Relating Code Equivalence to Other Isomorphism Problems

Huck Bennett[*] Kaung Myat Htay Win[†]

December 15, 2024

Abstract

We study the complexity of the *Code Equivalence Problem* on linear error-correcting codes by relating its variants to isomorphism problems on other discrete structures—graphs, lattices, and matroids. Our main results are a fine-grained reduction from the Graph Isomorphism Problem to the Linear Code Equivalence Problem over any field $\mathbb{F}$, and a reduction from the Linear Code Equivalence Problem over any field $\mathbb{F}_p$ of prime, polynomially bounded order p to the Lattice Isomorphism Problem. Both of these reductions are simple and natural. We also give reductions between variants of the Code Equivalence Problem, and study the relationship between isomorphism problems on codes and linear matroids.

[*]University of Colorado Boulder. huck.bennett@colorado.edu. This work was supported in part by NSF grant No. 2432132. Part of this work was completed while the author was at Oregon State University.

[†]Oregon State University. htaywink@oregonstate.edu.

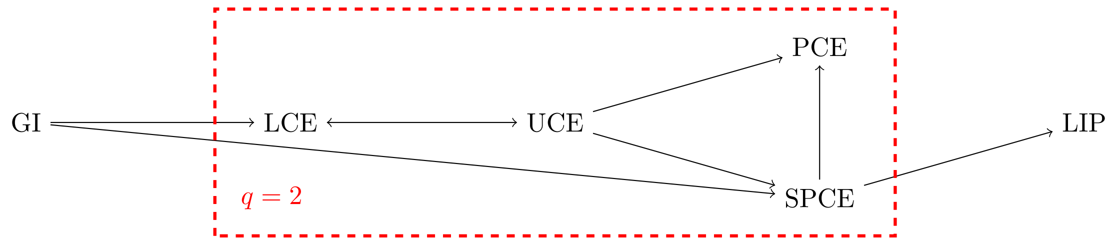

Figure 1: Reductions between the Graph Isomorphism Problem (GI), variants of Code Equivalence (CE), and the Lattice Isomorphism Problem (LIP). When working over $\mathbb{F}_q$ for $q = 2$, all four variants of CE are the same, and when $q = 3$, LCE and SPCE are the same. Several of these reductions were already known, or known with worse parameters. Our first main result, given in Theorem 1.1 and formally in Corollary 3.7, is a fine-grained reduction from GI to Linear Code Equivalence (LCE) and Signed Permutation Code Equivalence (SPCE) over any field $\mathbb{F}$. Our second main result, given in Theorem 1.2 and formally in Theorem 5.1, is a reduction from SPCE (and LCE) over fields $\mathbb{F}_p$ of prime order to the Lattice Isomorphism Problem (LIP).

1 Introduction

A *linear error-correcting code* (or simply *code*) $\mathcal{C} \subseteq \mathbb{F}^n$ is a linear subspace of $\mathbb{F}^n$, where $\mathbb{F}$ is a (typically finite) field. Variants of the *Code Equivalence Problem* (CE) ask, given two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ as input, whether $\mathcal{C}_1$ and $\mathcal{C}_2$ are "the same" in some sense. Specifically, Permutation Code Equivalence (PCE) asks if the codes are the same up to permutation of coordinates of codewords, and Linear Code Equivalence (LCE) asks if they are the same up to permutation and scaling of coordinates of codewords.

Perhaps the best-studied isomorphism problem is the *Graph Isomorphism Problem* (GI), which asks whether two input graphs are the same up to permutation of their vertices. In addition to isomorphism problems on fundamental discrete structures like graphs and codes being of inherent interest, the code equivalence problem appears prominently in cryptography. Indeed, several code-based cryptosystems are directly inspired by the code equivalence problem, including the well-known McEliece public-key cryptosystem [McE78] from the late 1970s; a recent instantiation ("Classic McEliece") of the McEliece cryptosystem, submitted to the NIST post-quantum standardization process [ABC$^+$22]; and the recent LESS identification scheme [BMPS20, BBPS21]. Furthermore, cryptosystems like LESS are not only inspired by CE, but rely on its hardness for their security. Accordingly, understanding the *complexity* of CE is important for cryptography.

Another active line of more recent work has studied the *Lattice Isomorphism Problem* (LIP). A lattice is the set of all integer linear combinations of some n linearly independent vectors in $\mathbb{R}^n$. Two lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ are said to be isomorphic if there is an orthogonal linear transformation mapping one to the other (informally, if one is a rotation of the other). In particular, in the last few years, a number of LIP-inspired cryptosystems have been proposed including a KEM [DvW22], a public-key cryptosystem [BGPS23], a digital signature scheme [DPPvW22], and [BCK23]. Although these "LIP cryptosystems" are analogous to and in part inspired by "CE cryptosystems," very little work has directly compared these cryptosystems, or directly compared CE and LIP.

1.1 Our Results

In this work, we study the complexity of the Code Equivalence Problem by relating it to isomorphism problems on other discrete structures. We next summarize our results, while deferring formal definitions to Section 2.

1.1.1 Reducing Graph Isomorphism to Code Equivalence

Our first main result is a fine-grained reduction from the Graph Isomorphism Problem (GI) to Linear Code Equivalence over any field $\mathbb{F}$. See Section 3.2 for formal statements.

Theorem 1.1 (GI to LCE; informal)**.** *There is a polynomial-time reduction from* GI *on graphs with n vertices and m edges to* LCE *over any field $\mathbb{F}$ on codes of dimension $n + O(1)$ and block length $m + O(n)$.*

Theorem 1.1 is not the first reduction showing GI-hardness of code equivalence. Indeed, such a reduction was first given by Petrank and Roth [PR97], and additional reductions were given by Kaski and Östergård [KO06] and Grochow [Gro12].

However, our reduction has several advantages. First, our reduction is fine-grained: it outputs codes of dimension roughly n whereas the reductions in [PR97, Gro12] output codes of dimension m. (The block length of their output codes, which is $3m + n$, is also nearly always higher than for our codes.) Second, our reduction holds over all fields $\mathbb{F}$, and in particular all finite fields $\mathbb{F}_q$. The reductions of [PR97, KO06] are only for binary codes. Third, our reduction is simple and natural: if the input graphs are sufficiently "well-connected" we just map them to their (transposed) incidence matrices, which are generator matrices for the output codes. The reduction in [KO06] is similar to ours and also simple, but ours is more efficient and general. Fourth, our reduction is to LCE and not just PCE as in [Gro12]. There is a known dimension-preserving reduction from LCE to PCE ([SS13b]; see also Corollary 4.5), and so showing hardness of LCE is at least as strong.

We note that despite being fine-grained our reduction only shows relatively mild hardness of LCE simply because Babai's quasipolynomial-time algorithm for GI [Bab16] shows that GI is not that hard. I.e., even if Babai's algorithm were optimal, our reduction would not imply (sub)exponential hardness of LCE. Nevertheless, understanding the relationship between GI and CE is a fundamental question, and our reduction shows a strong connection between the problems.

1.1.2 Reducing Code Equivalence to Lattice Isomorphism

Our second main result is an efficient reduction from LCE over fields of prime, polynomially bounded order p to the Lattice Isomorphism Problem (LIP) on point lattices. See Section 5 for formal statements.

Theorem 1.2 (LCE to LIP; informal)**.** *Let $n \in \mathbb{Z}^+$ and $p = p(n) \leq \mathrm{poly}(n)$ be a prime. There is a polynomial-time reduction from* LCE *on codes over $\mathbb{F}_p$ of block length n to* LIP*.*

The $p = 2$ case of Theorem 1.2 was already given by Regev in unpublished work [Reg14], modulo some technical details. We follow his reduction approach, which is quite simple and natural. Indeed, his reduction maps linear codes $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_2^n$ with sufficiently large minimum distance to the lattices $\mathcal{L}_1 := \mathcal{C}_1 + 2\mathbb{Z}^n$, $\mathcal{L}_2 := \mathcal{C}_2 + 2\mathbb{Z}^n$ (the so-called Construction-A lattices obtained from $\mathcal{C}_1$ and $\mathcal{C}_2$). However, there are issues that arise when trying to generalize this approach to the $p > 2$ case. The main issue is that the isometry groups of $\mathcal{C}_1$ and $\mathcal{C}_2$ and $\mathcal{L}_1$ and $\mathcal{L}_2$ no longer "align nicely." When $p = 2$, PCE and LCE are the same problem, but this is no longer true for $p > 2$, and so there is also the question of which variant of CE to reduce from. We handle these issues by reducing a different variant of CE called the Signed Permutation Code Equivalence Problem (SPCE) to LIP, and then reducing the more standard LCE to SPCE.

We also note that combining Theorems 1.1 and 1.2 shows GI-hardness of LIP, giving an alternative to a previous such reduction appearing in [DSV09]. (In fact, Regev [Reg14] observed that combining the reduction in [PR97] and the $p = 2$ case of Theorem 1.2 already shows this.)

1.1.3 Other Results

In part to give a modular reduction from LCE to SPCE, we introduce the Universal Code Equivalence Problem (UCE). UCE is a promise version of CE in which the input codes $\mathcal{C}_1, \mathcal{C}_2$ are either promised to be permutationally equivalent (YES instances), or not even linearly equivalent (NO instances). The promise excludes the case where $\mathcal{C}_1, \mathcal{C}_2$ are linearly but not permutationally equivalent. We then show that taking the *closures* of the input codes $\mathcal{C}_1$, $\mathcal{C}_2$ gives a reduction from LCE to UCE, where the closure of a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is the code $\mathcal{C} \otimes (a)_{a \in \mathbb{F}_q^*}$.[1] This builds on a result of Sendrier and Simos [SS13b] who showed that computing the closures of $\mathcal{C}_1, \mathcal{C}_2$ gives a reduction from LCE to PCE. Indeed, because UCE trivially reduces to each of PCE, SPCE, and LCE (preserving the dimension and block length of the underlying codes), our work strengthens the result of [SS13b].

Finally, we discuss connections between codes and linear matroids. In particular, we note that there is a close connection between equivalent matroid representations and equivalent codes, and specifically that codes with parity-check matrices A_1 and A_2 being (semi-)linearly equivalent implies that the linear matroids $M[A_1]$ and $M[A_2]$ are isomorphic. We did not see this connection discussed elsewhere. See Section 6.

1.2 Related Work on Matroid Theory and its Applications to Reductions

We focus on saying more about work related to the hardness reduction used to show Theorem 1.1, and sketching how it is useful to us. First, we note that our reduction is similar to and was inspired by a reduction from GI to LIP in work of Dutour Sikirić, Schürmann, and Vallentin [DSV09]. This in turn is related to an earlier reduction from GI to CE over $\mathbb{F}_2$ appearing in [KO06, Theorem 11.55]. Although the reduction in [KO06] uses similar ideas to ours, it is not fine-grained as ours is. ([KO06] uses less efficient graph gadgets to ensure "well-connectedness" and does not observe that it is more efficient to work with the dual codes to the ones they output.) So, our work is the first to achieve the parameters in Theorem 1.1 even in the important case of binary codes ($\mathbb{F} = \mathbb{F}_2$), although [KO06] could have achieved this with a bit more work. More importantly, our reduction works for general fields $\mathbb{F}$ and not just $\mathbb{F}_2$.

We note that the main technical tools in our hardness reduction corresponding to Theorem 1.1 and in [KO06, DSV09] come from matroid theory. (Although our reduction is simpler and achieves better parameters than the one used by Petrank and Roth [PR97], its correctness analysis requires heavier machinery.) Many of these results are stated in Oxley's textbook on matroids [Oxl11], and we have written our reduction in a way that relies on results written there as much as possible.

In particular, Theorem 1.1 and the reductions in [KO06, DSV09] use the work of Whitney [Whi33] on the connection between isomorphic graphs and isomorphic graphic matroids, and the theory Tutte [Tut71] developed about *chain groups*. Tutte uses the algebraic topological view of the $|V| \times |E|$ incidence matrix A of a graph $G = (V, E)$ as the *boundary operator*, which is a linear map from 1-chains (vectors indexed by edges in a graph) to 0-chains (vectors indexed by vertices in a graph). He also studies the transposed incidence matrix A^T or *co-boundary operator*, which maps 0-chains to 1-chains.

By definition, the kernel of A is called the *cycle group* of G, and because of this codes $\mathcal{C} := \ker(A)$ for such A are called *cycle codes*. Tutte showed that the set of supports of primitive vectors in such a cycle code $\mathcal{C}$ (i.e, of non-zero vectors in $\mathcal{C}$ with inclusion-wise minimal support) is in bijective correspondence with the set of simple cycles in G. The set of simple cycles of G determines the graphic matroid $M(G)$ of G, which in turn determines the graph G (up to isomorphism, for "well-connected" G) by Whitney's work [Whi33].

1.3 Other Related Work

A substantial amount of additional work has studied algorithmic and cryptanalytic aspects of CE including [Sen00, BCGQ11, Beu20, BBPS23]. We note that the Support Splitting Algorithm of [Sen00], which works for PCE but not LCE, also works for SPCE since $(SC)^{\perp} = SC^{\perp}$ for a signed permutation matrix S (i.e., applying a signed permutation to a code commutes with taking the dual of the code). This gives some sense of the practical complexity of SPCE, which is less common than LCE and PCE but naturally appears in both of our main reductions. Additionally, [SS13a, SS13b, BD22, DG23] study the relationship between variants of code equivalence. Specifically, [SS13a, SS13b] give a reduction from LCE to PCE (mentioned above), [DG23] uses a similar idea to [SS13b] to give a reduction from SPCE to PCE, and [BD22] shows that two codes are equivalent under a very general notion of "equivalence" if and only if they are semi-linearly equivalent.

[1]The notation $(a)_{a \in \mathbb{F}_q^*}$ denotes a vector of the elements of $\mathbb{F}_q^*$ (in arbitrary order).

Our main reductions are from GI to CE and from CE to LIP. We note two prior works that in some sense give reductions in the opposite direction of ours for special cases. Specifically, [BOS19] gives an efficient reduction from PCE to GI when the codes in the PCE instance have a trivial hull (the hull of a code $\mathcal{C}$ with dual code $\mathcal{C}^{\perp}$ is $\mathcal{C} \cap \mathcal{C}^{\perp}$). [DG23] gives an efficient algorithm for LIP on Construction-A lattices built from codes with trivial hulls using oracles for both SPCE and ZLIP, which is LIP restricted to rotations of $\mathbb{Z}^n$. To the best of our knowledge, [Reg14, DG23] are the only prior works considering the relationship between CE and LIP.

In terms of other work on the complexity of CE, [PR97] shows GI-hardness of PCE on binary codes (as noted above), and gives an interactive proof system for the problem, showing that it is unlikely to be NP-hard (we discuss this proof system in Section 6). Moreover, recent work [BM23] gives a search-to-decision reduction for PCE.

The main algorithm for the general LIP is [HR14], although some other work has studied LIP on structured families of lattices, mainly by reducing it to the Shortest Vector Problem on those lattices. In [CGG17], the authors study when Construction-A lattices built from binary and ternary codes have orthogonal bases (i.e., are isomorphic, up to scaling, to $\mathbb{Z}^n$).

Finally, we note the work of Rao and Sarma [RS11], which to the best of our knowledge is the only systematic study of computational isomorphism problems on matroids. We discuss this work further in Section 6.

1.4 Paper Organization

In Section 2, we give definitions and background material about graphs, codes, lattices, and matroids, as well as computational problems on them. The main technical results of the paper are in Sections 3 to 5. In Section 3, we give our reduction from the Graph Isomorphism Problem to variants of the Code Equivalence Problem; in Section 4, we give reductions between variants of the Code Equivalence Problem; and in Section 5 we give our reduction from variants of the Code Equivalence Problem to the Lattice Isomorphism Problem. The main result in Section 5 relies on a result proved in Section 4, but we note that otherwise these sections are essentially independent. Finally, we discuss connections between the Code Equivalence Problem and isomorphism problems on matroids in Section 6.

1.5 Acknowledgments

We would like to thank Jean-François Biasse, Léo Ducas, Josh Grochow, Amir Nayyeri, Lars Ran, Oded Regev, Noah Stephens-Davidowitz, and Wessel van Woerden for helpful comments and discussions. We specifically thank Lars and Wessel for allowing us to include an illustrative example they found [RvW24] in this work; see Section 5.1. We would also like to thank Raghavendra Rao for answering our question about [RS11]. Finally, we thank the anonymous reviewers for their helpful comments.

2 Preliminaries

We use $\mathbf{1}_n$ to denote the vector of length n consisting of all 1s, or simply write $\mathbf{1}$ when n is clear from context. We use I_n to denote the $n \times n$ identity matrix.

We use $\|\boldsymbol{x}\|_0$ to denote the Hamming weight of a vector, and $\|\boldsymbol{x}\|$ to denote the Euclidean norm of a vector in $\boldsymbol{x} \in \mathbb{R}^n$. We will use the notation $A_{I,J}$ to denote the $|I| \times |J|$ submatrix of an $m \times n$ matrix A formed by taking the rows indexed by $I \subseteq [m]$ and columns indexed by $J \subseteq [n]$ of A.

We use $N_G(v)$ to denote the neighborhood of a vertex v in a graph G, and use $\deg_G(v) = |N_G(v)|$ to denote its degree. We extend the definition of a neighborhood to a set of vertices S in G by defining $N_G(S) := \cup_{v \in V} N_G(v)$. We omit the subscript when the graph G is clear from context.

2.1 Algebra

We use $\mathbb{F}$ to denote a field and $\mathbb{F}_q$ to denote a finite field of order q, where q is a prime power. Given a commutative ring R, we use R^* to denote its multiplicative subgroup. When R is a field $\mathbb{F}$, $\mathbb{F}^* = \mathbb{F} \setminus \{0\}$. We use $\mathrm{Sym}(S)$ to denote the symmetric group on a finite set S, and define $S_n := \mathrm{Sym}([n])$.

2.1.1 Matrix Groups

Let R be a commutative ring. We define $\mathcal{P}_n(R)$ to be the set of $n \times n$ permutation matrices over R, $\mathcal{SP}_n(R)$ to be the set of $n \times n$ signed permutation matrices over R, $\mathcal{M}_n(R)$ to be the set of $n \times n$ monomial matrices over R, and $\mathrm{GL}_n(R)$ to be the set of invertible $n \times n$ matrices over R. A *signed permutation matrix* $S \in \mathcal{SP}_n(R)$ is a matrix with $S = DP$ for a diagonal matrix $D = \mathrm{diag}(d_1, \ldots, d_n)$ with $d_1, \ldots, d_n \in \{-1, 1\}$ and $P \in \mathcal{P}_n(R)$. A *monomial matrix* $M \in \mathcal{M}_n(R)$ is a matrix with $M = DP$ for a diagonal matrix $D = \mathrm{diag}(d_1, \ldots, d_n)$ with $d_1, \ldots, d_n \in R^*$ and $P \in \mathcal{P}_n(R)$. We note that $\mathcal{P}_n(R) \subseteq \mathcal{SP}_n(R) \subseteq \mathcal{M}_n(R) \subseteq \mathrm{GL}_n(R)$ for any R, and that each of these sets of matrices forms a group under matrix multiplication. For fields $\mathbb{F}$, $\mathcal{M}_n(\mathbb{F})$ corresponds to the set of linear isometries on $\mathbb{F}^n$ with respect to Hamming distance.

An *orthogonal matrix* is a matrix $O \in \mathbb{R}^{n \times n}$ satisfying $O^T O = I_n$. We define $\mathcal{O}_n$ to be the set of $n \times n$ orthogonal matrices. We note that $\mathcal{O}_n$ is the set of linear isometries on $\mathbb{R}^n$ with respect to Euclidean distance, and so for any $O \in \mathcal{O}_n$ and $\boldsymbol{x} \in \mathbb{R}^n$, $\|O\boldsymbol{x}\| = \|\boldsymbol{x}\|$.

2.2 Codes

A *linear code* (or simply *code*) is a linear subspace $\mathcal{C} \subseteq \mathbb{F}^n$, where $\mathbb{F}$ is a field. (Typically $\mathbb{F}$ is taken to be a finite field $\mathbb{F}_q$, but codes are well-defined over general fields and one of our main results holds for such general fields.) Here n is called the *block length* of $\mathcal{C}$, the *dimension* k of $\mathcal{C}$ is its dimension as a vector space, and the *minimum distance* of the code is defined as

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

A code over $\mathbb{F}$ with block length n, dimension k, and minimum distance at least d is called an $[n, k, d]_{\mathbb{F}}$ code. We sometimes drop d or $\mathbb{F}$ and write things like $[n, k, d]$ or $[n, k]_{\mathbb{F}}$ when we do not wish to specify a code's distance or field of definition. For codes over finite fields $\mathbb{F} = \mathbb{F}_q$ of order q, we write $[n, k, d]_q$ and $[n, k]_q$.

There are two standard ways to represent $[n, k]_{\mathbb{F}}$ codes, both of which we will use. The first way to represent a code is as the linear span of the columns of a *generator matrix* $G \in \mathbb{F}^{n \times k}$ with linearly independent columns (equivalently, the image of G):[2]

$$\mathcal{C}(G) := \{G\boldsymbol{x} : \boldsymbol{x} \in \mathbb{F}^k\} \ .$$

The second, dual representation is as the kernel of a *parity check matrix* $H \in \mathbb{F}^{(n-k) \times n}$ with linearly independent rows:

$$\mathcal{C}^{\perp}(H) := \{\boldsymbol{x} \in \mathbb{F}^n : H\boldsymbol{x} = \boldsymbol{0}\} \ .$$

We note that $G \in \mathbb{F}^{n \times k}$ of full column rank and $H \in \mathbb{F}^{(n-k) \times n}$ of full row rank satisfy $\mathcal{C}(G) = \mathcal{C}^{\perp}(H)$ if and only if $HG = 0$.

For convenience, we will in fact allow slightly more general versions of generator (respectively, parity-check) matrices, in which the columns (respectively, rows) are not required to be linearly independent. I.e., we allow generator matrices of $[n, k]$ codes with more than k vectors, and parity-check matrices of such codes with more than $n - k$ parity-check constraints. It is straightforward to compute a "standard" generator matrix with exactly k vectors or parity-check matrix with exactly $n - k$ constraints efficiently from an "overcomplete" generator or a parity-check matrix of an $[n, k]$ code using Gaussian elimination. Additionally, it is efficient to compute a generator matrix from a parity-check matrix and vice-versa.

[2]In coding theory codes are often generated by the *rows* of generator matrices. Here we match the column-basis convention commonly used for lattices.

Given a code $\mathcal{C} \subseteq \mathbb{F}^n$, we define its *dual code* as

$$\mathcal{C}^\perp := \{\boldsymbol{x} \in \mathbb{F}^n : \forall \boldsymbol{y} \in \mathcal{C}, \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0\} \ .$$

If G is a generator matrix of a code $\mathcal{C}$ and H is a parity-check matrix of $\mathcal{C}$ (i.e., $\mathcal{C} = \mathcal{C}(G) = \mathcal{C}^\perp(H)$) then the dual code $\mathcal{C}^\perp$ of $\mathcal{C}$ satisfies $\mathcal{C}^\perp = \mathcal{C}(H^T) = \mathcal{C}^\perp(G^T)$. If $\mathcal{C}$ is an $[n,k]$ code then $\mathcal{C}^\perp$ is an $[n, n-k]$ code.

For a code $\mathcal{C} \subseteq \mathbb{F}^{m_1}$ with generator matrix G and a vector $\boldsymbol{y} \in \mathbb{F}^{m_2}$, we define

$$\mathcal{C} \otimes \boldsymbol{y} := \mathcal{C}(G \otimes \boldsymbol{y}) = \{\boldsymbol{x} \otimes \boldsymbol{y} : \boldsymbol{x} \in \mathcal{C}\} \subseteq \mathbb{F}^{m_1 m_2} \ ,$$

where $\otimes$ denotes the *Kronecker product*.[3] Given a code $\mathcal{C} \subseteq \mathbb{F}^n$, we call the code $\mathcal{C}_I := \{(x_i)_{i \in I} : \boldsymbol{x} \in \mathcal{C}\} \subseteq \mathbb{F}^{|I|}$ for $I \subseteq [n]$ a *punctured code* with *puncture pattern* I. I.e., $\mathcal{C}_I$ is the code obtained from $\mathcal{C}$ by projecting each codeword of $\mathcal{C}$ onto the coordinates I.

2.3 Lattices

A *lattice* $\mathcal{L}$ is the set of all integer linear combinations of some n linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^m$. The matrix $B := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ with these vectors as its columns is called a basis of $\mathcal{L}$. That is,

$$\mathcal{L} = \mathcal{L}(B) := \{B\boldsymbol{z} : \boldsymbol{z} \in \mathbb{Z}^n\} = \{\sum_{i=1}^{n} z_i \boldsymbol{b}_i : z_1, \ldots, z_n \in \mathbb{Z}\} \ .$$

Here n is the *rank* of the lattice, and m is its *ambient dimension*. The *minimum distance* of a lattice $\mathcal{L}$ is defined as

$$\lambda_1(\mathcal{L}) := \min_{\substack{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{L}, \\ \boldsymbol{x} \neq \boldsymbol{y}}} \|\boldsymbol{x} - \boldsymbol{y}\| = \min_{\boldsymbol{x} \in \mathcal{L} \setminus \{\boldsymbol{0}\}} \|\boldsymbol{x}\| \ .$$

We note that codes and lattices are defined analogously. In particular, the dimension of a code is analogous to the rank of a lattice and the block length of a code is analogous to the ambient dimension of a lattice. The minimum distance of a code is defined in terms of Hamming distance, while the minimum distance of a lattice is defined in terms of Euclidean distance.

Furthermore, we note that two generator matrices $G_1, G_2 \in \mathbb{F}_q^{m \times n}$ generate the same code if and only if $G_2 = G_1 U$ for some $U \in \mathrm{GL}_n(\mathbb{F}_q)$, and similarly that two basis matrices $B_1, B_2 \in \mathbb{R}^{m \times n}$ generate the same lattice if and only if $B_2 = B_1 U$ for some $U \in \mathrm{GL}_n(\mathbb{Z})$. Matrices in $\mathrm{GL}_n(\mathbb{Z})$ are called *unimodular* matrices.

2.3.1 Construction A

There is a simple way to define a lattice $\mathcal{L}_A(\mathcal{C})$ from a code $\mathcal{C} \subseteq \mathbb{F}_p^n$ for prime p, which is called a *Construction-A lattice* [CS99].[4] Specifically, the Construction-A lattice associated with such a code $\mathcal{C}$ is the lattice defined as

$$\mathcal{L}_A(\mathcal{C}) := \mathcal{C} + p\mathbb{Z}^n \ . \tag{1}$$

I.e., $\mathcal{L}_A(\mathcal{C})$ is the lattice obtained by lifting $\mathcal{C}$ to the integers, and allowing for adding integer multiples of p to each coordinate. (When lifting $\mathbb{F}_p$ to $\mathbb{Z}$ and mixing arithmetic between elements in $\mathbb{F}_p$ and $\mathbb{Z}$, we associate the elements of $\mathbb{F}_p$ with the integers $\{0, 1, \ldots, p-1\}$ in the natural way.) We note that $\mathcal{L}_A(\mathcal{C})$ has rank and ambient dimension n. We also note that $\mathcal{C}$ is isomorphic to $\mathcal{L}_A(\mathcal{C})/(p\mathbb{Z}^n)$ as an additive group.

Let $\boldsymbol{e}_i$ be the ith standard normal basis vector. The following lemma gives a sufficient condition for the $2n$ vectors $\pm p\boldsymbol{e}_i$ to be the (only) shortest non-zero vectors in $\mathcal{L}_A(\mathcal{C})$ when $\mathcal{C} \subset \mathbb{F}_p^n$.

Lemma 2.1. *Let p be a prime, and let $\mathcal{C} \subset \mathbb{F}_p^n$ be a linear code with minimum distance $d > p^2$. Then $\lambda_1(\mathcal{L}_A(\mathcal{C})) = p$ and $\boldsymbol{x} \in \mathcal{L}_A(\mathcal{C})$ satisfies $\|\boldsymbol{x}\| = p$ if and only if $\boldsymbol{x} = \pm p\boldsymbol{e}_i$ for some $i \in [n]$.*

[3]One can more generally define the tensor product of codes $\mathcal{C}_1$ and $\mathcal{C}_2$ with generator matrices G_1 and G_2 as $\mathcal{C}_1 \otimes \mathcal{C}_2 = \mathcal{C}(G_1 \otimes G_2)$, but we will only use the case where $\mathcal{C}_2$ is one-dimensional (generated by a single vector $\boldsymbol{y}$).

[4]Technically, the definition of Construction-A lattices in [CS99] is only for the $p = 2$ case, but it immediately generalizes to arbitrary primes p.

Proof. By definition, $\pm p\boldsymbol{e}_i \in \mathcal{L}_A(\mathcal{C})$ for every i, which implies that $\lambda_1(\mathcal{L}_A(\mathcal{C})) \le p$. It remains to show that any vector $\boldsymbol{x} \in \mathcal{L}_A(\mathcal{C}) \setminus \{\boldsymbol{0}, \pm p\boldsymbol{e}_1, \dots, \pm p\boldsymbol{e}_n\}$ is such that $\|\boldsymbol{x}\| > p$. By definition of $\mathcal{L}_A(\mathcal{C})$, we can write any such $\boldsymbol{x}$ as $\boldsymbol{x} = \boldsymbol{c} + \boldsymbol{y}$ for $\boldsymbol{c} \in \mathcal{C}$ and $\boldsymbol{y} \in p\mathbb{Z}^n$. If $\boldsymbol{c} = \boldsymbol{0}$, then $\boldsymbol{x} \in p\mathbb{Z}^n \setminus \{\boldsymbol{0}, \pm p\boldsymbol{e}_1, \dots, \pm p\boldsymbol{e}_n\}$, and so $\|\boldsymbol{x}\| \ge \sqrt{2}p > p$. Otherwise, if $\boldsymbol{c} \ne \boldsymbol{0}$, $\|\boldsymbol{x}\| \ge \sqrt{\|\boldsymbol{c}\|_0} \ge \sqrt{d} > p$. The lemma follows. □

2.4 Matroids

We next give background on matroids. We refer the reader to the book of Oxley [Oxl11] for a comprehensive resource on matroids. We will generally use the notation in [Oxl11] and will cite a number of results from it.

Definition 2.2. A *matroid* $M = (E, \mathcal{I})$ is a pair consisting of a *ground set* E and a collection $\mathcal{I}$ of subsets of E that satisfy the following three properties:

1. $\emptyset \in \mathcal{I}$.
2. If $I \in \mathcal{I}$ and $I' \subseteq I$, then $I' \in \mathcal{I}$.
3. If I_1 and I_2 are in $\mathcal{I}$ and $|I_1| < |I_2|$, then there is an element $e \in I_2 \setminus I_1$ such that $I_1 \cup \{e\} \in \mathcal{I}$.

We call the second and third items the *hereditary* and *independence augmentation* properties of matroids, respectively. We call sets $I \in \mathcal{I}$ *independent sets* and sets $S \subseteq E, S \notin \mathcal{I}$ *dependent sets* of M. We call a maximal independent set $I \in \mathcal{I}$ a *basis* of M, and a minimal dependent set $S \subseteq E$ of M a *circuit* of M. I.e., adding any element to a basis makes it dependent, and removing any element from a circuit makes it independent.

Two matroids $M_1 = (E_1, \mathcal{I}_1)$ and $M_2 = (E_2, \mathcal{I}_2)$ are said to be *isomorphic* if there is a bijection $\pi : E_1 \to E_2$ such that for every set $S \subseteq E_1$, $S \in \mathcal{I}_1$ if and only if $\pi(S) = \{\pi(s) : s \in S\} \in \mathcal{I}_2$.

2.4.1 Classes of Matroids

The two most important classes of matroids (which inspire much of the terminology about matroids) are *graphic matroids* and *linear matroids*.

Definition 2.3. Given an undirected graph $G = (V, E)$, the *graphic matroid* (or *cycle matroid*) $M(G)$ derived from G is defined as follows. Its ground set is the set E of edges in G, and $S \subseteq E$ is an independent set if and only if the subgraph $G' = (V, S)$ of G is acyclic (i.e., a forest).

For connected graphs G, the bases of $M(G)$ are spanning trees of G. The circuits of $M(G)$ are the (sets of edges of) cycles of G.

Definition 2.4. Given a matrix $A = (\boldsymbol{a}_1, \dots, \boldsymbol{a}_m) \in \mathbb{F}^{n \times m}$ over a field $\mathbb{F}$, the *linear matroid* $M[A]$ is defined as follows. Its ground set is the set of indices $[m]$ of columns of A, and $S \subseteq [m]$ is independent if and only if the (multi)set of columns indexed by S is linearly independent over $\mathbb{F}$.

A matroid M is called $\mathbb{F}$-*representable* if there exists a matrix A over $\mathbb{F}$ such that $M = M[A]$. Such a matrix A is called an $\mathbb{F}$-*representation* of M. A matroid M is called *representable* (or simply *linear*) if it is $\mathbb{F}$-representable for some $\mathbb{F}$. We next show that every graphic matroid is $\mathbb{F}$-representable for every field $\mathbb{F}$.

Given a simple graph $G = (V, E)$, we call a directed graph $D = D(G) = (V, E')$ an *orientation* of G if for every undirected edge $\{u, v\} \in E$, exactly one of the directed edges (u, v) and (v, u) is in E'. Furthermore, given a directed graph $D = (V, E)$, we define the *incidence matrix* $A = A(D) \in \{-1, 0, 1\}^{V \times E}$ of D as follows.[5] Let $v \in V$ and let $e = (u, w) \in E$ be a directed edge. Then

$$A_{v,e} := \begin{cases} 1 & \text{if } v = w \ , \\ -1 & \text{if } v = u \ , \\ 0 & \text{otherwise} \ . \end{cases} \tag{2}$$

[5]We use notation like $A \in \{-1, 0, 1\}^{V \times E}$ to indicate that A is an $|V| \times |E|$ matrix whose rows and columns are indexed by V and E, respectively.

We note that A is well-defined over any field $\mathbb{F}$ (with $-1 = 1$ if $\mathbb{F}$ is a characteristic-2 field), that each column of A has exactly two non-zero entries (corresponding to the head w and tail u of each edge in D), and that if D_1 and D_2 are orientations of the same undirected graph G that $A(D_1) = A(D_2) \cdot F$ for some diagonal matrix F with ± 1 entries on the main diagonal.

The following theorem says that incidence matrices of (orientations of) graphs G are representations of $M(G)$.

Theorem 2.5 ([Oxl11, Proposition 5.1.2 and Lemma 5.1.3])**.** *If G is a simple graph, then $M(G)$ is representable over any field $\mathbb{F}$. Moreover, if $D = D(G)$ is an (arbitrary) orientation of G, then $A(D) \in \mathbb{F}^{V \times E}$ is an $\mathbb{F}$-representation of $M(G)$.*

We will use the following standard fact.

Lemma 2.6. *Let $G = (V, E)$ be a connected graph and let $\mathbb{F}$ be a field. Then the incidence matrix $A \in \mathbb{F}^{V \times E}$ of an arbitrary orientation of G has row rank $|V| - 1$.*

Proof. Let $n := |V|$. Because each column of A contains one 1, one -1, and otherwise 0s, we have that $\mathbf{1}^T A = \mathbf{0}$. So, the row rank of A is at most $n - 1$.

We now prove that the row rank of A is at least $n - 1$. Let $T = (V, E_T)$ be a spanning tree of G with incidence matrix A_T. Because A_T is a submatrix of A, it suffices to show that A_T has rank (exactly) $n - 1$. Specifically, we claim that for every edge $f \in E_T$, $\boldsymbol{e}_f^T$ is in the row span of A_T, where $\boldsymbol{e}_f \in \{0,1\}^{E_T}$ is the standard normal basis vector indexed by f. Fix such an edge f. Then removing f from E_T partitions T into two connected components with respective vertex sets W and $\overline{W}$. Define $\boldsymbol{x} \in \{0,1\}^V$ by

$$x_v := \begin{cases} 0 & \text{if } v \in W \ , \\ 1 & \text{if } v \in \overline{W} \ . \end{cases}$$

Then $\boldsymbol{x}^T A_T \in \{\pm \boldsymbol{e}_f^T\}$, as needed. □

2.5 Isomorphism Problems

We next define (the decision versions of) several computational isomorphism problems.

2.5.1 Problems on Graphs

We first define the Graph Isomorphism Problem (GI).

Definition 2.7 (Graph Isomorphism Problem)**.** The Graph Isomorphism Problem (GI) is the decision problem defined as follows. Given graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ with $n := |V|$ as input, decide whether there exists $\sigma \in S_n$ such that for all $i, j \in [n], i \neq j$, v_i and v_j are adjacent in G_1 if and only if $v_{\sigma(i)}$ and $v_{\sigma(j)}$ are adjacent in G_2.

2.5.2 Problems on Codes

We say that codes $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}^n$ are *permutationally* (respectively, *signed permutationally*, *linearly*) *equivalent* if there exists $P \in \mathcal{P}_n(\mathbb{F})$ (respectively, $S \in \mathcal{SP}_n(\mathbb{F})$, $M \in \mathcal{M}_n(\mathbb{F})$) such that $P\mathcal{C}_1 = \mathcal{C}_2$ (respectively, $S\mathcal{C}_1 = \mathcal{C}_2$, $M\mathcal{C}_1 = \mathcal{C}_2$). We use $\mathcal{C}_1 \overset{P}{\approx} \mathcal{C}_2$, $\mathcal{C}_1 \overset{SP}{\approx} \mathcal{C}_2$, and $\mathcal{C}_1 \overset{L}{\approx} \mathcal{C}_2$ to denote that codes are permutationally, signed permutationally, and linearly equivalent, respectively. We recall that the matrix U in the following definitions corresponds to a change of basis. We define the following three corresponding computational problems.

Definition 2.8 (Code Equivalence Problem Variants)**.** For a field $\mathbb{F}$, the *Permutation Code Equivalence Problem* (respectively, *Signed Permutation Code Equivalence Problem*, *Linear Code Equivalence Problem* over $\mathbb{F}$, denoted $\mathrm{PCE}_{\mathbb{F}}$ (respectively, $\mathrm{SPCE}_{\mathbb{F}}$, $\mathrm{LCE}_{\mathbb{F}}$), is the decision problem defined as follows. Let $n, k \in \mathbb{Z}^+$. On input generator matrices $G_1, G_2 \in \mathbb{F}^{n \times k}$, decide whether there exist $P \in \mathcal{P}_n(\mathbb{F})$ (respectively, $S \in \mathcal{SP}_n(\mathbb{F})$, $M \in \mathcal{M}_n(\mathbb{F})$) and $U \in \mathrm{GL}_k(\mathbb{F})$ such that $PG_1U = G_2$ (respectively, $SG_1U = G_2$, $MG_1U = G_2$).

We note that all three problems are the same in the $\mathbb{F} = \mathbb{F}_2$ case since $\mathcal{P}_n(\mathbb{F}_2) = \mathcal{SP}_n(\mathbb{F}_2) = \mathcal{M}_n(\mathbb{F}_2)$, and that SPCE and LCE are the same in the $\mathbb{F} = \mathbb{F}_3$ case since $\mathcal{SP}_n(\mathbb{F}_3) = \mathcal{M}_n(\mathbb{F}_3)$.

We will also use the following promise problem, which we call the Universal Code Equivalence Problem (UCE). It has the YES instances of PCE and the NO instances of LCE.

Definition 2.9 (Universal Code Equivalence Problem)**.** For a field $\mathbb{F}$, the Universal Code Equivalence Problem over $\mathbb{F}$ (UCE_q) is the decision problem defined as follows. Let $n, k \in \mathbb{Z}^+$. On input generator matrices $G_1, G_2 \in \mathbb{F}^{n \times k}$, decide which of the following holds:

1. (YES instance.) There exist $P \in \mathcal{P}_n(\mathbb{F})$ and $U \in \mathrm{GL}_k(\mathbb{F})$ such that $G_2 = PG_1U$.
2. (NO instance.) There do not exist $M \in \mathcal{M}_n(\mathbb{F})$ and $U \in \mathrm{GL}_k(\mathbb{F})$ such that $G_2 = MG_1U$.

We note that UCE trivially reduces to each of the other three variants of code equivalence that we have defined.

Lemma 2.10. *Let $n, k, d \in \mathbb{Z}^+$ and let $\mathbb{F}$ be a field. There is a* $\mathrm{poly}(n)$*-time reduction from* $\mathrm{UCE}_{\mathbb{F}}$ *on* $[n, k, d]$ *codes to each of* $\mathrm{LCE}_{\mathbb{F}}$, $\mathrm{SPCE}_{\mathbb{F}}$, *and* $\mathrm{PCE}_{\mathbb{F}}$ *on* $[n, k, d]$ *codes.*

Proof. The reduction is the identity mapping. The lemma follows by noting that the YES and NO instances of $\mathrm{UCE}_{\mathbb{F}}$ are subsets of the respective YES and NO instances of each of $\mathrm{LCE}_{\mathbb{F}}$, $\mathrm{SPCE}_{\mathbb{F}}$, and $\mathrm{PCE}_{\mathbb{F}}$. □

We will use the following lemma, which says that two codes are equivalent (in any of the above senses) if and only if their dual codes are equivalent.

Lemma 2.11. *Let $\mathbb{F}$ be a field. If two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ over $\mathbb{F}$ are permutationally (respectively, signed permutationally, linearly) equivalent, then their dual codes $\mathcal{C}_1^\perp$ and $\mathcal{C}_2^\perp$ are permutationally (respectively, signed permutationally, linearly) equivalent.*

Proof. Suppose that $G_1, G_2 \in \mathbb{F}^{n \times k}$ are respective generator matrices of $\mathcal{C}_1, \mathcal{C}_2$, that H_1 is a parity-check matrix for $\mathcal{C}_1$, and that $\mathcal{C}_1 \overset{P}{\approx} \mathcal{C}_2$. Then by assumption, $H_1G_1 = 0$ and there exist $P \in \mathcal{P}_n(\mathbb{F})$ and $U \in \mathrm{GL}_n(\mathbb{F})$ such that $PG_1U = G_2$. We claim that $H_2 := H_1P^{-1}$ is a parity-check matrix for $\mathcal{C}_2$. Indeed, this follows because $H_2G_2 = H_1P^{-1}PG_1U = 0$. Therefore, H_1^T, H_2^T are generator matrices for $\mathcal{C}_1^\perp, \mathcal{C}_2^\perp$, respectively, and $H_2^T = (P^{-1})^T H_1^T$. It follows that $\mathcal{C}_1^\perp \overset{P}{\approx} \mathcal{C}_2^\perp$.

A similar argument works with signed permutational or linear equivalence in place of permutational equivalence. □

Additionally, we note that even if $G_1, G_2 \in \mathbb{F}^{n \times k'}$ are overcomplete generator matrices of $[n, k]_{\mathbb{F}}$ codes $\mathcal{C}_1, \mathcal{C}_2$ (i.e., if $k' > k$), it still holds that $\mathcal{C}_1 \overset{P}{\approx} \mathcal{C}_2$ if and only if there exist a permutation matrix $P \in \mathcal{P}_n(\mathbb{F})$ and a full-rank matrix $U \in \mathrm{GL}_{k'}(\mathbb{F})$ such that $PG_1U = G_2$. Similar statements are true for signed permutationally and linearly equivalent codes.

Finally, we note that two codes $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}^n$ over a field $\mathbb{F}$ are called *semi-linearly equivalent* if there exist $M \in \mathcal{M}_n(\mathbb{F})$ and a field automorphism α of $\mathbb{F}$ such that $\mathcal{C}_2 = M\mathcal{C}_1'$, where

$$\mathcal{C}_1' := \{(\alpha(x_1), \ldots, \alpha(x_n))^T : \boldsymbol{x} = (x_1, \ldots, x_n)^T \in \mathcal{C}_1\} \ .$$

2.5.3 Problems on Lattices

We now define the Lattice Isomorphism Problem (LIP).

Definition 2.12 (Lattice Isomorphism Problem)**.** The Lattice Isomorphism Problem (LIP) is the decision problem defined as follows. Let $n \in \mathbb{Z}^+$. On input bases $B_1, B_2 \in \mathbb{Q}^{n \times n}$, decide whether there exist $O \in \mathcal{O}_n$ and $U \in \mathrm{GL}_n(\mathbb{Z})$ such that $B_2 = OB_1U$.

3 From Graphs to Codes

We now present our fine-grained reduction from the Graph Isomorphism Problem to Linear Code Equivalence over any field. Our reduction uses matroid theory, and we use results about matroids from Oxley's textbook [Oxl11].

3.1 Results about Isomorphic Matroids

We start by collecting useful results about when matroids are isomorphic.

The following result addresses the relationship between when two graphs G_1 and G_2 are isomorphic and when their corresponding graphic matroids $M(G_1)$ and $M(G_2)$ are isomorphic. In fact, Whitney's famous 2-isomorphism theorem [Whi33] gives a complete characterization of when $M(G_1)$ and $M(G_2)$ are isomorphic, but we will only consider the special case when the graphs are 3-connected. A graph G is called *k-vertex-connected* or simply *k-connected* if it has at least k vertices and it remains connected when fewer than k vertices and their incident edges are removed.

Theorem 3.1 ([Oxl11, Lemma 5.3.2])**.** *If G_1 and G_2 are 3-connected graphs and $M(G_1)$ and $M(G_2)$ are isomorphic, then G_1 and G_2 are isomorphic.*

We note that the converse statement to Theorem 3.1—if G_1 and G_2 are isomorphic then $M(G_1)$ and $M(G_2)$ are isomorphic—holds in general, regardless of whether G_1 and G_2 are 3-connected. We will use this in the sequel.

We next give an efficient reduction from the Graph Isomorphism Problem on general graphs to the Graph Isomorphism Problem on k-connected graphs, which shows that the assumption about 3-connectivity in Theorem 3.1 is essentially without loss of generality. A very similar reduction was used in a similar context to ours in [DSV09].

Lemma 3.2. *For any $k, n \in \mathbb{Z}^+$ with $n \geq k$, there is a $\mathrm{poly}(n)$-time reduction from GI on graphs with n vertices and m edges to GI on k-connected graphs with $n + k$ vertices and $m + kn$ edges.*

Proof. On input graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, the reduction constructs G_i' from G_i for $i = 1, 2$ as follows. It adds a set S of k new vertices to G_i, and adds an edge between each of these k new vertices and each of the original vertices $v \in V$. It is straightforward to check that G_1' and G_2' are k-connected, and that if G_1 and G_2 are isomorphic then so are G_1' and G_2'.

Now, consider an isomorphism π from G_1' and G_2'. If $\pi(S) = S$ then the restriction of π to V is an isomorphism from G_1 to G_2. Otherwise, there must exist a set S' of k vertices in G_2 such that $\pi(S) = S'$ and for every $v \in S'$, $N_{G_2}(v) \cap S' = \emptyset$ and $\deg_{G_2}(v) = n - k$. I.e., S' is an independent set of size k in G_2 and each of its vertices is connected to all vertices in $V \setminus S'$.[6] In that case, there is an automorphism σ on G_2' that swaps S and S', and the restriction of $\sigma \circ \pi$ to V is an isomorphism from G_1 to G_2. It follows that if G_1' and G_2' are isomorphic then G_1 and G_2 are also isomorphic. □

We next present a result about when two matrices represent the *same* linear matroid.

Theorem 3.3 ([Oxl11, Proposition 6.3.12])**.** *Let $n, k \in \mathbb{Z}^+$, let $\mathbb{F}$ be a field, and let $A_1, A_2 \in \mathbb{F}^{k \times n}$. Suppose that there exist $U \in \mathrm{GL}_k(\mathbb{F})$ and a non-singular $n \times n$ diagonal matrix D such that $UA_1D = A_2$. Then $M[A_1] = M[A_2]$.*

From this, we get the following corollary saying that two codes being isomorphic implies that the linear matroids defined by their parity-check matrices are isomorphic.

Corollary 3.4. *Let $n, k \in \mathbb{Z}^+$, let $\mathbb{F}$ be a field, and let $A_1, A_2 \in \mathbb{F}^{k \times n}$. If $\mathcal{C}^{\perp}(A_1) \overset{L}{\approx} \mathcal{C}^{\perp}(A_2)$ then $M[A_1]$ and $M[A_2]$ are isomorphic.*

[6]If there is such a set S' then G_2 was already k-connected.

Proof. Using the assumption that $\mathcal{C}^\perp(A_1) \overset{L}{\approx} \mathcal{C}^\perp(A_2)$ and the fact that codes are linearly equivalent if and only if their dual codes are (Lemma 2.11), there exist $U \in \mathrm{GL}_k(\mathbb{F})$ and $M \in \mathcal{M}_n(\mathbb{F})$ such that $M^T A_1^T U^T = A_2^T$ (and therefore also $UA_1M = A_2$), with $M = DP$ for a non-singular $n \times n$ diagonal matrix D and for $P \in \mathcal{P}_n(\mathbb{F}_q)$. Let $A_1' := UA_1D$. Then $M[A_1] = M[A_1']$ by Theorem 3.3, and $M[A_1']$ and $M[A_2]$ are isomorphic since $A_2 = A_1'P$. The result follows. □

3.2 The Reduction

We are now ready for our main reduction from Graph Isomorphism to Code Equivalence.

Theorem 3.5. *For any field $\mathbb{F}$, there is a $\mathrm{poly}(n)$-time reduction from the Graph Isomorphism Problem on 3-connected graphs with n vertices and m edges to each of $\mathrm{SPCE}_{\mathbb{F}}$ and $\mathrm{LCE}_{\mathbb{F}}$ on codes of dimension $n-1$ and block length m.*

Proof. On input 3-connected graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, each with n vertices and m edges, the reduction works as follows. It constructs arbitrary orientations of G_1 and G_2 with respective incidence matrices A_1 and A_2. It then outputs their transposed incidence matrices $A_i^T \in \{-1, 0, 1\}^{E_i \times V} \subseteq \mathbb{F}^{E_i \times V}$ (recall Equation (2)) for $i = 1, 2$ as generator matrices for the codes $\mathcal{C}_1 := \mathcal{C}(A_1^T)$ and $\mathcal{C}_2 := \mathcal{C}(A_2^T)$, respectively.

It is clear that the reduction is efficient and that the output codes have block length m. Moreover, because the input graphs are (3-)connected, the incidence matrices A_1 and A_2 have row rank $n-1$ by Lemma 2.6, and so the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ have dimension $n-1$. It remains to show correctness.

Assume that G_1 and G_2 are isomorphic via an isomorphism $\pi : V \to V$. Then there exists $U \in \mathcal{P}_n(\mathbb{F}_q)$ corresponding to the permutation $v \mapsto \pi(v)$ for $v \in V$, $P \in \mathcal{P}_m(\mathbb{F})$ corresponding to the bijection $\{u, v\} \mapsto \{\pi(u), \pi(v)\}$ from edges in E_1 to edges in E_2, and $D = \mathrm{diag}(d_1, \ldots, d_m)$ for $d_1, \ldots, d_m \in \{-1, 1\}$ such that $DPA_1^TU = A_2^T$.[7] It follows that $\mathcal{C}_1 \overset{SP}{\approx} \mathcal{C}_2$ and $\mathcal{C}_1 \overset{L}{\approx} \mathcal{C}_2$.

Now, assume that $\mathcal{C}_1 \overset{L}{\approx} \mathcal{C}_2$ (which is implied by $\mathcal{C}_1 \overset{SP}{\approx} \mathcal{C}_2$). By Lemma 2.11, this implies that the dual codes $\mathcal{C}_1^\perp = \mathcal{C}^\perp(A_1)$ and $\mathcal{C}_2^\perp = \mathcal{C}^\perp(A_2)$ are also linearly equivalent. By Corollary 3.4, it then holds that the linear matroids $M[A_1]$ and $M[A_2]$ are isomorphic, which in turn implies that the graphic matroids $M(G_1)$ and $M(G_2)$ are isomorphic by Theorem 2.5. Finally, using Theorem 3.1 and the fact that G_1 and G_2 are 3-connected, we have that G_1 and G_2 are isomorphic. □

Remark 3.6. We make two remarks about Theorem 3.5 and its proof. First, we remark that it is not really necessary to switch back and forth between considering the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ and their dual codes in the proof above. However, working with $\mathcal{C}_1$ and $\mathcal{C}_2$ directly implies that the corresponding *dual matroids* of $M^*(G_1)$ and $M^*(G_2)$, called the *cographic* or *bond matroids* of G_1 and G_2, are isomorphic instead of that $M(G_1)$ and $M(G_2)$ are isomorphic directly. It is not difficult to show both that codes are equivalent and matroids are isomorphic if their duals are, but we need to use duals either for matroids or codes in order to apply results from [Oxl11] as they are stated. (The reduction in [DSV09] works with bond matroids instead of cycle matroids directly when reducing GI to LIP.)

Second, we remark that not only does the proof of Theorem 3.5 work for codes defined over arbitrary fields, it largely work for codes defined over arbitrary *rings*. Indeed, the theory of chain groups developed by Tutte in [Tut71], which underlies this reduction, allows for using rings rather than fields. However, codes defined over rings (which are modules instead of vectors spaces) are not as nice algebraically and do not allow for some of what we have done above.

We conclude with the following corollary about reducing Graph Isomorphism on arbitrary graphs to Linear Code Equivalence.

Corollary 3.7. *For any field $\mathbb{F}$, there is a $\mathrm{poly}(n)$-time reduction from GI on graphs with n vertices and m edges to $\mathrm{LCE}_{\mathbb{F}}$ on codes of dimension $n+2$ and block length $m+3n$.*

[7]Here D fixes inconsistent orientations of edges $e \in E_1$ and $\pi(e) \in E_2$ in the incidence matrices A_1 and A_2.

Proof. If $n < 3$, solve the GI instance directly and output a (fixed) YES or NO instance of $\mathrm{LCE}_{\mathbb{F}}$ depending on the result. Otherwise, combine the reduction in Lemma 3.2, setting $k = 3$, with Theorem 3.5. $\square$

4 Reductions Between Code Equivalence Variants

In this section we give reductions between variants of the Code Equivalence Problem, which we will use in the next section.

4.1 Distance-Increasing Self-Reductions for LCE and PCE

The following lemma says that every permutation σ on the Cartesian product $[m] \times [n]$ induces a permutation σ' on $[m]$ such that σ' is a "projection" of σ.

Lemma 4.1. *Let $m, n \in \mathbb{Z}^+$ and let $\sigma \in \mathrm{Sym}([m] \times [n])$. Then there exists $\sigma' \in \mathrm{Sym}([m])$ such that for every $i \in [m]$ there exist $j, j' \in [n]$ such that $\sigma(i, j) = (\sigma'(i), j')$.*

Proof. Construct the bipartite graph $G = (V = L \sqcup R, E)$, where $L = \{v_1, \dots, v_m\}$, $R = \{w_1, \dots, w_m\}$, and there is an edge from $v_i \in L$ to $w_{i'} \in R$ if and only if there exist $j, j' \in [n]$ such that $\sigma(i, j) = (i', j')$.

Notice that to prove the lemma it suffices to show that G contains a perfect matching (where an edge from v_i to $w_{i'}$ in the perfect matching corresponds to the mapping $\sigma'(i) = i'$). We claim that for every $S \subseteq L$, $|N(S)| \geq |S|$. Assuming this claim, G contains a perfect matching by Hall's Theorem (see, e.g., [Got21]) and the lemma follows.

It remains to prove the claim. Let $S \subseteq L$, and suppose that $|N(S)| < |S|$. Let $T := \{(i, j) : v_i \in S, j \in [n]\}$ and $T' := \{(i', j') : w_{i'} \in N(S), j' \in [n]\}$. Then $\sigma(T) \subseteq T'$, and so $|\sigma(T)| \leq |T'| = n\,|N(S)| < n\,|S| = |T|$. However, σ is a permutation and hence injective, so $|\sigma(T)| < |T|$ is a contradiction. $\square$

We now get the following theorem, which gives efficient distance-increasing but dimension-preserving reductions for LCE and PCE.

Theorem 4.2. *Let q be a prime power, let $r \in \mathbb{Z}^+$, let $\boldsymbol{y} \in (\mathbb{F}_q^*)^r$, let $\mathcal{C}_1, \mathcal{C}_2$ be $[n, k, d]_q$ codes for some $n, k, d \in \mathbb{Z}^+$, and let $\mathcal{C}_1', \mathcal{C}_2'$ be the $[rn, k, rd]_q$ codes defined by $\mathcal{C}_1' := \mathcal{C}_1 \otimes \boldsymbol{y}, \mathcal{C}_2' := \mathcal{C}_2 \otimes \boldsymbol{y}$, respectively. Then:*

1. *$\mathcal{C}_1 \overset{L}{\approx} \mathcal{C}_2$ if and only if $\mathcal{C}_1' \overset{L}{\approx} \mathcal{C}_2'$.*
2. *If $\boldsymbol{y} = \mathbf{1}_r$, then $\mathcal{C}_1 \overset{P}{\approx} \mathcal{C}_2$ if and only if $\mathcal{C}_1' \overset{P}{\approx} \mathcal{C}_2'$.*

Proof. We first prove the forward direction of the items, in which we assume that $\mathcal{C}_1$ and $\mathcal{C}_2$ are equivalent. For Item 1, suppose that $M \in \mathcal{M}_n(\mathbb{F}_q)$ is such that $M(\mathcal{C}_1) = \mathcal{C}_2$. We claim that $M' := M \otimes I_r \in \mathcal{M}_{nr}(\mathbb{F}_q)$ is such that $M'(\mathcal{C}_1') = \mathcal{C}_2'$. Because $\mathcal{C}_1'$ and $\mathcal{C}_2'$ are codes of the same dimension and M' is injective, to show that $\mathcal{C}_1' \overset{L}{\approx} \mathcal{C}_2'$ it suffices to show that for every $\boldsymbol{c}_1' \in \mathcal{C}_1'$, $M'\boldsymbol{c}_1' \in \mathcal{C}_2'$. By assumption, such a codeword $\boldsymbol{c}_1' = \boldsymbol{c}_1 \otimes \boldsymbol{y}$ for some codeword $\boldsymbol{c}_1 \in \mathcal{C}_1$, and moreover $\boldsymbol{c}_2 := M\boldsymbol{c}_1 \in \mathcal{C}_2$. So,

$$M'\boldsymbol{c}_1' = (M \otimes I_r)(\boldsymbol{c}_1 \otimes \boldsymbol{y}) = (M\boldsymbol{c}_1) \otimes (I_r\boldsymbol{y}) = \boldsymbol{c}_2 \otimes \boldsymbol{y} \in \mathcal{C}_2' \ ,$$

where the second equality uses the mixed-product property of Kronecker products. The claim follows. The forward direction of Item 2 follows from a very similar argument with $P \in \mathcal{P}_n(\mathbb{F}_q)$ and $P' := P \otimes I_r \in \mathcal{P}_{nr}(\mathbb{F}_q)$ in place of M and M', respectively.

Now, we prove the backward direction of the items, in which we assume that $\mathcal{C}_1'$ and $\mathcal{C}_2'$ are equivalent. For Item 1, we have by assumption that there exists $M' = D'P' \in \mathcal{M}_{nr}(\mathbb{F}_q)$ such that $M'\mathcal{C}_1' = \mathcal{C}_2'$ where $D' \in \mathbb{F}_q^{nr \times nr}$ is a full-rank diagonal matrix and where $P' \in \mathcal{P}_{nr}(\mathbb{F}_q)$. Then by identifying elements $(i, j) \in [n] \times [r]$ with indices $(i-1)r + j \in [nr]$ and applying Lemma 4.1, there exists a submatrix $P = (P')_{I,J}$ of P' such that $P \in \mathcal{P}_n(\mathbb{F}_q)$, $J := \{(i-1)r + j_i : i \in [n]\}$, and $I := \{(\sigma(i) - 1)r + j_i' : i \in [n]\}$ for some permutation $\sigma \in S_n$ and $j_1, \dots, j_n, j_1', \dots, j_n' \in [r]$. Let $M := (M')_{I,J}$ and note that $M \in \mathcal{M}_n(\mathbb{F}_q)$.

Define the punctured codes

$$\widehat{\mathcal{C}}_1 := \{\boldsymbol{c}_J : \boldsymbol{c} \in \mathcal{C}_1'\} \ , \quad \widehat{\mathcal{C}}_2 := \{\boldsymbol{c}_I : \boldsymbol{c} \in \mathcal{C}_2'\} \ ,$$

and notice that $M\widehat{\mathcal{C}}_1 = \widehat{\mathcal{C}}_2$, which in particular implies that $\widehat{\mathcal{C}}_1 \overset{L}{\approx} \widehat{\mathcal{C}}_2$. Because linear equivalence is a transitive relation on codes, to show that $\mathcal{C}_1 \overset{L}{\approx} \mathcal{C}_2$, it therefore suffices to show that $\mathcal{C}_1 \overset{L}{\approx} \widehat{\mathcal{C}}_1$ and $\mathcal{C}_2 \overset{L}{\approx} \widehat{\mathcal{C}}_2$. Let $D_1 := \mathrm{diag}(y_{j_1'}, \dots, y_{j_n'})$ and let $D_2 := \mathrm{diag}(y_{j_1}, \dots, y_{j_n})$. Then $D_1\mathcal{C}_1 = \widehat{\mathcal{C}}_1$ and $D_2\mathcal{C}_2 = \widehat{\mathcal{C}}_2$, from which it follows that $\mathcal{C}_1 \overset{L}{\approx} \widehat{\mathcal{C}}_1$ and $\mathcal{C}_2 \overset{L}{\approx} \widehat{\mathcal{C}}_2$, as needed.[8]

A similar argument to the one above shows that the backward direction of Item 2 holds. It suffices to run the same argument and note that (1) if $M' = D'P'$ for $D' = I_{nr}$ then $M = I_nP$ for $P \in \mathcal{P}_n(\mathbb{F}_q)$, and (2) using the assumption that $\boldsymbol{y} = \mathbf{1}_r$, we have that $D_1 = D_2 = I_n$. Indeed, (1) implies that $\widehat{\mathcal{C}}_1 \overset{P}{\approx} \widehat{\mathcal{C}}_2$, and (2) implies that $\widehat{\mathcal{C}}_1 = \mathcal{C}_1$ and $\widehat{\mathcal{C}}_2 = \mathcal{C}_2$. The result follows. □

The following corollary is immediate.

Corollary 4.3. *Let q be a prime power, and let $n, k, d, r \in \mathbb{Z}^+$. Then there is a* $\mathrm{poly}(n, r, \log q)$*-time reduction from* LCE *(respectively,* PCE*) on $[n, k, d]_q$ codes to* LCE *(respectively,* PCE*) on $[rn, k, rd]_q$ codes.*

4.2 A Reduction from LCE to UCE

We next give a reduction from LCE_q to PCE_q appearing in [SS13b]. The reduction is to map the input codes $\mathcal{C}_i$ for $i = 1, 2$ to $\mathcal{C}_i' := \mathcal{C}_i \otimes (a)_{a\in\mathbb{F}_q^*}$, which [SS13b] calls the *closure* of $\mathcal{C}_i$. The conference version of [SS13b] does not include a correctness proof, although its full version does (specifically, as Lemma 4 and Theorem 1 in the full version). For completeness, here we give a slightly different proof as the proof of Theorem 4.4. In fact, we only prove correctness for one direction of the reduction since Theorem 4.2 already implies the converse and in fact something stronger; see Corollary 4.5 below.

Theorem 4.4. *Let q be a prime power, let $n, k, d \in \mathbb{Z}^+$, let $\boldsymbol{y} := (a)_{a\in\mathbb{F}_q^*}$, let $\mathcal{C}_1, \mathcal{C}_2$ be $[n, k, d]_q$ codes, and let $\mathcal{C}_1', \mathcal{C}_2'$ be the $[(q-1)n, k, (q-1)d]_q$ codes defined by $\mathcal{C}_1' := \mathcal{C}_1 \otimes \boldsymbol{y}, \mathcal{C}_2' := \mathcal{C}_2 \otimes \boldsymbol{y}$, respectively. If $\mathcal{C}_1 \overset{L}{\approx} \mathcal{C}_2$, then $\mathcal{C}_1' \overset{P}{\approx} \mathcal{C}_2'$.*

Proof. For any $a \in \mathbb{F}_q^*$, multiplication by a induces a permutation on the elements in $\mathbb{F}_q^*$. Therefore, for every $a \in \mathbb{F}_q^*$, there exists a matrix $Q_a \in \mathcal{P}_{q-1}(\mathbb{F}_q)$ such that

$$Q_a\boldsymbol{y} = a\boldsymbol{y} \ . \tag{3}$$

Suppose that $M = DP \in \mathcal{M}_n(\mathbb{F}_q)$ for $D = \mathrm{diag}(a_1, \dots, a_n)$ and $P \in \mathcal{P}_n(\mathbb{F}_q)$ is such that $M(\mathcal{C}_1) = \mathcal{C}_2$. Let $Q \in \mathbb{F}_q^{(q-1)n\times(q-1)n}$ be the block diagonal matrix defined as

$$Q := \begin{pmatrix} Q_{a_1} & 0 & \cdots & 0 \\ 0 & Q_{a_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_{a_n} \end{pmatrix} ,$$

where each Q_{a_i} is as defined in Equation (3). Let $P' := Q \cdot (P \otimes I_n)$, and note that $P' \in \mathcal{P}_{(q-1)n}(\mathbb{F}_q)$. We claim that $P'\mathcal{C}_1' = \mathcal{C}_2'$, which implies the theorem. To prove this, it suffices to show that for every vector $\boldsymbol{c}_1' \in \mathcal{C}_1'$, $P'\boldsymbol{c}_1' \in \mathcal{C}_2'$. By assumption, such a vector $\boldsymbol{c}_1'$ is of the form $\boldsymbol{c}_1' = \boldsymbol{c}_1 \otimes \boldsymbol{y}$ for some $\boldsymbol{c}_1 \in \mathcal{C}_1$, and $\boldsymbol{c}_2 := M\boldsymbol{c}_1$ is contained in $\mathcal{C}_2$.

[8] We note that full-rank diagonal matrices such as D_1 and D_2 are themselves monomial matrices.

Then, using the mixed-product property of the Kronecker product several times,

$$\begin{aligned} P'\boldsymbol{c}_1' &= Q \cdot (P \otimes I_n) \cdot (\boldsymbol{c}_1 \otimes \boldsymbol{y}) \\ &= Q \cdot ((P\boldsymbol{c}_1) \otimes (I_n \boldsymbol{y})) \\ &= Q \cdot ((D^{-1}\boldsymbol{c}_2) \otimes \boldsymbol{y}) \\ &= Q \cdot (D^{-1} \otimes \boldsymbol{y}) \cdot \boldsymbol{c}_2 \\ &= (I_n \otimes \boldsymbol{y}) \cdot \boldsymbol{c}_2 \\ &= \boldsymbol{c}_2 \otimes \boldsymbol{y} \ , \end{aligned}$$

which is contained in $\mathcal{C}_2'$, as needed. The penultimate equality follows from the fact that

$$Q \cdot (D^{-1} \otimes \boldsymbol{y}) = \begin{pmatrix} Q_{a_1}(a_1^{-1}\boldsymbol{y}) & 0 & \cdots & 0 \\ 0 & Q_{a_2}(a_2^{-1}\boldsymbol{y}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_{a_n}(a_n^{-1}\boldsymbol{y}) \end{pmatrix} = I_n \otimes \boldsymbol{y} \ ,$$

where we have used the definition of the matrices Q_{a_i} from Equation (3). □

We now get the following corollary, which says that LCE_q efficiently reduces to UCE_q. We note that although the reduction is the same as in [SS13b], our result is stronger than the corresponding result in [SS13b], which says that LCE_q efficiently reduces to PCE_q instead of UCE_q. This strengthening requires Theorem 4.2, but once equipped with that is just an observation.

Corollary 4.5. *Let q be a prime power, and let $n, k, d \in \mathbb{Z}^+$. Then there is a $\mathrm{poly}(n, q)$-time reduction from* LCE *on $[n, k, d]_q$ codes to* UCE *on $[(q-1)n, k, (q-1)d]_q$ codes.*

Proof. Let $\boldsymbol{y} := (a)_{a \in \mathbb{F}_q^*}$. On input generator matrices G_1 and G_2, the reduction outputs the generator matrices $G_1' := G_1 \otimes \boldsymbol{y}$ and $G_2' := G_2 \otimes \boldsymbol{y}$. The reduction clearly runs in $\mathrm{poly}(n, q)$ time. Furthermore, if $\mathcal{C}(G_1) \overset{L}{\approx} \mathcal{C}(G_2)$ then $\mathcal{C}(G_1') \overset{P}{\approx} \mathcal{C}(G_2')$ by Theorem 4.4, and if $\mathcal{C}(G_1') \overset{L}{\approx} \mathcal{C}(G_2')$ then $\mathcal{C}(G_1) \overset{L}{\approx} \mathcal{C}(G_2)$ by Theorem 4.2, Item 1. □

5 From Codes to Lattices

In this section, we give a simple, efficient reduction from the Signed Permutation Code Equivalence Problem on prime fields $\mathbb{F}_p$ (SPCE_p) with sufficiently large minimum distance to the Lattice Isomorphism Problem (LIP). Using reductions from the previous sections, this also yields an efficient reduction from LCE_p to LIP. We again note that the $p = 2$ case of this reduction is due to Regev [Reg14].

Theorem 5.1. *Let p be a prime and let $n \in \mathbb{Z}^+$. There is a $\mathrm{poly}(n, \log p)$-time reduction from SPCE_p on codes with block length n and minimum distance at least $p^2 + 1$ to* LIP *on lattices of rank n.*

Proof. On input generator matrices G_1 and G_2, the reduction outputs bases of $\mathcal{L}_1 := \mathcal{L}_A(\mathcal{C}_1)$ and $\mathcal{L}_2 := \mathcal{L}_A(\mathcal{C}_2)$, where $\mathcal{C}_1 := \mathcal{C}(G_1)$ and $\mathcal{C}_2 := \mathcal{C}(G_2)$. It is efficient to compute bases of $\mathcal{L}_1$ and $\mathcal{L}_2$. Indeed, they have respective generating sets $(G_1 \mid pI_n)$ and $(G_2 \mid pI_n)$, and these can be converted into bases by computing their Hermite Normal Formal in $\mathrm{poly}(n, \log p)$ time (see, e.g., [MG02]).

It remains to show correctness. First, suppose that $\mathcal{C}_1 \overset{SP}{\approx} \mathcal{C}_2$, and that $S\mathcal{C}_1 = \mathcal{C}_2$ for $S \in \mathcal{SP}_n(\mathbb{F}_p)$. We then get that $S\mathcal{L}_1 = S(\mathcal{C}_1 + p\mathbb{Z}^n) = \mathcal{C}_2 + p\mathbb{Z}^n = \mathcal{L}_2$.

Now, suppose that $\mathcal{L}_1$ and $\mathcal{L}_2$ are isomorphic, and that $O\mathcal{L}_1 = \mathcal{L}_2$ for $O \in \mathcal{O}_n$. By Lemma 2.1 we have that $\pm p\boldsymbol{e}_i \in \mathcal{L}_1$ and $\pm p\boldsymbol{e}_i \in \mathcal{L}_2$ for all $i \in [n]$, and, using the assumption that $\mathcal{C}_1$ and $\mathcal{C}_2$ each have minimum distance at least $p^2 + 1$, that all other non-zero vectors in $\mathcal{L}_1$ and $\mathcal{L}_2$ have norm greater than p. Because multiplication by O is norm-preserving, it follows that for every $i \in [n]$, $O(p\boldsymbol{e}_i) = \pm p\boldsymbol{e}_j$ for some j, and that $O(p\boldsymbol{e}_i) \neq O(\pm p\boldsymbol{e}_{i'})$ for $i \neq i'$ (since in particular O is non-singular). This implies that $O \in \mathcal{SP}_n(\mathbb{R})$.

Let $S \in \mathcal{SP}_n(\mathbb{F}_p)$ be the signed permutation matrix obtained from O by mapping the elements $-1, 0, 1 \in \mathbb{Z}$ to the elements $-1, 0, 1 \in \mathbb{F}_p$ entry-wise in the natural way. We claim that $S\mathcal{C}_1 = \mathcal{C}_2$. Indeed, for every $\boldsymbol{x} \in \mathcal{C}_1$ we have that $O\boldsymbol{x} \in \mathcal{L}_2$, $S\boldsymbol{x} = O\boldsymbol{x} \bmod p\mathbb{Z}^n$,[9] and $\mathcal{C}_2 = \mathcal{L}_2 \bmod p\mathbb{Z}^n$. So, $S\boldsymbol{x} \in \mathcal{C}_2$ as needed. □

We now get the following corollary, which is our main result relating the complexity of (Linear) Code Equivalence and the Lattice Isomorphism Problem.

Corollary 5.2. *Let p be a prime, and let $n, k \in \mathbb{Z}^+$. Then there is a* $\mathrm{poly}(n,p)$*-time reduction from* LCE_p *on* $[n,k]_p$ *codes to* LIP *on lattices of rank R, where $R := 5n$ if $p = 2$ and $R := (p-1)(p+2)n = (p^2+p-2)n$ for $p > 2$.*

Proof. If $p = 2$, apply Corollary 4.3 with $r := 5$ to reduce LCE_2 on $[n,k]_2$ codes to LCE_2 on $[5n,k,5]_2$ codes, and then reduce to LIP using Theorem 5.1 (where we note that $5 > p^2$ and that $\mathrm{LCE}_2 = \mathrm{SPCE}_2$).

If $p > 2$, first apply Corollary 4.3 with $r := p+2$ to reduce LCE_p on $[n,k]_p$ codes to LCE_p on $[(p+2)n, k, p+2]_p$ codes. Second, apply Corollary 4.5 to reduce these LCE_p instances to UCE_p instances on $[dn,k,d]_p$ codes, where $d := (p-1)(p+2) = p^2+p-2 > p^2$. Third, apply Lemma 2.10 to reduce these UCE_p instances to SPCE_p instances on $[dn,k,d]_p$ codes. Finally, apply Theorem 5.1 to reduce these SPCE_p instances to LIP on lattices of rank $R = dn$. □

5.1 Examples Showing Non-Reductions

We conclude this section with examples showing non-reductions. Specifically, we show that the simple "Construction-A" reduction in Theorem 5.1 does not (directly) work as a reduction from either of PCE_p of LCE_p to Lattice Isomorphism, which motivates reducing from SPCE_p instead.

We first show that the reduction does not work for PCE_p. Let $\mathcal{C}_1, \mathcal{C}_2 \subset \mathbb{F}_3^{10}$ be the dimension-1 codes with generator matrices $G_1 := \begin{pmatrix}1\\1\end{pmatrix} \otimes \mathbf{1}_5$ and $G_2 := \begin{pmatrix}1\\-1\end{pmatrix} \otimes \mathbf{1}_5$, respectively. Then $\mathcal{C}_1, \mathcal{C}_2$ have minimum distance greater than 9, $\mathcal{C}_1 \overset{P}{\not\approx} \mathcal{C}_2$, and $\mathcal{L}_A(\mathcal{C}_1)$ is isomorphic to $\mathcal{L}_A(\mathcal{C}_2)$. The fact that $\mathcal{C}_1 \overset{P}{\not\approx} \mathcal{C}_2$ follows from the definition of Permutation Code Equivalence and the fact that all codewords in $\mathcal{C}_1$ (but not $\mathcal{C}_2$) have coordinates with consistent signs. The fact that $\mathcal{L}_A(\mathcal{C}_1)$ and $\mathcal{L}_A(\mathcal{C}_2)$ are isomorphic follows by noting that there is a signed permutation that maps $\mathcal{C}_1$ to $\mathcal{C}_2$ (and fixes $3\mathbb{Z}^{10}$).

We now show that the reduction does not work for LCE_p. Let $\mathcal{C}_1, \mathcal{C}_2 \subset \mathbb{F}_5^{26}$ be the dimension-1 codes with generator matrices $G_1 := \begin{pmatrix}1\\1\end{pmatrix} \otimes \mathbf{1}_{13}$ and $G_2 := \begin{pmatrix}1\\2\end{pmatrix} \otimes \mathbf{1}_{13}$, respectively. Then $\mathcal{C}_1, \mathcal{C}_2$ have minimum distance greater than 25, $\mathcal{C}_1 \overset{L}{\approx} \mathcal{C}_2$, and $\mathcal{L}_A(\mathcal{C}_1)$ is not isomorphic to $\mathcal{L}_A(\mathcal{C}_2)$. The fact that $\mathcal{C}_1$ and $\mathcal{C}_2$ are linearly equivalent follows from the definition of Linear Code Equivalence. The fact that $\mathcal{L}_A(\mathcal{C}_1)$ and $\mathcal{L}_A(\mathcal{C}_2)$ are not isomorphic follows by noting that $\mathcal{L}_A(\mathcal{C}_1)$ contains a vector of norm $\sqrt{26}$ (specifically, $\mathbf{1}_{26}$), whereas $\mathcal{L}_A(\mathcal{C}_2)$ contains no vector of norm $\sqrt{26}$. This latter fact follows by noting that all vectors in $\mathcal{L}_A(\mathcal{C}_2) \setminus 5\mathbb{Z}^{26}$ have all non-zero coordinates, and half of these coordinates have magnitude at least 2.

The necessity of a minimum distance lower bound. Finally, we give an example found by Ran and van Woerden [RvW24] of a pair of self-dual $[6,3]_5$ codes $\mathcal{C}_1, \mathcal{C}_2$, each with distance less than p^2+1, such that $\mathcal{C}_1 \overset{SP}{\not\approx} \mathcal{C}_2$, but such that $\mathcal{L}_A(\mathcal{C}_1)$ and $\mathcal{L}_A(C_2)$ are isomorphic. This shows the necessity of having a lower bound on minimum distance in Theorem 5.1.

We will use the following fact relating self-dual codes and lattices. A code $\mathcal{C}$ is self-dual if $\mathcal{C} = \mathcal{C}^\perp$, and a lattice $\mathcal{L}$ is self-dual if $\mathcal{L} = \mathcal{L}^*$, where $\mathcal{L}^* := \{\boldsymbol{y} \in \mathrm{span}(\mathcal{L}) : \forall \boldsymbol{x} \in \mathcal{L}, \langle \boldsymbol{x}, \boldsymbol{y}\rangle \in \mathbb{Z}\}$. The $p = 2$ case of the fact is given in [CS99, Ch. 7, Sec. 2, Thm. 2., Item ii], and this generalizes to larger prime p.

[9]We interpret $\bmod p\mathbb{Z}^n$ as a map from $\mathbb{Z}^n$ to $\mathbb{F}_p^n$, and note that $\boldsymbol{y} \bmod p\mathbb{Z}^n$ is equivalent to applying the mod p operation coordinate-wise to $\boldsymbol{y}$.

Fact 5.3. *Let p be a prime, and let $\mathcal{C} \subseteq \mathbb{F}_p^n$ be a self-dual code. Then $\mathcal{L} := \frac{1}{\sqrt{p}} \cdot \mathcal{L}_A(\mathcal{C})$ is a self-dual lattice.*

The example is as follows. Let

$$G_1 := \begin{pmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \end{pmatrix}, \qquad G_2 := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & 4 & 2 \\ 1 & 3 & 3 \\ 2 & 2 & 4 \end{pmatrix},$$

$\mathcal{C}_1 := \mathcal{C}(G_1)$, and $\mathcal{C}_2 := \mathcal{C}(G_2)$. We note that both $\mathcal{C}_1$ and $\mathcal{C}_2$ are over $\mathbb{F}_5$, are self-dual, and appear (specified by the generator matrices G_1, G_2 above) in the database of self-dual codes created by Harada and Munemasa [HM15].[10]

The fact that $\mathcal{C}_1 \overset{SP}{\not\approx} \mathcal{C}_2$ follows by noting that $\mathcal{C}_1$ has minimum distance 2, whereas $\mathcal{C}_2$ has minimum distance 4. On the other hand, $\frac{1}{\sqrt{p}} \cdot \mathcal{L}_A(\mathcal{C}_1)$ and $\frac{1}{\sqrt{p}} \cdot \mathcal{L}_A(\mathcal{C}_2)$ both have rank 6 and are both self-dual lattices by Fact 5.3. But, up to isomorphism there is a single self-dual lattice of rank 6 (see, e.g., [Kin03, Table 3]), and so $\frac{1}{\sqrt{p}} \cdot \mathcal{L}_A(\mathcal{C}_1)$ and $\frac{1}{\sqrt{p}} \cdot \mathcal{L}_A(\mathcal{C}_2)$ must be isomorphic. Scaling by the same factor does not change whether two lattices are isomorphic, and so $\mathcal{L}_A(\mathcal{C}_1)$ and $\mathcal{L}_A(\mathcal{C}_2)$ must also be isomorphic.

6 Isomorphism Problems on Matroids and their Connection to Code Equivalence

In this section, we discuss connections between isomorphism problems on matroids and their connections to code equivalence. We focus on the (dis)similarities between the Linear Matroid Isomorphism (defined formally below) and Code Equivalence.

In Section 3, we used results about when graphic and linear matroids were isomorphic in our main reduction, but we did not study the corresponding computational problems in their own right. We now define these problems, more or less following the definitions given in [RS11].

Definition 6.1 (Graphic Matroid Isomorphism)**.** The Graphic Matroid Isomorphism (GMI) Problem is defined as follows. Given two graphs $G_1 = (V, E_1)$ and $G = (V, E_2)$ as input, decide whether $M(G_1)$ is isomorphic to $M(G_2)$.

Definition 6.2 (Linear Matroid Isomorphism)**.** For a field $\mathbb{F}$, the Linear Matroid Isomorphism Problem (LMI$_\mathbb{F}$) is defined as follows. Give two matrices $A_1, A_2 \in \mathbb{F}^{k \times n}$ for some $k, n \in \mathbb{Z}^+$, decide whether $M[A_1]$ is isomorphic to $M[A_2]$.

First, we note that combining Theorem 3.1, the converse to Theorem 3.1 that we remark on immediately below it, and Lemma 3.2 yields a reduction from Graph Isomorphism to Graphic Matroid Isomorphism. (We emphasize that without the 3-connectedness assumption in Theorem 3.1, it is *not* true that two graphs need be isomorphic if their corresponding graphic matroids are.) Second, we note that Theorem 2.5 gives a reduction from GMI to LMI$_\mathbb{F}$ for any field $\mathbb{F}$. Combining these results implies that GMI and LMI$_\mathbb{F}$ over any field $\mathbb{F}$ are GI-hard. In what follows, we discuss the complexity of LMI itself and its connection to code equivalence.

6.1 The relationship between CE and LMI

The definition of a linear matroid $M[A]$ from a matrix A is similar to the parity-check definition of a code $\mathcal{C}^\perp(A)$, and so it is natural to ask how the two objects are related. More specifically, it is natural to ask how CE and LMI are related.

[10]Specifically, these are the only two codes listed in [HM15] over $\mathbb{F}_5$ with block length $n = 6$.

Question 6.3. *Let A_1 and A_2 be matrices. What is the relationship between the codes $\mathcal{C}^\perp(A_1)$ and $\mathcal{C}^\perp(A_2)$ being equivalent and the linear matroids $M[A_1]$ and $M[A_2]$ being isomorphic?*

To make this question precise, we need to specify what variant of code equivalence we mean and what field $\mathbb{F}$ we are working over. However, it is interesting for essentially any choices of these things.

Equivalent matroid representations. Oxley [Oxl11, Chapter 6.3] defines two representations A and A' of a linear matroid to be *projectively equivalent* if there exist an invertible matrix U and a non-singular diagonal matrix D such that $UAD = A'$. He defines two representations to be *equivalent* if it is possible to apply a field automorphism to each entry in UAD for such U and D to obtain A' (the same field automorphism is applied to each entry). We note that projective equivalence is stronger than equivalence. (In fact, Oxley *defines* both terms in terms of applying a sequence of operations, but then proves that these definitions are equivalent to the descriptions given above in [Oxl11, Propositions 6.3.12 and 6.3.13]; we stated Proposition 6.3.12 as Theorem 3.3 in the previous section.)

Relating equivalent representations and equivalent codes. We note the close correspondence between projective equivalence (respectively, equivalence) of matroid representations and linear equivalence (respectively, semi-linear equivalence) of codes. Indeed, the only difference is that there is no permutation matrix in the definition of (projective) equivalence, i.e., (projectively) equivalent representations generate *the same* matroid, not just isomorphic matroids. However, if we extend the definitions in the natural way to isomorphisms, we recover the definitions of linear and semi-linear code equivalence.

Define two linear matroids M_1 and M_2 represented by matrices A_1 and A_2 to be *linearly isomorphic* if there exists a matrix A_1' such that (1) A_1 and A_1' are projectively equivalent representations of M_1, and (2) $A_1'P = A_2$ for a permutation matrix P. Define M_1 and M_2 to be *semi-linearly isomorphic* in the same way, but with "projectively equivalent" replaced with "equivalent" in condition (1). Then $M[A_1]$ and $M[A_2]$ are linearly isomorphic (respectively, semi-linearly isomorphic) if and only if $\mathcal{C}^\perp(A_1)$ and $\mathcal{C}^\perp(A_2)$ are linearly equivalent (respectively, semi-linearly equivalent). In fact, we already formalized the linear equivalence part of this in Theorem 3.3 and Corollary 3.4 in the previous section but without phrasing it this way.

6.2 Known Hardness of LMI

Oxley and Welsh [OW02] use a result by Khachiyan [Kha95], which proves NP-hardness of a problem he calls Linear Degeneracy. The problem is to decide whether every size-k subset of n rational input points in k dimensions is linearly independent. Specifically, [OW02] notes that this result implies that it is coNP-hard to decide whether the linear matroid generated by $M[A]$ for a (single) matrix $A \in \mathbb{Q}^{k\times n}$ is isomorphic to the uniform matroid $U_{k,n}$ (i.e., the matroid on a ground set of size n whose bases are all subsets of size k).[11] They call the problem of checking whether a linear matroid is isomorphic to $U_{k,n}$ the *Uniform Matroid Isomorphism Problem.*

The work of Rao and Sarma. In [RS11], Rao and Sarma formally define computational problems on matroids and study the complexity of these problems. In particular, they define and study LMI. As part of they work, they note that it is possible to represent $U_{k,n}$ over any field with at least n elements using a Vandermonde matrix, which implies that Uniform Matroid Isomorphism is efficiently reducible to Linear Matroid Isomorphism over such fields. Combining this with [OW02], this implies that $\mathrm{LMI}_{\mathbb{Q}}$ is coNP-hard.

However, [RS11] also claims more. In its abstract, [RS11] claims that LMI is coNP-hard over "polynomially growing fields," and discussion in the paper seems to imply that LMI is coNP-hard on all fields with at least n non-zero elements. (The relevant theorem statement in their work, [RS11, Proposition 3.4], just claims that "LMI is coNP-hard" without explicitly specifying for which fields or giving a proof past the discussion above.)

[11] In fact, they state their result as showing NP-hardness rather than coNP-hardness, presumably under Turing reductions.

In fact, the key result in [Kha95] on which [OW02, RS11] rely, is stated *only* for infinite fields (specifically, for the rationals). So, the coNP-hardness proof for LMI over finite fields in [RS11] at a minimum contains a gap (we thank Raghavendra Rao for confirming this [Rao24]). It appears to be wrong for fields of polynomial order, but likely to be correct for prime fields of sufficiently large exponential order.

The hardness reduction in [Kha95] is a reduction from Subset Sum over the integers, to Linear Degeneracy over the rational numbers (the output Linear Degeneracy instance is in fact integral). However, the reduction seems likely to work more generally as a reduction from Subset Sum over fields $\mathbb{F}$ of size at least roughly n (where n is the number of values in the Subset Sum instance) to Linear Degeneracy over $\mathbb{F}$.

Subset Sum is in fact still NP-hard on prime fields $\mathbb{F}_p$ of sufficiently large exponential order (fields of order $2^{\Omega(n^c)}$ for some constant $c > 0$). One can see this by upper bounding the (largest possible) values output by the standard reduction from 3-SAT to Subset Sum, and then choosing p to be larger than their sum. This ensures that there is no "wrap-around," i.e., subset sums of values are valid over $\mathbb{F}_p$ if and only if they are valid over the integers. On the other hand, Subset Sum is polynomial-time solvable over fields of polynomial order by using a dynamic programming algorithm similar to the one for knapsack on bounded-weight items. So, Khachiyan's reduction only proves vacuous hardness over such fields, and therefore Rao and Sarma's claim about coNP-hardness of LMI over such fields is incorrect.

6.3 A Complexity Discrepancy Between LCE and LMI

Assuming that the coNP-hardness reduction in [RS11] does work for prime fields of order $p = 2^{O(n^c)}$, this shows a gap between the hardness of LMI and LCE over $\mathbb{F}_p$. Indeed, LCE_p for such p is clearly in NP, and is in coAM using a slight variant of a protocol introduced by Petrank and Roth [PR97] (see below). From this it follows that LCE_p is not NP-hard or coNP-hard unless the polynomial hierarchy collapses.

The coAM protocol for LCE_q is as follows (q need not be a prime). On input generator matrices $G_0, G_1 \in \mathbb{F}_q^{n\times k}$, Arthur samples a uniformly random bit b (i.e., samples $b \sim \{0,1\}$), uniformly random $M \sim \mathcal{M}_n(\mathbb{F}_q)$, and uniformly random $U \sim \mathrm{GL}_k(\mathbb{F}_q)$. He then sends MG_bU to Merlin, and asks what b is. The protocol in [PR97] uses the same idea, but is for PCE. This protocol only runs in polynomial time for LCE over finite fields $\mathbb{F}_q$ with $\log q \leq \mathrm{poly}(n)$, and does not work for LCE over infinite fields.

References

[ABC$^+$22] Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, , and Wen Wang. Classic McEliece, 2022. NIST Post-Quantum Cryptography Standardization Project submission. 2

[Bab16] László Babai. Graph isomorphism in quasipolynomial time. In *STOC*, 2016. 3

[BBPS21] Alessandro Barenghi, Jean-François Biasse, Edoardo Persichetti, and Paolo Santini. LESS-FM: fine-tuning signatures from the code equivalence problem. In *PQCrypto*, 2021. 2

[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. 4

[BCGQ11] László Babai, Paolo Codenotti, Joshua A. Grochow, and Youming Qiao. Code equivalence and group isomorphism. In *SODA*, 2011. 4

[BCK23] Alessandro Budroni, Jesús-Javier Chi-Domínguez, and Mukul Kulkarni. Lattice isomorphism as a group action and hard problems on quadratic forms. *IACR Cryptol. ePrint Arch.*, page 1093, 2023. 2

[BD22] Simeon Ball and James Dixon. The equivalence of linear codes implies semi-linear equivalence. *Des. Codes Cryptogr.*, 90(7):1557–1565, 2022. 4

[Beu20] Ward Beullens. Not enough LESS: an improved algorithm for solving code equivalence problems over $\mathbb{F}_q$. In *SAC*, 2020. 4

[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. 2

[BM23] Jean-François Biasse and Giacomo Micheli. A search-to-decision reduction for the permutation code equivalence problem. In *ISIT*, 2023. 5

[BMPS20] Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. LESS is more: Code-based signatures without syndromes. In *AFRICACRYPT*, volume 12174, pages 45–65. Springer, 2020. 2

[BOS19] Magali Bardet, Ayoub Otmani, and Mohamed Saeed-Taha. Permutation code equivalence is not harder than graph isomorphism when hulls are trivial. In *ISIT*, 2019. 5

[CGG17] Karthekeyan Chandrasekaran, Venkata Gandikota, and Elena Grigorescu. Deciding orthogonality in construction-A lattices. *SIAM J. Discret. Math.*, 31(2):1244–1262, 2017. 5

[CS99] John Conway and Neil J. A. Sloane. *Sphere packings, lattices, and groups.* Springer, 1999. 7, 16

[DG23] Léo Ducas and Shane Gibbons. Hull attacks on the lattice isomorphism problem. In *PKC*, 2023. 4, 5

[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 *ASIACRYPT*, 2022. 2

[DSV09] Mathieu Dutour Sikirić, Achill Schürmann, and Frank Vallentin. Complexity and algorithms for computing Voronoi cells of lattices. *Mathematics of Computation*, 78(267):1713–1731, sep 2009. 3, 4, 11, 12

[DvW22] Léo Ducas and Wessel P. J. van Woerden. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In *EUROCRYPT*, 2022. 2

[Got21] Felix Gotti. Matching and Hall's theorem, 2021. Lecture notes. Available at `https://math.mit.edu/~fgotti/docs/Courses/Combinatorial%20Analysis/30.%20Matchings%20and%20Hall%27s%20Theorem/Matching%20and%20Hall%27s%20Theorem.pdf`. 13

[Gro12] Joshua A. Grochow. Matrix isomorphism of matrix lie algebras. In *CCC*, 2012. 3

[HM15] Masaaki Harada and Akihiro Munemasa. Database of self-dual codes, 2015. Available at `https://www.math.is.tohoku.ac.jp/~munemasa/selfdualcodes.htm`. 17

[HR14] Ishay Haviv and Oded Regev. On the lattice isomorphism problem. In *SODA*, 2014. 5

[Kha95] Leonid Khachiyan. On the complexity of approximating extremal determinants in matrices. *Journal of Complexity*, 11(1):138–153, 1995. 18, 19

[Kin03] Oliver D. King. A mass formula for unimodular lattices with no roots. *Mathematics of Computation*, 72(242):839–863, 2003. 17

[KO06] Petteri Kaski and Patric R.J. Östergård. *Classification Algorithms for Codes and Designs.* Springer-Verlag, 2006. 3, 4

[McE78] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory, 1978. DSN Progress Report. 2

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems – a cryptographic perspective*, volume 671 of *The Kluwer international series in engineering and computer science*. Springer, 2002. 15

[OW02] James G. Oxley and Dominic J. A. Welsh. Chromatic, flow and reliability polynomials: The complexity of their coefficients. *Comb. Probab. Comput.*, 11(4):403–426, 2002. 18, 19

[Oxl11] James Oxley. *Matroid Theory*. Oxford University Press, 02 2011. Second Edition. 4, 8, 9, 11, 12, 18

[PR97] E. Petrank and R.M. Roth. Is code equivalence easy to decide? *IEEE Transactions on Information Theory*, 43(5):1602–1604, 1997. 3, 4, 5, 19

[Rao24] Raghavendra Rao, 2024. Personal communication. 19

[Reg14] Oded Regev, 2014. Personal communication. 3, 5, 15

[RS11] B. V. Raghavendra Rao and Jayalal Sarma. On the complexity of matroid isomorphism problem. *Theory Comput. Syst.*, 49(2):246–272, 2011. Preliminary version in CSR 2009. 5, 17, 18, 19

[RvW24] Lars Ran and Wessel van Woerden, 2024. Personal communication. 5, 16

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

[SS13a] Nicolas Sendrier and Dimitris Simos. How easy is code equivalence over $\mathbb{F}_q$? In *International Workshop on Coding and Cryptography (WCC)*, 2013. 4

[SS13b] Nicolas Sendrier and Dimitris E. Simos. The hardness of code equivalence over $\mathbb{F}_q$ and its application to code-based cryptography. In *PQCrypto*, 2013. Full version available at `https://inria.hal.science/hal-00790861v2`. 3, 4, 14, 15

[Tut71] W. T. Tutte. Introduction to the theory of matroids, 1971. RAND Technical Report. 4, 12

[Whi33] Hassler Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55(1):245–254, 1933. 4, 11