ASYMPTOTIC IMPROVEMENTS TO PROVABLE ALGORITHMS FOR THE CODE EQUIVALENCE PROBLEM

HUCK BENNETT[1], DRISANA BHATIA[2], JEAN-FRANÇOIS BIASSE[3], MEDHA DURISHETI[5], LUCAS LABUFF[4], VINCENZO PALLOZZI LAVORANTE[3], AND PHILIP WAITKEVITCH[3]

ABSTRACT. We present several new provable algorithms for two variants of the code equivalence problem on linear error-correcting codes, the Linear Code Equivalence Problem (LCE) and the Permutation Code Equivalence Problem (PCE). Specifically, for arbitrary codes of block length n and dimension k over any finite field $\mathbb{F}_q$, we show:

(1) A deterministic algorithm running in $2^{n+o(n+q)}$ time for LCE.
(2) A randomized algorithm running in $2^{n/2+o(n+q)}$ time for LCE and PCE.
(3) A quantum algorithm running in $2^{n/3+o(n+q)}$ time for LCE and PCE.

The first algorithm complements the deterministic roughly 2^n-time algorithm of Babai (SODA 2011) for PCE. The second two algorithms improve on recent work of Nowakowski (PQCrypto 2025), which gave algorithms with similar running times, but only for code equivalence on *random* codes and only over fields of order $q \geq 7$.

1. INTRODUCTION

Let $\mathcal{C}_1, \mathcal{C}_2$ be two linear codes over the finite field $\mathbb{F}_q$. In this paper, we present new algorithms with improved provable time complexity for the problem of finding a linear isometry (for the Hamming distance) τ such that $\mathcal{C}_2 = \tau(\mathcal{C}_1)$ when such an isometry exists. This problem is (the search version of) the *Linear Code Equivalence Problem* (LCE). Because isometries preserve distances, equivalent codes have the same geometry and in particular the same weight enumerators. Deciding if two linear codes are equivalent, and if so computing a corresponding isometry between them are longstanding computational problems.

The focus of this work is on *provable* algorithms for LCE and the related Permutation Code Equivalence Problem (PCE), where τ is required to be a permutation. There are relatively few works focusing on provable algorithms for code equivalence. The first non-trivial such algorithm is due to Babai [BCGQ11].[1] It consists of reducing an instance of PCE to $\binom{n}{k}$ instances of the Graph Isomorphism Problem (GI), where n is the length of the codes in the PCE instance and k is their dimension. I.e., the codes are $[n,k]_q$ codes. Combining this reduction with the quasipolynomial-time GI algorithm of Babai [Bab16] gives an algorithm for PCE running in $\binom{n}{k} \cdot \text{quasipoly}(n) \cdot \text{poly}(\log q) \leq 2^{n+o(n+q)}$ time. Furthermore, it is possible to use Babai's PCE algorithm to solve LCE by combining it with the reduction from LCE to PCE of [SS13]. However, the reduction in [SS13] increases the length of the codes (which are over $\mathbb{F}_q$) by a factor of $q-1$, and so this only gives a $2^{(q-1)n+o(n)}$-time algorithm for LCE. This is substantially slower than the corresponding algorithm for PCE, especially when q is large.

To the best of our knowledge, the other main provable algorithm for code equivalence appears in recent work of Nowakowski [Now25]. It uses ideas due to Chou, Persichetti and Santini [CPS23] to design a randomized algorithm to solve a special case of LCE in time and space roughly $2^{n/2}$, improving over the roughly 2^n-time algorithm in [BCGQ11] in this case. However, the algorithm in [Now25] has two substantial restrictions:

(1) It only works on *random* codes rather than arbitrary codes. Specifically, it solves the search version of LCE on the distribution of codes $\mathcal{C}_1, \mathcal{C}_2$, where $\mathcal{C}_1$ is generated by a uniformly random generator matrix $\boldsymbol{G}_1 \in \mathbb{F}_q^{k \times n}$ and $\mathcal{C}_2 = \tau(\mathcal{C}_1)$ for a uniformly random linear isometry τ.
(2) It only works when the field size q satisfies $q \geq 7$.

[1] We note that although this paper is by four authors, the algorithm for code equivalence appears in the appendix and is attributed solely to Babai.

Nowakowski additionally sketched how to give a quantum version of his algorithm running in time roughly $2^{n/3}$. He also explicitly asked whether it was possible to remove the second restriction, and to solve LCE on codes over fields of order $q < 7$.

1.1. **Our Contribution.** In this paper, we answer Nowakowski's question in the affirmative and remove both of the restrictions of his algorithm by giving randomized and quantum algorithms that run in time roughly $2^{n/2}$ and $2^{n/3}$, respectively, for both PCE and LCE. That is, our algorithms work on *worst-case* input codes over *arbitrary* finite fields $\mathbb{F}_q$, including the important special cases of $q \in \{2, 3, 5\}$ not handled by [Now25]. Additionally, we extend the deterministic PCE algorithm of Babai [BCGQ11], which runs in roughly $\binom{n}{k} \leq 2^n$ time, to an algorithm for LCE on codes over arbitrary fields $\mathbb{F}_q$ with similar running time. This improves on the algorithm obtained by combining [BCGQ11] and the LCE to PCE reduction in [SS13], which runs in time roughly $\binom{(q-1)n}{k}$ (where $\binom{(q-1)n}{k} \gg \binom{n}{k}$ for large values of q). More specifically, our results are as follows:

(1) A deterministic algorithm running in $2^{n+o(n+q)}$ time for LCE (Theorem 4.2).
(2) A randomized algorithm running in $2^{n/2+o(n+q)}$ time for LCE and PCE (Theorem 5.17).
(3) A quantum algorithm running in $2^{n/3+o(n+q)}$ time for LCE and PCE (Theorem 6.2).

1.2. **Acknowledgements.** Most of this work was performed during the Research Experiences for Undergraduates (REU-Site) program "Cryptography and Coding Theory at the University of South Florida" which ran from May 27th to August 2nd 2024 (usf-crypto.org/reu-program/). This program is funded by the U.S. National Science Foundation under Grant #2244488. H.B. is funded in part by NSF Grant #2432132. H.B. would also like to thank Chinmay Nirkhe for telling him about [BHT97] and Tselil Schramm for sharing her lecture notes on matroid basis sampling [Sch22]. J.-F. B. thanks Delaram Karobaei, Alexander May and Benjamin Wesolowski for useful conversations about the collision problem with non-uniform distributions that took place at the Post-Quantum Algebraic Cryptography semester at the Institut Henri Pointcaré in Fall 2024. J.-F. B. also thanks Toshio Nakata for pointing out useful references on the collision problem.

1.3. **Additional Related Work.** In early work on code equivalence, Leon [Leo82] presented an algorithm for the computation of the automorphism group of a code. Its analysis is heuristic, and its bottleneck is the computation of a large enough set of weight-w codewords, which is typically done through the Information Set Decoding (ISD) algorithm. The ISD algorithm was originally introduced by Prange [Pra62], and was improved in many subsequent works including [BJMM12, BM18, DEEK24, ES24, MMT11, Pet10, Ste88]. The original approach of Leon to solve LCE with the use of ISD was further improved by Beullens [Beu20] and by Barenghi, Biasse, Persichetti and Santini [BBPS23].

Despite the use of further heuristics, the complexity of the PCE and LCE algorithms resulting from this line of work is exponential. Yet, as observed by Petrank and Roth [PR97], the code equivalence problem is unlikely to be NP-hard because this would imply the collapse of the polynomial hierarchy. Petrank and Roth also proved a reduction from GI to PCE. At the time, this reduction seemed to indicate that the code equivalence problem could not be too easy to solve, but subsequent work of Babai [Bab16] showed that GI could be solved in quasipolynomial time.

Other works have focused on identifying easy instances of the code equivalence problem, with a special focus on PCE with input codes having a trivial or small hull (the hull of a code $\mathcal{C}$ is the intersection $\mathcal{C} \cap \mathcal{C}^\perp$ of $\mathcal{C}$ with its dual code $C^\perp$). Most notably, the Support Splitting Algorithm (SSA) originally described by Sendrier [Sen00] efficiently solves random instances of PCE. Additional work on this special case includes the work of Bardet, Otmani and Saeed-Taha [BOS19], which leverages a reduction from PCE to GI, and of Saeed-Taha [Sae18] who gave a reduction from the code equivalence problem to the problem of solving a system of polynomial equations.

We also note that the *hardness* of the code equivalence problem is relevant to the security of code-based cryptographic schemes. Indeed, in the McEliece [McE78] and in the Niederreiter [Nie86] schemes, the public key is a linear code that is permutationally equivalent to a secret code where an efficient decoding algorithm is known. More recently, Biasse, Micheli, Persichetti and Santini described a Zero-Knowledge proof protocol whose security provably relies on the code equivalence problem [BMPS20]. Via the Fiat-Shamir heuristic [FS86], this protocol gives rise to the LESS digital signature scheme, which recently moved to Round 2 of the Additional Digital Signature Schemes NIST standardization process [BBB+23]. Cryptosystems based on

the hardness of the code equivalence problem belong to a broader class of schemes that encompasses isomorphism problems: lattice isomorphism [DvW22, DPPvW22, BGPS23], matrix code equivalence [CNP+23b], and tensor isomorphism [JQSY19]. Due to their apparent resistance to efficient quantum attacks, these cryptosystems are promising candidates for post-quantum digital signatures, and they were submitted to the NIST standardization process [BBB+23, BBD+23, CNP+23a].

1.4. **Technical Overview.** To sketch our algorithms, it is helpful to first review Babai's algorithm for PCE [BCGQ11], for which we first recall some terminology. A *generator matrix* $\boldsymbol{G} \in \mathbb{F}_q^{k\times n}$ for an $[n,k]_q$ code $\mathcal{C}$ has rows whose $\mathbb{F}_q$-span are equal to $\mathcal{C}$. A generator matrix $\boldsymbol{G}$ is said to be in *systematic form* if it is of the form $\boldsymbol{G} = \begin{pmatrix} I_k & \boldsymbol{G}' \end{pmatrix}$ for some matrix $\boldsymbol{G}' \in \mathbb{F}_q^{k\times(n-k)}$, i.e., if its $k \times k$ prefix submatrix is the identity matrix. Let SF be the algorithm that maps a generator matrix $\boldsymbol{G} = \begin{pmatrix} \boldsymbol{G}' & \boldsymbol{G}'' \end{pmatrix}$ with $\boldsymbol{G}'$ non-singular to its corresponding systematic form matrix by left-multiplying by $(\boldsymbol{G}')^{-1}$. I.e., $\mathrm{SF} : \begin{pmatrix} \boldsymbol{G}' & \boldsymbol{G}'' \end{pmatrix} \mapsto (\boldsymbol{G}')^{-1} \cdot \begin{pmatrix} \boldsymbol{G}' & \boldsymbol{G}'' \end{pmatrix} = \begin{pmatrix} I_k & (\boldsymbol{G}')^{-1} \cdot \boldsymbol{G}'' \end{pmatrix}$. An *information set* $I = \{i_1, \ldots, i_k\} \subseteq [n]$ for an $[n,k]_q$ code $\mathcal{C}$ with generator matrix $\boldsymbol{G} \in \mathbb{F}_q^{k\times n}$ is a set of k coordinates such that the $k \times k$ submatrix obtained by restricting $\boldsymbol{G}$ to columns indexed by I is non-singular.

The search version of PCE is as follows: given generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2 \in \mathbb{F}_q^{k\times n}$ of permutationally equivalent codes as input, find a full-rank matrix $\boldsymbol{S} \in \mathbb{F}_q^{k\times k}$ and a permutation matrix $\boldsymbol{P} \in \mathbb{F}_q^{n\times n}$ such that $\boldsymbol{S}\boldsymbol{G}_1\boldsymbol{P} = \boldsymbol{G}_2$. Similarly, the search version of LCE is as follows: given generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2 \in \mathbb{F}_q^{k\times n}$ of linearly equivalent codes as input, find a full-rank matrix $\boldsymbol{S} \in \mathbb{F}_q^{k\times k}$ and a monomial matrix $\boldsymbol{M} \in \mathbb{F}_q^{n\times n}$ such that $\boldsymbol{S}\boldsymbol{G}_1\boldsymbol{M} = \boldsymbol{G}_2$. A *monomial matrix* $\boldsymbol{M} = \boldsymbol{P}\boldsymbol{D}$ is a matrix that is the product of a permutation matrix $\boldsymbol{P}$ and a full-rank diagonal matrix $\boldsymbol{D}$.

Babai's PCE algorithm. The idea of Babai's PCE algorithm is as follows. Let $\boldsymbol{G}_1, \boldsymbol{G}_2 \in \mathbb{F}_q^{k\times n}$ be respective generator matrices $[n,k]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$ such that there exists a permutation σ with $\sigma(\mathcal{C}_1) = \mathcal{C}_2$ (in particular, $\mathcal{C}_1$ and $\mathcal{C}_2$ are permutationally equivalent). Then, σ must map every information set I_1 of $\mathcal{C}_1$ to some information set I_2 of $\mathcal{C}_2$. We call a pair of information sets I_1 of $\mathcal{C}_1$ and I_2 of $\mathcal{C}_2$ with $\sigma(I_1) = I_2$ for some permutation σ with $\sigma(\mathcal{C}_1) = \mathcal{C}_2$ *matching information sets*.

Suppose that matching information sets I_1, I_2 for permutationally equivalent codes $\mathcal{C}_1, \mathcal{C}_2$ with generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$ are known, and let $\boldsymbol{P}_{I_1}, \boldsymbol{P}_{I_2}$ be $n \times n$ permutation matrices corresponding to permutations that map I_1, I_2 to $\{1, \ldots, k\}$, respectively. Then there must exist a full-rank matrix $\boldsymbol{S} \in \mathbb{F}_q^{k\times k}$, a permutation matrix $\boldsymbol{P}_1 \in \mathbb{F}_q^{k\times k}$, and a permutation matrix $\boldsymbol{P}_2 \in \mathbb{F}_q^{(n-k)\times(n-k)}$ such that

$$\boldsymbol{S} \cdot \mathrm{SF}(\boldsymbol{G}_1) \cdot \begin{pmatrix} \boldsymbol{P}_1^{-1} & \\ & \boldsymbol{P}_2 \end{pmatrix} = \mathrm{SF}(\boldsymbol{G}_2) \ . \tag{1.1}$$

Furthermore, $\mathrm{SF}(\boldsymbol{G}_1) = \begin{pmatrix} I_k & \boldsymbol{G}_1' \end{pmatrix}$ and $\mathrm{SF}(\boldsymbol{G}_2) = \begin{pmatrix} I_k & \boldsymbol{G}_2' \end{pmatrix}$ for some $\boldsymbol{G}_1', \boldsymbol{G}_2' \in \mathbb{F}_q^{k\times(n-k)}$. It follows by Equation (1.1) that $\boldsymbol{S} = \boldsymbol{P}_1$, and therefore $\boldsymbol{P}_1\boldsymbol{G}_1'\boldsymbol{P}_2 = \boldsymbol{G}_2'$.

So, by the preceding discussing, to solve PCE it suffices (1) to know a pair of matching information sets I_1, I_2, and (2) to be able to find permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that

$$\boldsymbol{P}_1\boldsymbol{G}_1'\boldsymbol{P}_2 = \boldsymbol{G}_2' \ . \tag{1.2}$$

This latter problem is equivalent to the graph isomorphism problem (GI) on $\mathbb{F}_q$-edge-labeled bipartite graphs (where we interpret $\boldsymbol{G}_1', \boldsymbol{G}_2'$ as the corresponding adjacency matrices for these graphs), a problem that Babai showed to be solvable in quasipolynomial time in another seminal paper [Bab16].[2] So, Babai's PCE algorithm works by:

(1) Finding an arbitrary information set I_1 of $\mathcal{C}_1$.
(2) "Guessing" an information set I_2 of $\mathcal{C}_2$ matching I_1 by enumerating all $\binom{n}{k}$ size-k subsets of $[n]$.
(3) Solving GI on each of the $\binom{n}{k}$ resulting instances $\boldsymbol{G}_1', \boldsymbol{G}_2'$, using [Bab16].

This algorithm therefore takes $\binom{n}{k} \cdot \mathrm{quasipoly}(n) \cdot \mathrm{poly}(\log(q)) \leq 2^{n+o(n+q)}$ time.

From PCE to LCE. Let $\lambda = (\lambda_1, \ldots, \lambda_{q-1})$ where $\mathbb{F}_q = \{\lambda_1, \ldots, \lambda_{q-1}\}$, and let $\mathcal{C}_1, \mathcal{C}_2$ be $[n,k]_q$ codes corresponding to an instance of LCE. The reduction from LCE to PCE in [SS13] works by outputting the

[2] In fact, [BCGQ11] preceded [Bab16] and so Babai's PCE algorithm uses an older subexponential-time algorithm for GI. Despite being slower, this GI algorithm is still not the main bottleneck in the running time of [BCGQ11].

closures $\mathcal{C}_1 \otimes \lambda, \mathcal{C}_2 \otimes \lambda$ of the input codes, where $\otimes$ is the Kronecker product. These closures are $[(q-1)n, k]_q$ codes, and so there are $\binom{(q-1)n}{k} \approx 2^{(q-1)n \cdot H(k/((q-1)n))}$ possible information sets I_2 to enumerate when running Babai's PCE algorithm on $\mathcal{C}_1 \otimes \lambda, \mathcal{C}_2 \otimes \lambda$, which is far larger than 2^n especially for large values of q (here H is the binary entropy function).

So, we take a different approach to solving LCE. Using similar reasoning to that described above for Babai's PCE algorithm, to solve LCE on $[n, k]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$, it suffices (1) to know a pair of matching information sets I_1, I_2 for $\mathcal{C}_1, \mathcal{C}_2$, and (2) to be able to find a $k \times k$ monomial matrix $\boldsymbol{M}_1$ and an $(n-k) \times (n-k)$ monomial matrix $\boldsymbol{M}_2$ such that

$$\boldsymbol{M}_1 \boldsymbol{G}'_1 \boldsymbol{M}_2 = \boldsymbol{G}'_2 \ , \tag{1.3}$$

where $\boldsymbol{G}'_1, \boldsymbol{G}'_2$ are defined as above using knowledge of I_1, I_2.

The problem of finding such $\boldsymbol{M}_1, \boldsymbol{M}_2$ was recently introduced and studied by Chou, Persichetti, and Santini [CPS23], who named it the *Left-Right Linear Equivalence Problem* (LRL). Accordingly, we call matrices $\boldsymbol{G}'_1, \boldsymbol{G}'_2$ such that there exist monomial matrices $\boldsymbol{M}_1, \boldsymbol{M}_2$ satisfying Equation (1.3) *LRL equivalent*. This problem was also used in Nowakowski's work [Now25]. We note the close correspondence between Equations (1.2) and (1.3). Indeed, the only difference is that in Equation (1.2) $\boldsymbol{P}_1, \boldsymbol{P}_2$ are permutation matrices and in Equation (1.3) $\boldsymbol{M}_1, \boldsymbol{M}_2$ are monomial matrices.

Call the problem of finding permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ satisfying Equation (1.2) the *Left-Right Permutation Equivalence Problem* (LRP). We call matrices $\boldsymbol{G}'_1, \boldsymbol{G}'_2$ such that there exist monomial matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ satisfying Equation (1.2) *LRP equivalent*. As mentioned above, LRP is equivalent to GI on $\mathbb{F}_q$-edge-labeled bipartite graphs, and is therefore solvable in quasipoly(n) time when $\boldsymbol{G}'_1, \boldsymbol{G}'_2 \in \mathbb{F}_q^{k \times (n-k)}$ for some $k \leq n$.

The key to our algorithm for LCE is a reduction from LRL to LRP, which works in a similar way to the closure-based LCE to PCE reduction of [SS13]. Given $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$, we define the *expanded matrix*

$$\widehat{\boldsymbol{A}} := \lambda^T \otimes \boldsymbol{A} \otimes \lambda \in \mathbb{F}_q^{(q-1)m \times (q-1)n} \ ,$$

where $\lambda \in (\mathbb{F}_q^*)^{q-1}$ is a vector consisting of the $q-1$ distinct elements in $\mathbb{F}_q^*$, as before.

Given an instance $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{F}_q^{m \times n}$ of LRL, our reduction to LRP simply maps $\boldsymbol{A}_1, \boldsymbol{A}_2$ to their respective expanded matrices, i.e., it maps $\boldsymbol{A}_1 \mapsto \widehat{\boldsymbol{A}}_1$, $\boldsymbol{A}_2 \mapsto \widehat{\boldsymbol{A}}_2$, where $\widehat{\boldsymbol{A}}_1, \widehat{\boldsymbol{A}}_2 \in \mathbb{F}_q^{(q-1)m \times (q-1)n}$. It is clear that this reduction is efficient (it runs in poly$(m+n+q)$ time). See Corollary 3.10 for a correctness proof.

Furthermore, when $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{F}_q^{m \times n}$, we can solve LRP on $\widehat{\boldsymbol{A}}_1, \widehat{\boldsymbol{A}}_2$ in quasipoly$(m+n+q)$ time, from which it is possible to find monomial matrices $\boldsymbol{M}_1, \boldsymbol{M}_2$ satisfying Equation (1.3) (with $\boldsymbol{A}_1 = \boldsymbol{G}'_1$, $\boldsymbol{A}_2 = \boldsymbol{G}'_2$) in quasipoly$(n+q)$ time. So, overall, our modification to Babai's algorithm runs in $\binom{n}{k} \cdot \text{quasipoly}(n+q) \leq 2^{n+o(n+q)}$ time.

A faster algorithm using randomness. We next turn to describing our randomized algorithm for PCE (which can also be adapted to LCE using ideas similar to those above). The algorithm uses the same basic approach as in [Now25], which in turn builds on [CPS23]. However, unlike [Now25], our algorithm works for arbitrary, worst-case instances of PCE over arbitrary finite fields $\mathbb{F}_q$. Our algorithm is also provable, whereas the algorithm in [CPS23] is heuristic, and is substantially simpler than [CPS23, Now25].

The key idea behind the algorithm in [Now25] is to reduce PCE to collision finding in a canonical form function F.[3] A *canonical form* for LRP is a function $F : \mathbb{F}_q^{m \times n} \to \mathbb{F}_q^{m \times n}$ such that for all $\boldsymbol{A}, \boldsymbol{A}' \in \mathbb{F}_q^{m \times n}$, (1) $\boldsymbol{A}$ and $F(\boldsymbol{A})$ are LRP equivalent, and (2) $F(\boldsymbol{A}) = F(\boldsymbol{A}')$ if and only if $\boldsymbol{A}$ and $\boldsymbol{A}'$ are LRP equivalent.

The most technical part of [Now25] is defining and analyzing an elaborate canonical form F for LRL. Our algorithm is substantially simpler, and leverages the fact that LRP is simply a special case of ($\mathbb{F}_q$-edge labeled) GI, as noted in [BCGQ11]. It then takes advantage of yet another work of Babai: a quasipolynomial-time computable canonical form for graphs [Bab19], which is a follow-up to his quasipolynomial-time GI algorithm [Bab16].[4]

Furthermore, Equation (1.2) and the surrounding discussion shows the correspondence between matching information sets I_1, I_2 and LRP-equivalent matrices $\boldsymbol{G}'_1, \boldsymbol{G}'_2$. In particular, given knowledge of such I_1, I_2 it

[3] In fact, [Now25] studies LCE rather than PCE. The main ideas described in our work apply to both problems.

[4] Technically, LRP is equivalent to GI on $\mathbb{F}_q$-edge-labeled bipartite graphs, whereas [Bab19] is stated "natively" for unlabeled graphs. However, there is an efficient reduction from GI on edge-labeled graphs to GI on unlabeled graphs [ZKT85], and so this distinction is immaterial. See Remark 2.9 for a more thorough discussion.

is possible to efficiently compute LRP-equivalent matrices $\boldsymbol{G}_1', \boldsymbol{G}_2'$ corresponding to the "non-identity parts" of systematic form generator matrices $\begin{pmatrix} I_k & \boldsymbol{G}_1' \end{pmatrix}, \begin{pmatrix} I_k & \boldsymbol{G}_2' \end{pmatrix}$ of the respective input codes $\mathcal{C}_1, \mathcal{C}_2$. From such $\boldsymbol{G}_1', \boldsymbol{G}_2'$ it is in turn possible to recover an isometry mapping $\mathcal{C}_1$ to $\mathcal{C}_2$ in quasipolynomial time using [Bab16].

We may therefore model the problem of finding matching information sets as the problem of finding values $x, y \in [N]$ such that $F_1(x) = F_2(y)$ for functions $F_1, F_2 : [N] \to \mathbb{F}_q^{m \times n}$, where F_i for $i = 1, 2$ corresponds to a map from the information sets of $\mathcal{C}_i$ to the graph $F(\boldsymbol{G}_i')$ output by Babai's corresponding canonical form F for graphs from [Bab19]. Such a pair (x, y) is called a *claw*. Furthermore, if $\mathcal{C}_1, \mathcal{C}_2$ are permutationally equivalent, then they have the same number N of information sets, and there are at least N pairs of matching information sets total. This in turn implies that F_1, F_2 have at least N claws (x, y).

Because of this, the probability that uniformly random $x, y \sim [N]$ are such that $F_1(x) = F_2(y)$ is at least $1/N$, and so if we sample m values $x_1, \ldots, x_m, y_1, \ldots, y_m \sim [N]$ the expected number of claws (x_i, y_j) we obtain is at least m^2/N. Therefore, setting $m = \Omega(\sqrt{N})$ we get at least $\Omega(1)$ claws in expectation. Furthermore, it is possible to show using Chebyshev's inequality that the total number of claws concentrates around its expectation. Since $N \leq \binom{n}{k} \leq 2^n$, this means that it always suffices to take $m \approx 2^{n/2}$. Accordingly, the resulting algorithm finds a claw (which is the crux for solving PCE) with high probability in $2^{n/2+o(n+q)}$ time.

There is an issue with the algorithm sketched above, however: it assumes that it is possible to efficiently sample a (truly) uniformly random information set. On the contrary, it is not at all clear how to do this. Because of this, [Now25] used the fact that the input codes were random to argue that a random size-k subset of indices is an information set with high probability. However, we are able to bypass any assumptions about the underlying codes by using an efficient algorithm for *almost* uniformly random matroid basis sampling (of which information set sampling is a special case) from [ALOV19]; see Algorithm 5 and Theorem 5.8.

An even faster quantum algorithm. Finally, we sketch how to extend the previous randomized PCE algorithm to a quantum algorithm. First, sample $m = \binom{n}{k}^{1/3} \cdot \log\left(\binom{n}{k}^{1/3}\right)$ many information sets from $\mathcal{C}_1$. If the number of distinct information sets among those sampled is less than $\binom{n}{k}^{1/3}$, then via a coupon collector argument one can show that with good probability the number of distinct information sets is $N \leq \binom{n}{k}^{1/3}$ and that all N of these appear among the m total information sets sampled. In this case, any information set of $\mathcal{C}_2$ will match one of the information sets sampled from $\mathcal{C}_1$.

Otherwise, keep $\binom{n}{k}^{1/3}$ distinct information sets of $\mathcal{C}_1$. By Grover search, it is then possible to find an information set for $\mathcal{C}_2$ matching one of those kept for $\mathcal{C}_1$ in roughly $\sqrt{\binom{n}{k} / \binom{n}{k}^{1/3}} = \binom{n}{k}^{1/3} \leq 2^{n/3}$ quantum time. This use of Grover search for claw finding is similar to and inspired by the quantum collision finding algorithm of [BHT97]. We also again note that [Now25] sketched a quantum algorithm along similar lines, although it did not provide many details.

1.5. **Organization of the Paper.** In Section 2, we present standard background on linear codes and the code equivalence problem. In Section 3, we introduce our method for solving the LRL problem. In Section 4, we apply this method to solve LCE with a modification of Babai's algorithm. Furthermore, in Section 5, we give a randomized algorithm using a meet-in-the-middle strategy to solve PCE and LCE, and in Section 6 we give a quantum algorithm to solve PCE and LCE.

2. Background

An $[n, k, d]_q$ code $\mathcal{C}$ is a k-dimensional vector space over $\mathbb{F}_q^n$ such that the minimum Hamming distance between two distinct elements (codewords) is d. In particular, every $[n, k, d]_q$ code is a linear code. An $[n, k]_q$ code is a code that is an $[n, k, d]_q$ code for some d. A matrix $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$ whose rows generate an $[n, k]_q$ code $\mathcal{C}$ is called a *generator matrix* for $\mathcal{C}$.

Definition 2.1 (Information set)**.** Let $\mathcal{C}$ be an $[n, k, d]_q$ code. We say that the indices $\{i_1, \ldots, i_k\} \subseteq [1, \ldots, n]$ are an *information set* for $\mathcal{C}$ if the $k \times k$ matrix made up of columns indexed by $i_1, \cdots, i_k$ of a generator matrix for $\mathcal{C}$ is invertible.

The coordinates of a codeword $c \in \mathcal{C}$ of indices belonging to an information set uniquely identify c. There are up to $\binom{n}{k} \sim 2^{nH(k/n)}$ possible information set for an $[n, k, d]_q$-linear code where H denotes the

binary entropy function. When the first k columns of the matrix $\boldsymbol{A} = \begin{pmatrix}\boldsymbol{A}_0 & \boldsymbol{A}_1\end{pmatrix} \in \mathbb{F}_q^{k\times n}$ are linearly independent (i.e., $\{1,\dots,k\}$ is an information set of $\mathcal{C}$), we denote by $\mathrm{SF}(\boldsymbol{A})$ the systematic form defined by $\mathrm{SF}(\boldsymbol{A}) = \boldsymbol{A}_0^{-1}\boldsymbol{A}$ which has the shape $\begin{pmatrix}I_k & \boldsymbol{A}_1'\end{pmatrix}$ for $\boldsymbol{A}_1' = \boldsymbol{A}_0^{-1}\boldsymbol{A}_1$. Working with a generator matrix in systematic form allows us to uniquely identify codewords with their first k components.

Linear maps that preserve the Hamming distance leave certain essential properties of linear codes intact—in particular, those related to their decoding capacities. We introduce the notion of equivalence of codes by first defining permutations of code. We say that two $[n,k,d]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$ are *permutationally equivalent* (which we denote by $\mathcal{C}_1 \overset{P}{\approx} \mathcal{C}_2$) if there is a permutation $\pi \in \mathcal{S}_n$ such that $\pi(\mathcal{C}_1) = \mathcal{C}_2$, i.e., every codeword of $\mathcal{C}_2$ arises as the permutation of the entries of a codeword of $\mathcal{C}_1$ according to π.

Definition 2.2 (Permutation matrix)**.** Let $\pi \in \mathcal{S}$. The permutation matrix $\boldsymbol{P} \in \mathbb{F}_q^{n\times n}$ satisfying $P_{j,i} = 1$ if $\pi(i) = j$ and $P_{i,j} = 0$ otherwise acts via a right multiplication by permuting columns according to π. More specifically, if $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{F}_q^{k\times n}$, and $\boldsymbol{B} = \boldsymbol{AP}$, then the column of index $\pi(i)$ of $\boldsymbol{B}$ is the column of index i of $\boldsymbol{A}$.

With the above definition, two linear codes $\mathcal{C}_1, \mathcal{C}_2$ with generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$ are permutationally equivalent if and only if there exist an invertible matrix $\boldsymbol{S}$ and a permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{G}_2 = \boldsymbol{SG}_1\boldsymbol{P}$.

Definition 2.3 (PCE)**.** Let $\mathcal{C}_1, \mathcal{C}_2$ be $[n,k,d]_q$ codes. The Permutation Code Equivalence problem (PCE) is the task of finding $\pi \in \mathcal{S}_n$ such that $\mathcal{C}_2 = \pi(\mathcal{C}_1)$. Equivalently, if $\boldsymbol{G}_1, \boldsymbol{G}_2$ are generator matrices for $\mathcal{C}_1$ and $\mathcal{C}_2$, PCE is the task of finding an invertible matrix $\boldsymbol{S}$ and a permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{G}_2 = \boldsymbol{SG}_1\boldsymbol{P}$ when $\mathcal{C}_1 \overset{P}{\approx} \mathcal{C}_2$.

The definition of PCE we give above corresponds to the search variant of this problem. In many references, PCE is defined as a decision problem: given $\mathcal{C}_1, \mathcal{C}_2$, decide whether $\mathcal{C}_1$ and $\mathcal{C}_2$ are permutationally equivalent. As shown in [BM23], search and decision are polynomially equivalent. In the scope of this paper, we focus on the search variant.

Permutation equivalence can be extended to codes that are image of each other by a linear isometry. Two such codes $\mathcal{C}_1, \mathcal{C}_2$ are said to be linearly equivalent, and this property is denoted by $\mathcal{C}_1 \overset{L}{\approx} \mathcal{C}_2$.

Proposition 2.4. *All linear isometries of $\mathbb{F}_q^n$ can be uniquely identified by $(\sigma, \boldsymbol{v}) \in \mathcal{S}_n \rtimes \mathbb{F}_q^{*n}$. The image $y \in \mathbb{F}_q^n$ of $x \in \mathbb{F}_q^n$ is given by*

$$y_{\sigma(i)} = v_{\sigma(i)} x_i.$$

Hence, an isometry necessarily acts via a permutation of the columns, followed by the scaling of the entries by non-zero scalars. From a matrix point of view, a linear isometry $\tau = (\sigma, \boldsymbol{v}) \in \mathcal{S}_n \rtimes \mathbb{F}_q^{*n}$ acts on the columns of $\boldsymbol{A} \in \mathbb{F}_q^{k\times n}$ via right multiplication by the permutation matrix $\boldsymbol{P}$ corresponding to σ, followed by the right multiplication by the diagonal matrix $\boldsymbol{D} = \mathrm{diag}(v_1, \dots, v_n)$: $\boldsymbol{APD}$. A matrix of the form $\boldsymbol{PD}$ is a *monomial matrix*.

Proposition 2.5 (LCE)**.** *Let $\mathcal{C}_1, \mathcal{C}_2$ be $[n,k,d]_q$-linear codes. The Linear Code Equivalence problem (LCE) is the task of finding $\tau \in \mathcal{S}_n \rtimes \mathbb{F}_q^{*n}$ such that $\mathcal{C}_2 = \tau(\mathcal{C}_1)$. Equivalently, if $\boldsymbol{G}_1, \boldsymbol{G}_2$ are generator matrices for $\mathcal{C}_1$ and $\mathcal{C}_2$, LCE is the task of finding an invertible matrix $\boldsymbol{S}$, a permutation matrix $\boldsymbol{P}$, and a diagonal matrix $\boldsymbol{D}$ such that $\boldsymbol{G}_2 = \boldsymbol{SG}_1\boldsymbol{PD}$.*

The work of Sendrier and Simos [SS13] introduced a strategy to reduce PCE to LCE through the concept of the *closure* of a linear code. In a nutshell, given an $[n,k,d]_q$-linear code $\mathcal{C}$, once constructs an $[n(q-1),k,d]_q$ code $\tilde{\mathcal{C}}$ via the operation

$$\begin{aligned}(c_1,..,c_n) &\mapsto (c_1,\dots,c_n) \otimes (\lambda_1,\dots,\lambda_{q-1}) \\ &= (\lambda_1 c_1, \lambda_2 c_1, \dots, \lambda_{q-1}c_1, \lambda_1 c_2, \lambda_2 c_2, \dots, \lambda_{q-1}c_2, \dots, \lambda_1 c_n, \lambda_2 c_n, \dots, \lambda_{q-1}c_n),\end{aligned}$$

where $\mathbb{F}_q^* = \{\lambda_1, \dots, \lambda_{q-1}\}$ and $\otimes$ denotes the Kronecker product. The following lemma is from [SS13]. See also the stronger statement about the closure of a code and the formal proof of these statements in [BW24].

Lemma 2.6 ([SS13])**.** *Two $[n,k,d]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$ are linearly equivalent if and only if their closures $\tilde{\mathcal{C}}_1$ and $\tilde{\mathcal{C}}_2$ are permutationally equivalent.*

We note that the disadvantage of the reduction in Lemma 2.6 is that it does not preserve the length of the codes $\mathcal{C}_1, \mathcal{C}_2$. We now turn our attention to the deterministic PCE algorithm due to Babai. It consists of reducing PCE to many instances of the Graph Isomorphism Problem (GI). Assume the two input codes $\mathcal{C}_1, \mathcal{C}_2$ have generator matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ respectively. Without loss of generality, we can assume that $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are in systematic form: $\boldsymbol{G}_1 = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$, and $\boldsymbol{G}_2 = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$. The goal of Babai's code equivalence algorithm is to reduce this instance of Code Equivalence to an instance of GI on $\mathbb{F}_q$-colored graphs defined by $\boldsymbol{A}_1, \boldsymbol{A}_2$.

Definition 2.7 ($\mathbb{F}_q$-colored bipartite graphs)**.** A bipartite graph $\mathcal{G}$ is a graph whose vertices can be partitioned into two subsets A and B such that each edge of $\mathcal{G}$ has one endpoint in A and one endpoint in B. The graph is $\mathbb{F}_q$-colored if each edge is labeled by an element of $\mathbb{F}_q$. The adjacency matrix $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$ of $\mathcal{G}$ has a coefficient $a_{i,j}$ if and only if the i-th element of A is connected to the j-th element of B via an edge of label $a_{i,j}$.

Two graphs are said to be *isomorphic* if their vertices can be permuted in a way that preserves edges. From the standpoint of adjacency matrices, let $\mathcal{G}_1$ and $\mathcal{G}_2$ be $\mathbb{F}_q$-colored bipartite graphs with adjacency matrices $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{F}_q^{m \times n}$. $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic if and only if there are permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that $\boldsymbol{A}_2 = \boldsymbol{P}_1 \boldsymbol{A}_1 \boldsymbol{P}_2$.[5]

Theorem 2.8 ([Bab16])**.** *There is a quasipolynomial-time algorithm that solves the graph isomorphism problem.*

Remark 2.9. A graph is called *edge-labeled* if its edges are each assigned a value, and two edge-labeled graphs are isomorphic if and only if there is a vertex permutation that not only preserves adjacency but maps edges of one graph to edges of the other graph with the same label. In particular, we will need to solve the Graph Isomorphism problem on $\mathbb{F}_q$-colored (bipartite) graphs—graphs that are edge-labeled with labels from $\mathbb{F}_q$.

It is unclear if Babai's quasipolynomial-time algorithm for GI [Bab16] (given in Theorem 2.8) and follow-up work giving a quasipolynomial-time computable canonical form for graphs [Bab19] (given in Theorem 5.4) work "natively" on edge-labeled graphs. (In a remark in [BCGQ11], Babai claims that edge labeling only speeds up a prior subexponential-time algorithm for GI.) However, there is is a poly(n)-time reduction from graph isomorphism on undirected, edge-labeled graphs to graph isomorphism on undirected, non-edge-labeled graphs—see [ZKT85], which calls edge-labeled graphs "color-graphs". So, using [ZKT85], Theorems 2.8 and 5.4 imply quasipoly(n)-time algorithms for GI on $\mathbb{F}_q$-colored graphs. Because of this, we state Theorems 2.8 and 5.4 and calls to GI algorithms without worrying about the distinction between edge-labeled and non-edge-labeled graphs.

Finally, we remark that the size of the graphs output by the reduction in [ZKT85] applied to $\mathbb{F}_q$-colored graphs can be upper bounded by a function only of n, the number of vertices, and not q, the number of colors. This is because, regardless of q, there are only $\binom{n}{2} = O(n^2)$ possible distinct edge labels, and n and the number of distinct edge labels are the only things that matter for the size of the output graphs.

To proceed with the reduction from PCE to GI, assume first that one knows the image of $\{1, \ldots, k\}$ under the permutation σ defined by the permutation matrix $\boldsymbol{P}$.

Theorem 2.10 (Th. 7.1 of [BCGQ11])**.** *Let $\boldsymbol{G}_1 = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$, and $\boldsymbol{G}_2 = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$ be generating matrices of two permutation-equivalent codes $\mathcal{C}_1, \mathcal{C}_2$ of dimension k and length n over $\mathbb{F}_q$. Assume $\sigma(\{1, \ldots, k\})$ is known for a secret $\sigma \in \mathcal{S}_n$ such that $\sigma(\mathcal{C}_1) = \mathcal{C}_2$. Then there is an efficient reduction from the problem of computing $\boldsymbol{S} \in \mathrm{GL}_k(\mathbb{F}_q)$ and a permutation matrix $\boldsymbol{P}$ with $\boldsymbol{G}_2 = \boldsymbol{S}\boldsymbol{G}_1\boldsymbol{P}$ to the Graph Isomorphism problem.*

Proof. Assume that $\boldsymbol{P}'$ is the permutation matrix corresponding to $\sigma' \in \mathcal{S}_n$ such that $\sigma' \circ \sigma(\{1, \ldots, k\}) = \{1, \ldots, k\}$. Such a permutation can be efficiently computed from the knowledge of $\sigma(\{1, \ldots, k\})$, and we get the identity

$$\begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix} \boldsymbol{P}' = \boldsymbol{S} \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix} \boldsymbol{P}\boldsymbol{P}'.$$

[5] Technically an undirected bipartite graph $G = (V = (L \sqcup R), E)$ with $m = |L|, n = |R|$ and $\mathbb{F}_q$-labeled edges is represented by an $(m+n) \times (m+n)$ adjacency matrix A' of the form $A' = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}$. However, throughout the paper we follow the convention of representing G simply by the block $A \in \mathbb{F}_q^{m \times n}$, and refer to this as the adjacency matrix of G.

The permutation matrix $\boldsymbol{P}\boldsymbol{P}'$ corresponds to $\sigma' \circ \sigma$. Hence, it fixes $\{1, \ldots, k\}$. The left $k \times k$ submatrix $\boldsymbol{S}\boldsymbol{P}\boldsymbol{P}'$ of the right-hand side is invertible. Hence, so is the left $k \times k$ submatrix of the left-hand side, and after computation of the systematic form of the left hand side matrix, we have an identity of the form

$$\begin{pmatrix} I_k & \boldsymbol{A}_2' \end{pmatrix} = \boldsymbol{S}' \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix} \boldsymbol{P}\boldsymbol{P}', \tag{2.1}$$

where $\boldsymbol{S}'$ is invertible. Since the permutation corresponding to $\boldsymbol{P}\boldsymbol{P}'$ fixes $\{1, \ldots, k\}$, it has the form $\boldsymbol{P}\boldsymbol{P}' = \begin{pmatrix} \boldsymbol{P}_1 & \\ & \boldsymbol{P}_2 \end{pmatrix}$ where $\boldsymbol{P}_1$ is a permutation matrix of $\mathcal{S}_k$. By Equation (2.1), we obtain:

- $I_k = \boldsymbol{S}'\boldsymbol{P}_1$.
- $\boldsymbol{A}_2' = \boldsymbol{S}'\boldsymbol{A}_1\boldsymbol{P}_2$.

The first item shows that $\boldsymbol{S}' = \boldsymbol{P}_1^{-1}$ is a permutation matrix, and since $\boldsymbol{A}_2' = \boldsymbol{S}'\boldsymbol{A}_1\boldsymbol{P}_2$, we can recover $\boldsymbol{S}'$ and $\boldsymbol{P}_2$ by solving the Graph Isomorphism problem on the $\mathbb{F}_q$-colored bipartite graphs defined by interpreting $\boldsymbol{A}_2'$ and $\boldsymbol{A}_1$ as adjacency matrices. □

Without the knowledge of $\sigma(\{1, \ldots, k\})$, we need to enumerate the $\binom{n}{k}$ possibilities for $\sigma(1), \ldots, \sigma(k)$. For each possible choice, we proceed with the method outlined above and attempt to solve PCE from its reduction to GI. The algorithm can be summed up with the pseudocode described in Algorithm 1:

Require: Generator matrices $\boldsymbol{G}_1 = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}, \boldsymbol{G}_2 = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$ of two permutationally equivalent $[n, k, d]_q$ codes.
Ensure: $\boldsymbol{S}$ invertible and a permutation matrix $\boldsymbol{P}$ with $\boldsymbol{G}_2 = \boldsymbol{S}\boldsymbol{G}_1\boldsymbol{P}$.

```
for all size-k subsets {i_1, ..., i_k} ⊆ {1, ..., n} do
    Compute a permutation matrix P' that maps i_1, ..., i_k to {1, ..., k}.
    if the first k columns of G_2 P' are not independent then
        break;
    end if
    Compute the systematic form (I_k  A_2') of G_2 P'.
    Find permutation matrices P_1, P_2 with A_2' = P_1 A_1 P_2 via Babai's GI algorithm if they exist.
end for
return S = P_1, P = ( P_1^{-1}      )
                    (        P_2    ).
```

Algorithm 1: Original Babai deterministic PCE algorithm.

Theorem 2.11. *Algorithm 1 is correct, and it solves PCE using* $\binom{n}{k} \sim 2^{nH(k/n)}$ *calls to an oracle for GI.*

Since the complexity to solve GI is $2^{\log(n)^c}$ for a constant c, the overall complexity of Babai's PCE algorithm is in $2^{nH(k/n)(1+o(1))}$. Note that it is a deterministic algorithm.

3. A quasipolynomial-time algorithm for LRL

In this section, we present a quasipolynomial time algorithm to solve the so-called *Left-Right Linear equivalence* problem (LRL): given two matrices $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{F}_q^{n \times n}$, find monomial matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ such that $\boldsymbol{A}_2 = \boldsymbol{Q}_1\boldsymbol{A}_1\boldsymbol{Q}_2$. The LRL equivalence problem was introduced by Chou, Persichetti and Santini [CPS23] who proposed a heuristic algorithm to solve it for large field sizes q. Recently, Nowakowski [Now25] described an algorithm with proven time complexity that solves LRL equivalence on random codes when $q \geq 7$. Our approach to solve LRL equivalence relies on an efficient reduction to GI. This method for the resolution of LRL equivalence is the key technical ingredient that enables our extension of Babai's code equivalence algorithm to LCE in Section 4.

3.1. **Block permutations.** Here, we reuse ingredients of the reduction from LCE to PCE. Simply put, we can solve linear code equivalence of two codes by finding a permutation between their closures. Suppose that $(\sigma, \boldsymbol{v}) \in \mathcal{S}_n \times (\mathbb{F}_q^*)^n$ defines a linear isometry between k-dimensional codes $\mathcal{C}_1$ and $\mathcal{C}_2$ over $\mathbb{F}_q^n$. The corresponding permutation $\pi \in \mathcal{S}_{n(q-1)}$ is such that $\widetilde{C}_2 = \pi(\widetilde{C}_1)$ acts block by block, where each of the n blocks of $q-1$ coordinates in $\widetilde{C}_1, \widetilde{C}_2$ corresponds to a single coordinate in $\mathcal{C}_1, \mathcal{C}_2$. We will use this property to solve LRL.

Definition 3.1 (Block of indices)**.** The indices of the closure $\tilde{\mathcal{C}}$ of a code $\mathcal{C}$ of length n over $\mathbb{F}_q$ are divided into n blocks. For $i \leq n$, the block B_i is defined as

$$B_i = \{(q-1)(i-1)+1, (q-1)(i-1)+2, \ldots, (q-1)(i-1)+(q-1)\}.$$

The indices of the block B_i are mapped to that of the block $B_{\sigma(i)}$. Within the block, entries are also re-ordered according to a permutation of the $q-1$ entries that is defined by $v_{\sigma(i)}$. More specifically, the i-th block $(\lambda_1 c_i, \lambda_2 c_i, \ldots, \lambda_{q-1} c_i)$ first gets mapped to the $\sigma(i)$-th block, and then the following transformation takes place:

$$(\lambda_1 c_i, \lambda_2 c_i, \ldots, \lambda_{q-1} c_i) \mapsto (\lambda_1 v_{\sigma(i)} c_i, \lambda_2 v_{\sigma(i)} c_i, \ldots, \lambda_{q-1} v_{\sigma(i)} c_i) = (\lambda_{\pi_{v_{\sigma(i)}}(1)} c_i, \lambda_{\pi_{v_{\sigma(i)}}(2)} c_i, \ldots, \lambda_{\pi_{v_{\sigma(i)}}(q-1)} c_i),$$

where the permutation $\pi_v \in \mathcal{S}_{q-1}$ for $v \in \mathbb{F}_q^*$ is induced by the bijection of $\mathbb{F}_q^*$ given by $x \mapsto vx$, that is: $v\lambda_l = \lambda_{\pi_v(l)}$. This means that the permutation $\pi \in \mathcal{S}_{n(q-1)}$ corresponding to $(\sigma, \boldsymbol{v})$ acts in the following way for all $i \leq n$, and $l \leq q-1$:

$$(q-1)(i-1) + l \longmapsto (q-1)(\sigma(i)-1) + \pi_{v_{\sigma(i)}}(l).$$

Definition 3.2 (Block permutation)**.** Let $\pi_1, \ldots, \pi_n \in \mathcal{S}_{q-1}$ and $\sigma \in \mathcal{S}_n$. The *block permutation* defined by σ and $(\pi_i)_{i \leq n}$ has the $n(q-1) \times n(q-1)$ permutation matrix $(\boldsymbol{P}_\sigma \otimes I_{q-1}) \cdot \mathrm{diag}(\boldsymbol{P}_{\pi_1}, \ldots, \boldsymbol{P}_{\pi_n})$, where $\boldsymbol{P}_\pi$ is the permutation matrix corresponding to the permutation π. The block permutation performs the following operation:

$$(q-1)(i-1) + l \longmapsto (q-1)(\sigma(i)-1) + \pi_{\sigma(i)}(l).$$

One can immediately check that the following holds:

Lemma 3.3. *Suppose $(\sigma, \boldsymbol{v}) \in \mathcal{S}_n \times \mathbb{F}_q^{*n}$ defines a linear isometry between k-dimensional codes $\mathcal{C}_1$ and $\mathcal{C}_2$ over $\mathbb{F}_q^n$. The corresponding permutation $\pi \in \mathcal{S}_{n(q-1)}$ such that $\pi(\tilde{\mathcal{C}}_1) = \tilde{\mathcal{C}}_2$ satisfies*

$$\boldsymbol{P}_\pi = (\boldsymbol{P}_\sigma \otimes I_{q-1}) \cdot \mathit{diag}(\boldsymbol{P}_{\pi_{v_1}}, \ldots, \boldsymbol{P}_{\pi_{v_n}})$$

3.2. **Reducing LRL to GI.** Given input matrices $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{F}_q^{m \times n}$, search for monomial matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ such that $\boldsymbol{A}_2 = \boldsymbol{Q}_1 \boldsymbol{A}_1 \boldsymbol{Q}_2$. We reduce this task to the Graph Isomorphism problem by expanding rows and columns of $\boldsymbol{A}_1, \boldsymbol{A}_2$ in such a way that $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ act by block permutation on the rows (resp. columns) of the expanded matrices. This strategy directly follows the methods to produce the closure of a code [SS13]. However, we cannot directly reuse the rigorous proofs provided in [BW24] as our statements apply directly to matrices, and include row expansion in addition the usual column expansion.

Definition 3.4 (Column-expanded matrix)**.** Let $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$ and $\{\lambda_1, \ldots, \lambda_{q-1}\} = \mathbb{F}_q^*$. The column-expanded matrix of $\boldsymbol{A}$ (denoted $\tilde{\boldsymbol{A}}$) is the $m \times n(q-1)$ matrix over $\mathbb{F}_q$ where each coefficient $a_{i,j}$ of $\boldsymbol{A}$ is replaced by the $1 \times (q-1)$ block

$$(\lambda_1 a_{i,j}, \lambda_2 a_{i,j}, \ldots, \lambda_{q-1} a_{i,j}),$$

in other words, $\tilde{\boldsymbol{A}} = \boldsymbol{A} \otimes (\lambda_1, \ldots, \lambda_{q-1})$.

Clearly, if $\boldsymbol{G}$ is a generator matrix for the linear code $\mathcal{C}$, then $\tilde{\boldsymbol{G}}$ is the generator matrix for its closure $\tilde{\mathcal{C}}$. The same operation can then be performed row-wise, i.e. the matrix can be expanded by replacing each of the rows with the block made of all its $q-1$ possible scalar multiples.

Definition 3.5 (Expanded matrix)**.** Let $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$ and $\{\lambda_1, \ldots, \lambda_{q-1}\} = \mathbb{F}_q^*$. The expanded matrix of $\boldsymbol{A}$ (denoted $\widehat{\boldsymbol{A}}$) is the $m(q-1) \times n(q-1)$ matrix over $\mathbb{F}_q$ consisting in replacing each row R_i of $\tilde{\boldsymbol{A}}$ by the block $(\lambda_1 R_i, \lambda_2 R_i, \ldots, \lambda_{q-1} R_i)$. In other words:

$$\widehat{\boldsymbol{A}} = \widetilde{\tilde{\boldsymbol{A}}^T}^T,$$

or equivalently $\widehat{\boldsymbol{A}} = (\lambda_1, \ldots, \lambda_{q-1})^T \otimes \tilde{\boldsymbol{A}} = (\lambda_1, \ldots, \lambda_{q-1})^T \otimes \boldsymbol{A} \otimes (\lambda_1, \ldots, \lambda_{q-1})$.

The following lemma is a similar assertion to the fact that one can reduce LCE to PCE by taking the closure of the input codes.

Lemma 3.6. *Let $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{F}_q^{m \times n}$. Then there exists a monomial matrix $\boldsymbol{Q}$ such that $\boldsymbol{A}_2 = \boldsymbol{A}_1\boldsymbol{Q}$ if and only if there exists a permutation matrix $\boldsymbol{P}$ such that $\tilde{\boldsymbol{A}}_2 = \tilde{\boldsymbol{A}}_1\boldsymbol{P}$.*

Proof. First, assume that there exists a monomial matrix $\boldsymbol{Q}$ such that $\boldsymbol{A}_2 = \boldsymbol{A}_1\boldsymbol{Q}$. Let $\sigma \in \mathcal{S}_n$ and $\boldsymbol{v} \in \mathbb{F}_q^{*n}$ be defined by the linear isometry corresponding to $\boldsymbol{Q}$. Then according to Lemma 3.3 the block permutation matrix

$$\boldsymbol{P} = (\boldsymbol{P}_\sigma \otimes I_{q-1}) \cdot \mathrm{diag}(\boldsymbol{P}_{\pi_{v_1}}, \ldots, \boldsymbol{P}_{\pi_{v_n}})$$

satisfies $\tilde{\boldsymbol{A}}_2 = \tilde{\boldsymbol{A}}_1\boldsymbol{P}$.

Now, assume that there exists a permutation matrix $\boldsymbol{P}$ such that $\tilde{\boldsymbol{A}}_2 = \tilde{\boldsymbol{A}}_1\boldsymbol{P}$. This means that the columns of $\tilde{\boldsymbol{A}}_2$ are a permutation of the columns of $\tilde{\boldsymbol{A}}_1$. Up to a reordering of the identical columns, we can assume that $\boldsymbol{P}$ is a block permutation matrix, i.e. that it is of the form

$$(\boldsymbol{P}_\sigma \otimes I_{q-1}) \cdot \mathrm{diag}(\boldsymbol{P}_{\pi_1}, \ldots, \boldsymbol{P}_{\pi_n}),$$

where $\sigma \in \mathcal{S}_n$ and $\pi_1, \ldots, \pi_n \in \mathcal{S}_{q-1}$. Let $C_1^1, \ldots, C_n^1$ be the columns of $\boldsymbol{A}_1$ and $C_1^2, \ldots, C_n^2$ be the columns of $\boldsymbol{A}_2$. Under the block permutation, the block of columns $(\lambda_1 C_i^1, \ldots, \lambda_{q-1} C_i^1)$ is mapped to the block of columns $(\lambda_1 C_{\sigma(i)}^2, \ldots, \lambda_{q-1} C_{\sigma(i)}^2)$. In particular, the first column of the block $\lambda_1 C_i^1$ is mapped to $\lambda_{\pi_{\sigma(i)}(1)} C_{\sigma(i)}^2$. Hence we necessarily have $C_{\sigma(i)}^2 = v_{\sigma(i)} C_i^1$ for $v_{\sigma(i)} := \lambda_1 \lambda_{\pi_{\sigma(i)}(1)}^{-1}$. By the definition of $\pi_{v_{\sigma(i)}}$, we have that

$$(\lambda_1 C_{\sigma(i)}^2, \ldots, \lambda_{q-1} C_{\sigma(i)}^2) = (v_{\sigma(i)}\lambda_1 C_i^1, \ldots, v_{\sigma(i)}\lambda_{q-1} C_i^1) = (\lambda_{\pi_{v_{\sigma(i)}}(1)} C_i^1, \ldots, \lambda_{\pi_{v_{\sigma(i)}}(q-1)} C_i^1).$$

This means that $\pi_i = \pi_{v_i}$ for all i. Hence $\boldsymbol{A}_2 = \boldsymbol{A}_1\boldsymbol{Q}$ where $\boldsymbol{Q}$ is the monomial matrix corresponding to the linear isometry defined by $(\sigma, \boldsymbol{v})$. □

Corollary 3.7. *Given the column expanded matrices $\tilde{\boldsymbol{A}}_1, \tilde{\boldsymbol{A}}_2$ of $\boldsymbol{A}_1, \boldsymbol{A}_2$, and a permutation matrix $\boldsymbol{P}$ such that $\tilde{\boldsymbol{A}}_2 = \tilde{\boldsymbol{A}}_1\boldsymbol{P}$, there is an efficient procedure to compute a monomial matrix $\boldsymbol{Q}$ such that $\boldsymbol{A}_2 = \boldsymbol{A}_1\boldsymbol{Q}$.*

Proof. Assume that there is a permutation matrix $\boldsymbol{P}$ such that $\tilde{\boldsymbol{A}}_2 = \tilde{\boldsymbol{A}}_1\boldsymbol{P}$. As mentioned in the proof of Lemma 3.6, there is a block permutation matrix $\boldsymbol{P}'$ such that $\tilde{\boldsymbol{A}}_2 = \tilde{\boldsymbol{A}}_1\boldsymbol{P}'$. It can be obtained from the re-ordering of the action of $\boldsymbol{P}$ on identical columns of $\tilde{\boldsymbol{A}}_1$. Let us show that this procedure is efficient. The matrices $\tilde{\boldsymbol{A}}_1$ and $\tilde{\boldsymbol{A}}_2$ are made of n blocks of $q-1$ columns. The columns of a given block are the scalar multiples of a single element. To create a block permutation $\boldsymbol{P}'$ from $\boldsymbol{P}$, we start with the first block B_1 of indices. The first column $\lambda_1 C_1^1$ of $\tilde{A}_1$ is mapped to $\lambda_l C_i^2$. Since $\boldsymbol{P}$ is not necessarily a block permutation, the rest of the columns of the first block of $\tilde{A}_1$ are not necessarily mapped to the i-th block of $\tilde{\boldsymbol{A}}_2$. However, if one of them $\lambda_{l'} C_1^1$ is mapped to an element of the j-th block of $\tilde{A}_2$ for $j \neq i$, we necessarily have that the columns of $\tilde{\boldsymbol{A}}_2$ of indices in B_i are equal to the ones of index in B_j. Hence we can swap the pre-images of the columns of index i with the columns of $\tilde{\boldsymbol{A}}_1$ of index in B_1. This way, we ensure that all columns of $\tilde{\boldsymbol{A}}_1$ of index in B_1 are mapped to a column of $\tilde{\boldsymbol{A}}_2$ of index in B_i. We set $\sigma(1) := i$, and repeat the process for the blocks of index $2, \ldots, n$. At the end of the procedure, we have a block permutation matrix $\boldsymbol{P}' = (\boldsymbol{P}_\sigma \otimes I_{q-1}) \cdot \mathrm{diag}(\boldsymbol{P}_{\pi_1}, \ldots, \boldsymbol{P}_{\pi_n})$. Finally, for all i, we set $v_{\sigma(i)} := \lambda_1 \lambda_{\pi_{\sigma(i)}(1)}^{-1}$. The monomial matrix $\boldsymbol{Q}$ corresponding to the linear isometry defined by $(\sigma, \boldsymbol{v})$ satisfies $\boldsymbol{A}_2 = \boldsymbol{A}_1\boldsymbol{Q}$. □

Lemma 3.8. *Let $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$ and $\boldsymbol{B} \in \mathbb{F}_q^{n \times l}$. Then we have*

$$\widetilde{\boldsymbol{AB}} = \boldsymbol{A}\tilde{\boldsymbol{B}}.$$

Proof. This follows from the fact that $(AB) \otimes \lambda = (\boldsymbol{AB}) \otimes (I_n \lambda) = (\boldsymbol{A}I_n) \otimes (\boldsymbol{B} \otimes \lambda) = \boldsymbol{A}(\boldsymbol{B} \otimes \lambda)$ for $\lambda = (\lambda_1, \ldots, \lambda_{q-1})$. □

Corollary 3.9. *Let $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{F}_q^{m \times n}$. There exist monomial matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ such that $\boldsymbol{A}_2 = \boldsymbol{Q}_1\boldsymbol{A}_1\boldsymbol{Q}_2$ if and only if there exist a permutation matrix $\boldsymbol{P}_2$ and a monomial matrix $\boldsymbol{Q}_1$ such that*

$$\tilde{\boldsymbol{A}}_2 = \boldsymbol{Q}_1\tilde{\boldsymbol{A}}_1\boldsymbol{P}_2.$$

Proof. We apply Lemma 3.6 to $\widetilde{\boldsymbol{Q}_1\tilde{\boldsymbol{A}}_1} = \boldsymbol{Q}_1\tilde{\boldsymbol{A}}_1$ and $\tilde{\boldsymbol{A}}_2$. □

Corollary 3.10. *Let $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{F}_q^{m\times n}$. There exist monomial matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ such that $\boldsymbol{A}_2 = \boldsymbol{Q}_1\boldsymbol{A}_1\boldsymbol{Q}_2$ if and only if there exist permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ with*

$$\widehat{\boldsymbol{A}_2} = \boldsymbol{P}_1\widehat{\boldsymbol{A}_1}\boldsymbol{P}_2.$$

Proof. By applying Corollary 3.9 a first time, we have $\boldsymbol{A}_2 = \boldsymbol{Q}_1\boldsymbol{A}_1\boldsymbol{Q}_2$ if and only if there is a permutation matrix $\boldsymbol{P}_2$ and monomial matrix $\boldsymbol{Q}_1$ such that $\tilde{\boldsymbol{A}}_2 = \boldsymbol{Q}_1\tilde{\boldsymbol{A}}_1\boldsymbol{P}_2$. By taking the transpose on both sides, this means that

$$\boldsymbol{P}_2^T\ \tilde{\boldsymbol{A}}_1{}^T\ \boldsymbol{Q}_1^T = \tilde{\boldsymbol{A}}_2{}^T$$

We apply Corollary 3.9 a second time to state that this is equivalent to the existence of a permutation matrix $\boldsymbol{P}_1$ satisfying

$$\boldsymbol{P}_2^T\ \widetilde{\tilde{\boldsymbol{A}}_1{}^T}\ \boldsymbol{P}_1 = \widetilde{\tilde{\boldsymbol{A}}_2{}^T}.$$

We transpose again to conclude that this is equivalent to

$$\boldsymbol{P}_1^T\ \widetilde{\tilde{\boldsymbol{A}}_1{}^T}^T\ \boldsymbol{P}_2 = \widetilde{\tilde{\boldsymbol{A}}_2{}^T}^T,$$

i.e. $\boldsymbol{P}_1^T\widehat{\boldsymbol{A}_1}\boldsymbol{P}_2 = \widehat{\boldsymbol{A}_2}$ since for all $\boldsymbol{A}$, we have $\widehat{\boldsymbol{A}} = \widetilde{\tilde{\boldsymbol{A}}^T}^T$. □

Algorithm 2 is correct thanks to Corollary 3.10, and it takes advantage of Corollary 3.7 to solve LRL by reducing it to an instance of GI given by the extended matrices of the input matrices. Its runtime is dominated by that of Babai's GI algorithm (Step 1). Then, with $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that $\widehat{\boldsymbol{A}_2} = \boldsymbol{P}_1\widehat{\boldsymbol{A}_1}\boldsymbol{P}_2$, we have that

$$\boldsymbol{P}_2^T\widehat{\boldsymbol{A}_1}{}^T\boldsymbol{P}_1^T = \boldsymbol{P}_2^T\widetilde{\tilde{\boldsymbol{A}}_1{}^T}\boldsymbol{P}_1^T = \widetilde{\boldsymbol{P}_2^T\tilde{\boldsymbol{A}}_1{}^T}\boldsymbol{P}_1^T = \widehat{\boldsymbol{A}_2}^T = \widetilde{\tilde{\boldsymbol{A}}_2{}^T}.$$

A first use of Corollary 3.7 gives us a monomial matrix $\boldsymbol{Q}_1$ with $\tilde{\boldsymbol{A}}_2{}^T = \boldsymbol{P}_2^T\ \tilde{\boldsymbol{A}}_1{}^T\ \boldsymbol{Q}_1$. By transposing again, we see that

$$\boldsymbol{Q}_1^T\tilde{\boldsymbol{A}}_1\boldsymbol{P}_2 = \widetilde{\boldsymbol{Q}_1^T\boldsymbol{A}_1}\boldsymbol{P}_2 = \tilde{\boldsymbol{A}}_2.$$

Another use of Corollary 3.7 gives us a monomial matrix $\boldsymbol{Q}_2$ with $\boldsymbol{Q}_1^T\boldsymbol{A}_1\boldsymbol{Q}_2 = \boldsymbol{A}_2$.

Require: Matrices $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{F}_q^{m\times n}$
Ensure: Monomial matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ such that $\boldsymbol{A}_2 = \boldsymbol{Q}_1\boldsymbol{A}_1\boldsymbol{Q}_2$, or $\perp$ if no such matrices exist.
1: Use Babai's GI algorithm on the $\mathbb{F}_q$-colored bipartite graphs $\mathcal{G}_1, \mathcal{G}_2$ defined by $\widehat{\boldsymbol{A}_1}$ and $\widehat{\boldsymbol{A}_2}$.
2: **if** $\mathcal{G}_1$ is not isomorphic to $\mathcal{G}_2$ **then**
3: **return** $\perp$
4: **else**
5: Let permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that $\widehat{\boldsymbol{A}_2} = \boldsymbol{P}_1\widehat{\boldsymbol{A}_1}\boldsymbol{P}_2$.
6: Use Corollary 3.7 to compute a monomial matrix $\boldsymbol{Q}_1$ with $\tilde{\boldsymbol{A}}_2{}^T = \boldsymbol{P}_2^T\ \tilde{\boldsymbol{A}}_1{}^T\ \boldsymbol{Q}_1$.
7: Use Corollary 3.7 to compute a monomial matrix $\boldsymbol{Q}_2$ with $\boldsymbol{Q}_1^T\boldsymbol{A}_1\boldsymbol{Q}_2 = \boldsymbol{A}_2$.
8: **end if**
9: **return** $\boldsymbol{Q}_1^T, \boldsymbol{Q}_2$.

Algorithm 2: Resolution of LRL equivalence.

4. A variant of Babai's algorithm for solving LCE

Assume that two input codes $\mathcal{C}_1, \mathcal{C}_2$ satisfy $\mathcal{C}_2 = \tau(\mathcal{C}_1)$ for a linear isometry τ defined by $(\sigma, \boldsymbol{v}) \in \mathcal{S}_n \rtimes (\mathbb{F}_q^*)^n$. We propose a variant of Babai's PCE algorithm to retrieve τ. Our strategy consists in reducing the search for τ to the resolution of an instance of LRL. First, assume we know $\sigma(\{1, \dots, k\})$. As before, we also assume that the input codes are given in systematic form, i.e., with generator matrices of the form $\boldsymbol{G}_1 = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$, and $\boldsymbol{G}_2 = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$. We obtain the following result which is similar to Theorem 2.10, but for LCE instead of PCE.

Lemma 4.1 (LCE to LRL reduction). *Let* $\boldsymbol{G}_1 = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$, *and* $\boldsymbol{G}_2 = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$ *be generator matrices of two codes* $\mathcal{C}_1, \mathcal{C}_2$ *of dimension* k *and length* n *over* $\mathbb{F}_q$ *that are linearly equivalent under the action of the linear isometry given by* $(\sigma, \boldsymbol{v}) \in \mathcal{S}_n \times (\mathbb{F}_q^*)^n$. *Assuming* $\sigma(\{1, \dots, k\})$ *is known, there is an efficient reduction from the problem of computing* $\sigma, \boldsymbol{v}$ *to the search for permutation matrices* $\boldsymbol{P}_1, \boldsymbol{P}_2$ *and diagonal matrices* $\boldsymbol{D}_1, \boldsymbol{D}_2$ *with non-zero entries such that*

$$\boldsymbol{A}_2 = \boldsymbol{D}_1 \boldsymbol{P}_1 \boldsymbol{A}_1 \boldsymbol{P}_2 \boldsymbol{D}_2.$$

Proof. We can easily construct a permutation $\sigma' \in \mathcal{S}_n$ such that $\sigma' \circ \sigma(\{1, \dots, k\}) = \{1, \dots, k\}$. Let $\boldsymbol{P}_\sigma, \boldsymbol{P}_{\sigma'}$ be the $n \times n$ permutation matrices corresponding to σ, σ'. Solving LCE corresponds to finding $\boldsymbol{P}_\sigma$ and a diagonal matrix with non-zero entries $\boldsymbol{D}$ such that $\boldsymbol{G}_2 = \boldsymbol{S}\boldsymbol{G}_1 \boldsymbol{P}_\sigma \boldsymbol{D}$ for some invertible matrix $\boldsymbol{S}$. We multiply by $\boldsymbol{P}_{\sigma'}$ on both sides to obtain

$$\boldsymbol{G}_2 \boldsymbol{P}_{\sigma'} = \boldsymbol{S}\boldsymbol{G}_1 \boldsymbol{P}_\sigma \boldsymbol{D} \boldsymbol{P}_{\sigma'} = \boldsymbol{S}\boldsymbol{G}_1 \boldsymbol{P}_\sigma \boldsymbol{P}_{\sigma'} \boldsymbol{D}_{\sigma'},$$

where $\boldsymbol{D}_{\sigma'}$ is the diagonal matrix whose entries are those of $\boldsymbol{D}$ under the permutation σ'. The linear isometry defined by the monomial matrix $\boldsymbol{P}_\sigma \boldsymbol{P}_{\sigma'} \boldsymbol{D}_{\sigma'}$ maps the indices $\{1, \dots, k\}$ to $\{1, \dots, k\}$ (and scales the corresponding entries). The two main consequences that are:

(1) $\boldsymbol{P}_\sigma \boldsymbol{P}_{\sigma'} \boldsymbol{D}_{\sigma'} = \begin{pmatrix} \boldsymbol{P}_1 \boldsymbol{D}_1 & \\ & \boldsymbol{P}_2 \boldsymbol{D}_2 \end{pmatrix}$ where $\boldsymbol{P}_1, \boldsymbol{P}_2$ are permutation matrix, $\boldsymbol{D}_1, \boldsymbol{D}_2$ are diagonal matrices with non zero entries, and the upper left block is of size $k \times k$.

(2) $\{1, \dots, k\}$ is an information set of $\boldsymbol{G}_2 \boldsymbol{P}_{\sigma'}$.

According to Point 2, we can multiply by an invertible matrix on the left to put $\boldsymbol{G}_2 \boldsymbol{P}_{\sigma'}$ in systematic form. Hence, there exist a $k \times k$ invertible matrix $\boldsymbol{S}'$ and a $k \times (n-k)$ matrix $\boldsymbol{A}'_2$ such that

$$\begin{pmatrix} I_k & \boldsymbol{A}'_2 \end{pmatrix} = \boldsymbol{S}' \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix} \begin{pmatrix} \boldsymbol{P}_1 \boldsymbol{D}_1 & \\ & \boldsymbol{P}_2 \boldsymbol{D}_2 \end{pmatrix}$$

By expanding the above product block by block, we obtain the following identities:

(1) $\boldsymbol{S}' \boldsymbol{P}_1 \boldsymbol{D}_1 = I_k$.

(2) $\boldsymbol{S}' \boldsymbol{A}_1 \boldsymbol{P}_2 \boldsymbol{D}_2 = \boldsymbol{A}'_2$.

Hence $\boldsymbol{S}' = \boldsymbol{D}_1^{-1} \boldsymbol{P}_1^{-1}$ is a monomial matrix, and we have $\boldsymbol{D}_1^{-1} \boldsymbol{P}_1^{-1} \boldsymbol{A}_1 \boldsymbol{P}_2 \boldsymbol{D}_2 = \boldsymbol{A}_2$. □

We can use the reduction in Lemma 4.1 from LCE to LRL (when the image of an information set by σ is known) to devise an analogue of Babai's algorithm that solves LCE. We assume that the two input codes $\mathcal{C}_1, \mathcal{C}_2$ have generator matrices $\boldsymbol{G}_1 = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$ and $\boldsymbol{G}_2 = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$ respectively. Assume that the linear isometry from $\mathcal{C}_1$ to $\mathcal{C}_2$ is defined by $(\sigma, \boldsymbol{v}) \in \mathcal{S}_n \times (\mathbb{F}_q^*)^n$. We enumerate the $\binom{n}{k}$ possibilities for $\{\sigma(1), \dots, \sigma(k)\}$. For each possible choice, we proceed with the method outlined above and attempt to solve LCE from its reduction to GI. The algorithm can be summed up with the following pseudocode:

Require: Generator matrices $\boldsymbol{G}_1 = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}, \boldsymbol{G}_2 = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$ of two linearly equivalent $[n, k, d]_q$ codes.
Ensure: $\boldsymbol{S}$ invertible and a monomial matrix $\boldsymbol{Q}$ with $\boldsymbol{G}_2 = \boldsymbol{S}\boldsymbol{G}_1 \boldsymbol{Q}$.

for all size-k subsets $\{i_1, \dots, i_k\} \subset \{1, \dots, n\}$ **do**
  Compute a permutation matrix $\boldsymbol{P}'$ that maps $i_1, \dots, i_k$ to $\{1, \dots, k\}$.
  if the first k columns of $\boldsymbol{G}_2 \boldsymbol{P}'$ are not independent **then**
    break.
  end if
  Compute $\begin{pmatrix} I_k & \boldsymbol{A}'_2 \end{pmatrix} = \mathrm{SF}(\boldsymbol{G}_2 \boldsymbol{P}')$.
  Use Algorithm 2 to look for monomial matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ with $\boldsymbol{A}'_2 = \boldsymbol{Q}_1 \boldsymbol{A}_1 \boldsymbol{Q}_2$.
  if Algorithm 2 does not return $\perp$ **then**
    break.
  end if
end for
return $\boldsymbol{S} = \boldsymbol{Q}_1$, $\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_1^{-1} & \\ & \boldsymbol{Q}_2 \end{pmatrix}$.

Algorithm 3: Variant of Babai's algorithm for LCE.

Theorem 4.2. *Algorithm 3 is correct, and it solves LCE after* $\binom{n}{k} \sim 2^{nH(k/n)}$ *calls to an oracle that solves GI between two* $\mathbb{F}_q$*-colored bipartite graphs with* $(q-1)n$ *vertices.*

Since from [Bab16], the complexity to solve GI on input graphs of size m is in $2^{\log(m)^c}$ for a constant c, and since by [ZKT85], we can reduce GI between $\mathbb{F}_q$-colored bipartite graphs of size $(q-1)n$ to GI on a graph of size $\mathrm{poly}(n+q)$, the overall time complexity of Babai's LCE algorithm is in $\mathrm{quasipoly}(n+q)\cdot\binom{n}{k} \leq 2^{n+o(n+q)}$. Note that it is a deterministic algorithm.

5. A meet-in-the-middle variants to solve PCE and LCE

In this section, we show how to solve the code equivalence problem using a meet-in-the-middle strategy combined with Babai's algorithm in time and space complexity bounded by $2^{n/2+o(n+q)}$. We begin by showing how to solve PCE, and then extend this to solve LCE.

5.1. Meet-in-the-middle approach to solve PCE.

The general approach of our variant of Babai's algorithm for finding π such that $\pi(\mathcal{C}_1) = \mathcal{C}_2$ consists in attempting to find information sets I_1 of $\mathcal{C}_1$ and I_2 of $\mathcal{C}_2$ such that after applying a permutation mapping $I_1 \mapsto [1,\dots,k]$ to $\mathcal{C}_1$ and a permutation mapping $I_2 \mapsto [1,\dots,k]$, PCE readily reduces to *GI*.

Lemma 5.1. *Let* $\boldsymbol{G}_1$ *be a generator matrix for* $\mathcal{C}_1$ *and* $\boldsymbol{G}_2$ *be a generator matrix for* $\mathcal{C}_2 = \pi(\mathcal{C}_1)$*. For all information sets* I_1 *of* $\mathcal{C}_1$ *and* $I_2 := \pi(I_1)$ *of* $\mathcal{C}_2$*, and for all permutations* σ_1 *mapping* I_1 *to* $[1,\dots,k]$ *and* σ_2 *mapping* I_2 *to* $[1,\dots k]$*, there exist permutation matrices* $\boldsymbol{P}_1 \in \mathbb{F}_q^{k\times k}, \boldsymbol{P}_2 \in \mathbb{F}_q^{(n-k)\times(n-k)}$*, and an invertible matrix* $\boldsymbol{S}$ *such that*

$$\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \boldsymbol{S}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\begin{pmatrix}\boldsymbol{P}_1 & \\ & \boldsymbol{P}_2\end{pmatrix},$$

where $\boldsymbol{P}_{I_1}$ *is the matrix of* σ_1 *and* $\boldsymbol{P}_{I_2}$ *is the matrix of* σ_2*.*

Proof. Assume $\boldsymbol{G}_2 = \boldsymbol{S}_0\boldsymbol{G}_1\boldsymbol{P}$ for some invertible matrix $\boldsymbol{S}_0$ and permutation matrix $\boldsymbol{P}$. Let I_1 be an information set for $\mathcal{C}_1$, and $\boldsymbol{P}_{I_1}$ be the matrix of a permutation that maps I_1 to $[1,\dots,k]$. Then, there is an invertible matrix $\boldsymbol{S}_1$ such that $\boldsymbol{S}_1\boldsymbol{G}_1\boldsymbol{P}_{I_1} = \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})$. This means that

$$\begin{aligned}\boldsymbol{G}_2 &= \boldsymbol{S}_0\boldsymbol{S}_1^{-1}\left(\boldsymbol{S}_1\boldsymbol{G}_1\boldsymbol{P}_{I_1}\right)\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}\\ &= \boldsymbol{S}_0\boldsymbol{S}_1^{-1}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}.\end{aligned}$$

Note that the permutation matrix $\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}$ corresponds to the permutation that maps $[1,\dots,k]$ to I_1, and then to $\pi(I_1) := I_2$ where $\pi(\mathcal{C}_1) = \mathcal{C}_2$. Hence I_2 is an information set of $\mathcal{C}_2$. Let P_{I_2} be the matrix of a permutation mapping I_2 to $[1,\dots,k]$. In particular, the permutation corresponding to $\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}\boldsymbol{P}_{I_2}$ maps $[1,\dots,k]$ to itself and hence has the shape

$$\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}\boldsymbol{P}_{I_2} = \begin{pmatrix}\boldsymbol{P}_1 & \\ & \boldsymbol{P}_2\end{pmatrix}.$$

By multiplying our previous identity by $\boldsymbol{P}_{I_2}$ on both sides, we obtain

$$\boldsymbol{G}_2\boldsymbol{P}_{I_2} = \boldsymbol{S}_0\boldsymbol{S}_1^{-1}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\begin{pmatrix}\boldsymbol{P}_1 & \\ & \boldsymbol{P}_2\end{pmatrix}.$$

Let $\boldsymbol{S}_2$ be the invertible matrix such that $\boldsymbol{S}_2\boldsymbol{G}_2\boldsymbol{P}_{I_2} = \mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2})$, and $\boldsymbol{S} := \boldsymbol{S}_2\boldsymbol{S}_0\boldsymbol{S}_1^{-1}$. We have

$$\boldsymbol{S}_2\boldsymbol{G}_2\boldsymbol{P}_{I_2} = \mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \boldsymbol{S}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\begin{pmatrix}\boldsymbol{P}_1 & \\ & \boldsymbol{P}_2\end{pmatrix}.$$

□

The above formalizes a property of an information set I_1 of $\mathcal{C}_1$ and its image $I_2 = \pi(I_1)$. We denote this by saying that I_1 and $\pi(I_1)$ are *matching information sets*.

Definition 5.2 (Matching information sets)**.** Let $\mathcal{C}_1, \mathcal{C}_2$ be two $[n,k]_q$ codes with generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$. We say that information sets I_1, I_2 for $\mathcal{C}_1, \mathcal{C}_2$, respectively, are *matching information sets* if there exists a permutation $\pi \in S_n$ such that $\pi(C_1) = \mathcal{C}_2$ and $\pi(I_1) = I_2$.

If I_1 and I_2 are matching information sets for $\mathcal{C}_1$ and $\mathcal{C}_2$, then there exist permutation matrices $\boldsymbol{P}_1 \in \mathbb{F}_q^{k\times k}$, $\boldsymbol{P}_2 \in \mathbb{F}_q^{(n-k)\times(n-k)}$, and an invertible matrix $\boldsymbol{S}$ such that

$$\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \boldsymbol{S}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) \begin{pmatrix} \boldsymbol{P}_1 & \\ & \boldsymbol{P}_2 \end{pmatrix}, \tag{5.1}$$

where $\boldsymbol{P}_{I_1}$ is the permutation matrix of a σ_1 that maps I_1 to $[1,\dots,k]$ and $\boldsymbol{P}_{I_2}$ is the permutation matrix of a σ_2 that maps I_2 to $[1,\dots,k]$. Moreover, from Equation (5.1) that if we know matching information sets I_1 and I_2 for $\mathcal{C}_1$ and $\mathcal{C}_2$ then we can efficiently find $\boldsymbol{A}_1, \boldsymbol{A}_2$ such that $\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$, $\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$, and

$$\begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix} = \boldsymbol{S} \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix} \begin{pmatrix} \boldsymbol{P}_1 & \\ & \boldsymbol{P}_2 \end{pmatrix}.$$

As in Babai's algorithm for solving PCE, this yields

- $\boldsymbol{S} = \boldsymbol{P}_1^{-1}$,
- $\boldsymbol{P}_1^{-1}\boldsymbol{A}_1\boldsymbol{P}_2 = \boldsymbol{A}_2$.

This means that the knowledge of two matching information sets allows us to efficiently reduce PCE to GI, just like in Babai's PCE algorithm. Next, we turn our attention to the design of a method to find two matching information sets I_1 and I_2. For this, we use properties of the $\mathbb{F}_q$-colored bipartie graphs defined by $\boldsymbol{A}_i$ where $\mathrm{SF}(\boldsymbol{G}_i\boldsymbol{P}_{I_i}) = \begin{pmatrix} I_k & \boldsymbol{A}_i \end{pmatrix}$. Indeed, Babai's strategy solves PCE when these graphs are isomorphic. To decide if it is the case, we use the notion of canonical form of a graph for which Babai described a quasipolynomial algorithm in [Bab19].

Definition 5.3 (Canonical form of a graph ([Bab19, Section 1]))**.** Let C be a class of finite graphs. A *canonical form* for the class C is a function $F : C \to C$ such that

(1) $\forall X \in C,\ X \simeq F(X)$.
(2) $\forall X, Y \in C,\ X \simeq Y \Leftrightarrow F(X) = F(Y)$.

The main result of [Bab19] is that computing a canonical form of a graph is doable in quasipolynomial time.

Theorem 5.4 ([Bab19, Corollary 2.3])**.** *There is a canonical form of graphs that can be computed in quasipolynomial time.*

We can reduce the resolution of PCE to the *Claw Finding Problem.*

Definition 5.5 (Claw Finding Problem)**.** Let A_1, A_2, B be finite sets and $f_1 : A_1 \to B$, $f_2 : A_2 \to B$ be two functions with the same range B. The *claw finding problem* consists of finding a pair $(x, y) \in A_1 \times A_2$ such that $f_1(x) = f_2(y)$ when it exists. Such a pair (x, y) is called a *claw.*

Given two input codes $\mathcal{C}_1, \mathcal{C}_2$ such that $\mathcal{C}_2 = \pi(\mathcal{C}_1)$ and of generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$, use the claw finding problem framework by letting A_1 be the set of all information sets of $\mathcal{C}_1$ and A_2 be the set of all information sets of $\mathcal{C}_2$. There is a quasipoly(n)-time computable canonical form for $\mathbb{F}_q$-colored graphs (and in particular, such bipartite graphs) by Theorem 5.4 and the reduction from GI on edge-colored graphs to non-edge-colored graphs in [ZKT85]. Let F be such a canonical form.

In the following, we identify a graph and its adjacency matrix. We define the set B as

$$B := \{F(\boldsymbol{A}) \ \mid\ \exists \text{ information set } I,\ \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_I) = \begin{pmatrix} I_k & \boldsymbol{A} \end{pmatrix}\},$$

where I is an information set of $\mathcal{C}_1$, and $\boldsymbol{P}_I$ the permutation matrix of a permutation that maps I to $[1,\dots,k]$.

Lemma 5.6. *Let $\mathcal{C}_1, \mathcal{C}_2$ be two $[n,k]_q$ codes such that there exists $\pi \in S_n$ with $\mathcal{C}_2 = \pi(\mathcal{C}_1)$. Let $\boldsymbol{G}_1, \boldsymbol{G}_2$ be generator matrices of $\mathcal{C}_1, \mathcal{C}_2$, respectively. We have*

$$B := \{F(\boldsymbol{A}_1) \ \mid\ \exists \boldsymbol{P}_{I_1},\ \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}\} = \{F(\boldsymbol{A}_2) \ \mid\ \exists \boldsymbol{P}_{I_2},\ \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_2}) = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}\},$$

where I_j is an information set for $\mathcal{C}_j$, and $\boldsymbol{P}_{I_j}$ denotes the matrix of a permutation that maps I_j to $[1,\dots,k]$ for $j = 1, 2$. Additionally, we have

$$|B| \leq \#\{\textit{information sets of } \mathcal{C}_1\} = \#\{\textit{information sets of } \mathcal{C}_2\}$$

Proof. To show the first claim, notice that every information set I_2 of $\mathcal{C}_2$ is of the form $I_2 = \pi(I_1)$ for some information set I_1 of $\mathcal{C}_1$. Hence for any permutation matrices $\boldsymbol{P}_{I_1}, \boldsymbol{P}_{I_2}$ mapping I_1 to $[1, \dots, k]$ (resp. I_2 to $[1, \dots, k]$), we must have $F(\boldsymbol{A}_1) = F(\boldsymbol{A}_2)$ where $\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$ and $\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$ (because I_1 and I_2 are matching information sets). Hence, since all information sets of $\mathcal{C}_1$ can be matched with one of $\mathcal{C}_2$ (and vice-versa), we must have

$$\{F(\boldsymbol{A}_1) \mid \exists \boldsymbol{P}_{I_1},\ \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}\} = \{F(\boldsymbol{A}_2) \mid \exists \boldsymbol{P}_{I_2},\ \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_2}) = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}\}.$$

Next, we need to argue that $|B|$ is bounded by the number of information sets. Let $j \in \{1, 2\}$, let I be an information set of $\mathcal{C}_j$, let $\boldsymbol{P}_I, \boldsymbol{P}'_I$ be the permutation matrices of two permutations that map I to $[1, \dots k]$ and let $\boldsymbol{A}, \boldsymbol{A}'$ such that $\mathrm{SF}(\boldsymbol{G}_j\boldsymbol{P}_I) = \begin{pmatrix} I_k & \boldsymbol{A} \end{pmatrix}$ and $\mathrm{SF}(\boldsymbol{G}_j\boldsymbol{P}'_I) = \begin{pmatrix} I_k & \boldsymbol{A}' \end{pmatrix}$. We need to prove that $F(\boldsymbol{A}) = F(\boldsymbol{A}')$ despite the fact that $\boldsymbol{P}_I$ and $\boldsymbol{P}'_I$ (and hence $\boldsymbol{A}$ and $\boldsymbol{A}'$) may be different. Let $\boldsymbol{S}, \boldsymbol{S}'$ be invertible matrices such that

$$\begin{aligned} \boldsymbol{S}\boldsymbol{G}_j\boldsymbol{P}_I &= \begin{pmatrix} I_k & \boldsymbol{A} \end{pmatrix} \\ \boldsymbol{S}'\boldsymbol{G}_j\boldsymbol{P}'_I &= \begin{pmatrix} I_k & \boldsymbol{A}' \end{pmatrix}. \end{aligned}$$

This means that we have the identity:

$$\boldsymbol{S}^{-1} \begin{pmatrix} I_k & \boldsymbol{A} \end{pmatrix} \boldsymbol{P}_I^{-1} = {\boldsymbol{S}'}^{-1} \begin{pmatrix} I_k & \boldsymbol{A}' \end{pmatrix} {\boldsymbol{P}'}_I^{-1}.$$

It immediately follows that $\begin{pmatrix} I_k & \boldsymbol{A}' \end{pmatrix} = \boldsymbol{S}'\boldsymbol{S}^{-1} \begin{pmatrix} I_k & \boldsymbol{A} \end{pmatrix} \boldsymbol{P}_I^{-1}\boldsymbol{P}'_I$. The matrix $\boldsymbol{P}_I^{-1}\boldsymbol{P}'_I$ is the permutation matrix of a permutation that maps $[1, \dots, k]$ to itself. Hence, it must factor as $\begin{pmatrix} \boldsymbol{P}_1 & \\ & \boldsymbol{P}_2 \end{pmatrix}$, where $\boldsymbol{P}_1$ is a $k \times k$ permutation matrix and $\boldsymbol{P}_2$ is a $(n-k) \times (n-k)$ permutation matrix. In particular, this means that $\boldsymbol{S}\,{\boldsymbol{S}'}^{-1} = \boldsymbol{P}_1$ and $\boldsymbol{A}' = \boldsymbol{P}_1^{-1}\boldsymbol{A}\boldsymbol{P}_2$, i.e., $F(\boldsymbol{A}) = F(\boldsymbol{A}')$. □

Proposition 5.7 (Reduction of PCE to the claw finding problem)**.** *Let $\mathcal{C}_1, C_2$ be two $[n, k]_q$ codes with respective generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$ such that $\mathcal{C}_2 = \pi(\mathcal{C}_1)$ for $\pi \in \mathcal{S}_n$. Let F be a canonical form for the class of $\mathbb{F}_q$-colored $(k, n-k)$-bipartite graphs. Let $j \in \{1, 2\}$ and let $\mathcal{I}_j$ be the set of information sets for $\mathcal{C}_j$, and*

$$B := \{F(\boldsymbol{A}_1) \mid \exists \boldsymbol{P}_{I_1},\ \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}\} = \{F(\boldsymbol{A}_2) \mid \exists \boldsymbol{P}_{I_2},\ \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_2}) = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}\},$$

where I_j is an information set for $\mathcal{C}_j$, and $\boldsymbol{P}_{I_j}$ denotes the matrix of a permutation that maps I_j to $[1, \dots, k]$ for $j = 1, 2$. We define $f_j : \mathcal{I}_j \to B$ by the following procedure on input $I \in \mathcal{I}_j$:

(1) Let $I = \{i_1, \dots, i_k\}$ where $i_1 < i_2 < \dots < i_k$.
(2) Let $\sigma \in \mathcal{S}_n = (1, i_1)(2, i_2) \dots (k, i_k)$.
(3) Let $\boldsymbol{P}_I$ be the permutation matrix corresponding to σ.
(4) $f_j(I) = F(\boldsymbol{A})$ where $\mathrm{SF}(\boldsymbol{G}_j\boldsymbol{P}_I) = \begin{pmatrix} I_k & \boldsymbol{A} \end{pmatrix}$

Then solving PCE efficiently reduces to finding a claw between f_1 and f_2.

Proof. Assume $(I_1, I_2) \in \boldsymbol{A}_1 \times \boldsymbol{A}_2$ satisfies $f_1(I_1) = f_2(I_2)$. First, compute:

$$\begin{aligned} \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) &= \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix} \\ \mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) &= \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}. \end{aligned}$$

Then, with Babai's algorithm for the resolution of the graph isomorphism problem [Bab16], Compute permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that $\boldsymbol{P}_1\boldsymbol{A}_1\boldsymbol{P}_2 = \boldsymbol{A}_2$. There exist invertible matrices $\boldsymbol{S}_1, \boldsymbol{S}_2$ such that

$$\boldsymbol{S}_2\boldsymbol{G}_2\boldsymbol{P}_{I_2} = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix} = \begin{pmatrix} I_k & \boldsymbol{P}_1\boldsymbol{A}_1\boldsymbol{P}_2 \end{pmatrix} = \boldsymbol{P}_1 \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix} \begin{pmatrix} \boldsymbol{P}_1^{-1} & \\ & \boldsymbol{P}_2 \end{pmatrix} = \boldsymbol{P}_1\boldsymbol{S}_1\boldsymbol{G}_1\boldsymbol{P}_{I_1} \begin{pmatrix} \boldsymbol{P}_1^{-1} & \\ & \boldsymbol{P}_2 \end{pmatrix}.$$

This means that $\boldsymbol{G}_2 = \boldsymbol{S}_2^{-1}\boldsymbol{P}_1\boldsymbol{S}_1\boldsymbol{G}_1\boldsymbol{P}$ for

$$\boldsymbol{P} = \boldsymbol{P}_{I_1} \begin{pmatrix} \boldsymbol{P}_1^{-1} & \\ & \boldsymbol{P}_2 \end{pmatrix} \boldsymbol{P}_{I_2}^{-1}.$$

Since $\boldsymbol{S} := \boldsymbol{S}_2^{-1}\boldsymbol{P}_1\boldsymbol{S}_1$ is invertible, we have $\boldsymbol{S}, \boldsymbol{P}$ such that $\boldsymbol{G}_2 = \boldsymbol{S}\boldsymbol{G}_1\boldsymbol{P}$, i.e., we solved PCE. □

Following the meet-in-the-middle approach (in a similar way as Nowakowski's work [Now25]), we solve the claw finding problem for f_1, f_2 and hence solve PCE. We draw m information sets from $\mathcal{C}_1$ according to a distribution $\mathcal{D}_1$ and store then in a list L, then we draw m information sets from $\mathcal{C}_2$ according to $\mathcal{D}_2$. For each information set I_2 of $\mathcal{C}_2$ we draw, we check if $f_2(I_2)$ corresponds to $f_1(I_1)$ for $I_1 \in L$. We specify m and $\mathcal{D}_I$ in the following section. This procedure is formalized in Algorithm 4.

Require: Generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$ of two equivalent $[n,k,d]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$, $m > 0$, and distributions $\mathcal{D}_1, \mathcal{D}_2$ on the information sets of $\mathcal{C}_1$ (resp. $\mathcal{C}_2$).
Ensure: A permutation matrix $\boldsymbol{P}$ satisfying $\mathrm{SF}(\boldsymbol{G}_2) = \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P})$.

for m iterations **do**
  Sample an information set I_1 of $\mathcal{C}_1$ from $\mathcal{D}_1$.
  $L \leftarrow L \cup (I_1, f_1(I_1))$.
end for
for m iterations **do**
  Sample an information set I_2 of $\mathcal{C}_2$ from $\mathcal{D}_2$.
  if there is $(I_1, f_1(I_1)) \in L$ with $f_1(I_1) = f_2(I_2)$ **then**
    Compute $\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$ and $\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$.
    Compute permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that $\boldsymbol{P}_1\boldsymbol{A}_1\boldsymbol{P}_2 = \boldsymbol{A}_2$ with Babai's algorithm [Bab16].
    break
  end if
end for
return $\boldsymbol{P} = \boldsymbol{P}_{I_1} \begin{pmatrix} \boldsymbol{P}_1^{-1} & \\ & \boldsymbol{P}_2 \end{pmatrix} \boldsymbol{P}_{I_2}^{-1}$

Algorithm 4: Meet-in-the-middle variant of Babai's PCE algorithm.

5.2. **Analysis of the runtime of Algorithm 4.** The correctness of Algorithm 4 readily derives from Proposition 5.7. To analyze its run time and its probability of success, we need to specify the distributions $\mathcal{D}_1, \mathcal{D}_2$ and m. In a nutshell, we use the distribution output by Algorithm 5, which is very close to the uniform distribution when T is large (in fact, it will suffice to take $T = \mathrm{poly}(n)$). Then we show that $m \in \Theta(\sqrt{N})$ samples is enough to guarantee a failure probability upper bounded by a constant, where $N \leq \binom{n}{k}$ is the number of information sets of $\mathcal{C}_1, \mathcal{C}_2$.

Require: An information set I_0 of $\boldsymbol{G} \in \mathbb{F}_q^{k\times n}$ and an integer $T > 0$.
Ensure: An information set of $\boldsymbol{G}$.

for $t = 1, 2, \ldots, T$ **do**
  Choose a uniformly random index $i \in I_{t-1}$.
  Let I_t to be an information set chosen uniformly at random among those containing $I_{t-1} \setminus \{i\}$.
end for
return I_T.

Algorithm 5: Basis exchange algorithm.

The following theorem asserts that the natural "down-up random walk" sampling algorithm in Algorithm 5 converges to the uniform distribution very quickly. It is from relatively recent work of Anari, Liu, Oveis Gharan, and Vinzant [ALOV19]; see also the lecture notes of Schramm [Sch22].

Theorem 5.8 ([ALOV19, Corollary 1.3]). *Let $\varepsilon > 0$, $N \leq \binom{n}{k}$ be the number of information sets, and $(p_i)_{i\leq N}$ be the probability distribution of the output of Algorithm 5 (i.e., p_i is the probability of outputting the ith information set). Then for $T \geq k \ln \frac{k}{\varepsilon}$, we have*

$$\sum_{i\leq N} \left| \frac{1}{N} - p_i \right| \leq \varepsilon.$$

For a relatively small number of steps T, the above algorithm produces an information set sampled from a distribution close to the uniform distribution.

In the following, we show that $\varepsilon < \frac{1}{N^2}$ is sufficient to achieve the desired bound on the success probability for some choice of $m \in \Theta(\sqrt{N})$. Our success probability is lower bounded by the probability that lists L_1 of m information sets of $\mathcal{C}_1$ independently drawn according to $\mathcal{D}_1$ and L_2 of m information sets of $\mathcal{C}_2$ independently drawn according to $\mathcal{D}_2$ contain $(I_1, \pi(I_1)) \in L_1 \times L_2$. Let π be a permutation such that $\mathcal{C}_2 = \pi(\mathcal{C}_1)$. For $i, j \leq m$, denote by:

$$X_{i,j} := \begin{cases} 1 & \text{if the } i\text{th element } I_i^1 \text{ of } L_1 \text{ and the } j\text{th element } I_j^2 \text{ of } L_2 \text{ satisfy } I_j^2 = \pi(I_i^1), \\ 0 & \text{otherwise.} \end{cases}$$

The number of matching information sets is then lower bounded by $X := \sum_{i,j} X_{i,j}$. The probability that $X_{i,j} = 1$ is $q := \sum_{i \leq N} p_i^2$. Let $\varepsilon_i := p_i - \frac{1}{N}$. In particular, $|\varepsilon_i| \leq \varepsilon$.

Lemma 5.9. *Assuming $\varepsilon < \frac{1}{N^2}$, we have $q = \frac{1}{N} + o\left(\frac{1}{N}\right)$.*

Proof. We have

$$\begin{aligned} q &= \sum_i p_i^2 \\ &= \sum_i \left(\frac{1}{N} + \varepsilon_i\right)^2 \\ &= \sum_i \frac{1}{N^2} + \sum_i \left(2\frac{\varepsilon_i}{N} + \varepsilon_i^2\right) \\ &= \frac{1}{N} + o\left(\frac{1}{N}\right). \end{aligned}$$

□

The mean and variance of X satisfy:

$$\begin{aligned} \mathrm{E}[X] &= m^2 \mathrm{E}[X_{i,j}] = m^2 q, \\ \mathrm{Var}[X] &= \sum_{i,j} \mathrm{Var}[X_{i,j}] + \sum_{(i,j) \neq (i',j')} \mathrm{Cov}(X_{i,j}, X_{i',j'}) \\ &= m^2 q(1-q) + \sum_{i,j,j' \neq j} 2\mathrm{Cov}(X_{i,j}, X_{i,j'}). \end{aligned}$$

The reduction on the number of cross terms comes from the fact that if i, i', j, j' are all different, $X_{i,j}$ and $X_{i',j'}$ are independent, and therefore $\mathrm{Cov}(X_{i,j}, X_{i',j'}) = 0$.

Lemma 5.10. *We have the following inequality:*

$$\Pr(\textit{no pair of matching information sets is obtained}) = \Pr(X = 0) \leq \frac{\mathrm{Var}[X]}{q^2 m^4}.$$

Proof. Let $\mu = \mathrm{E}[X]$, Chebyshev's inequality tells us that for $t > 0$, we have

$$P\left(|X - \mu| \geq t\right) \leq \frac{\mathrm{Var}[X]}{t^2}.$$

The event $\{X = 0\}$ is included in the event $\{|X - \mu| \geq t\}$ for $t = \mu = qm^2$. In this case, $\frac{\mathrm{Var}[X]}{t^2} = \frac{\mathrm{Var}[X]}{q^2 m^4}$. □

Given the above lemma, we want to argue that $\mathrm{Var}[X] = O(m^2 q)$. To do so, we bound the term $2\mathrm{Cov}(X_{i,j}, X_{i,j'})$.

Lemma 5.11. *Assume $|\varepsilon_i| < \frac{1}{N^2}$ for all $i \leq N$, and $m \leq N$. Then we have*

$$\mathrm{Var}[X] \leq m^2 q + o(m^2 q).$$

Proof. First, notice that

$$\begin{aligned}\mathrm{Cov}(X_{i,j}, X_{i,j'}) &= \mathrm{E}[X_{i,j}X_{i,j'}] - \mathrm{E}[X_{i,j}]\mathrm{E}[X_{i,j'}] \\ &= \Pr\left(x_i = y_j = y_{j'}\right) - \Pr(x_i = y_j)\Pr(x_i = y_{j'}) \\ &= \sum_{i \le N} p_i^3 - q^2\end{aligned}$$

Then:

$$\begin{aligned}\sum_{i \le N} p_i^3 &= \sum_i \left[\frac{1}{N^3} + 3\frac{\varepsilon_i}{N^2} + 3\frac{\varepsilon_i^2}{N} + \varepsilon_i^3\right] \\ &= \frac{1}{N^2} + 3\frac{\varepsilon_i}{N} + 3\varepsilon_i^2 + N\varepsilon_i^3 \\ &= \frac{1}{N^2} + o\left(\frac{1}{N^2}\right).\end{aligned}$$

Additionally, since $q = \frac{1}{N} + o\left(\frac{1}{N}\right)$, we have $q^2 = \frac{1}{N^2} + o\left(\frac{1}{N^2}\right)$. Therefore, $\mathrm{Cov}(X_{i,j}, X_{i,j'}) = o\left(\frac{1}{N^2}\right)$ and $\mathrm{Cov}(X_{i,j}, X_{i,j'}) = o\left(\frac{q}{N}\right)$. Moreover, we assumed that $m \le N$, and so this implies that $\mathrm{Cov}(X_{i,j}, X_{i,j'}) = o\left(\frac{q}{m}\right)$. This gives us

$$\begin{aligned}\mathrm{Var}[X] &= m^2 q(1-q) + 2\sum_{i,j,j' \neq j} \mathrm{Cov}(X_{i,j}, X_{i,j'}) \\ &= m^2 q(1-q) + 2m^2(m-1)\mathrm{Cov}(X_{i,j}, X_{i,j'}) \\ &\le m^2 q + 2m^2(m-1)\mathrm{Cov}(X_{i,j}, X_{i,j'}) \\ &= m^2 q + o(m^2 q)\ ,\end{aligned}$$

where the last equality follows from the fact that $\mathrm{Cov}(X_{i,j}, X_{i,j'}) = o\left(\frac{q}{m}\right)$. □

Theorem 5.12. *There is $m \in \Theta(\sqrt{N})$ such that if we sample m information sets in $\mathcal{C}_1$ and $\mathcal{C}_2$ using Algorithm 5 with $T > k\log(kN^2)$, we obtain a constant probability of success.*

Proof. If we pick $m \approx 2\sqrt{N}$, we have the probability of not drawing a collision being upper bounded by

$$\frac{\mathrm{Var}[X]}{qm^4} \approx \frac{2}{qm^2} \approx \frac{1}{2}.$$

□

Corollary 5.13. *Assume we use the distribution $\mathcal{D}_1, \mathcal{D}_2$ given by Algorithm 5 on input codes $\mathcal{C}_1, \mathcal{C}_2$ with parameter $T = \lceil k\log(kN^2)\rceil$, and the parameter $m := \left\lceil 2\sqrt{\binom{n}{k}}\right\rceil \le 2^{n/2+o(n)}$ as inputs of Algorithm 4, then the probability of solving PCE is bounded from below by a constant, and the running time is at most $\left\lceil 2\sqrt{\binom{n}{k}}\right\rceil \cdot \mathrm{quasipoly}(n) \cdot \mathrm{poly}(\log q) \le 2^{n/2+o(n+q)}$.*

5.3. **Meet-in-the-middle approach to solve LCE.** The above strategy readily extends to the resolution of LCE. Lemma 5.1 can be adapted by replacing the permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ by monomial matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2$.

Lemma 5.14. *Let $\boldsymbol{G}_1$ be a generator matrix for $\mathcal{C}_1$ and $\boldsymbol{G}_2$ be a generator matrix for $\mathcal{C}_2 = \tau(\mathcal{C}_1)$ for τ a linear isometry defined by $\tau = (\pi, \boldsymbol{v}) \in \mathcal{S}_n \rtimes \mathbb{F}_q^{*n}$. For all information sets I_1 of $\mathcal{C}_1$ and $I_2 := \pi(I_1)$ of $\mathcal{C}_2$, and for all permutations σ_1 mapping I_1 to $[1, \ldots, k]$ and σ_2 mapping I_2 to $[1, \ldots k]$, there exist monomial matrices $\boldsymbol{Q}_1 \in \mathbb{F}_q^{k\times k}, \boldsymbol{Q}_2 \in \mathbb{F}_q^{(n-k)\times(n-k)}$, and an invertible matrix $\boldsymbol{S}$ such that*

$$\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \boldsymbol{S}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\begin{pmatrix}\boldsymbol{Q}_1 & \\ & \boldsymbol{Q}_2\end{pmatrix},$$

where $\boldsymbol{P}_{I_1}$ is the matrix of σ_1 and $\boldsymbol{P}_{I_2}$ is the matrix of σ_2.

Proof. Assume $\boldsymbol{G}_2 = \boldsymbol{S}_0\boldsymbol{G}_1\boldsymbol{P}\boldsymbol{D}$ for some invertible matrix $\boldsymbol{S}_0$, permutation matrix $\boldsymbol{P}$, and diagonal matrix $\boldsymbol{D}$. Let I_1 be an information set for $\mathcal{C}_1$, and $\boldsymbol{P}_{I_1}$ be the matrix of a permutation that maps I_1 to $[1,\dots,k]$. Then, there is an invertible matrix $\boldsymbol{S}_1$ such that $\boldsymbol{S}_1\boldsymbol{G}_1\boldsymbol{P}_{I_1} = \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})$. This means that

$$\begin{aligned}\boldsymbol{G}_2 &= \boldsymbol{S}_0\boldsymbol{S}_1^{-1}\left(\boldsymbol{S}_1\boldsymbol{G}_1\boldsymbol{P}_{I_1}\right)\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}\boldsymbol{D}\\ &= \boldsymbol{S}_0\boldsymbol{S}_1^{-1}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}\boldsymbol{D}.\end{aligned}$$

Note that the permutation matrix $\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}$ corresponds to the permutation that maps $[1,\dots,k]$ to I_1, and then to $\pi(I_1) := I_2$ where $\tau = (\pi, \boldsymbol{v})$. Hence I_2 is an information set of $\mathcal{C}_2$. Let P_{I_2} be the matrix of a permutation mapping I_2 to $[1,\dots,k]$. In particular, the permutation corresponding to $\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}\boldsymbol{P}_{I_2}$ maps $[1,\dots,k]$ to itself and hence has the shape

$$\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}\boldsymbol{P}_{I_2} = \begin{pmatrix}\boldsymbol{P}_1 & \\ & \boldsymbol{P}_2\end{pmatrix}.$$

By multiplying our previous identity by $\boldsymbol{P}_{I_2}$ on both sides, we obtain

$$\begin{aligned}\boldsymbol{G}_2\boldsymbol{P}_{I_2} &= \boldsymbol{S}_0\boldsymbol{S}_1^{-1}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}\boldsymbol{D}\boldsymbol{P}_{I_2}\\ &= \boldsymbol{S}_0\boldsymbol{S}_1^{-1}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\boldsymbol{P}_{I_1}^{-1}\boldsymbol{P}\boldsymbol{P}_{I_2}\boldsymbol{D}'\\ &= \boldsymbol{S}_0\boldsymbol{S}_1^{-1}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\begin{pmatrix}\boldsymbol{P}_1 & \\ & \boldsymbol{P}_2\end{pmatrix}\boldsymbol{D}'\\ &= \boldsymbol{S}_0\boldsymbol{S}_1^{-1}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\begin{pmatrix}\boldsymbol{Q}_1 & \\ & \boldsymbol{Q}_2\end{pmatrix},\end{aligned}$$

where $\boldsymbol{D}'$ is the diagonal matrix whose entries are the permutation of the entries of $\boldsymbol{D}$ by the permutation represented by $\boldsymbol{P}_{I_2}$, and $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ are the monomial matrices $\boldsymbol{P}_1\boldsymbol{D}_1, \boldsymbol{P}_2\boldsymbol{D}_2$ with $\boldsymbol{D}_1$ the diagonal matrix whose entries are the first k entries of $\boldsymbol{D}'$ and $\boldsymbol{D}_2$ is the diagonal matrix made of the last $n-k$ entries of $\boldsymbol{D}'$. Let $\boldsymbol{S}_2$ be the invertible matrix such that $\boldsymbol{S}_2\boldsymbol{G}_2\boldsymbol{P}_{I_2} = \mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2})$, and $\boldsymbol{S} := \boldsymbol{S}_2\boldsymbol{S}_0\boldsymbol{S}_1^{-1}$. We have

$$\boldsymbol{S}_2\boldsymbol{G}_2\boldsymbol{P}_{I_2} = \mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \boldsymbol{S}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\begin{pmatrix}\boldsymbol{Q}_1 & \\ & \boldsymbol{Q}_2\end{pmatrix}.$$

□

We adapt Definition 5.2 to the case of matching sets for LCE:

Definition 5.15 (Monomially matching information sets)**.** Let $\mathcal{C}_1, \mathcal{C}_2$ be two $[n,k]_q$ codes with generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$. We say that information sets I_1, I_2 for $\mathcal{C}_1$ (resp. $\mathcal{C}_2$) are *monomially matching information sets* if there exists a linear isometry $\tau = (\pi, \boldsymbol{v})$ such that $\mathcal{C}_2 = \tau(\mathcal{C}_1)$ and $I_2 = \pi(I_2)$.

When I_1 and I_2 are matching information sets, there are monomial matrices $\boldsymbol{Q}_1 \in \mathbb{F}_q^{k\times k}, \boldsymbol{Q}_2 \in \mathbb{F}_q^{(n-k)\times(n-k)}$, and an invertible matrix $\boldsymbol{S}$ such that

$$\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \boldsymbol{S}\,\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1})\begin{pmatrix}\boldsymbol{Q}_1 & \\ & \boldsymbol{Q}_2\end{pmatrix},$$

Let $\boldsymbol{A}_1, \boldsymbol{A}_2$ such that $\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix}I_k & \boldsymbol{A}_1\end{pmatrix}$ and $\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \begin{pmatrix}I_k & \boldsymbol{A}_2\end{pmatrix}$. We have the identity

$$\begin{pmatrix}I_k & \boldsymbol{A}_1\end{pmatrix} = \boldsymbol{S}\begin{pmatrix}I_k & \boldsymbol{A}_2\end{pmatrix}\begin{pmatrix}\boldsymbol{Q}_1 & \\ & \boldsymbol{Q}_2\end{pmatrix},$$

and as in the original method from Babai to solve PCE, this yields

- $\boldsymbol{S} = \boldsymbol{Q}_1^{-1}$,
- $\boldsymbol{Q}_1^{-1}\boldsymbol{A}_1\boldsymbol{Q}_2 = \boldsymbol{A}_2$.

As in our variant of Babai's algorithm (and as in the independent work [Now25]), once a collision is found, we need to find monomial matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ such that $\boldsymbol{A}_2 = \boldsymbol{Q}_1\boldsymbol{A}_1\boldsymbol{Q}_2$. The main difference of our approach with that of [Now25] is that we reduce this task to GI via matrix expansion. This procedure, summarized in Algorithm 4, consists in finding permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that $\widehat{\boldsymbol{A}_1} = \boldsymbol{P}_1\widehat{\boldsymbol{A}_2}\boldsymbol{P}_2$ and then recovering $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ from $\boldsymbol{P}_1, \boldsymbol{P}_2$. Hence, we immediately have the following reduction of LCE to the resolution of the claw finding problem.

Proposition 5.16 (Reduction of LCE to the claw finding problem)**.** *Let $\mathcal{C}_1, C_2$ be two (n,k) linear codes over $\mathbb{F}_q$ with generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$ such that $\mathcal{C}_2 = \tau(\mathcal{C}_1)$ for $\tau = (\pi, \boldsymbol{v}) \in \mathcal{S}_n \rtimes \mathbb{F}_q^{*n}$. Let F be an assignment for the class of $\mathbb{F}_q$-colored $(qk, q(n-k))$-bipartite graphs. Let $j \in \{1,2\}$ and let A_j be the set of information sets for $\mathcal{C}_j$, and*

$$B := \{F(\widehat{\boldsymbol{A}_1}) \mid \exists \boldsymbol{P}_{I_1},\ \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}\} = \{F(\widehat{\boldsymbol{A}_2}) \mid \exists \boldsymbol{P}_{I_2},\ \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_2}) = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}\},$$

where I_j is an information set for $\mathcal{C}_j$, and $\boldsymbol{P}_{I_j}$ denotes the matrix of a permutation that maps I_j to $[1,\dots,k]$ for $j = 1,2$. We define $g_j : A_j \to B$ by the following procedure on input $I \in A_j$:

(1) Let $I = \{i_1, \dots, i_k\}$ where $i_1 < i_2 < \dots < i_k$.
(2) Let $\sigma \in \mathcal{S}_n = (1, i_1)(2, i_2)\dots(k, i_k)$.
(3) Let $\boldsymbol{P}_I$ be the permutation matrix corresponding to σ.
(4) $g_j(I) = F(\widehat{\boldsymbol{A}})$ where $\mathrm{SF}(\boldsymbol{G}_j\boldsymbol{P}_I) = \begin{pmatrix} I_k & \boldsymbol{A} \end{pmatrix}$

Then solving LCE efficiently reduces to finding a claw between g_1 and g_2.

Require: Generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$ of two linearly equivalent $[n,k,d]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$, $m > 0$, and distributions $\mathcal{D}_1, \mathcal{D}_2$ on the information sets of $\mathcal{C}_1$ (resp. $\mathcal{C}_2$).
Ensure: A monomial matrix $\boldsymbol{Q}$ satisfying $\mathrm{SF}(\boldsymbol{G}_2) = \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{Q})$.

for m iterations **do**
  Sample an information set I_1 of $\mathcal{C}_1$ from $\mathcal{D}_1$.
  $L \leftarrow L \cup (I_1, g_1(I_1))$.
end for
for m iterations **do**
  Sample an information set I_2 of $\mathcal{C}_2$ from $\mathcal{D}_2$.
  if there is $(I_1, g_1(I_1)) \in L$ with $g_1(I_1) = g_2(I_2)$ **then**
    Compute $\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$ and $\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$.
    Compute permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that $\boldsymbol{P}_1\widehat{\boldsymbol{A}_1}\boldsymbol{P}_2 = \widehat{\boldsymbol{A}_2}$ using Babai's algorithm [Bab16].
    Deduce monomial matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ such that $\boldsymbol{A}_2 = \boldsymbol{Q}_1\boldsymbol{A}_1\boldsymbol{Q}_1$ using Algorithm 2.
    break
  end if
end for
return $\boldsymbol{Q} = \boldsymbol{P}_{I_1} \begin{pmatrix} \boldsymbol{Q}_1^{-1} & \\ & \boldsymbol{Q}_2 \end{pmatrix} \boldsymbol{P}_{I_2}^{-1}$

Algorithm 6: Meet-in-the-middle variant of Babai's LCE algorithm

Theorem 5.17. *Assume we use the distribution on information sets given by Algorithm 5 with parameter $T = \lceil k \log(kN^2) \rceil$, and the parameter $m := \left\lceil 2\sqrt{\binom{n}{k}} \right\rceil \leq 2^{n/2+o(n)}$ as inputs of Algorithm 6, then the probability of solving LCE is bounded from below by a constant, and the run time is bounded by $\left\lceil 2\sqrt{\binom{n}{k}} \right\rceil \cdot \mathrm{quasipoly}(n+q) \leq 2^{n/2+o(n+q)}$.*

6. A quantum variant of the meet in the middle approach

In this section, we present a quantum variant of our meet-in-the-middle strategy to solve PCE and LCE. We adapt the quantum collision-finding algorithm of Brassard, Høyer and Tapp [BHT97] (the BHT algorithm) to the claw-finding problem in the setting where there are many claws (as in Section 5, a claw corresponds to a pair of matching information sets). Specifically, if two $[n,k]_q$ codes are permutationally or linearly equivalent then there are at least N pairs of matching information sets out of $N^2 \leq \binom{n}{k}^2$ pairs total. We note that the BHT algorithm is based on the Grover search algorithm [Gro96].

We now assume that we are given two $[n,k]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$ such that there is a linear isometry τ satisfying $\mathcal{C}_2 = \tau(\mathcal{C}_1)$. Let $\boldsymbol{G}_1, \boldsymbol{G}_2$ be generator matrices of $\mathcal{C}_1, \mathcal{C}_2$, and g_1, g_2 be functions defined as in Proposition 5.16. From a high level standpoint, we proceed as follows:

(1) Draw a list L of $\binom{n}{k}^{1/3}$ distincts information sets I_1 of $\mathcal{C}_1$.
(2) Search for an information set I_2 of $\mathcal{C}_2$ such that there is $I_1 \in L$ with $g_2(I_2) = g_1(I_1)$.
(3) Deduce a monomial matrix $\boldsymbol{Q}$ with $\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{Q}) = \mathrm{SF}(\boldsymbol{G}_2)$ similarly to Steps 8-10 of Algorithm 6.

The main technical difficulty with this approach is that we do not know the number N of distinct information sets of $\mathcal{C}_1$ (and $\mathcal{C}_2$). If there are more than $\binom{n}{k}^{1/3}$ distinct information sets, then the above method works in quantum time bounded by $\binom{n}{k}^{1/3} \cdot \mathrm{quasipoly}(n+q)$. If there are fewer information sets ($N \leq \binom{n}{k}^3$ of them), we solve PCE or LCE classically in $O(N)$ time. To do this, we construct a list L of all N pairs $(I_1, g_1(I_1))$ where I_1 is an information set of $\mathcal{C}_1$, and then for an arbitrary information set I_2 of $\mathcal{C}_2$, classically search for $(I_1, g_1(I_1)) \in L$ such that $g_1(I_1) = g_2(I_2)$.

We also need a way to estimate the number of distinct information sets N in the codes. To decide whether the quantum search or the classical enumeration method should be performed, we draw $\binom{n}{k}^{1/3} \log\left(\binom{n}{k}\right)$ information sets through Algorithm 5. From the analysis of the coupon collector problem, we know that if there are $N < \binom{n}{k}^{1/3}$ distinct information sets, then the list contains them all with high probability. On the other hand, if there are $N \geq \binom{n}{k}^{1/3}$ distinct information sets, then the list contains at least $\binom{n}{k}^{1/3}$ distinct information sets with high probability. Hence, we decide whether to opt for the classical method of the quantum search based on whether we obtain at least $\binom{n}{k}^{1/3}$ distinct information sets after drawing $\binom{n}{k}^{1/3} \log\left(\binom{n}{k}\right)$ information sets from Algorithm 5. We describe our PCE procedure in Algorithm 7 and the LCE variant in Algorithm 8.

Require: Generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$ of two permutationally equivalent $[n,k,d]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$.
Ensure: A permutation matrix $\boldsymbol{P}$ satisfying $\mathrm{SF}(\boldsymbol{G}_2) = \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P})$.

for $\binom{n}{k}^{1/3} \log\left(\binom{n}{k}\right)$ iterations **do**
  Sample an information set I_1 of $\mathcal{C}_1$ with Algorithm 5.
  If $(I_1, f_1(I_1)) \notin L$; The if statement is unnecessary. $L \leftarrow L \cup (I_1, f_1(I_1))$.
end for
if $|L| < \binom{n}{k}^{1/3}$ **then**
  Choose any information set I_2 of $\mathcal{C}_2$, and find $(I_1, f_1(I_1)) \in L$ such that $f_2(I_2) = f_1(I_1)$.
  If no such information set exists, **return** $\perp$.
else
  Keep exactly $\binom{n}{k}^{1/3}$ elements in L.
  let $f : \{k-\text{tuples of indices of columns of } \boldsymbol{G}_2\} \to \{0,1\}$ such that $f(I) = 1$ if I is an information set of $\mathcal{C}_2$ and $\exists (I_1, f_1(I_1)) \in L$ with $f_2(I_2) = f_1(I_1)$.
  Using Grover's algorithm [Gro96], find I_2 such that $f(I_2) = 1$.
  Find $(I_1, f_1(I_1)) \in L$ such that $f_2(I_2) = f_1(I_1)$.
  If no I_2 was found, **return** $\perp$.
end if
Compute $\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$ and $\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$.
Compute permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that $\boldsymbol{P}_1\boldsymbol{A}_1\boldsymbol{P}_2 = \boldsymbol{A}_2$ using Babai's algorithm [Bab16].
return $\boldsymbol{P} = \boldsymbol{P}_{I_1} \begin{pmatrix} \boldsymbol{P}_1^{-1} & \\ & \boldsymbol{P}_2 \end{pmatrix} \boldsymbol{P}_{I_2}^{-1}$

Algorithm 7: Quantum Meet-in-the-middle variant of Babai's PCE algorithm

Note that Algorithm 7 and Algorithm 8 require that the list L be stored in quantumly addressable classical memory (QRACM). To analyze their time and memory costs, we rely on the following lemma.

Lemma 6.1. *Let $\mathcal{C}$ be a $[n,k]_q$ code. Let N be the number of information sets of $\mathcal{C}_1$. We draw $m := \binom{n}{k}^{1/3} \log\left(\binom{n}{k}\right)$ information sets with Algorithm 5 on input $\mathcal{C}, \varepsilon = \frac{1}{\binom{n}{k}}$.*

(1) If $N < \binom{n}{k}^{1/3}$, then $P(\text{we draw all information sets of } \mathcal{C}) = 1 - o(1)$.
(2) If $N \geq \binom{n}{k}^{1/3}$, then $P\left(\text{we draw } \binom{n}{k}^{1/3} \text{ distinct information sets of } \mathcal{C}\right) = 1 - o(1)$.

Require: Generator matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$ of two linearly equivalent $[n,k,d]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$.
Ensure: A monomial matrix $\boldsymbol{Q}$ satisfying $\mathrm{SF}(\boldsymbol{G}_2) = \mathrm{SF}(\boldsymbol{G}_1\boldsymbol{Q})$.

for $\binom{n}{k}^{1/3} \log\left(\binom{n}{k}\right)$ iterations **do**
  Sample an information set I_1 of $\mathcal{C}_1$ with Algorithm 5.
  If $(I_1, g_1(I_1)) \notin L$; The if statement is unnecessary. $L \leftarrow L \cup (I_1, g_1(I_1))$.
end for
if $|L| < \binom{n}{k}^{1/3}$ **then**
  Choose any information set I_2 of $\mathcal{C}_2$, and find $(I_1, g_1(I_1)) \in L$ such that $g_2(I_2) = g_1(I_1)$.
  If no such information set exists, **return** $\perp$.
else
  Keep exactly $\binom{n}{k}^{1/3}$ elements in L.
  let $f : \{k-\text{tuples of indices of columns of } \boldsymbol{G}_2\} \to \{0,1\}$ such that $f(I) = 1$ if I is an information set of $\mathcal{C}_2$ and $\exists (I_1, g_1(I_1)) \in L$ with $g_2(I_2) = g_1(I_1)$.
  Using Grover's algorithm [Gro96], find I_2 such that $f(I_2) = 1$.
  Find $(I_1, g_1(I_1)) \in L$ such that $g_2(I_2) = g_1(I_1)$.
  If no I_2 was found, **return** $\perp$.
end if
Compute $\mathrm{SF}(\boldsymbol{G}_1\boldsymbol{P}_{I_1}) = \begin{pmatrix} I_k & \boldsymbol{A}_1 \end{pmatrix}$ and $\mathrm{SF}(\boldsymbol{G}_2\boldsymbol{P}_{I_2}) = \begin{pmatrix} I_k & \boldsymbol{A}_2 \end{pmatrix}$.
Compute permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that $\boldsymbol{P}_1 \widehat{\boldsymbol{A}_1} \boldsymbol{P}_2 = \widehat{\boldsymbol{A}_2}$ using Babai's algorithm [Bab16].
Deduce monomial matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ such that $\boldsymbol{A}_2 = \boldsymbol{Q}_1 \boldsymbol{A}_1 \boldsymbol{Q}_1$ using Algorithm 2.
return $\boldsymbol{Q} = \boldsymbol{P}_{I_1} \begin{pmatrix} \boldsymbol{Q}_1^{-1} & \\ & \boldsymbol{Q}_2 \end{pmatrix} \boldsymbol{P}_{I_2}^{-1}$

Algorithm 8: Quantum Meet-in-the-middle variant of Babai's LCE algorithm

Proof. Let p be the probability of drawing a particular information set I. The probability of not drawing I after drawing m information sets is upper bounded by $(1-p)^m \leq e^{-mp}$.

Case (1): Given the analysis of Algorithm 5, the probability p of drawing a given information set satisfies

$$p \geq \frac{1}{N} - \varepsilon \geq \frac{1}{\binom{n}{k}^{1/3}} - \frac{1}{\binom{n}{k}} := p_0.$$

By the union bound on the probability of not obtaining a given I, the probability of obtaining all N information sets is at least $1 - Ne^{-p_0 m} \geq 1 - \binom{n}{k}^{1/3} e^{-mp_0}$. Now, given that $m = \binom{n}{k}^{1/3} \log\left(\binom{n}{k}\right)$, we get

$$\begin{aligned}
\binom{n}{k}^{1/3} e^{-mp_0} &= \binom{n}{k}^{1/3} e^{-\binom{n}{k}^{1/3} \log\left(\binom{n}{k}\right)\left(\frac{1}{\binom{n}{k}^{1/3}} - \frac{1}{\binom{n}{k}}\right)} \\
&= \binom{n}{k}^{1/3} e^{-\log\left(\binom{n}{k}\right)(1+o(1))} \\
&= \binom{n}{k}^{-2/3+o(1)} \xrightarrow[n\to\infty]{} 0.
\end{aligned}$$

Case (2): Let N be the number of information sets of $\mathcal{C}$. We assume that $N \geq N' := \binom{n}{k}^{1/3}$. Assume that we draw m information sets from a distribution where the probability of drawing any individual element is at least $p > 0$. To estimate the probability of drawing N' distinct information sets, we consider the more restricted case of drawing at least an element within each of the N' sets of size at least $\left\lfloor \frac{N}{N'} \right\rfloor$ of a partition of the N information sets. The probability of drawing an information set within any set of the partition is at least $p' := p\left\lfloor \frac{N}{N'} \right\rfloor$. After drawing m information sets, the probability of not having an element of a given element of the partition is upper bounded by $e^{-mp'}$. By the union bound, the probability of missing of not drawing an information set from each set of the partition is less than $N'e^{-mp'}$. Now, we assume that $m = \binom{n}{k}^{1/3} \log\left(\binom{n}{k}\right) = N' \log(N'^3)$, and that $\varepsilon = \frac{1}{\binom{n}{k}} = \frac{1}{N'^3}$. This means that the probability of not drawing

at least an element in each set of the partition is less than

$$\begin{aligned} N'e^{-mp'} &= N'e^{-N'\log(N'^3)(\frac{1}{N}-\frac{1}{N'^3})\lfloor\frac{N}{N'}\rfloor} \\ &= N'e^{-\log(N'^3)\left(\frac{N'}{N}\lfloor\frac{N}{N'}\rfloor+o(1)\right)} \\ &\leq \frac{N'}{N'^{3(\frac{1}{2}+o(1))}} \xrightarrow[n\to\infty]{} 0. \end{aligned}$$

Since we are less likely of not obtaining N' distinct information sets than to not obtain one element in each set of the partition, our claim follows. □

Theorem 6.2. *Algorithm 7 runs in classical and quantum time bounded by* $\binom{n}{k}^{1/3}\cdot\text{quasipoly}(n)\cdot\text{poly}(\log q) \leq 2^{n/3+o(n+q)}$ *and* $\binom{n}{k}^{1/3}\cdot\text{poly}(n) \leq 2^{n/3+o(n)}$ *quantumly addressable classical memory. Algorithm 8 runs in classical and quantum time bounded by* $\binom{n}{k}^{1/3}\cdot\text{quasipoly}(n+q) \leq 2^{n/3+o(n+q)}$ *and* $\binom{n}{k}^{1/3}\cdot\text{poly}(n+q) \leq 2^{n/3+o(n+q)}$ *quantumly addressable classical memory. Both algorithms return the correct answer with a probability lower bounded by a constant.*

Proof. The creation of L requires at most $O\left(\log\left(\binom{n}{k}\right)\binom{n}{k}^{1/3}\right)$ calls to f_1, f_2, g_1, g_2. Each call to f_1, f_2 costs $\text{quasipoly}(n)\cdot\text{poly}(\log q)$, while a call to g_1, g_2 costs $\text{quasipoly}(n+q)$. Hence, in Algorithm 7, the creation of L costs $\binom{n}{k}^{1/3}\cdot\text{quasipoly}(n)\cdot\text{poly}(\log q)$ time, while in Algorithm 8, it costs $\binom{n}{k}^{1/3}\cdot\text{quasipoly}(n+q)$ time. In Algorithm 7, each element of L has size $\text{poly}(n)$ while in Algorithm 8, elements of L have size $\text{poly}(n+q)$.

If $|L| < \binom{n}{k}^{1/3}$, the remaining classical steps are performed in $\tilde{O}(|L|)$. If $|L| \geq \binom{n}{k}^{1/3}$, the number of calls to an oracle evaluating f (respectively g) through Grover's search algorithm is in

$$O\left(\sqrt{\frac{|\{k-\text{tuples of indices of columns of } \boldsymbol{G}_2\}|}{|f^{-1}(1)|}}\right) \leq O\left(\sqrt{\frac{\binom{n}{k}}{\binom{n}{k}^{1/3}}}\right) = O\left(\binom{n}{k}^{1/3}\right).$$

Since the bottleneck of a call to f is the cost of f_2, while the cost of g is given by that of g_2, the claim on the quantum time of Algorithm 7 and Algorithm 8 follows. □

References

[ALOV19] Nima Anari, Kuikui Liu, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials II: high-dimensional walks and an FPRAS for counting bases of a matroid. In *STOC*, 2019. 5, 16

[Bab16] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697. ACM, 2016. 1, 2, 3, 4, 5, 7, 13, 15, 16, 20, 21, 22

[Bab19] László Babai. Canonical form for graphs in quasipolynomial time: preliminary report. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1237–1246. ACM, 2019. 4, 5, 7, 14

[BBB+23] Marco Baldi, Alessandro Barenghi, Luke Beckwith, Jean-François Biasse, Andre Esser, Kris Gaj, Kamyar Mohajerani, Gerardo Pelosi, Edoardo Persichetti, Markku-Juhani O. Saarinen, Paolo Santini, and Robert Wallace. Less, 2023. Additional Digital Signature Schemes - Round 1 Submissions. 2, 3

[BBD+23] Joppe W. Bos, Olivier Bronchain, Léo Ducas, Serge Fehr, Yu-Hsuan Huang, Thomas Pornin, Eamonn W. Postlethwaite, Thomas Prest, Ludo N. Pulles, and Wessel van Woerden. Hawk, 2023. Additional Digital Signature Schemes - Round 1 Submissions. 3

[BBPS23] 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. 2

[BCGQ11] László Babai, Paolo Codenotti, Joshua A. Grochow, and Youming Qiao. Code equivalence and group isomorphism. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1395–1408. SIAM, 2011. 1, 2, 3, 4, 7

[Beu20] Ward Beullens. Not enough LESS: an improved algorithm for solving code equivalence problems over F_q. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*, volume 12804 of *Lecture Notes in Computer Science*, pages 387–403. Springer, 2020. 2

[BGPS23] Huck Bennett, Atul Ganju, Pura Peetathawatchai, and Noah Stephens-Davidowitz. Just how hard are rotations of $\mathbb{Z}^n$? algorithms and cryptography with the simplest lattice. In *EUROCRYPT*, 2023. 3

[BHT97] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. *SIGACT News*, 28(2):14–19, 1997. 2, 5, 20

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20: How 1 + 1 = 0 improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 520–536. Springer, 2012. 2

[BM18] Leif Both and Alexander May. Decoding linear codes with high error rate and its impact for LPN security. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings*, volume 10786 of *Lecture Notes in Computer Science*, pages 25–46. Springer, 2018. 2

[BM23] Jean-François Biasse and Giacomo Micheli. A search-to-decision reduction for the permutation code equivalence problem. In *IEEE International Symposium on Information Theory, ISIT 2023, Taipei, Taiwan, June 25-30, 2023*, pages 602–607. IEEE, 2023. 6

[BMPS20] Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. LESS is more: Code-based signatures without syndromes. In Abderrahmane Nitaj and Amr M. Youssef, editors, *Progress in Cryptology - AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20-22, 2020, Proceedings*, volume 12174 of *Lecture Notes in Computer Science*, pages 45–65. Springer, 2020. 2

[BOS19] Magali Bardet, Ayoub Otmani, and Mohamed Saeed-Taha. Permutation code equivalence is not harder than graph isomorphism when hulls are trivial. In *IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7-12, 2019*, pages 2464–2468. IEEE, 2019. 2

[BW24] Huck Bennett and Kaung Myat Htay Win. Relating code equivalence to other isomorphism problems. *Cryptology ePrint Archive*, 2024. To appear in Designs, Codes, and Cryptography. 6, 9

[CNP+23a] Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Lars Ran, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Meds, 2023. Additional Digital Signature Schemes - Round 1 Submissions. 3

[CNP+23b] Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Take your MEDS: digital signatures from matrix code equivalence. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *Progress in Cryptology - AFRICACRYPT 2023 - 14th International Conference on Cryptology in Africa, Sousse, Tunisia, July 19-21, 2023, Proceedings*, volume 14064 of *Lecture Notes in Computer Science*, pages 28–52. Springer, 2023. 3

[CPS23] Tung Chou, Edoardo Persichetti, and Paolo Santini. On linear equivalence, canonical forms, and digital signatures. *IACR Cryptol. ePrint Arch.*, page 1533, 2023. 1, 4, 8

[DEEK24] Léo Ducas, Andre Esser, Simona Etinski, and Elena Kirshanova. Asymptotics and improvements of sieving for codes. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, volume 14656 of *Lecture Notes in Computer Science*, pages 151–180. Springer, 2024. 2

[DPPvW22] Léo Ducas, Eamonn W. Postlethwaite, Ludo N. Pulles, and Wessel P. J. van Woerden. Hawk: Module LIP makes lattice signatures fast, compact and simple. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part IV*, volume 13794 of *Lecture Notes in Computer Science*, pages 65–94. Springer, 2022. 3

[DvW22] Léo Ducas and Wessel P. J. van Woerden. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 643–673. Springer, 2022. 3

[ES24] Andre Esser and Paolo Santini. Not just regular decoding: Asymptotics and improvements of regular syndrome decoding attacks. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VI*, volume 14925 of *Lecture Notes in Computer Science*, pages 183–217. Springer, 2024. 2

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986. 2

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996. 20, 21, 22

[JQSY19] Zhengfeng Ji, Youming Qiao, Fang Song, and Aaram Yun. General linear group action on tensors: A candidate for post-quantum cryptography. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part I*, volume 11891 of *Lecture Notes in Computer Science*, pages 251–281. Springer, 2019. 3

[Leo82] Jeffrey Leon. Computing automorphism groups of error-correcting codes. *IEEE Trans. Inf. Theory*, 28(3):496–510, 1982. 2

[McE78] Robert McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44:114–116, January 1978. 2

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{\mathcal{O}}(20.054n)$. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2011. 2

[Nie86] Harald Niederreiter. Knapsack-Type Cryptosystems and Algebraic Coding Theory. *Problems of Control and Information Theory*, 15:159–166, 1986. 2

[Now25] Julian Nowakowski. An improved algorithm for code equivalence. In *PQCrypto*, 2025. 1, 2, 4, 5, 8, 16, 19

[Pet10] Christiane Peters. Information-set decoding for linear codes over f_q. In Nicolas Sendrier, editor, *Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings*, volume 6061 of *Lecture Notes in Computer Science*, pages 81–94. Springer, 2010. 2

[PR97] Erez Petrank and Ron M. Roth. Is code equivalence easy to decide? *IEEE Trans. Inf. Theory*, 43(5):1602–1604, 1997. 2

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Inf. Theory*, 8(5):5–9, 1962. 2

[Sae18] Mohamed Ahmed Saeed-Taha. *Algebraic approach to Code Equivalence*. Phd thesis, University of Rouen Normandie, January 2018. Available at https://theses.hal.science/tel-01678829/file/saeedtahamohamed3.pdf. 2

[Sch22] Tselil Schramm. Lecture 5: Approximate sampling of spanning trees via matroid basis exchange, 2022. Lecture Notes. Available at https://tselilschramm.org/random-processes/05-approximate-matroid-sampling.pdf. 2, 16

[Sen00] Nicolas Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Trans. Inf. Theory*, 46(4):1193–1203, 2000. 2

[SS13] Nicolas Sendrier and Dimitris E. Simos. The hardness of code equivalence over and its application to code-based cryptography. In Philippe Gaborit, editor, *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings*, volume 7932 of *Lecture Notes in Computer Science*, pages 203–216. Springer, 2013. 1, 2, 3, 4, 6, 9

[Ste88] Jacques Stern. A method for finding codewords of small weight. In Gérard D. Cohen and Jacques Wolfmann, editors, *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1988. 2

[ZKT85] V. N. Zemlyachenko, N. M. Korneenko, and R. I. Tyshkevich. Graph isomorphism problem. *Journal of Soviet Mathematics*, 29(4):1426–1481, May 1985. 4, 7, 13, 14

[1]Department of Computer Science, University of Colorado Boulder

[2]University of California Berkeley

[3]Center for Cryptographic Research, University of South Florida

[5]Virginia Polytechnic Institute and State University

[4]University of Maryland at College Park