGapHGPDist$_{\tau,\epsilon}$ *unless* P = NP.

Proof. Suppose otherwise. Then there is a deterministic polynomial-time algorithm that takes an input instance (H, t) of MinDist (where H is a binary parity-check matrix and t is a non-negative integer) to an instance (H_1, H_2, t') of AddGapHGPDist in a way that preserves the respective decision problems' output. Say, H_1 and H_2 are full-rank and corresponding code parameters are $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$. Recall that the hypergraph product codes corresponding to parity-check matrices H_1 and H_2 with full rank and corresponding code parameters $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$ have parameters $[[n_1n_2 + (n_1 - k_1)(n_2 - k_2), k_1k_2, \min\{d_1,\ d_2\}]]$. Due to the symmetry of the construction, we can assume that $d_1 = \min\{d_1,\ d_2\}$.

We reiterate a subtle point: n_1 and n_2 are functions of n and hence really should be expressed as $n_1(n)$ and $n_2(n)$, but we will suppress this dependency in order to avoid overloading our notation. Now consider the following cases:

1. For all (H_1, H_2, t'), $\min\{n_1, n_2\} = O(1)$: If we assume that $n_1 = \min\{n_1, n_2\} = O(1)$ then one can just perform an efficient brute-force search to decide if there is a non-zero vector x with the properties needed. While if $n_2 \leq n_1$, then using the fact that $\min\{d_1, d_2\} \leq \min\{n_1, n_2\} = O(1)$, one can check if there is a non-zero vector x with $|x| \leq n_2 = O(1)$ such that $H_1x^T = 0$. This would show that MinDist admits a poly-time algorithm, which contradicts $P \neq NP$.
2. $n_1, n_2 = \omega(1)$: Consider a NO instance (H, t) of MinDist. By the definition of Karp reduction, it follows $(H_1, H_2, t^{'})$ is a NO instance of AddGapHGPDist$_{\tau,\epsilon}$. under the assumption $d_1 = \min\{d_1\ d_2\}$,this gives the following set of inequalities:

   $$d_1 > t^{'} + \tau(n_1n_2 + (n_1 - k_1)(n_2 - k_2))^{\epsilon} \geq t' + \tau(n_1n_2)^{\epsilon}.$$

   Assume $n_1 \leq n_2$ and $d_1 = \min\{d_1,\ d_2\}$. The other case can be treated similarly. Then $d_1 \geq t' + \tau {n_1}^{2\epsilon}$ and so $d_1/n_1 \geq t'/n_1 + n_1^{2\epsilon-1}$. Since $d_1 \leq n_1$, one obtains a contradiction for large enough n_1. ◀

There are a number of very intriguing questions that Proposition 22 stimulates. Do there exist an $1/2 < \epsilon \leq 1$ and a $0 < \tau < 1$ for which AddGapHGPDist$_{\tau,\epsilon}$ is in coNP? Or P? Or maybe BQP? We discuss the most intriguing question in the next subsection.

5.2 A new square-root barrier?

On one hand, our technique for proving the hardness of AddGapCSSDist used hypergraph product codes, and resulted in an additive approximation with a square-root error term. Moreover, Proposition 22 shows that there is, in some sense, no way to improve on this square-root term using hypergraph product codes. On the other hand, the proof of [19] was based on rather different methods that exploited graphs with certain extremal properties, and yet arrived at the same kind of square-root error term. While there is no analog of Proposition 22 found in [19], it appears that the graphs they use in their reduction are essentially optimal [16]. It is interesting that the same square-root term in the approximation appears in both reductions, and one inevitably wonders if there is a new kind of square-root barrier at play.

5.3 Can we get a linear approximation for minimum graph state distance and linear codes with rate 1/2?

Recall that we were able to prove hardness for approximating the minimum graph state distance and the hardness of approximating the distance of linear codes with rate equal to 1/2 with a cube root additive term. Can we improve it? Following our proof strategy, a natural way to get past the cube root barrier and get an improvement, say a square root approximation, would be to find an infinite family of efficiently computable codes $\{C_m\}$ for which $\min\{d(C_m),\ d(C_m^\perp)\} = \Omega(n_m)$, where n_m is the length of C_m. Whether such an infinite family of codes exists is a possible open direction. The possibility of achieving the hardness results without the need of such a family of codes is also a direction to pursue.

5.4 Quantum "nearest codeword problem"

In the classical setting, the nearest codeword problem (NCP) has played a useful role in understanding the complexity of distances of codes. This is because NCP is essentially a non-homogeneous version of the minimum distance problem. So it is natural to wonder about quantum analogs of the question. One subtlety is that the codewords of a quantum code are, strictly speaking, *not* vectors over $\mathbb{F}_2$, but rather vectors in some subspace of the Hilbert space $(\mathbb{C}^2)^{\otimes n}$. (We have avoided discussing this point much in this paper, but see Section 2.) From this perspective, the quantum nearest codeword problem should be a question about computing the orthogonal projections of states onto the codespace. There are other, discrete variants one can imagine that happen inside the symplectic $\mathbb{F}_2$ vector space that contains the stabilizers of a code. These questions would not be about codewords *per se*, but about Pauli error operators, e.g. given a Pauli error, what is the closest logical error (either in Hamming weight or symplectic weight)?

References

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

2 E. R. Berlekamp, R. J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

3 Vijay Bhattiprolu, Venkatesan Guruswami, Euiwoong Lee, and Xuandi Ren. Inapproximability of finding sparse vectors in codes, subspaces, and lattices. *arXiv:2410.02636. To appear* FOCS25, 2025.

4 Vijay Bhattiprolu and Euiwoong Lee. Inapproximability of sparsest vector in a real subspace. *arXiv preprint arXiv:2410.02636*, 2024. doi:10.48550/arXiv.2410.02636.

5 S. Bravyi, B. M. Terhal, and B. Leemhuis. Majorana fermion codes. *New Journal of Physics*, 12(8):083039, 2010.

6 Sergey Bravyi and Matthew B. Hastings. Homological product codes. *In Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, pages 273–282, 2014. doi:10.1145/2591796.2591870.

7 Adán Cabello, Lars Eirik Danielsen, Danielsen, Antonio J. López-Tarrida, and José R. Portillo. Optimal preparation of graph states. *Physical Review A*, 83(4), 2011.

8 A. Robert Calderbank, Eric M. Rains Rains, Peter M. Shor, and Neil JA Sloane. Quantum error correction via codes over gf (4). *IEEE Transactions on Information Theory*, 44(4):1369–1387, 1998. doi:10.1109/18.681315.

9 A. Robert Calderbank and Peter W. Shor. Hardness of approximating the minimum distance of a linear code. *Physical Review A*, 54(2), 1996.

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

11 Andrew Cross, Zhiyang He, Anand Natarajan, Mario Szegedy, and Guanyu Zhu. Quantum locally testable code with constant soundness. *Quantum*, 8, 2024. doi:10.22331/Q-2024-10-18-1501.

12 Andrew Cross, Graeme Smith, John A. Smolin, and Bei Zeng. Codeword stabilized quantum codes. *IEEE International Symposium on Information Theory*, pages 364–368, 2008. doi:10.1109/ISIT.2008.4595009.

13 Nicolas Delfosse and Matthew B. Hastings. Union-find decoders for homological product codes. *Quantum*, 5:406, 2021. doi:10.22331/Q-2021-03-10-406.

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

15 A. R. P. Erdős and V. T. Sós. On a problem of graph theory. *Studia Scientiarum Mathematicarum Hungarica*, pages 215–235, 1966.

16 Zoltán Füredi and Miklós Simonovits. The history of degenerate (bipartite) extremal graph problems. In *Erdős centennial*, pages 169–264. Springer, 2013.

17 Daniel Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, 1997.

18 Marc Heinn, Jens Eisert, and Hans J. Breigel. Multiparty entanglement in graph states. *Physical Review A*, 69(6), 2004. doi:10.1103/PhysRevA.69.062311.

19 Upendra Kapshikar and Srijita Kundu. On the hardness of the minimum distance problem of quantum codes. *IEEE Transactions on Information Theory*, 69(10):6293–6302, 2023. doi:10.1109/TIT.2023.3286870.

20 Tali Kaufman and Ran J. Tessler. New cosystolic expanders from tensors imply explicit quantum LDPC codes with $\Omega(\sqrt{n \log kn})$ distance. *In Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1317–1329, 2021.

21 Hai Liu, Chengju Li, and Cunsheng Ding. Five infinite families of binary cyclic codes and their related codes with good parameters. *Finite Fields and Their Applications*, 91, 2023. doi:10.1016/J.FFA.2023.102270.

22 Damian Markham and Barry C. Sanders. Graph states for quantum secret sharing. *Physical Review A*, 78, 2008.

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

24 Xuandi Ren. Private communication, 2025.

25 Andrew M. Steane. Error correcting codes in quantum theory. *Physical Review A*, 77(2), 1996.

26 J. P. Tillich and G. Zémor. Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Transactions on Information Theory*, 60(2):1193–1202, 2013. doi:10.1109/TIT.2013.2292061.

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