Generalizing the Rate Bound of the Hermitian-Lifted Code

Na'ama Nevo

Fall 2022

Abstract

Hermitian-Lifted Codes were first described in a paper by Lopez, Malmskog, Matthews, Piñero-Gonzales, and Wootters, which are advantageous for being locally recoverable and evaluated on a large set of functions. The paper proves that the rate of the code is bounded below by a positive constant for Hermitian Curve on $q = 2^l$. This paper generalizes the theorem to show that the rate of Hermitian-Lifted Codes is bounded below by a positive constant when $q = p^l$ for any odd prime p.

1 Introduction

Codes are locally recoverable if an index of a codeword can be recovered by a subset of the other symbols in the codeword. Locally recoverable codes have many advantages such as protecting against data loss and allowing for data to be accessed in multiple ways, and are therefore ideal to study to find ways to optimize their parameters. Algebraic codes, which are also referred to as codes on curves, use geometric structures to increase the length of codes while maintaining smaller fields. Lifted evaluation codes increase the number of functions the code can be evaluated on, thus increasing the number of codewords.

This paper builds on a theorem from a paper by Lopez, Malmskog, Matthews, Piñero-Gonzales, and Wootters, which describes a specific code construction called the Hermitian-Lifted Code. The Hermitian curve is $y^{q+1} = x^q + x$ where $q = p^l$ for any prime p and any positive integer l. The paper proves that when $q = 2^l$, the rate of the Hermitian Lifted code is bounded below by a positive constant independent of the size of l. This paper follows a very similar proof to extend the theorem to show that when $q = p^l$ for all primes p, the rate of the Hermitian-Lifted code is still bounded below by a positive constant, rather than tending to 0.

Although the theorem proves a lower bound that is very close to 0, this result is significant because it shows that the rate of this code is better than the rate of the one-point code $\mathcal{C}$, which tends to 0 as q increases. This implies that the set of functions included in the Hermitian-Lifted code that are not in the set of functions for $\mathcal{C}$ is actually very large, significant enough to increase the lower bound of the rate of the function. Additionally, the lower bound found in this theorem is not a tight bound. The bound calculated in the theorem finds a sufficient number of functions that yields a rate with a positive bound, but does not aim to find all the functions or the actual dimension of the code. This discovery is significant because the Hermitian-Lifted Code improves the rate of the Hermitian one-point code while maintaining the same locality and availability.

The rest of the paper will first cover the necessary background on codes and algebraic geometry codes, and then introduce the construction for Hermitian-Lifted Codes described in [5]. In Section 4, we will prove the generalized version of the theorem proved in [5] by following a very similar proof structure to the one used to prove the $q = 2^l$ case. Finally, we will show an example of the proof in the $p = 3$ case and an example of a good monomial.

2 Important Background and Notation

2.1 Error Correcting and Detecting Codes

Error Correcting Codes are algorithms used to maximize accurate data transmission in networks between a sender and receiver. Data transmission is often hindered by random errors which can flip bits of a message to the wrong symbols. The goals of error correcting codes is both to be able to detect as many errors as possible and to be able correct as many errors as possible. Coding Theory investigates the limits of how effective and efficient a code can be, motivating the continuous search for new strategies of encoding messages that yield optimal detecting and correcting abilities.

Error detecting codes are prevalent in every day life. One of the simplest examples that is commonly seen is the ISBN code used to identify books [6]. Books that were published before 2007 are identified by a unique 10-digit sequence (13-digit ISBN codes were introduced after 2007). However, only the first 9 numbers of the ISBN code provide information about the book. The last digit is calculated based on the first 9 digits by the following formula: for the first nine digits of an ISBN code $a_1a_2a_3 \dots a_9$, the last digit is calculated as $a_{10} = a_1 + 2a_2 + 3a_3 + \cdots + 9a_9 \mod 11$. If $a_{10} = 10$, then it is represented as the character X.

This method is able to detect a single error in an ISBN code, where one digit is mistyped with the wrong number. If the wrong digit is the last one, then the formula fails since there is only one correct number between 1 and 10 that satisfies the formula and it will be evident that there is a mistake. Additionally, if any of the first 9 digits are accidentally switched, the mistake would also be detected: if the digit a_i is accidentally typed as some other digit a'_i for $i < 10$, then the sum before modding out by 11 differs from the correct sum by $i(a'_i - a_i)$. Since both i and $a'_i - a_i$ have values between 1 and 9, then the product cannot possibly be divisible by 11. Therefore, the value of a_{10} would be incorrect and the formula would fail.

Although one incorrect digit can always be detected, the same is not true for two or more mistakes. While the difference from the real sum cannot be divisible by 11 when there is only one mistake, it is possible for the difference to have a factor of 11 when there are two or more mistakes. This would result in the same value modulo 11, so the equation would be satisfied despite the mistakes. Additionally, when a single mistake occurs and is detected, it is not clear which digit caused the error so the code cannot be corrected.

A different intuitive strategy for detecting errors is to send a message multiple times. Whenever the same symbol is sent in one position in each instance of the message, it is safe to assume that the symbol is correct. If the symbol in one position varies between the message instances, then an error is detected. The repetition strategy also allows errors to be corrected: the symbol that appears the most times in each position would be chosen as the most probable correct symbol. The main disadvantage of sending a message many times is that it is an inefficient use of space. Messages would take a long time to send and most of the transmitted information would be redundant.

Different types of codes vary in how many errors they can detect, how many errors they can correct, and the amount of extra information that needs to be transmitted. The challenge of coding theory is to find codes that can be most effective at correcting errors while also minimizing the amount symbols that need to be transmitted. While the ISBN code can only detect one error and correct none, it is useful for recognizing typing mistakes, because ISBN codes are relatively short and can easily be recopied if there is a mistake. As messages get longer and the likelihood of mistakes increases, it is ideal to have codes that are both efficient and are capable of correcting the message in the event that there are many errors.

2.2 Linear Codes

We will now introduce the formal definition of a code. A *code* $\mathcal{C}$ over an alphabet A is a subset of A^n for a positive integer n. In other words, $\mathcal{C} \in A^n$. In most of this paper, the alphabet A will be a finite field. Every element of code is called a *codeword*, and the *length* of the code is n, which is the number of symbols in each codeword.

For any two codewords $\vec{x} = (x_1, x_2, \ldots, x_n)$ and $\vec{y} = (y_1, y_2, \ldots, y_n)$, the *Hamming Distance* between any $\vec{x}$ and $\vec{y}$ is defined as $d(\vec{x}, \vec{y}) = \#\{i | x_i \neq y_i\}$, which is the number of spots that $\vec{x}$ and $\vec{y}$ have differing symbols. Every code has a *minimum distance* d, which is the smallest Hamming Distance between any two distinct codewords in the code. The minimum distance of a code determines how many errors in a message the code can correct. If a message $m \in A^n$ is transmitted but $m \notin \mathcal{C}$, then it would be assumed that the intended transmission was the codeword in $\mathcal{C}$ with the smallest Hamming Distance from m. Thus, correcting a message by finding the closest codeword is accurate only if the transmission has at most $\frac{d-1}{2}$ errors.

The *relative minimum distance* of a code is defined as $\delta = d/n$. Since the number of errors in a transmission the code can accurately correct is limited by the minimum distance, it is ideal for the relative minimum distance to be as close as possible to 1 to maximize the correcting abilities of the code.

If A is a finite field, then a code $\mathcal{C}$ is a *linear code* if it is a vector subspace of A^n. Recall that a vector subspace is closed under any linear combinations of elements in the subspace. A vector subspace also has a set of basis elements, which is a subset of the vectors in the subspace that can generate all the vectors in the set through linear combinations. The *minimum weight* in a linear code is the smallest distance between a nonzero codeword and the zero codeword, and the minimum weight is equal to the minimum distance. For a linear code, the *dimension* of the code k is equal to the dimension of the vector subspace, or the number of elements in the basis.

The *information rate* of a code is given by $R = k/n$. Out of the n symbols in a codeword, only k symbols provide information about the message, while the other $n - k$ symbols are transmitted to assist with error detection. Therefore, maximizing the rate of a code minimizes the number of extraneous symbols that need to be transmitted, which increases the efficiency of transmission. A rate as close as possible to 1 is ideal.

Although maximizing both the relative minimum distance and the information rate of a code is ideal, when the two values are maximized they have an inverse relationship. Increasing one causes a decrease in the other, which means that there are trade offs when constructing a code. Depending on the application, a code may be more useful with a higher correcting ability, or it may be more useful with efficient transmission. Discovering different types of codes that optimize each value in different amounts is important to find good codes for a variety of contexts.

Linear codes with length n and dimension k can be represented by a $k \times n$ generator matrix, where each row of the matrix one of the k basis elements of the code's vector space.

Example 1. Consider the linear code $\mathcal{C}$ over the alphabet $A = \mathbb{F}_2 = \{0, 1\}$ with the following generator matrix.

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

It is easy to see that the code $\mathcal{C}$ has length $n = 6$ and dimension $k = 3$. Note that the minimum distance of the set of basis vectors is not equal to the minimum distance of the entire code, which often makes it very difficult to calculate the minimum distance of large codes. Since this example is small and only has 8 codewords, we can use brute force to find the minimum distance. Writing out and comparing all the elements of the code shows that the minimum distance of $\mathcal{C}$ is 3.

Now consider that the message $(1, 1, 0, 1, 0, 1)$ is received. Since this vector is not a linear combination of any of the rows of the generator matrix, then the message cannot be a codeword in $\mathcal{C}$. Therefore, an error must have occurred in the transmission.

A minimum distance of 3 means this code can correct at most $\frac{3-1}{2} = 1$ error in a codeword. Thus, in order to correct the error, we must find a codeword in $\mathcal{C}$ that has a Hamming Distance of 1 from the received message. The codeword $(1, 1, 0, 1, 0, 0)$ is generated by adding the first two rows of the matrix, and has a Hamming Distance of 1 from the received message. Then this codeword is the closest to the message, so the code algorithm would assume that the intended message was $(1, 1, 0, 1, 0, 0)$.

While the minimum distance can be used to correct messages that are not in the code, there is no confirmation that the correction is accurate because the exact number of mistakes in the transmission is unknown. In the event that more errors occured in a transmission than a code is able to correct, then the closest codeword to the transmission would not necessarily be the intended message. Maximizing the minimum distance allows room for more errors to occur and be correctly fixed.

The need for a large minimum distance gives rise to the following question: what is the greatest possible minimum distance a code could have given its parameters? The Singleton Bound provides the answer to this exact question by bounding the minimum distance with respect to the dimension and the length of the code.

Theorem 1. Singleton Bound Let $\mathcal{C}$ be a code with dimension k, length n, and minimum distance d. Then $d \leq n - k + 1$.

Any code with a minimum distance that has the maximum value provided by the Singleton Bound is called a *Maximal Distance Separable* Code, or MDS Code. One example of an MDS Code is the Reed-Solomon Code. The Reed-Solomon Code is one of the most widely used codes, with many applications including CDs and space transmission. In order to define the Reed-Solomon Code, we will use the definition $L_k = \{f \in \mathbb{F}_q[x] | \deg f \leq k\}$.

Definition 1. *Let* $\mathbb{F}_q = \{\alpha_0, \alpha_1, \ldots, \alpha_{q-1}\}$ *and let* $0 < k \leq q$. *Then the Reed-Solomon Code is defined as*

$$RS(k, q) = \{(f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{q-1})) | f \in L_{k-1}\}.$$

The Reed-Solomon Code $RS(k, q)$ is defined over the alphabet $\mathbb{F}_q$, the finite field of size q, and has a length of q and dimension k. Since the minimum distance is equal to the minimum weight, the minimum distance d can be calculated by finding the minimum weight. A function of degree $k-1$ can have at most $k-1$ roots. Since the length of each codeword is q, then the minimum number of nonzero symbols in a codeword is $q-(k-1)=q-k+1$. This implies that $d \geq q-k+1=n-k+1$. By the Singleton Bound, the minimum distance of $d \geq q-k=n-k+1$, so the minimum distance of $RS(k,q)$ satisfies $d=n-k+1$. Therefore, $RS(k,q)$ is an MDS code.

The code allows messages of length k to be sent and encoded. Consider the message $(m_0, m_1, \ldots, m_{k-1})$. Then the function $f \in \mathbb{F}_q[x]$ used to encode the message is $f(x)=m_0+m_1x+\cdots+m_{k-1}x^{k-1}$, and the transmitted codeword would be $(f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{q-1}))$. By solving a linear system of equations, the original message can then be decoded from the results of the transmitted codeword.

Example 2. Consider using the code $RS(4,7)$ to send the message $(3,2,1,4)$. Working over the field $\mathbb{F}_7=\{0,1,2,3,4,5,6\}$, the function used to encode the message is $f=3+2x+x^2+4x^3$. Therefore the codeword will be $(3,3,1,0,3,6,5)$. The minimum distance of $RS(4,7)$ is $d=q-k=7-4=3$. Therefore, it can correct up to 1 error. In the case that an error has occurred, the code can correct the error by solving a system of equations.

The Reed-Solomon code has an optimal relative minimum distance, which makes it a very effective code in transmitting accurate messages across channels. For that reason, it is widely used across a variety of applications. The drawback of an MDS code is that it only allows for very short messages to be sent. In the example of the Reed-Solomon code, the length of the message is limited by the dimension k, which is limited by the size of the field, q. Therefore, to increase the length of messages, the size of the field must be increased as well, which becomes computationally expensive. Since the minimum distance is $q-k+1$, decreasing the dimension increases the minimum distance. However, that simultaneously decreases the length and the number of the messages that can be encoded. In cases where many different types of messages need to be sent, using different types of codes that are not MDS but that have more flexibility with the length of the message and size of the code becomes necessary. This paper will describe algebraic codes that use different strategies to increase the possible number of codewords a code can have.

2.3 Algebraic Geometry Codes

Algebraic Geometry Codes allow a wider variety of codewords to be sent than the Reed-Solomon Code without increasing the size of the field by replacing values in the finite field by points on a curve defined over a finite field. Increasing the size of the field slows down computation time and can lead to storage issues. Whereas the length of a Reed-Solomon code is limited by q, the size of the field, the length of codes on curves is instead limited by the number of solutions on a given curve that are in the field. Additionally, there can be more functions on the curve, allowing for not only longer codewords but more codewords.

The definitions in this section come from Judy Walker's textbook, *Codes and Curves* [6].

2.3.1 Divisors and Riemann Roch Space of Functions

Before defining algebraic geometry codes, we will introduce the necessary components for their construction. Algebraic geometry codes are constructed from a curve and field, which together are used to generate a set of points and a set of functions that build specific codewords. The curves used must be *nonsingular projective plane curves*, which we will define next.

Definition 2. For a field k, the *projective plane* $\mathbb{P}^2(k)$ is

$$\mathbb{P}^2(k) = (k^3 \setminus \{(0,0,0)\})/\sim,$$

where $(X_0, Y_0, Z_0) \sim (X_1, Y_1, Z_1)$ if and only if there is some nonzero element α in k that satisfies $X_1 = \alpha X_0$, $Y_1 = \alpha Y_0$, and $Z_1 = \alpha Z_0$.

An equivalence class of points in the projective plane is written as $(X_0 : Y_0 : Z_0)$, which includes all the points that are a nonzero element in k multiple of (X_0, Y_0, Z_0). A projective plane curve in this paper refers to the curve $F(X, Y, Z) = 0$ where F is a polynomial in $k[X, Y, Z]$ for a field k. Let K be a field that contains the field k as a subfield. A *K-rational point* over the curve C is a point $(X_0 : Y_0 : Z_0) \in \mathbb{P}^2(K)$ that is a solution to the projective plane curve, satisfying $F(X_0, Y_0, Z_0) = 0$. We will denote the set of all K-rational points as $C(K)$.

The requirement for curves on codes is that they are nonsingular. A *singular point* on a curve is a point where the function and all of its partial derivatives are equal to 0. More specifically, if $F(X, Y, Z) \in k[X, Y, Z]$, then $(X_0 : Y_0 : Z_0)$ is a singular point if $F(X_0 : Y_0 : Z_0) = 0$, $F_X(X_0 : Y_0 : Z_0) = 0$, $F_Y(X_0 : Y_0 : Z_0) = 0$, and $F_Z(X_0 : Y_0 : Z_0) = 0$. A *nonsingular* or *smooth* curve is a curve that does not have any singular points. The smooth condition of a curve is necessary for many useful theorems about curves over finite fields because a smooth curve does not have any nodes or self intersections. Therefore, nonsingular curves have many advantages over singular curves when used in codes on curves.

Definition 3. Let $f(x, y)$ be a function of degree d in $k[x, y]$ where k is a field. Then the *homogenization of* f is the function $F(X, Y, Z) = Z^d f(X/Z, Y/Z)$.

The homogenization of a function results in a function where every monomial has the same total degree. Setting $Z = 0$ in a homogenized function leaves only the monomials from the original function with the highest degree, which is the same process used to find the limit of a function as it approaches infinity. A point (X_0, Y_0, Z_0) where $Z_0 = 0$ is thus called a *point at infinity*, denoted P_∞. Every other point is called an *affine* point.

Definition 4. (Divisor) Let C be a curve over $\mathbb{F}_q$, and let Q be the set of points on C over $\mathbb{F}_q$. Then a *divisor* D on C over $\mathbb{F}_q$ is an element of the free abelian group with basis Q.

In other words, a divisor is a linear combination of the elements in Q. If all the coefficients of D are nonnegative, then we call D and *effective* divisor and say that $D \geq 0$. The *support* of D, denoted supp(D), is the set of Q which have a nonzero coefficient in D. The *intersection divisor* of two curves C and C', both over $\mathbb{F}_q$, is the sum of all their points of intersection, denoted $C \cap C'$.

Definition 5. Let $F[X, Y, Z]$ be the polynomial which defines the nonsingular projective plane curve C over $\mathbb{F}_q$. Then the *field of rational functions on C* is

$$\mathbb{F}_q(C) = \left(\left\{ \frac{g(X,Y,Z)}{h(X,Y,Z)} \,\middle|\, \substack{g,h \in \mathbb{F}_q[X,Y,Z] \\ \text{are homogenous} \\ \text{of the same degree}} \right\} \cup \{0\} \right) / \sim,$$

where $g/h \sim g'/h'$ if and only if $gh' - g'h \in \langle F \rangle \subset \mathbb{F}_q[X, Y, Z]$.

The $\langle F \rangle$ refers to the ideal generated by F where $F[X, Y, Z] = 0$.

Definition 6. Let C be a curve over $\mathbb{F}_q$ and let $f = g/h \in \mathbb{F}_q(C)$. Then the *divisor of* f is $\text{div}(f) = \sum P - \sum Q$, where $\sum P$ is the intersection divisor $C \cap C_g$ and $\sum Q$ is the intersection divisor $C \cap C_h$.

In the function $f = g/h$, the intersection divisor $C_g \cap C$ is the zeros of f, while the intersection divisor $C_h \cap C$ is the *poles* of f. The divisor of f can be thought of as the zeros of f minus the poles of f.

Finally, we can combine definitions 4, 5, and 6 to describe the set of functions that forms the Riemann-Roch Space, also called the space of rational functions associated to divisor D. The Reimann-Roch Space is an important component of the construction of codes on curves.

Definition 7. Let D be a divisor of C over $\mathbb{F}_q$. Then the *space of rational functions associate to* D is

$$L(D) = \{f \in \mathbb{F}_q(C) | \text{div}(f) + D \geq 0\} \cup \{0\}.$$

Recall that a divisor is effective, or greater than or equal to 0, when all the coefficients of the points are positive. Since, the coefficients that are negative in $\text{div}(f)$ are the poles of f, then the set $L(D)$ contains all the functions that have at most D poles in order for the negative coefficients to cancel out.

2.3.2 Codes on Curves

Our original definition for the Reed-Solomon code was

$$RS(k, q) = \{(f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{q-1})) | f \in L_{k-1}\}.$$

We can redefine this code using the new definitions introduced above. The Reed-Solomon code can also be defined as

$$RS(k, q) = \{(f(P_1), \ldots, f(P_{q-1})) | f \in L((k-1)P_\infty)\},$$

where $L((k-1)P_\infty)$ is the space of rational functions associated to the divisor $(k-1)P_\infty$. The set $L((k-1)P_\infty)$ includes all functions f which have at most $k-1$ poles at P_∞. This is equivalent to including all functions that have degree at most $k-1$. Therefore, the space of rational functions associated to D when we set $D = (k-1)P_\infty$ is equal to L_{k-1}, which maintains the definition of the Reed-Solomon code.

As mentioned before, the Reed-Solomon code optimizes the minimum distance at the expense of allowing long messages to be transmitted. We now introduce Goppa's code construction, which generalizes the Reed-Solomon code to allow for longer codewords and messages. We will now be shifting the notation used in previous sections and will let X be a projective, nonsingular plane curve over $\mathbb{F}_q$ and C be the code. Let D be a divisor on X. For a curve X, defined over field k, and extension field K, $X(K)$ denotes the set of points on X defined over the field K. Then let $\mathcal{P} = \{P_1, \ldots, P_n\} \in X(\mathbb{F}_q)$ be a set of n distinct rational points on X over $\mathbb{F}_q$. Now we can finally define an algebraic geometry code.

Definition 8. Given nonsingular projective plane curve X over field $\mathbb{F}_q$, a set of points $\mathcal{P} \subset X(\mathbb{F}_q)$, and divisor D on X where $\mathcal{P} \cap \text{supp}(D) = \emptyset$, the *algebraic geometry code* associated with X, $\mathcal{P}$, and D is

$$C(X, \mathcal{P}, D) = \{(f(P_1), \ldots, f(P_n)) | f \in L(D)\} \subset \mathbb{F}_q^n.$$

Algebraic geometry codes are linear codes. In Goppa's code construction, the length of the code is no longer limited to the size of the field q. Instead, the length of the code is equal to the number of points n in the set $\mathcal{P}$, which is restricted by $0 < n \leq |X(\mathbb{F}_q)|$. Therefore, curves that have more rational points and a larger set $X(\mathbb{F}_q)$ allow for longer possible codewords. The size of the Reed-Solomon code is q^k, which is the total number of polynomials with degree at most $k-1$ over the q possible coefficients. In algebraic geometry codes, the number of possible messages that can be sent, or the size of the code, is $|L(D)|$, which is the number of functions in the Reimann-Roch space associated with D.

One of the simplest subsets of algebraic geometry codes is the *one-point code*, which uses a multiple of one rational point as its divisor.

Definition 9. *([5]) Let X be a smooth curve defined over $\mathbb{F}_q$. Let P be a point on $X(\mathbb{F}_q)$ and m be a natural number. Let $B = \{P_1, P_2, \ldots, P_n\}$ be a set of points on $X(\mathbb{F}_q)$ not containing P, and let $D = P_1 + P_2 + \cdots + P_n$ be a divisor. Let $L(mP)$ be the Riemann-Roch space of functions on X only with poles P with order at most m. The one-point code $C(D, mP)$ is the set $\{(f(P_1), f(P_2), \ldots, f(P_n)) \in (\mathbb{F}_q)^n : f \in L(mP)\}$.*

The one-point code can be explained in terms of the original definition of algebraic geometry codes: the one-point code is $C(X, \mathcal{P}, mP)$ where m is a natural number and $P \notin \mathcal{P}$. The set $L(mP)$ is the set of rational functions on X which do not have more poles than mP.

Since we want to have codes with long words, we want to use curves that have a maximal number of rational points. The Hasse-Weil bound uses the *genus* of a curve to bound the number of points possible for a curve over a finite field, which can be used to check whether curves are maximal. The genus of a curve is a positive integer measure of complexity of curves. It is calculated by the formula $(d-1)(d-2)/2 = d(d-1)/2$, where d is degree of the curve.

Theorem 2. *(Hasse-Weil Bound) For a smooth, projective curve X with genus g over finit field $\mathbb{F}_q$ with cardinality q, the number of possible points on X over $\mathbb{F}_q$ is*

$$q + 1 - 2g\sqrt{q} \leq |X(\mathbb{F}_q)| \leq q + 1 + 2g\sqrt{q}.$$

Note that the genus is only calculated for nonsingular curves. Having a positive genus increases the upper bound of the number of points on a curve, which is another reason nonsingular curves are ideal over singular curves.

The focus of this paper is on codes on the Hermitian Curve, which is a maximal curve based on the Hasse-Weil Bound. In the next section we will describe properties of the Hermitian Curve that make it an ideal curve to study.

2.4 The Hermitian-Lifted Code

2.4.1 The Hermitian Curve

The Hermitian Curve $\mathcal{H}_q$ is the equation $x^q + x = y^{q+1}$, defined over the field $\mathbb{F}_{q^2}$ where $q = p^l$ for some prime p. We can check that this curve is maximal by using the Hasse-Weil Bound. First, we calculate the genus using the formula. The degree of the $\mathcal{H}_q$ is d, so we get that $g = q(q-1)/2$, and q^2 is the size of the field. The upper bound for the size of $\mathcal{H}_q(\mathbb{F}_{q^2})$ is then

$$q^2 + 1 + 2(q(q-1)/2)\sqrt{q^2} = q^2 + 1 + q^3 - q^2 = q^3 + 1.$$

Next, we need to calculate $\mathcal{H}_q(\mathbb{F}_{q^2})$. By homogenizing the curve, we see that there is one unique point at infinity at $(0:1:0)$. Consider for the rest of the solutions that $Z=1$. Since $X^{q+1}=N(X)\in\mathbb{F}_q$ for all values of $X\in\mathbb{F}_{q^2}$, then there are q^2 options for X with only q corresponding unique solutions to Y, leading to $q(q^2)=q^3$ additional solutions. In total, there are q^3+1 rational points on the Hermitian Curve over $\mathbb{F}_{q^2}$, so $\mathcal{H}_q(\mathbb{F}_q)$ is equal to the upper bound of the Hasse-Weil Bound. Therefore, the Hermitian Curve is maximal. This property makes codes on the Hermitian Curve appealing to study because they have the largest possible length of codewords and also a large genus.

Now we define the one-point code on the Hermitian curve. We take mP_∞ to be the pole divisor. The Riemann-Roch space $L(mP_\infty)$ is given by $\{x^iy^j : 0\leq j\leq q-1, iq+j(q+1)\leq m\}$. Using the notation in the original definition of an algebraic geometry code, the Hermitian one-point code is $C(\mathcal{H}_q,\mathcal{H}_q(\mathbb{F}_{q^2})\setminus\{P_\infty\},mP_\infty)$. We will use the notation $C_{q,m}$ to denote the Hermitian one-point code with divisor mP_∞ and curve $\mathcal{H}_q$.

The length of the Hermitian one-point code is q^3. When $m>2g-2$ then the dimension is equal to $m-g+1$. In general, the dimension satisfies $k=\dim(L(mP_\infty))\geq m-g+1$. The minimum distance is given by $n-m$.

2.4.2 Locality and Availability of Hermitian One-Point Code

Locally recoverable codes are codes that can recover a single erased point in a codeword from a subset of the remaining points. A code C has *locality* r and *availability* t if for each $i\in[n]$ there are t disjoint repair sets for i in C each of size at most r. Locally Recoverable Codes were first introduced by Balaji and Krishnan in [2].

The ability to recover a missing piece of data from the available data is extremely useful for preventing the loss of information. For example, information stored in the cloud is all stored somewhere in physical machines. Assume for simplicity that one symbol is stored on each machine. Although the probability of any single machine being destroyed is low, cloud storage is made up of many machines, making it likely that at least some machines will malfunction. Having the ability to use a small group of other machines to recover the erasure of a single point provides protection against inevitable accidents, mistakes, or natural disasters. Without locally recoverable codes, information would frequently be permanently lost due to random damaging events.

Curves and other geometric objects have structures that allow for convenient recovery sets, such as the Hermitian Curve [3]. In the Hermitian one-point, the recovery sets are the intersections of lines with the curve, where any point on a line can be recovered by the remaining points on the line.

Theorem 3. *([5]) Hermitian one-point code has locality q and availability q^2-1.*

Proof. Let each index i correspond to a point P_i in $H_q(\mathbb{F}_{q^2})\setminus\{P_\infty\}$. For any $\alpha,\beta\in\mathbb{F}_{q^2}$, let $L_{\alpha,\beta}:\mathbb{F}_{q^2}\to(\mathbb{F}_{q^2})^2$ so that $L_{\alpha,\beta}(t)=(\alpha t+\beta,t)$. For any line $IM(L_{\alpha,\beta})$ passing through P_i that is not tangent, then let $R_{i,\alpha}=(H_q(\mathbb{F}_{q^2})\cap IM(L_{\alpha,\beta}))\setminus\{P_i\}$. Note that $|R_{i,\alpha}|=q$ and there are q^2-1 disjoint sets for each i.

Any codeword in $C_{q,m}$ is a linear combination of x^ay^b satisfying $b\leq q-1$ and $aq+b(q+1)\leq q^2-1$. This gives

$$a+b+b\frac{1}{q}\leq q-\frac{1}{q}$$

$$a+b\leq q-\frac{b+1}{q}$$

$$a + b \leq q - 1$$

This gives a univariate polynomial of degree at most $q-1$. This means that if one point is removed there are q points left on the line that can help recover it, which is sufficient. □

The high availability of the Hermitian one-point code along with the fact that it has the maximal number of rational points relative to its genus makes it a very appealing code to study. The Hermitian-Lifted code shares many properties with the Hermitian one-point code, but we will prove that it has the advantage of a higher rate.

3 The Code Construction

The Hermitian-Lifted code is similar to the Hermitian one-point code, but it extends the set of functions that can be used to form codewords. Lifted codes were first introduced by Guo, Kopparty, and Sudan [4]. The definitions in this section come directly from [5], where the Hermitian-Lifted code was first described.

Definition 10. *For polynomials $f \in \mathbb{F}_{q^2}[x,y]$ and $g \in \mathbb{F}_{q^2}[t]$ and for function $L : \mathbb{F}_{q^2}[t] \to \mathbb{F}_{q^2}^2$ we say that $f \circ L$ agrees with g on X if $f(L(t)) = g(t)$ for all $t \in \mathbb{F}_{q^2}$ with $L(t) \in X$.*

Recall that $X = H_q(\mathbb{F}_{q^2}) \setminus \{P_\infty\}$. The function L takes in a t and then outputs the coordinate $(\alpha t + \beta, t)$ for the α and β corresponding to the specific L. So $L(t)$ is a line on the curve H_q over the field $\mathbb{F}_{q^2}$ and plugging that line into f yields $g(t)$ which is a univariate polynomial.

Definition 11. *Given a prime power q, let*

$$\mathcal{F} = \begin{cases} f \in \mathbb{F}_{q^2}[x,y] : & \text{for every } L \in \mathcal{L} \text{ there exists } g \in \mathbb{F}_{q^2}[t] \\ & \quad \text{so that } deg(g) \leq q-1 \text{ and so that } f \circ L \\ & \quad \text{agrees with } g \text{ on } X \end{cases}$$

where $\mathcal{L} = \{L_{\alpha,\beta} : \alpha, \beta \in \mathbb{F}_{q^2}\}$ is the set of all lines of the form $L_{\alpha,\beta}(t) = (\alpha t + \beta, t)$.

In other words, $\mathcal{F}$ is the set of all functions such that for every line L, plugging in $f(L(t))$ yields a univariate polynomial with degree at most $q-1$, which here is called g. Assuming $f \in \mathbb{F}_{q^2}[x,y]$ then $f(L(t))$ will also be over the appropriate field, so the main restriction is that the total degree does not exceed $q-1$ after simplification.

Now we define Hermitian-Lifted Codes.

Definition 12. *Let q be a prime power and let $\mathcal{F}$ be defined as above. Then the Hermitian-Lifted Code $\mathcal{C} \subset (\mathbb{F}_{q^2})^{q^3}$ is the evaluation code*

$$\mathcal{C} = \{(f(x,y))_{(x,y)\in\mathcal{X}} : f \in \mathcal{F}\}.$$

The length of the code is q^3 because $|\mathcal{X}| = q^3$. Each codeword is the vector created by the inputs of $\mathcal{X}$ into a function f, and all the functions f satisfy that their total degree is at most $q-1$.

Horizontal lines are ignored here because they include the point at infinity which is not an evaluation point by construction of $\mathcal{X}$.

It is important to remember the difference between $C_{q,m}$, the Hermitian one-point code, and $\mathcal{C}$, the Hermitian-Lifted code. Both of these have codewords of length q^3 and the inputs are $\mathcal{X} = H_q(\mathbb{F}_{q^2}) \setminus \{P_\infty\}$. Also, both codes have locality q and availability $q^2 - 1$. The difference is the functions f that are valid. In the one-point code $C_{q,m}$, the functions f must satisfy $f \in L(mP_\infty)$, which is polynomials of total degree at most m. In the Hermitian-Lifted Code, the functions must satisfy $f \in \mathcal{F}$, which is all functions with degree at most $q - 1$ after they are parameterized. Although the distinction between these two codes is seemingly insignificant, the main theorem proves that the slight difference in the set of functions $\mathcal{F}$ and $L(mP_\infty)$ is enough to make the rate of each code converge to a different value as q goes to infinity.

4 Main Theorem and Proof

The authors of [5] prove that the rate of the Hermitian-Lifted Code as $q \to \infty$ when $q = 2^l$ is bounded below by a positive constant. The aim of this paper is to extend this theorem to show that the rate of the Hermitian-Lifted Code is also bounded below by a positive constant when $q = p^l$ when p is any odd prime. Recall that the Hermitian Lifted Code $\mathcal{C} \subset \mathbb{F}_{q^2}^{q^3}$ is defined as $\mathcal{C}\{(f(x,y))_{(x,y)\in\mathcal{X}} : f \in \mathcal{F}\}$. The statement of the theorem in [5] is the following:

Theorem 4. *Suppose $q = 2^l$ where $l \geq 2$ and let $\mathcal{C}$ be the Hermitian-Lifted Code. Then the rate of $\mathcal{C}$ is at least* 0.007.

Despite the fact that 0.007 is a very small lower bound, it demonstrates a significant difference between the Hermitian-Lifted code and the Hermitian one-point code, which has a rate that converges to 0 as q increases. Set $m = q^2 - 1$ and represent the Hermitian one-point code as $\mathcal{C}_{q,q^2-1}$. The one-point code is a subset of the lifted code because the set of functions corresponding to each code satisfies $L((q^2 - 1)P_\infty) \subset \mathcal{F}$. Since the one-point code is a subset of the lifted code, then the dimension of the lifted code must be at least the dimension of the one-point code.

Recall that when $m > 2g - 2$, then the dimension of the Hermitian one-point code is equal to $m - g + 1$. In the code C_{q,q^2-1}, we have $m = q^2 - 1$ and $2g - 2 = q^2 - q - 2$, so $m > 2g - 2$. Thus, the dimension of $\mathcal{C}_{q,q^2-1}$ is $\frac{q(q+1)}{2}$ and the length of the code is q^3. Therefore, the rate of Hermitian one-point code $\mathcal{C}$ is at least

$$\frac{q(q+1)}{2q^3} = \frac{q^2}{2q^3} + \frac{q}{2q^3} = \frac{1}{2q} + \frac{1}{2q^2},$$

which clearly converges to 0 as $q \to \infty$. Therefore, finding that the rate of the Hermitian-Lifted code converges to a number greater than 0 proves that the set of functions $\mathcal{F} \setminus L((q^2 - 1)P_\infty)$ is large. The proof of Theorem 4 in [5] follows the appoach of finding a large set of functions included in the lifted code but not in the one-point code that result in a sufficiently large dimension. By following a very similar process of finding a set of functions to the proof in [5], we will prove the following main result.

Theorem 5. *Suppose that $q = p^l$ where p is an odd prime and $l \geq 2$. Then the rate of $\mathcal{C}$ is at least*

$$\frac{.469}{p^4(p-1)(p^3 - p^2 + 1)}.$$

.

Although the bound decreases as p increases, the bound always remains positive. The remainder of this section will be the proof to Theorem 5. First, we will describe the set of functions that will provide a lower bound on the dimension of the code, a set which we call *good monomials*. Next, we will describe the conditions that must be satisfied for a monomial to be considered good, and provide some examples for when $q = 3^l$. Finally, we will count the minimum number of functions that must be good monomials, which will lead to a lower bound on a dimension. The final proof of the bound in Theorem 5 will conclude this section.

4.1 Good Monomials

Before proving the theorem, we will prove some lemmas assuming that $q = p^l$. In this section, we will denote the Hermitian Curve $\mathcal{H}_q$ by $\mathcal{X}$. The goal is to find a large set of monomials, $M_{a,b}(x,y) = x^a y^b \in \mathcal{F}$, that are linearly independent. By finding and counting the number of monomials that are linearly independent, we will find a lower bound on the dimension of the code, and thus also a lower bound on the rate of the code.

For a monomial $M_{a,b}(x,y) = x^a y^b$, the exponents are bounded by $a \leq q-1$ and $b \leq q^2-1$. The bound $a, b \leq q^2 - 1$ follows from the fact that we are working over the field $\mathbb{F}_{q^2}$. The bound on a follows from the reduction of the degrees from the Hermitian Curve. For example, assume $a = q$ and $b = 2$. From the equation of the Hermitian curve, we have $x^q = y^{q+1} - x$, so instead of $x^a y^b = x^q y^2$, we could write it as $(y^{q+1} - x)y^2 = y^{q+3} - y^{q+1}x$, which is a linear combination of monomials with $a \leq q - 1$. Therefore, the conditions $a \leq q - 1$ and $b \leq q^2 - 1$ include all the linearly independent monomials.

The purpose of the following lemma is to prove that the evaluation map of the monomials in the set is injective, which ensures that the size of the set is a lower bound on the dimension of the code.

Lemma 1. *Let $M_{a,b}(x,y) = x^a y^b$. Then the set of vectors $\{(M_{a,b}(x,y))_{(x,y)\in\mathcal{X}} : 0 \leq a \leq q-1, 0 \leq b \leq q^2 - 1\}$ are linearly independent.*

The proof of this lemma for all values of q is in Proposition 5 of [5].

Now that the we have proven that the evaluation map of the monomials is linearly independent, we can find a good set of monomials that will bound the dimension. Before we can define which monomials are good, we will introduce another definition. Given a line of the form $L_{\alpha,\beta}(t) = (\alpha t + \beta, t)$, the following definition gives the equation of the points of the Hermitian curve that intersect with the line.

Definition 13. *Given $\alpha, \beta \in \mathbb{F}_{q^2}$, define*

$$p_{\alpha,\beta}(t) = t^{q+1} + \alpha^q t^q + \alpha t + (\beta + \beta^q) = t^{q+1} + a^q t^q + \alpha t + \gamma.$$

For $g(t) \in \mathbb{F}_{q^2}[t]$, $\bar{g}(t)$ is the remainder when $g(t)$ is divided by $p_{\alpha,\beta}(t)$. Let $deg_{\alpha,\beta}(g) = deg(\bar{g}_{\alpha,\beta}(t))$. Note that $deg_{\alpha,\beta}(g) \leq q$ for all $g \in \mathbb{F}_{q^2}(t)$.

For a line $L_{\alpha,\beta}(t) = (\alpha t + \beta, t)$, we have $M_{\alpha,\beta} \circ L_{\alpha,\beta}$ agrees with polynomial g of degree strictly less than q on $\mathcal{X}$ if and only if $\deg_{\alpha,\beta}(M_{\alpha,\beta} \circ L_{\alpha,\beta}) < q$. Write

$$(M_{\alpha,\beta} \circ L_{\alpha,\beta})(t) = h(t)p_{\alpha,\beta}(t) + g(t)$$

for $\deg(g) \leq q$. Then since $t \in \mathcal{X}$, then t satisfies $0 = t^{q+1} + (\alpha + \beta t)^q + \alpha + \beta t$ so $p_{\alpha,\beta}(t) = 0$. Thus, $M_{\alpha,\beta} \circ L_{\alpha,\beta}$ agrees with $g(t)$. Note that there are $q+1$ values of t for polynomial of degree at most q, so $g(t)$ is a unique polynomial.

For a monomial $M_{a,b}(x, y)$, let $g_{a,b}(t) = M_{a,b}(x, y) \circ L_{\alpha,\beta}(t) = (\alpha t + \beta)^a t^b$. We define a *good monomial* to be a monomial $M_{a,b}(x, y)$ such that $g_{a,b}(t)$ satisfies $\deg_{\alpha,\beta}(g) \leq q - 1$.

What follows is that if $M_{a,b}$ is a good monomial, then $M_{a,b} \in \mathcal{F}$. Recall that $\mathcal{F}$ is defined to be the set of all functions that when parameterized yields a univariate polynomial of degree at most $q - 1$, so this observation follows by the definition of a good monomial.

4.2 Conditions of Good Monomials

The notation for the rest of this section will be as follows. The line $L_{\alpha,\beta}(t) = (\alpha t + \beta, t)$ is a line that goes through $q + 1$ points of the Hermitian curve $\mathcal{X}$ for $\alpha, \beta \in \mathbb{F}_{q^2}$. Also, $\gamma = \beta + \beta^q \in \mathbb{F}_q$. This is not used later in the proof, but is true because $(\beta + \beta^q) = (\beta^q + \beta^{q^2}) = \beta^q + \beta$, which by Fermat's Little Theorem implies that $\gamma \in \mathbb{F}_q$.

Let $\sigma_0, \dots, \sigma_q$ be the roots of $p(t)$. We know there are $q+1$ roots because there are $q+1$ points in the intersection of the line and the Hermitian curve. Thus,

$$p(t) = t^{q+1} + a^q t^q + \alpha t + \gamma = (t - \sigma_0) \cdots (t - \sigma_q) = c_0 t^{q+1} + c_1 t^q + \cdots + c_q t + c_q + 1$$

where $c_k = \sum_{S \subset 0,\dots,q, |S|=k} \prod_{l \in S} \sigma_l$ for $k = 0, \dots, q$. This is just given by the expansion of the product above.

For any $k \geq 0$ we define the element $P_k = \sum_{i=0}^{q} \sigma_i^k$. These values will be used to find a condition for good monomials.

Lemma 2. *Let q be a power of p and let $\alpha, \beta \in \mathbb{F}_{q^2}$. Then $P_{k+1} = -\alpha^q P_k$ if and only if $deg_{\alpha,\beta}(t^k) < q$.*

Proof. Write $t^k = g_k(t)p(t) + \bar{g}_k(t)$ for some polynomial $g_k(t)$ so that the polynomial $\bar{g}_k(t)$ has degree at most q. The goal is to show that $deg(\bar{g}_k(t)) < q$ if and only if $P_{k+1} = -\alpha^q P_k$.

Since $\sigma_0, \dots, \sigma_q$ are the roots of $p(t)$, we have $\bar{g}_k(\sigma_i) = \sigma_i^k$. Thus, we know $q + 1$ values of $\bar{g}_k$ by plugging in each σ_i for $i = \{0, \dots, q\}$. Since $\bar{g}_k$ has degree less than q, we can use Lagrange interpolation to write

$$\bar{g}_k(t) = \sum_{i=0}^{k} \sigma_i^k \prod_{j \neq i} \left(\frac{t - \sigma_j}{\sigma_i - \sigma_j} \right) = \left(\sum_{i=0}^{q} \sigma_i^k \prod_{j \neq i} \frac{1}{\sigma_i - \sigma_j} \right) t^q + r(t) \tag{1}$$

where $deg(r(t)) < q$. Since we are only concerned with checking when the degree of the whole polynomial is less than q, it is enough to check which conditions ensure that the coefficient of t^q is 0. Since

$$p(t) = t^{q+1} + \alpha^q t^q + \alpha t + \gamma = (t - \sigma_0) \cdots (t - \sigma_q)$$

we can take the derivative of both sides and get

$$p'(t) = t^q + \alpha = \sum_{i=0}^{q} \prod_{j \neq i} (t - \sigma_j).$$

Replacing t with σ_i yields

$$p'(\sigma_i) = \sigma_i^q + \alpha = \sum_{i=0}^{q} \prod_{j \neq i} (\sigma_i - \sigma_j). \tag{2}$$

Because σ_i is a root of $p(t)$ then $\sigma_i^q + 1 + \alpha^q\sigma_i^q + \alpha\sigma_i = -\gamma$, which can be factored as $(\sigma_i^q + \alpha)(\sigma_i + \alpha^q) = \alpha^{q+1} - \gamma$. Divide both sides of the equation by $(\sigma_i + \alpha^q)$ and use equation 2 to get

$$\prod_{j\neq i}(\sigma_i - \sigma_j) = \frac{\alpha^{q+1} - \gamma}{\sigma_i + \alpha^q}.$$

Using equation 1 we can calculate the coefficient of t^q in $\bar{g}_k(t)$ to be

$$\sum_{i=1}^{q} \frac{\sigma_i^k(\sigma_i + \alpha^q)}{\alpha^{q+1} - \gamma} = \frac{P_{k+1} + \alpha^q P_k}{\alpha^{q+1} - \gamma}$$

which is equal to zero exactly when $P_{k+1} = -\alpha^q P_k$. Thus, this is our condition for when $deg_{\alpha,\beta}(g) < q$. □

Now that we have a sufficient condition on P_k that gives good monomials, we need to find a condition on k that will satisfy $P_{k+1} = -\alpha^q P_k$. We will find a few more patterns before we find a condition on k.

Lemma 3. *Let q be a power of an odd prime p. For $0 \leq k < q$, $P_k = (-1)^k\alpha^{qk}$ and $P_{kq} = (-q)^k\alpha^k$.*

Proof. Since q is a multiple of p, then $P_0 = q+1 = 1$. Let $1 \leq k < q$. We get $P_k = -\alpha^q P_{k-1}$ from Lemma 2. By induction, we get that $P_k = (-1)^k\alpha^{qk}$. Because we are working over $\mathbb{F}_{q^2}$,

$$P_{kq} = \sum_{i=0}^{q} \sigma_i^{qk} = \left(\sum_{i=0}^{q} \sigma_i^k\right)^q = (\alpha^{qk})^q = (-1)^k\alpha^k.$$

Therefore, $P_k = (-1)^k\alpha^{qk}$ and $P_{kq} = (-1)^k\alpha^k$. □

Using Lemmas 2 and 3 above, we can find a useful relationship between the elements of the following matrix, which will be useful for the upcoming theorems.

$$\begin{pmatrix} P_0 & P_q & \cdots & P_{(q-1)q} \\ P_1 & P_{q+1} & \cdots & P_{(q-1)q+1} \\ \vdots & \vdots & & \vdots \\ P_{q-1} & P_{2q-1} & \cdots & P_{q^2-1} \end{pmatrix}.$$

For a root σ of $p(t) - t^{q+1} + \alpha^q t^q + \alpha t + \gamma$, we have $-\sigma^{q+1} = \alpha^q\sigma^q + \alpha\sigma + \gamma$. By multiplying both sides of the equation by σ^{k-q-1} we get $-\sigma^k = \alpha^q\sigma^{k-1} + \alpha\sigma^{k-q} + \gamma\sigma^{k-q-1}$. Thus, we obtain that the values of P_k satisfy the recurrence relation

$$-P_k = \alpha^q P_{k-1} + \alpha P_{k-q} + \gamma P_{k-q-1}. \tag{3}$$

Based on this formula, the (i, j) entry of the matrix is determined by the $(i-1, j)$, $(i, j-1)$, and $(i-1, j-1)$ entries of the matrix. As a result, the entire matrix is determined by the first row and first column of the matrix. Therefore, every 2×2 submatrix M must satisfy the recurrence relation

$$-M_{22} = \alpha^q M_{12} + \alpha M_{21} + \gamma M_{11}. \tag{4}$$

The next step of the proof will be to show that the above matrix can be written as a product of a specific product of matrices, using the fact that it is sufficient to show that the first row and column are equal and every submatrix satisfies the recurrence relation. The following definitions will provide the tools for defining the product.

Definition 14. *Let $A = [a_{ij}]$ be an $r \times s$ matrix and $B = [b_{ij}]$ an $m_1 \times m_2$ matrix. The Kronecker product of A and B is the $rm_1 \times sm_2$ matrix that can be expressed in block form as*

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1s}B \\ a_{21}B & a_{22}B & \cdots a_{2s}B & \\ \vdots & \vdots & & \vdots \\ a_{r1}B & a_{r2}B & \cdots & arsB \end{pmatrix}$$

Next, we describe the matrices that make up the product. Consider a $p \times p$ matrix B where every 2×2 submatrix satisfies the property in 4. The first row of the matrix is $\{1, -\alpha, \alpha^2, -\alpha^3, \ldots, \alpha^{p-1}\}$ and the first column is $\{1, -\alpha^1, \alpha^{2q}, -\alpha^{3q}, \ldots, \alpha^{q(p-1)}\}$. We can use this information to find every other element in the matrix.

Lemma 4. *The term of the matrix B_{ij} in row i and column j has the form*

$$B_{ij} = \sum_{n=0}^{\min(i,j)-1} (-1)^{i+j-n} \binom{i-1}{n} \binom{i+j-n-2}{i-1} \alpha^{(i-1-n)q+j-1-n} \gamma^n. \tag{5}$$

Proof. We will prove this formula by induction. First, we can easily verify that the first row and first column satisfy this formula. Now, assume that entries $B_{a,b}$, $B_{a+1,b}$, and $B_{a,b+1}$ satisfy the formula. We want to show that these imply that $B_{a+1,b+1}$ satisfies the formula. We can calculate the $B_{a+1,b+1}$ entry by applying the property for any 2×2 matrix contained in the $p \times p$ matrix.

$$\begin{aligned} B_{a+1,b+1} = -\alpha^q & \sum_{n=0}^{\min(a-1,b)} (-1)^{a+b+1-n} \binom{a-1}{n} \binom{a+b-n-1}{a-1} \alpha^{(a-1-n)q+b-n} \gamma^n \\ & + -\alpha \sum_{n=0}^{\min(a,b-1)} (-1)^{a+b+1-n} \binom{a}{n} \binom{a+b-n-1}{a} \alpha^{(a-n)q+b-1-n} \gamma^n \\ & + -\gamma \sum_{n=0}^{\min(a-1,b-1)} (-1)^{a+b-n} \binom{a-1}{n} \binom{a+b-n-2}{a-1} \alpha^{(a-1-n)q+b-1-n} \gamma^n \\ = & \sum_{n=0}^{\min(a-1,b)} (-1)^{a+b-n} \binom{a-1}{n} \binom{a+b-n-1}{a-1} \alpha^{(a-n)q+b-n} \gamma^n \\ & + \sum_{n=0}^{\min(a,b-1)} (-1)^{a+b-n} \binom{a}{n} \binom{a+b-n-1}{a} \alpha^{(a-n)q+b-n} \gamma^n \\ & + \sum_{n=0}^{\min(a-1,b-1)} (-1)^{a+b+1-n} \binom{a-1}{n} \binom{a+b-n-2}{a-1} \alpha^{(a-1-n)q+b-1-n} \gamma^{n+1}. \end{aligned}$$

This can be simplified differently depending on whether $a > b$, $a < b$, or $a = b$. First, consider that $a \geq b$. We can further simplify by combining the summations. Temporarily, let

$$X = \left(\binom{a-1}{n} \binom{a+b-1-n}{a-1} + \binom{a}{n} \binom{a+b-1-n}{a} + \binom{a-1}{n-1} \binom{a+b-1-n}{a-1} \right).$$

Then substituting the value of X we get

$$\begin{aligned}
B_{a+1,b+1} &= \sum_{n=1}^{b-1}(-1)^{a+b-n}X\alpha^{(a+n)q+b-n}\gamma^{n} \\
&+ (-1)^{a+b}\left(\binom{a-1}{0}\binom{a+b-1}{a-1}+\binom{a}{0}\binom{a+b-1}{a}\right)\alpha^{aq+b}\gamma^{0} \\
&+ (-1)^{a}\binom{a-1}{b}\binom{a-1}{a-1}\alpha^{(a-b)q}\gamma^{b} \\
&+ (-1)^{a}\binom{a-1}{b-1}\binom{a-1}{a-1}\alpha^{(a-b)q}\gamma^{b} \\
&= \sum_{n=0}^{b}(-1)^{a+b-n}X\alpha^{(a+n)q+b-n}\gamma^{n}.
\end{aligned}$$

Now consider that $a \leq b$ and use the same definition for X.

$$\begin{aligned}
B_{a+1,b+1} &= \sum_{n=1}^{a-1}(-1)^{a+b-n}X\alpha^{(a+n)q+b-n}\gamma^{n} \\
&+ (-1)^{a+b}\left(\binom{a-1}{0}\binom{a+b-1}{a-1}+\binom{a}{0}\binom{a+b-1}{a}\right)\alpha^{aq+b}\gamma^{0} \\
&+ (-1)^{b}\binom{a}{a}\binom{b-1}{a}\alpha^{b-a}\gamma^{a} \\
&+ (-1)^{b}\binom{a-1}{a-1}\binom{b-1}{a-1}\alpha^{b-a}\gamma^{a} \\
&= \sum_{n=0}^{a}(-1)^{a+b-n}X\alpha^{(a+n)q+b-n}\gamma^{n}.
\end{aligned}$$

Both cases result in the simplification

$$B_{a+1,b+1} = \sum_{n=0}^{\min(a,b)}(-1)^{a+b-n}X\alpha^{(a+n)q+b-n}\gamma^{n}.$$

Now that we have simplified the equation into one summation, we can simplify the value of X. Using known identities of binomial coefficients, the binomial coefficients can be rewritten as the following.

$$\begin{aligned}
\binom{a+b-1-n}{a-1} &= \frac{a}{a+b-n}\binom{a+b-n}{a} \\
\binom{a+b-1-n}{a} &= \frac{b-n}{a+b-n}\binom{a+b-n}{a} \\
\binom{a-1}{n-1} &= \frac{n}{a}\binom{a}{n} \\
\binom{a-1}{n} &= \frac{a-n}{a}\binom{a}{n}
\end{aligned}$$

Now we will substitute these identities into the equation for X.

$$\begin{aligned}
X &= \left(\binom{a-1}{n}\binom{a+b-1-n}{a-1} + \binom{a}{n}\binom{a+b-1-n}{a} + \binom{a-1}{n-1}\binom{a+b-1-n}{a-1}\right) \\
&= \binom{a}{n}\binom{a+b-n}{a}\left(\frac{a-n}{a}\cdot\frac{a}{a+b-n} + \frac{b-n}{a+b-n} + \frac{n}{a}\cdot\frac{a}{a+b-n}\right) \\
&= \binom{a}{n}\binom{a+b-n}{a}\left(\frac{a+b-n}{a+b-n}\right) \\
&= \binom{a}{n}\binom{a+b-n}{a}.
\end{aligned}$$

Plugging in the simplified value of X into the summation for $B_{a+1,b+1}$ yields

$$B_{a+1,b+1} = \sum_{n=0}^{b}(-1)^{a+b-n}\binom{a}{n}\binom{a+b-n}{a}\alpha^{(a+n)q+b-n}\gamma^{n},$$

which satisfies the formula. By induction, every entry in B satisfies 5, and the proof is complete. □

The formula for each term allows us to find the exact values of the last row and last column.

Lemma 5. *The last row of matrix B is $B_{pj} = (-1)^{j-1}\alpha^{(p-j)q}\gamma^{j-1}$ and the last column is $B_{ip} = (-1)^{p-1}\alpha^{p-i}\gamma^{i-1}$.*

Proof. The last row of the matrix is given by

$$B_{pj} = \sum_{n=0}^{j-1}(-1)^{p+j-n}\binom{p-1}{n}\binom{p+j-n-2}{p-1}\alpha^{(p-1-n)q+j-1-n}\gamma^{n}.$$

By binomial coefficient identities, $\binom{p-1}{n} = (-1)^n \mod p$. Also by binomial coefficient identities,

$$\binom{p+j-n-2}{p-1} = \frac{p}{j-n-1}\binom{p+j-n-2}{p}.$$

This implies that $\binom{p+j-n-2}{p-1} \equiv 0 \mod p$ except when $n = j-1$. Therefore, the only terms that are left in the last row are the terms with γ^{j-1}. Thus, $B_{pj} = (-1)^{j-1}\alpha^{(p-j)q}\gamma^{j-1}$, proving the lemma for the last row.

The last column of the matrix is given by

$$B_{ip} = \sum_{n=0}^{i-1}(-1)^{p+j-n}\binom{i-1}{n}\binom{i+p-n-2}{i-1}\alpha^{(i-1-n)q+p-1-n}\gamma^{n}.$$

Since

$$\binom{p+i-n-2}{p-1} = \frac{(p+i-n-2)!}{(i-1)!(p-n-1)!},$$

then $\binom{p+i-n-2}{p-1}$ is divisible by p as long as $i-n-2 \geq 0$. This is the case except when $n = i-1$, because $i-(i-1)-2 = -1$. Therefore, all the terms of the entries in the last column become 0 modulo p except when $n = i-1$. Therefore, the formula for the last column of the matrix is $B_{ip} = (-1)^{p-1}\alpha^{p-i}\gamma^{i-1}$, completing the proof of the lemma. □

Now that we have defined one of the matrices, we can define the sequence of matrices that will be in the Kronecker Product.

Definition 15. *The matrix B_h represents the $p \times p$ matrix where the (i, j) term is of the form*

$$\left(\sum_{n=0}^{\min(i,j)-1} (-1)^{i+j-n}\binom{i-1}{n}\binom{i+j-n-2}{i-1}\alpha^{(i-1-n)q+j-1-n}\gamma^{n}\right)^{p^{l-h}}.$$

Lemma 6. *Assume $q = p^l$. Then*

$$\begin{pmatrix} P_0 & P_q & \cdots & P_{(q-1)q} \\ P_1 & P_{q+1} & \cdots & P_{(q-1)q+1} \\ \vdots & \vdots & & \vdots \\ P_{q-1} & P_{2q-1} & \cdots & P_{q^2-1} \end{pmatrix} = B_1 \otimes B_2 \otimes \cdots \otimes B_l.$$

Proof. Denote the matrix on the left by Γ_q and the matrix on the right side by Γ'_q. The first row of Γ'_q is $(1, -\alpha, \alpha^2, -\alpha^3, \ldots, \alpha^{q-1})$ and the first column is $(1, -\alpha^q, \alpha^{2q}, \ldots, \alpha^{(q-1)q})$. By Lemmas 2 and 3, the first rows and first columns of Γ_q and Γ'_q are equal. Therefore, in order to show that $\Gamma_q = \Gamma'_q$, it is sufficient to show that every 2×2 matrix inside of Γ'_q satisfies 4.

We will proceed by induction. First, we know that the matrix B_l satisfies property 4 by construction and by Lemma 4. Now let $i > 1$ and assume that every 2×2 block of the matrix

$$B = B_{i+1} \otimes B_{i+2} \otimes \cdots \otimes B_l$$

satisfies 4. To complete the inductive step, the goal is to show that every 2×2 block of the matrix

$$B_i \otimes B = \left(\begin{array}{c|c|c|c} B & (-\alpha)^{p^{l-i}}B & \cdots & (\alpha^{p-1})^{p^{l-i}}B \\ \hline (-\alpha^{q})^{p^{l-i}}B & (2\alpha^{q+1}-\gamma)^{p^{l-i}}B & \cdots & (-\gamma\alpha^{p-2})^{p^{l-i}}B \\ \hline \vdots & \vdots & & \vdots \\ \hline \alpha^{(p-1)q})^{p^{l-i}}B & (-\gamma\alpha^{(p-2)q})^{p^{l-i}}B & \cdots & (\gamma^{p-1})^{p^{l-i}}B \end{array}\right)$$

also satisfies 4. There are four cases for where a 2×2 block may lie in the matrix above.

1. The block lies entirely in one of the 9 cells.
2. The block intersects four different cells of the matrix.
3. The block intersects two horizontally adjacent cells.
4. The block intersects two vertically adjacent cells.

Any 2×2 block in B satisfies the relation by the induction hypothesis. Therefore, any block in the first case will also satisfy the relation because multiplying by a constant will maintain the relation.

Note that by Lemma 5, the first and last columns and rows of B have the following structure:

$$\begin{pmatrix} 1 & -\alpha & \alpha^2 & \cdots & \alpha^{p^{i-1}} \\ -\alpha^q & & & & -\alpha^{p^{i-2}}\gamma \\ \alpha^{2q} & & & & \alpha^{p^{i-3}}\gamma^2 \\ \vdots & & & & \vdots \\ \alpha^{q(p^{i-1})} & -\alpha^{q(p^{i-2})}\gamma & \alpha^{q(p^{i-3})}\gamma^2 & \cdots & \gamma^{p^{i-1}} \end{pmatrix} \tag{6}$$

Now let w, x, y, z be constants from a 2×2 submatrix of B and thus satisfying $-z = \alpha^q x + \alpha y + \gamma w$. Then in the second case, the block will be of the form

$$\left(\begin{array}{c|c} \prod_{k=0}^{i-1} \gamma^{(p-1)p^k} w & \prod_{k=0}^{i-1} \alpha^{(p-1)qp^k} x \\ \hline \prod_{k=0}^{i-1} \alpha^{(p-1)p^k} y & z \end{array}\right) = \left(\begin{array}{c|c} \gamma^{p^i-1} w & \alpha^{q(p^i-1)} x \\ \hline \alpha^{p^i-1} y & z \end{array}\right).$$

We want to show that $-z = \alpha^{qp^i} x + \alpha^{p^i} y + \gamma^{p^i} w$. We can rewrite the relationship between the constants as $(-z^{1/p^i})^{p^i} = (\alpha^q x^{1/p^i} + \alpha y^{1/p^i} + \gamma w^{1/p^i})^{p^i}$ which by Theorem 8 implies that $-z = \alpha^{qp^i} x + \alpha^{p^i} y + \gamma^{p^i} w$. Thus, submatrices in the second case satisfy 4.

For the third case when the block intersects two horizontally adjacent cells, the block will have the form

$$\left(\begin{array}{c|c} (-1)^{j-1}\alpha^{(p^i-j)\gamma^{j-1}} x & (-1)^j \alpha^{(j-1)q} y \\ \hline (-1)^j \alpha^{p^i-j-1} y^j x & (-1)^j \alpha^{jq} y \end{array}\right)$$

for constants x and y and $1 \leq j \leq q-1$. Indeed, we have

$$\begin{aligned} -(-1)^j \alpha^{jq} y &= (-1)^{j-1}\alpha^{jq} y + (-1)^j \alpha^{(p^i-j)} \gamma^j x + (-1)^{j-1} \alpha^{(p^i-j)} \gamma^j x \\ &= (-1)^{j-1} \alpha^{jq} y, \end{aligned}$$

satisfying 4. Finally, in the fourth case of intersecting vertically adjacent cells, the block has the form

$$\left(\begin{array}{c|c} (-1)^{j-1}\alpha^{(p^i-j)q}\gamma^{j-1} x & (-1)^j \alpha^{(p^i-j-1)q} \gamma^j x \\ \hline (-1)^{j-1} \alpha^{j-1} y & (-1)^j \alpha^j y \end{array}\right)$$

which satisfies 4 because

$$\begin{aligned} -(-1)^j \alpha^j y &= (-1)^j \alpha^{(p^i-j)q} \gamma^j x + (-1)^{j-1} \alpha^j y + (-1)^{j-1} \alpha^{(p^i-j)q} \gamma^j x \\ &= (-1)^{j-1} \alpha^j y. \end{aligned}$$

Therefore, every 2×2 block of $B_i \otimes B$ satisfies 4. By induction, every 2×2 block in Γ'_q must also satisfy 4. Since the first rows and columns of Γ_q and Γ'_q are equal, and both satisfy the recurrence relation, then the entire matrices must be equal and $\Gamma_q = \Gamma'_q$. □

The matrix identity in Lemma 6 can be used to find a sufficient condition for k that satisfies $det_{\alpha,\beta}(t^k) < q$.

Lemma 7. *Assume $q = p^l$. Let $0 \leq k \leq q^2$ and write $k = wq + z$ where $z < q$. For $\alpha, \beta \in \mathbb{F}_{q^2}$, suppose either $w = 0$ or that there exists $1 \leq i \leq l$ such that $w \equiv 0 \mod p^i$ and $z \not\equiv -1 \mod p^i$. Then $deg_{\alpha,\beta}(t^k) < q$.*

Proof. Suppose $k = wq + z$ where w and z satisfy the conditions stated in the lemma. By Lemma 2, we just need to show that $P_{k+1} = -\alpha^q P_k$ in order to show that $deg_{\alpha,\beta}(t^k) < q$.

When $w = 0$, then $k = z < q$. This automatically gives that $deg_{\alpha,\beta}(t^k) < q$ because by Lemma 3, since $0 \leq k \leq q$ then $P_k = (-1)^k a^{qk}$.

Assume that there exists an i such that $w \equiv 0 \mod p^i$ and $z \not\equiv -1 \mod p^i$. Let

$$A = B_1 \otimes \cdots \otimes B_{l-i} \quad \text{and} \quad B = B_{l-i+1} \otimes \cdots \otimes B_l$$

where B_h is as in Definition 15. By Lemma 6, we have

$$A \otimes B = \begin{pmatrix} P_0 & P_q & \cdots & P_{(q-1)q} \\ P_1 & P_{q+1} & \cdots & P_{(q-1)q+1} \\ \vdots & \vdots & & \vdots \\ P_{q-1} & P_{2q-1} & \cdots & P_{q^2-1} \end{pmatrix} = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1s}B \\ a_{21}B & a_{22}B & \cdots & a_{2s}B \\ \vdots & \vdots & & \vdots \\ a_{s1}B & a_{s2}B & \cdots & a_{ss}B \end{pmatrix}$$

where $s = p^{l-i}$.

Suppose that P_k lies in block $a_{cd}B$ for some $c, d \in \{1, \ldots, p^{l-i}\}$. The fact that $w \equiv 0 \mod p^i$ means P_k is in the first column of $a_{cd}B$. Since $z \not\equiv -1 \mod p^i$ means that P_k is not in the last row of $a_{cd}B$. Therefore, P_{k+1} must also be in the same block $a_{cd}B$. By the structure of the first column of B shown in Equation 4, we get that $P_{k+1} = -\alpha^q P_k$. Since this is what we wanted to show by Lemma 2, we get that $\deg_{\alpha\beta}(t^k) < q$. □

4.3 Counting the number of Good Monomials

So far, we have proved that monomials of the form $M_{a,b}(x, y) = x^a y^b$ for $a \leq q - 1$ and $b \leq q^2 - 1$ are linearly independent and found conditions for when $deg_{\alpha,\beta}(t^k) < q$. Now, we want to count as many good monomials as possible that fit the conditions. If $a + b < q$, then it is clear the $M_{a,b}$ is good. If $a + b \geq q$, there are two ways that $M_{a,b}$ can be good. First, all the terms could reduce to degree less than q modulo $p_{\alpha,\beta}(t)$ without using finite field properties. Second, the coefficient in front of the term t^q could reduce modulo p. The formula for the coefficient can be found by expanding $M_{a,b} \circ L_{\alpha,\beta}$:

$$(M_{a,b} \circ L_{\alpha,\beta})(t) = M_{a,b}(\alpha t + \beta, t) = (\alpha t + \beta)^a t^b = \sum_{j \leq a} \binom{a}{j} \alpha^j \beta^{a-j} t^{b+j}.$$

If $\binom{a}{j} \equiv 0 \mod p$ for $j = q - b$, then the monomial is good. To count when this coefficient will disappear, we use Lucas' Theorem.

Definition 16. *Let a and b be integers between 0 and $p^d - 1$ for prime p, and let $p\!-\!ary(a) \in \{0, 1, \ldots, p-1\}^d$ denote the p-ary expansion of a. We say that a lies in the p-shadow of b, denoted $a \leq_p b$ if every digit of $p - ary(a)$ is less than or equal to the corresponding digit in $p - ary(b)$.*

Theorem 6. *(Lucas). Let $0 \leq a \leq b$ be integers. Then $\binom{b}{a}$ is zero mod p if and only if $a \not\leq_p b$, meaning a does not lie in the p-shadow of b.*

With these tools in mind, we can come up with some sufficient conditions for good monomials and count the number of monomials that satisfy those conditions to get a bound on the rate of the Hermitian-Lifted Code. Note that the properties in the following lemma do not cover all possible good monomials, but are plentiful enough to provide a positive lower bound. The following lemma and proof are identical to Claim 12 from [5], with a general prime p replacing every 2 from the original version. We define x_r to be the digit corresponding to p^r of the p-ary expansion of x, or the digit r positions from the right.

Lemma 8. ([5]) Suppose that $a \leq q-1$ and $b \leq q^2-1$ satisfy the following properties:

1. $b = wq + b'$ for some $w < q$ and some $b' < p^{l-1}$ so that $w \equiv 0 \mod (p^i)$ for some $1 \leq i \leq l$.
2. $a < p^{l-1}$
3. there is some $0 \leq s \leq i-1$ so that $a_s = b'_s = 0$

Then $M_{a,b}$ is good.

Proof. Suppose that a, b satisfy the properties above. Let $L_{\alpha,\beta}(t) = (\alpha t + \beta, t)$ be a line in $\mathcal{L}$ and write

$$(M_{a,b} \circ L_{\alpha,\beta})(t) = \sum_{j \leq a} \binom{a}{j} \alpha^j \beta^{a-j} t^{j+b} = \sum_{j \leq_p a} \alpha^j \beta^{a-j} t^{j+b} \tag{7}$$

using Lucas' theorem in the second equality. Notice that for any $j \leq_p a$, we have $j < p^{l-1}$ and $j_s = 0$, using properties (2) and (3). Then the only monomials that appear in 7 are of the form t^k where $k = wq + b' + j$ for w, b' as in property (1) and for $j \leq_p a$. Let i be as in (1), so that $w \equiv 0 \mod p^i$. We claim that $b' + j \not\equiv -1 \mod p^i$. Indeed, we can write

$$b' = p^{s+1} b'' + b''' \quad \text{and} \quad j = p^{s+1} j'' + j'''$$

for some $b''', j''' > p^s$, using the fact that $b'_s = j_s = 0$. Note that there exists some $c \leq p^i - p^{s+1}$ so that

$$p^{s+1}(b'' + j'') \equiv c \mod p^i.$$

Thus,

$$b'_j \equiv c + b''' + j''' \mod p^i.$$

Since $b''', j''' < p^s$, we have

$$c + b''' + j''' < (p^i - p^{s+1}) + (p^{s+1} - 1) = p^i - 1$$

which means that $b' + j \not\equiv -1 \mod p^i$, as claimed.

Thus, k is of the form $k = wq + z$ (where $z = b' + j$) so that $w \equiv 0 \mod p^i$ and $z \not\equiv -1$. By Lemma 7, k has the necessary conditions to satisfy $\deg_{\alpha,\beta}(t^k) < q$. Therefore, $\deg_{\alpha,\beta}(M_{a,b} \circ L_{\alpha,\beta} < q$ for all α and β, so $M_{a,b}$ is good. □

Finally, we prove Theorem 5.

Proof. We will count the number of pairs a, b that satisfy the sufficient conditions for $M_{a,b}$ to be good. We iterate over all s, where we take s to be the smallest index so that $a_s = b'_s = 0$. For a given s, there are $p^{2s} - (p^2-1)^s$ ways to assign the bits $a_0, \dots, a_{s-1}$ and $b'_0, \dots, b_s - 1'$, since there are only $(p^2-1)^s$ ways to never have $a_r = b'_r = 0$ for any $0 \leq r \leq s-1$. Then there are $p^{2(l-s-2)}$ ways to assign the bits $a_{s+1}, \dots, a_{l-2}, b'_{s+1}, \dots, b'_{l-2}$. Finally, there are p^{l-s-1} ways to assign the bits $w_{s+1}, \dots, 2_{l-1}$. We will choose $w_0, \dots, w_s = 0$, ensuring that

$w \equiv 0 \mod 3^{s+1}$, specifically $w \equiv 0 \mod 3^i$ for some $i > s$. Thus, the total number of monomials meeting the description in Lemma 8 when $l \geq 2$ is

$$\begin{aligned}
\sum_{s=0}^{l-1}(p^{2s}-(p^2-1)^s)p^{2(l-s-2)}p^{l-s-1} &= \sum_{s=0}^{l-1}(p^{2s}-(p^2-1)^s)p^{2(l-s-2)+l-s-1} \\
&= \sum_{s=0}^{l-1}(p^{2s}-(p^2-1)^s)p^{3l-3s-5} \\
&= \frac{p^{3l}}{p^5}\sum_{s=0}^{l-1}(p^{2s}-(p^2-1)^s)p^{-3s} \\
&= \frac{p^{3l}}{p^5}\sum_{s=0}^{l-1}\left(\frac{1}{p}^s - \frac{p^2-1}{p^3}^s\right) \\
&= \frac{p^{3l}}{p^5}\sum_{s=0}^{l-1}p^{-s} - \frac{p^2-1}{p^3}^s \\
&= \frac{p^{3l}}{p^5}\left(\frac{1-(\frac{1}{p})^l}{1-\frac{1}{p}} - \frac{1-(\frac{p^2-1}{p^3})^l}{1-\frac{p^2-1}{p^3}}\right) \\
&= \frac{p^{3l}}{p^5}\left(\frac{p}{p-1}\left(1-\left(\frac{1}{p}\right)^l\right) - \frac{p^3}{p^3-p^2+1}\left(1-\left(\frac{p^2-1}{p^3}\right)^l\left(\frac{1}{p}\right)\right)\right) \\
&= \frac{p^{3l}}{p^5}\left(\frac{p}{p-1} - \frac{p}{p-1}\left(\frac{1}{p}\right)^l - \frac{p^3}{p^3-p^2+1} + \frac{p^3}{p^3-p^2+1}\left(\frac{p^2-1}{p^3}\right)^l\right) \\
&= \frac{p^{3l}}{p^5}\left(\frac{p}{(p-1)(p^3-p^2+1)} - \frac{p}{p-1}\left(\frac{1}{p}\right)^l + \frac{p^3}{p^3-p^2+1}\left(\frac{p^2}{p^3-p^2+1}\right)^l\right) \\
&= \frac{p^{3l}}{p^5}\left(\frac{p}{(p-1)(p^3-p^2+1)}\right)(1-(p^3-p^2+1)\left(\frac{1}{p}\right)^l + (p^3-p^2)\left(\frac{p-1}{p^3}\right)^l) \\
&= q^3\left(\frac{1-(p^3-p^2+1)(\frac{1}{p})^l+(p^3-p^2)(\frac{p^2-1}{p^3})^l}{p^4(p-1)(p^3-p^2+1)}\right) \\
&\geq q^3\left(\frac{1-.531}{p^4(p-1)(p^3-p^2+1)}\right)
\end{aligned}$$

The last step comes from finding the lower bound on the numerator when $p \geq 3$ and $l \geq 2$. The rate of the code is $r = k/n$ where $n = q^3$. Therefore, we have that the rate when p is an odd prime is bounded below by

$$\frac{.469}{p^4(p-1)(p^3-p^2+1)},$$

which completes the proof. □

5 Examples

5.1 Example when $p = 3$

The proof applies to the general when q is the power of any odd prime, which makes it difficult to visualize some of the steps of the proof. Below, we will show some of the steps of the proof specifically when $p = 3$, and calculate the lower bound of the rate given by Theorem 5.

The structure of matrix B is the 3×3 matrix determined by the formula given in Lemma 4 modulo 3. Thus, the product $B_1 \otimes \cdots \otimes B_l$ when $p = 3$ is

$$\begin{pmatrix} 1 & (-\alpha)^{3^{l-1}} & (\alpha^2)^{3^{l-1}} \\ (-\alpha^q)^{3^{l-1}} & (2\alpha^{q+1} - \gamma)^{3^{l-1}} & (2\alpha\gamma)^{3^{l-1}} \\ (\alpha^{2q})^{3^{l-1}} & (2\alpha^q\gamma)^{3^{l-1}} & \gamma^{2(3^{l-1})} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 & -\alpha & \alpha^2 \\ -\alpha^q & 2\alpha^{q+1} - \gamma & 2\alpha\gamma \\ \alpha^{2q} & 2\alpha^q\gamma & \gamma^2 \end{pmatrix}.$$

It is straightforward to check that the matrix B satisfies the block relation 4 without applying Lemma 4 by only checking the 4 possible 2×2 submatrices.

The lower bound of the rate of the Hermitian-Lifted Code when $q = 3^l$ is given by

$$\frac{.469}{3^4(2)(3^3 - 3^2 + 1)} = \frac{.469}{3078} = .000152.$$

Note that this lower bound does not depend on the value of l. This is because the numerator .469 is calculated by finding the lower bound of $-(p^3 - p^2 + 1)(\frac{1}{p})^l + (p^3 - p^2)(\frac{p^2-1}{p^3})^l$, which increases as l increases for a fixed value of p. While this does not necessarily indicate that the actual rate of the Hermitian-Lifted Code increases as l increases, it does imply that the lower bound .000152 is not an optimal lower bound for all $q = 3^l$ given the number of good monomials that are defined in Lemma 7.

5.2 Example of a Good Monomial

We will use the conditions in Lemma 7 to find an example of a good monomial, and then walk through the algebraic steps to verify that it reduces down to a polynomial of degree at most than $q - 1$ after plugging in the line parameterization.

Suppose that $q = 3^3 = 27$. The Hermitian Curve $\mathbb{H}_27$ is $x^{27} + x = y^{28}$ over the field $\mathbb{F}_{27^2}$. Based on property (1) of Lemma 7, either $w \equiv 0 \mod 3$, $w \equiv 0 \mod 9$, or $w \equiv 0 \mod 27$, and $b' < 9$. Let $w = 9$ and $b' = 6$, which gives $b = 9(27) + 6 = 249$. By property (2), we have $a < 9$, so we can pick $a = 6$. Finally, we need to check that our choices of a and b' satisfy property (3). Since the ternary expansion of a and b' is $a = b' = 20$, the second digit satisfies $a_0 = b'_0 = 0$. Therefore, Lemma 7 asserts that the monomial $M_{6,249} = x^6y^{249}$ is good.

We could check algebraically that the monomial x^6y^{249} is good by showing that it reduces to a monomial of degree at most $q - 1 = 26$. However, this would be a very tedious process, so we will omit the proof but explain how to use the Hermitian Curve to reduce the degree of polynomials. Plugging in the parameterization of lines into the monomial gives

$$(M_{6,249} \circ L_{\alpha,\beta})(t) = (\alpha t + \beta)^6 t^{249} = \alpha t^{250} + \beta t^{243}.$$

Thus, we would need to show that t^{250} and t^{243} each individually reduce to monomials of degree at most 26.

From the Hermitian Curve itself, we get

$$t^{28} = (\alpha t + \beta)^{27} + (\alpha t + \beta) = \alpha^{27} t^{27} + \alpha t + c_1.$$

Since we are only concerned with the degree of the monomial, we can generalize many of the constants, in this case $c_1 = \beta^{27} + \beta$. We can use this relation to reduce every power of t to a monomial of degree at most 27. Below are the first few algebraic steps for the next largest degrees.

$$\begin{aligned} t^{29} &= \alpha^{27} t^{28} + \alpha t^2 + c_1 t = \alpha^{54} t^{27} + \alpha t + c_2 + \alpha t^2 + c_1 t \\ &= \alpha^{54} t^{27} + \alpha t^2 + c_3 t + c_2 \end{aligned}$$

$$\begin{aligned} t^{30} &= \alpha^{54} t^{28} + \alpha t^3 + c_3 t^2 + c_2 t = \alpha^{81} t^{27} + \alpha^{55} t + c_4 + \alpha t^3 + c_3 t^2 + c_2 t \\ &= \alpha^{81} t^{27} + \alpha t^3 + c_3 t^2 + c_3 t + c_4 \end{aligned}$$

$$\begin{aligned} t^{31} &= \alpha^{81} t^{28} + \alpha t^4 + c_3 t^3 + c_5 t^2 + c_4 t = \alpha^{108} t^{27} + \alpha^{82} t + c_6 + \alpha t^4 + c_3 t^3 + c_5 t^2 + c_4 t \\ &= \alpha^{108} t^{27} + \alpha t^4 + c_3 t^3 + c_5 t^2 + c_7 t + c_6 \end{aligned}$$

To confirm that $M_{6,243}$ is a good monomial, this process would be continued until reaching t^{243} and t^{250}. Although this would be a very long process to do by hand, Lemma 7 implies that the monomials t^{243} and t^{250} reduce down to a degree of at most t^{26}. This occurs because the coefficients of t^{27} reduce to 0 modulo 3.

Note that the values of a and b do not have to satisfy all the conditions in Lemma 7 in order to be a good monomial. The proof of the theorem counts a sufficient number of good monomials to reach a positive lower bound for the rate, but it does not count all the possible good monomials. Therefore, the actual rate of the Hermitian-Lifted Code is much higher than the bound given in this paper.

6 Conclusion

This paper proved an extension of the main theorem in [5] by following a similar proof strategy to conclude that all Hermitian-Lifted Codes have a rate bounded below by a positive constant, regardless of the value of q.

There are remaining unanswered questions regarding Hermitian-Lifted Codes and also lifted codes in general. In [1], the authors improve the lower bound given in [5] by using a different proof strategy for counting good monomials. A similar approach could be used to prove the general prime case to try to improve the bound given in Theorem 5. Further, it would be interesting to find the exact number of good monomials in order to find the exact value of the rate of the Hermitian-Lifted Code. Similar questions could also be studied for lifted codes on different types of curves. While Hermitian-Lifted Codes are not actually implemented in real world applications, discovering codes which are both locally recoverable and have good parameters could be useful for constructing similar codes which may eventually be implemented in the future.

7 Appendix: Finite Fields

This section will review important algebraic facts and finite field properties that are important for understanding facts about algebraic codes. Most of the following information can be found in the appendix of [6].

Most algebraic codes have a finite field as the alphabet. One of the most common examples of a finite field is the integers mod p, denoted $\mathbb{Z}_p$, for any prime p. Note that the integers mod n for a nonprime number n does not form a field, because a field cannot have any zero divisors. However, there are fields of order $q = p^n$ for any prime p and positive integer n, denoted $\mathbb{F}_q$, which is frequently used as the alphabet for codes in the paper.

Recall Fermat's Little Theorem, which is useful when working over the field $\mathbb{F}_p$.

Theorem 7. (Fermat's Little Theorem) If p is a prime number, then any integer a satisfies $a^p \equiv a \mod p$.

This theorem can be used to reduce polynomials modulo a prime p, because any exponent greater than or equal to p can be reduced. For example, in the Reed-Solomon code, the dimension k is bounded by the size of the field $q - 1$ because any polynomial of degree greater than $q - 1$ modulo q could be reduced to a polynomial with degree at most $q - 1$. The same is true for fields $\mathbb{F}_q$ when q is a prime power. Any element $a \in \mathbb{F}_q$ satisfies the property $a^q \equiv a$, so any polynomials in $\mathbb{F}_q$ can also be reduced to a polynomial of degree at most $q - 1$.

Another useful algebraic property is the binomial coefficient expansion modulo a prime.

Theorem 8. *The binomial expansion formula modulo prime p is $(a + b)^p = a^p + b^p$.*

This theorem makes binomial expansions over the field $\mathbb{F}_p$ easy and simple, and can be extended to polynomial expansions where $(a_1 + a_2 + \cdots + a_n)^p = a_1^p + a_2^p + \cdots + a_n^p$.

The code used in the main theorem has the alphabet $\mathbb{F}_{q^2}$, where $q = p^l$ for a prime p and positive integer l. The field $\mathbb{F}_q$ is a subfield of $\mathbb{F}_{q^2}$. While $a^{q^2} = a$ for all $a \in \mathbb{F}_{q^2}$, the relation $a^q = a$ is not always true for all $a \in \mathbb{F}_{q^2}$. Instead, $a^q = a$ is true only if $a \in \mathbb{F}_q \in \mathbb{F}_{q^2}$. This property can be used to check if an element in $\mathbb{F}_{q^2}$ is also an element in $\mathbb{F}_q$.

Theorem 9. *For an element $x \in \mathbb{F}_{q^2}$ where q is a prime power, $x^q - x = 0$ if and only if $x \in \mathbb{F}_q$.*

The *norm* of x is the function $N(x) : \mathbb{F}_{q^2} \to \mathbb{F}_q$ and is defined as $N(x) = x^{q+1}$. The *trace* of x is the function $Tr(x) : \mathbb{F}_{q^2} \to \mathbb{F}_q$ and is defined as $Tr(x) = x^q + x$. We can use Theorem 9 to verify that these functions in fact map elements from $\mathbb{F}_{q^2}$ to $\mathbb{F}_q$. In order to show that the norm of x maps elements to $\mathbb{F}_q$, we must show that $(x^{q+1})^q = x^{q+1}$. Indeed, we have $(x^{q+1})^q = x^{q^2+q} = x^{q+1}$, so $N(x) \in \mathbb{F}_q$. Similarly, the trace of x satisfies $(x^q + x)^q = x^{q^2} + x^q = x + x^q$, which also verifies the mapping of the trace function. These properties about norm and trace are useful for working with the Hermitian Curve, which is the trace of x on one side of the equation and the norm of y on the other side of the equation.

References

[1] Allen, A., Pabón-Cancel, E., Piñero-González, F., and Polanco, L. Improving the dimension bound of hermitian lifted codes, 2023.

[2] Babu, B. S., Krishnan, M. N., Vajha, M., Ramkumar, V., Sasidharan, B., and Kumar, P. V. Erasure coding for distributed storage: An overview. *CoRR abs/1806.04437* (2018).

[3] Barg, A., Tamo, I., and Vladut, S. G. Locally recoverable codes on algebraic curves. *CoRR abs/1603.08876* (2016).

[4] Guo, A., and Sudan, M. New affine-invariant codes from lifting. *CoRR abs/1208.5413* (2012).

[5] López, H. H., Malmskog, B., Matthews, G. L., Piñero-González, F., and Wootters, M. Hermitian-lifted codes. *CoRR abs/2006.05558* (2020).

[6] Walker, J. L. Codes and curves. *Student Mathematical Library 7* (2002), 66.