Baltic J. Modern Computing, Vol. 4 (2016), No. 4, pp. 753–788
DOI: http://dx.doi.org/10.22364/bjmc.2016.4.4.12

Quantum List Decoding of Classical Block Codes of Polynomially Small Rate from Quantumly Corrupted Codewords*

Tomoyuki YAMAKAMI

Faculty of Engineering, University of Fukui, 3-9-1 Bunkyo, Fukui, 910-8507, Japan

TomoyukiYamakami@gmail.com

Abstract. Given a classical error-correcting block code, the task of quantum list decoding is to produce from any quantumly corrupted codeword a short list containing all messages whose codewords exhibit high "presence" in the quantumly corrupted codeword. Efficient quantum list decoders have been used to prove a quantum hardcore property of classical codes. However, the code rates of all known families of efficiently quantum list-decodable codes are, unfortunately, too small for other practical applications. To improve those known code rates, we prove that a specific code family of polynomially small code rate over a fixed code alphabet, obtained by concatenating generalized Reed-Solomon codes as outer codes with Hadamard codes as inner codes, has an efficient quantum list-decoding algorithm if its codewords have relatively high codeword presence in a given quantumly corrupted codeword. As an immediate application, we use the quantum list decodability of this code family to solve a certain form of quantum search problems in polynomial time. When the codeword presence becomes smaller, in contrast, we show that the quantum list decodability of generalized Reed-Solomon codes with high confidence is closely related to the efficient solvability of the following two problems: the noisy polynomial interpolation problem and the bounded distance vector problem. Moreover, assuming that $\mathrm{NP} \not\subseteq \mathrm{BQP}$, we also prove that no efficient quantum list decoder exists for the generalized Reed-Solomon codes.

Keywords: quantum computation, block error-correcting code, quantum list decoding, quantumly corrupted codeword, quantum one-way function, generalized Reed-Solomon code, Hadamard code, concatenated code

---

* An early version appeared in the Proceedings of the 13th Computing: The Australasian Theory Symposium (CATS 2007), pp. 153–162, Ballarat, Australia, January 30–February 2, 2007. This work was in part supported by the Mazda Foundation.

1 Quantum List Decoding

Classical list decoding, whose notion is attributed to Elias (1957) and Wozencraft (1958) in late 1950s, has recently drawn significant attention after Sudan's (1997) discovery of an efficient list-decoding algorithm for well-studied Reed-Solomon codes beyond its "traditional" error-correction radius. List decoding has since then found useful applications to cryptography as well as computational complexity theory (see, e.g., survey articles of Sudan (2000) and Trevisan (2004)). For a wider range of applications to, in particular, quantum computations, an introduction of quantum analogue of such list decoding is an inevitable consequence.

In a seminal paper of Kawachi and Yamakami (2010) (following an early work of Adcock and Cleve (2002) on biased oracles) published first in 2006, a notion of *quantum list decoding* of classical block codes arose quite naturally in their study of *quantum hardcore functions* for arbitrary (strongly) quantum one-way functions. A goal of quantum list decoding in Kawachi and Yamakami's *implicit-input explicit-output model* is to produce a relatively short list of message candidates by means of oracle queries to a faulty quantum encoding procedure given as a form of *oracle*. This model of quantum list decoding slightly differs from a conventional transmission model between a sender and a receiver through a noisy channel, particularly, in the following aspects. Given an original message hidden to the receiver, assumed is the existence of a faulty quantum encoding procedure (called as a *quantum-computationally corrupted codeword* or *quantumly corrupted codeword*) that tries to generate a code symbol at each specified block location of a desired codeword induced from the original message. To recover the hidden message from this quantumly corrupted codeword, the receiver is allowed to access the quantumly corrupted codeword *repeatedly*, partly because he cannot duplicate "unknown" quantum states by a quantum-mechanical principle. The quantumly corrupted codeword is likely to behave "adversarially" and hinder the receiver's effort of recovering uniquely the original message. Quite often, however, it is sufficient to produce a reasonably short list of message candidates including all the messages whose corresponding codewords are in close proximity to the given quantumly corrupted codeword, and thus this list certainly contains the hidden message. This "closeness" is scaled by the notion of *codeword presence* (or *presence*, in short), which indicates the average probability of obtaining successfully each block symbol of the target codeword from the quantumly corrupted codeword (see Kawachi and Yamakami (2010) for an intuition behind this notion). Because of these differences, the classical list decodability does not generally imply the quantum list decodability. To construct hardcore functions is the primary purpose of quantum list decoding by Kawachi and Yamakami (2010), and their study of quantum list decoding was centered at a natural question of what types of classical block codes are efficiently quantum list decodable.

In the past literature showed several families of block codes that are classical/quantum list decodable in polynomial time. The first of such examples is a family of *Hadamard codes*. In the case of classical list decoding, Goldreich and Levin (1989) proved the classical list decodability of the binary Hadamard codes, and subsequently Goldreich, Rubinfeld, and Sudan (1995) presented a general list-decoding algorithm for the q-ary Hadamard codes. Concerning quantum list decoding, by contrast, Adcock and Cleve (2002) essentially proved that the binary Hadamard codes are

quantum list decodable in polynomial time. For the q-ary Hadamard codes, a fast quantum list-decoding algorithm was given by Kawachi and Yamakami (2010). They also presented two additional quantum list-decodable codes: *shifted Legendre symbol codes* and *pairwise equality codes*. A common feature of these codes is that they all have exponentially small code rate, where the *rate* of a code is a ratio of message length (or a dimension of the code) and codeword's block length. For instance, the rate of the binary Hadamard code is exactly $n/2^n$ for message length n. Notice that, in a practical setting, code rate and block length are important factors in designing error-correcting codes. In particular, a family of polynomial-time classical list-decodable codes of polynomially small rate over the binary code alphabet finds numerous applications in the fields of cryptography and computational complexity theory (refer to, e.g., survey articles by Sudan (2000) or Trevisan (2004)).

All known efficiently quantum list-decodable code families have so far *exponentially small* code rate, which is extremely smaller than the code rates of many practical codes. It is therefore natural to ask whether there exists an efficiently quantum list-decodable code of polynomially small rate and of fixed alphabet size for any given bias parameter. This paper is profoundly motivated by this intriguing question and, as its main theorem, it will successfully prove the existence of such a code family; more strongly, we will show the following statement.

Theorem 1. [Main Theorem] *Let q be any prime constant. For any constant $k \geq 1$, there exist a polynomially-time computable function $t : \mathbb{N} \to \mathbb{N}^+$ and a classical block $(t(n), n)_q$-code family C such that*

1. *C is polynomial-time classically list decodable with confidence $5/6$, and*
2. *C is polynomial-time quantumly list decodable with presence at least $1/q + 1/n^k$ and confidence $2/3$.*

This code family C has code rate $n/t(n)$, which is only polynomially small.

The rest of this paper is dedicated to proving this theorem and seeking its application.

To obtain the desired code family stated in the main theorem, we will initially seek a well-studied code family. A family of *generalized Reed-Solomon (GRS) codes* has relatively large code rate; however, it usually has large alphabet size. From this code family, we will build a family of codes of high code rate over a fixed code alphabet by an idea of Forney (1966). In this paper, we will use in Section 3 a concatenated code $C^{GRS\text{-}H}$ of Guruswami and Sudan (2000), which is obtained by concatenating the generalized Reed-Solomon codes with the Hadamard codes. Our key claim—Theorem 7—states that the codes $C^{GRS\text{-}H}$ (with an adequate choice of code parameters) are efficiently quantum list decodable as far as their codeword presence is relatively high. Theorem 1 follows immediately from this claim, because $C^{GRS\text{-}H}$ was already proven to be classically list decodable (Guruswami and Sudan, 2000). As the first step toward the proof of Theorem 7, we will demonstrate in Proposition 8 that this concatenated code family possesses efficient quantum list decodability, provided that the generalized Reed-Solomon codes are efficiently quantum list decodable. This claim will be proven in Section 3.2 by employing a technique of constructing an efficient "quantum reduction" between two quantumly corrupted codewords. An advantage of this proof technique is that it requires

no *soft information*, which is a key ingredient in the classical case of Guruswami and Sudan (1999, 2000).

Our next step is to show in Lemma 11 that the generalized Reed-Solomon codes are indeed efficiently quantum list decodable, by partially applying a *polynomial reconstruction algorithm* of Guruswami and Sudan (1999), as far as a target codeword has relatively high presence in a given quantumly corrupted codeword. Unfortunately, the use of such a classical algorithm makes the query complexity of our quantum list decoder quite high. On the contrary, as the presence becomes lower, it seems to become harder to solve efficiently the quantum list-decoding problem. For instance, when the presence is arbitrary close to a reciprocal of the code alphabet size, we can convert an efficient quantum list-decoding algorithm to an efficient quantum algorithm that even solves a certain NP-complete problem. This immediately leads to an unlikely consequence that every NP-problem can be solved efficiently on a quantum computer with high success probability. In a similar vein, we will present a direct connection between quantum list decodability of the generalized Reed-Solomon codes and the quantum solvability of two classical problems: the *noisy polynomial interpolation problem* (NPIP) of Naor and Pinkas (1999) and the *bounded distance vector problem* (BDVP), both of which will be defined in Sections 4.2–4.3. To be more precise, we will show that (1) if the generalized Reed-Solomon codes are quantumly list decodable, then the NPIP is quantumly solvable and (2) if the BDVP is quantumly solvable, then the generalized Reed-Solomon codes are quantumly list decodable.

Our quantum list-decoding algorithm for the aforementioned concatenated code finds an immediate application to certain types of problems. Our example in this paper is an *NBQP-search problem*, in which, given a polynomial-time quantum algorithm and an input instance, we want to find a classical witness of polynomial size that forces the algorithm to accept the input with high probability. We will show in Section 5 that solving this search problem on average implies solving it in worst case. This can be compared to a classical case of an NP-search problem of Kumar and Sivakumar (1999).

In line of the study on quantum list decoding, we will make a brief discussion in Section 6 on another notion of *local quantum list decoding* based on an *implicit-input implicit-output model* where an outcome of a list-decoding algorithm is a list of *descriptions* of quantum-circuit list decoders rather than a list of messages. Similarly to the classical case of Sudan, Trevisan, and Vadhan (2001), we can apply our quantum list decoder for generalized Reed-Solomon codes to conduct local quantum list decoding for the Reed-Müller codes. As an immediate consequence, we can prove the so-called *hardness amplification* of quantum circuits, following the argument of Sudan, Trevisan, and Vadhan (2001).

2 Foundations of Quantum List Decoding

This section explains basic notions and notation concerning quantum list decoding. Throughout this paper, let $\mathbb{N}$ denote the set of all *natural numbers* (i.e., nonnegative integers) and set $\mathbb{N}^+ = \mathbb{N} - \{0\}$. For any positive integers m and n with $m \leq n$, the notation $[m, n]_{\mathbb{Z}}$ means the integer set $\{m, m+1, m+2, \ldots, n\}$ and $[n]$ is the shorthand for $[1, n]_{\mathbb{Z}}$ whenever $n \geq 1$. For any number $q \in \mathbb{N}^+$, $\mathbb{F}_q$ (or $GF(q)$) denotes a *finite*

(Galois) field of size q. When q is a prime number, we often express the elements of $\mathbb{F}_q$ in terms of the numbers in $[0, q-1]_{\mathbb{Z}}$. We sometimes use a prime power q^m rather than a prime q. Conventionally, we also identify each vector in $(\mathbb{F}_q)^m$ with its corresponding element in $\mathbb{F}_{q^m}$. Let $\mathbb{Q}$ and $\mathbb{C}$ respectively denote the sets of all *rational numbers* and of all *complex numbers*. We further set $\mathbb{Q}^{\geq 0} = \{r \in \mathbb{Q} \mid r \geq 0\}$.

For a finite alphabet Σ, a *string* x over Σ is a finite sequence of symbols from Σ, and $|x|$ denotes the *length* of x (i.e., the number of all the occurrences of symbols in x).

2.1 Classical Block Codes

We briefly explain classical block (error-correcting) codes, which are key objects of our interest. Roughly speaking, a *(block) code* is a set of strings of the same length over a finite alphabet Σ and each string of a code is indexed by a message and is called a *codeword*. In this paper, we mostly deal with a *family of codes*, each of which corresponds to a different message length n in $\mathbb{N}$. Such a code family can be specified in general by a series $(\Sigma_n, I_n, \Gamma_n)$ of triplets composed of *message space* Σ_n, *index set* I_n, and *code alphabet* Γ_n for each *message length*[1] n (which serves as a "basis parameter" in this paper).

As standard nowadays in computational complexity theory, we view a code C (or $C^{(n)}$, to emphasize "n") for each fixed message length n as a "function" that maps $\Sigma_n \times I_n$ to Γ_n. For convenience, let the *code size* $N(n) = |\Sigma_n|$ and let the *code alphabet size* $q(n) = |\Gamma_n|$. It is also convenient to assume that $\Sigma_n = (\Sigma')^n$ for a certain fixed *message alphabet* Σ' so that n actually represents the *length* of messages in Σ_n over Σ'; in this case, $n = \log_{|\Sigma'|} N(n)$ holds for every length $n \in \mathbb{N}$. For instance, if $\Sigma' = \{0, 1\}$, then all messages can be expressed in binary. By abbreviating $C(x, y)$ as $C_x(y)$, we treat $C_x(\cdot)$ as a function mapping I_n to Γ_n and we call it a *codeword*, whose *block length* (or *code length*) $M(n)$ equals $|I_n|$. Since the elements in I_n serve as indices of block locations of a codeword, it is often assumed that $I_n = \{0, 1, \ldots, M(n)-1\}$ so that each element of I_n can be expressed in $\lceil \log_2 M(n) \rceil$ bits. For convenience, we also identify C_x with the vector $(C_x(0), C_x(1), \cdots, C_x(M(n)-1))$ in the *ambient space* $(\Gamma_n)^{M(n)}$ of dimension $M(n)$. Because we mainly work on a finite field, we often regard Γ_n as a finite field $\mathbb{F}_{q(n)}$ of order $q(n)$.

The *rate* of a code C is defined to be the ratio $n/M(n)$. The *(Hamming) distance* $d(C_x, C_y)$ between two codewords C_x and C_y is the number of non-zero components in the vector $C_x - C_y$. The *minimal distance* $d(C^{(n)})$ (or $d(n)$, in short) of the codes of message length n is the smallest distance between any pair of distinct codewords associated with the messages of length n. In contrast, $\Delta(C_x, C_y)$ denotes the *relative (Hamming) distance* $d(C_x, C_y)/M(n)$. The above-described code is simply called an $(M(n), n)_{q(n)}$-code[2] (or $(M(n), n, d(n))_{q(n)}$-code, to emphasize the minimal distance $d(n)$ of the code of message length n). For readability, we often drop a length parameter n from both subscripts and argument places whenever we discuss a set of codewords of

[1] This parameter is also known as the *dimension* or *information length* of a code.

[2] The reader should be aware that, in some literature, the notation $(M(n), \Gamma_n)_{q(n)}$ is used instead.

a "fixed" message length n. A *linear* $(M(n), n)_{q(n)}$*-code* forms a n-dimensional vector space in $\left(\mathbb{F}_{q(n)}\right)^{M(n)}$.

Hadamard Codes HAD. Let n be any message length used as a parameter, and let q be any prime number. A *q-ary Hadamard code family* $\mathrm{HAD}^{(q)} = \{\mathrm{HAD}^{(q,n)}\}_{n\in\mathbb{N}}$ consists of all $(q^n, n, q^n - q^{n-1})_q$-codes $\mathrm{HAD}^{(q,n)} : (\mathbb{F}_q)^n \times (\mathbb{F}_q)^n \to \mathbb{F}_q$ obtained as follows. For each message $x = (x_1, x_2, \ldots, x_n)$ in $(\mathbb{F}_q)^n$, $\mathrm{HAD}^{(q,n)}(x, r)$ equals $\sum_{i=1}^{n} x_i r_i \bmod q$, where $r = (r_1, r_2, \ldots, r_n)$ is in the index set $(\mathbb{F}_q)^n$.

(Normalized) Generalized Reed-Solomon Codes GRS. Let q be any prime number and let k and n be any two positive integers satisfying that $n \leq k \leq q$. A *(normalized) generalized Reed-Solomon code family* $\mathrm{GRS} = \{\mathrm{GRS}^{(k,n,q)}\}_{n,k\in\mathbb{N}}$ consists of all $(k, n, k-n+1)_q$-codes defined as follows. Let $x = (x_1, x_2, \ldots, x_n) \in (\mathbb{F}_q)^n$ be any message and let D_k be a fixed set of k distinct elements (called *code locators*) in $\mathbb{F}_q$. Let $\mathrm{GRS}^{(k,n,q)} : (\mathbb{F}_q)^n \times D_k \to \mathbb{F}_q$ be defined as $\mathrm{GRS}^{(k,n,q)}(x, r) = \sum_{i=1}^{n} x_i r^{i-1} \bmod q$, which is a polynomial of degree at most $n-1$ with $r \in D_k$. Occasionally, we expand the domain D_k of $\mathrm{GRS}_x^{(k,n,q)}$ to the entire field $\mathbb{F}_q$.

2.2 Quantumly Corrupted Codewords and Codeword Presence

A *quantum bit* (or a *qubit*, in short) is a unit vector in the complex space $\mathbb{C}^2$, and a *quantum state* is generally a tensor product of some of these qubits. To express such a quantum state, we customarily use Dirac's notation. For instance, a quantum state $|\phi\rangle$ of two qubits can be expressed as $|\phi_1\rangle \otimes |\phi_1\rangle$, where $|\phi_1\rangle$ and $|\phi_2\rangle$ are both qubits; however, we often abbreviate $|\phi_1\rangle \otimes |\phi_2\rangle$ as $|\phi_1\rangle|\phi_2\rangle$. An execution of a *quantum algorithm* on an input instance corresponds to a series of applications of unitary operations, and it is usually modeled by a "computation" of a *quantum Turing machine* (Bernstein and Vazirani, 1997; Yamakami, 1999, 2003) or a *quantum circuit* (Yao, 1993). We use the notation $\mathcal{A}(x)$ (or more formally, $\mathcal{A}|x\rangle$) to denote a quantum state obtained after executing quantum algorithm $\mathcal{A}$ on classical input x (which is formally given in the form of quantum state $|x\rangle$). When we refer to an *output* of $\mathcal{A}$ on x, we mean a classical string that is obtained by *measuring* (or *observing*) the quantum state $\mathcal{A}(x)$ in the standard computational basis, where a measurement is a projection onto a certain Hilbert space. For simplicity, we say that a quantum algorithm *runs in polynomial time* if its corresponding quantum Turing machine halts within time polynomial in the length of each input. Similar to the complexity classes P and NP, BQP denotes the collection of all (classical) decision problems that can be solved by quantum algorithms in polynomial time with success probability at least $2/3$. For more details on quantum computation, the reader may refer to, e.g., Nielsen and Chuang (2000).

Let us consider a quantum procedure that tries to encode a classical message into its codeword. In general, a quantum computation tends to interact with an outside system of a currently operating quantum system, causing a quantum corruption of the computation. Hence, our process of quantum encoding may be corrupted. A corrupted process of such quantum encoding can be described as an application of a certain form of unitary operator. As noted before, when $q(n)$ is a prime number, we represent each element in $\mathbb{F}_{q(n)}$ as an integer in $[0, q(n)-1]_{\mathbb{Z}}$, which is further expressed in binary. In their

2006 conference paper, Kawachi and Yamakami coined the terminology—a *quantum-computationally corrupted codeword* or *quantumly corrupted codeword*—to describe such a unitary operator O, with two fixed parameter functions $\ell(n)$ and $m(n)$ mapping $\mathbb{N}$ to $\mathbb{N}$, that satisfies the following condition: for any two strings $r \in I_n$ and $s \in \{0,1\}^{m(n)}$ and any number $\ell(n)$, there exists a quantum state $|\phi_{r,z}\rangle$ of $\ell(n)$ qubits such that

$$O|r\rangle|s\rangle|0^{\ell(n)}\rangle = \sum_{z\in\{0,1\}^{m(n)}} \alpha_{r,z}|r\rangle|s\oplus z\rangle|\phi_{r,z}\rangle, \tag{1}$$

where the notation $\oplus$ denotes the bitwise XOR, $|\phi_{r,z}\rangle$ indicates garbage information produced when we apply the operator O to the three registers, and the amplitudes $\{\alpha_{r,z}\}_{r,z}$ satisfy that $\sum_{z\in\{0,1\}^{m(n)}} |\alpha_{r,z}|^2 = 1$ for every index $r \in I_n$. Since O is a unitary operator, so is its *inverse* O^{-1}. Another important notion of Kawachi and Yamakami is "codeword presence" in O. The *presence* of codeword C_x in O, denoted $\mathrm{Pre}_O(C_x)$, is the average probability of obtaining the correct values $C_x(r)$ by a measurement over all indices $r \in I_n$; namely, $\mathrm{Pre}_O(C_x) = (1/M(n)) \sum_{r\in I_n} |\alpha_{r,C_x(r)}|^2$.

2.3 Asymptotic Behaviors of Codeword Presence

The value of *codeword presence* is a key to the performance of a quantum list decoder. We will briefly argue asymptotic behaviors of codeword presence for arbitrary quantumly corrupted codewords in a fashion similar to classical cases of Guruswami, Håstad, Sudan, and Zuckerman (2002). For this purpose, we need to expand the existing notions of presence and (Hamming) distance of codewords in a more general fashion. Notice that these generalized presence and distance are applied only to this subsection.

Let n be any message length and define W_n to be the set of all vectors $w = (w_{r,z})_{r\in I_n, z\in\mathbb{F}_{q(n)}} \in [0,1]^{q(n)M(n)}$ (where each $w_{r,z}$ may be viewed as the probability $|\alpha_{r,z}|^2$ of obtaining (r,z) after measuring a quantumly corrupted codeword) satisfying the restriction that $\sum_{z\in[0,q(n)-1]_{\mathbb{Z}}} w_{r,z} = 1$ for each index $r \in [0, M(n)-1]_{\mathbb{Z}}$, where $M(n) = |I_n|$. For every $w \in W_n$, it follows that $\sum_r \sum_z w_{r,z} = M(n)$. Next, we consider the set V_n of all codewords (viewed as a vector) $a = (a_r)_{r\in I_n} \in ([0, q(n)-1]_{\mathbb{Z}})^{M(n)}$. We embed each codeword a into W_n by the special mapping v, defined as $v(a) = (\delta^{(a)}_{r,z})_{r\in I_n, z\in\mathbb{F}_{q(n)}} \in \{0,1\}^{q(n)M(n)}$, where $\delta^{(a)}_{r,z}$ is 1 if $a(r) = z$, and 0 otherwise. Moreover, for any code (seen as a subset of V_n) $C^{(n)}$, let $v(C^{(n)}) = \{v(a) \mid a \in C^{(n)}\}$. Obviously, $v(V_n) \subseteq W_n$ holds.

Using the above notations, let us generalize the notions of distance and presence as follows. For any pair $v, w \in W_n$, we define $d(v,w) = M(n) - \langle v|w\rangle$, where $\langle\cdot|\cdot\rangle$ denotes the standard *inner product*. This generalized notion naturally expands the standard notion of the distance $d(\cdot,\cdot)$ because, for any $a, b \in V_n$, we have

$$d(v(a), v(b)) = M(n) - \langle v(a)|v(b)\rangle) = M(n) - |\{(r,z) \mid a(r) = b(r) = z\}| = d(a,b).$$

Moreover, for any two vectors $a \in V_n$ and $w \in W_n$, define $\mathrm{Pre}_w(a) = \frac{1}{M(n)}\langle v(a)|w\rangle$. We then obtain

$$\mathrm{Pre}_w(a) = \frac{M(n) - d(v(a), w)}{M(n)} = \frac{\langle v(a)|w\rangle}{M(n)} = 1 - \frac{d(v(a), w)}{M(n)}.$$

First, we wish to obtain an *asymptotic lower bound* of codeword presence in terms of minimal relative distance λ. For this purpose, we will introduce the notation $QL^{poly}(\lambda)$ for the minimal possible "presence" ε, with which, for an arbitrary family of block codes with minimal relative distance λ, the cardinality of all messages having codeword presence of at least ε is polynomially bounded. More precisely, let $C = \{C^{(n)}\}_{n\in\mathbb{N}}$ be any $(M(n), n, d(n))_{q(n)}$-code family and let $\Delta(C^{(n)}) = d(C^{(n)})/M(n)$ express the relative distance of $C^{(n)}$. For each pair $w \in W_n$ and $\varepsilon \in [0,1]$, we write $E(w,\varepsilon)$ for the set $\{a \in V_n \mid \mathrm{Pre}_w(a) \geq \varepsilon\}$. For any function $f : \mathbb{N} \to \mathbb{N}$ and any number $n \in \mathbb{N}$, the notation $presence(C,f)(n)$ denotes $\min\{\varepsilon \in \mathbb{R}^{\geq 0} \mid \forall w \in W_n[\,|E(w,\varepsilon) \cap C^{(n)}| \leq f(n)\,]\}$ and we set $Pre(C,f) = \limsup_{n\to\infty}\left\{\frac{presence(C,f)(n)}{M(n)}\right\}$. In addition, let $QL_f(\lambda) = \inf_{C:\Delta(C)\geq\lambda}\{Pre(C,f)\}$, where $\Delta(C) = \liminf_{n\to\infty}\{\Delta(C^{(n)})\}$. For each fixed constant $c \in \mathbb{N}$, we set $QL_c^{poly}(\lambda) = \sup_{a>0}\{QL_{f_a^{(c)}}(\lambda)\}$, where $f_a^{(c)}(n) = an^c$ for any number $n \in \mathbb{N}$. Finally, $QL^{poly}(\lambda)$ is set to be $\limsup_{c\to\infty}\{QL_c^{poly}(\lambda)\}$.

Proposition 2. *Let c be any positive constant and let λ be any number in $[0,1]$, representing a minimal relative distance. It holds that either $QL_c^{poly}(\lambda) \geq 1/q + (1-1/q)\left(1-\lambda/(1-1/q)+\lambda/an^c(1-1/q)\right)^{1/2}$ or $QL_c^{poly}(\lambda) \geq 1/q + (1-1/q)\left(1-\lambda/(1-1/q)\right)^{1/2}$. Therefore, $QL^{poly}(\lambda) \geq 1/q + (1-1/q)\left(1-\lambda(1-1/q)\right)^{1/2}$ follows.*

In certain extreme cases, it holds that $QL^{poly}(0) = 1$ and $QL^{poly}(1) \geq \sqrt{1/q} + (1-\sqrt{1/q})/q$. It remains open whether the equality $QL^{poly}(\lambda) = 1/q + (1-1/q)\left(1-\lambda(1-1/q)\right)^{1/2}$ holds or not.

Next, we will show an *asymptotic upper bound* of codeword presence, particularly, in terms of the rate of a "linear" $(M(n), n, d(n))_{q(n)}$-code family $C = \{C^{(n)}\}_{n\in\mathbb{N}}$. For convenience, we write $rate(C^{(n)})$ for the code rate $n/M(n)$ of $C^{(n)}$. Here, let R be any code rate in $[0,1]$ and let $f : \mathbb{N} \to \mathbb{N}$ be any function. We define $QU_f(R) = \sup_{C:rate(C)\geq R}\{Pre(C,f)\}$, where $rate(C) = \liminf_{n\to\infty}\{rate(C^{(n)})\}$. With this notation, for each constant $c > 0$, we write $QU_c^{const}(R)$ for $QU_{f_c}(R)$, where f_c is a constant function defined as $f_c(n) = c$ for all numbers $n \in \mathbb{N}$. Define $QU^{const}(R)$ to be $\limsup_{c\to\infty}\{QU_c^{const}(R)\}$.

Proposition 3. *Fix an odd prime number q. For every constant $c \in \mathbb{N}^+$ with $c > 2(q-1)$ and every code rate $R \in (0,1)$, it holds that $QU_c^{const}(R) \geq 1 - q^{-\frac{(1+2R)c-q}{(q-2)c}}$. Therefore, $QU^{const}(R) \geq 1 - q^{-\frac{1+2R}{q-2}}$ follows.*

For readability, we place the proofs of Propositions 2–3 in Appendix.

2.4 Kawachi-Yamakami Implicit-Input Explicit-Output Model

To formulate the notion of quantum list decoding, this paper deals with a specific model in which we *implicitly* take a quantumly corrupted codeword as a form of "oracle" and then we output a list of messages *explicitly* after accessing the oracle by way of oracle

queries. A process of making an oracle query and then receiving its oracle answer is conventionally assumed to take a unit time. Upon this *implicit-input explicit-output model*, the *quantum list-decoding problem* (QLDP) for a classical block code family C can be described as follows. First, let $C = \{C^{(n)}\}_{n\in\mathbb{N}}$ be any $(M(n), n, d(n))_{q(n)}$-code family with message space Σ_n and let $\mathcal{O}$ be any set of quantumly corrupted codewords for C. Taking a *bias parameter* $\varepsilon : \mathbb{N} \to [0, 1]$, we define the ε-QLDP as:

ε-QUANTUM LIST DECODING PROBLEM (ε-QLDP) FOR CODE FAMILY C WITH RESPECT TO $\mathcal{O}$

- INPUT: a message length n and a value $1/\varepsilon(n) > 0$.
- IMPLICIT INPUT: an oracle $O \in \mathcal{O}$ representing a quantumly corrupted codeword for $C^{(n)}$.
- OUTPUT: a list of messages including all messages $x \in \Sigma_n$ that satisfy the inequality $\mathrm{Pre}_O(C_x) \geq 1/q(n) + \varepsilon(n)$. For convenience, we refer to such a list as a *valid list* for the ε-QLDP.

Our goal is to solve the problem ε-QLDP for C using an efficient quantum algorithm that makes an oracle access to a given quantumly corrupted codeword in $\mathcal{O}$ with success probability at least $\delta(n)$, which is given as a *confidence parameter*. Here, let us formally introduce the notion of a *quantum list-decoding algorithm* (or simply, a *quantum list decoder*) that works with two parameters: bias ε and confidence δ.

Definition 4 (quantum list decoding). Let C be any code family, let $\varepsilon(n)$ be any bias parameter, and let $\delta(n)$ be any confidence parameter. A *quantum list-decoding algorithm* (or a *quantum list decoder*) for C with bias ε and confidence δ is a quantum algorithm $\mathcal{A}$ that solves the ε-QLDP for C with success probability at least $\delta(n)$. If $\mathcal{A}$ further runs in time polynomial in $(n, 1/\varepsilon(n), 1/\delta(n))$, it is called a *polynomial-time quantum list-decoding algorithm* for C.

The *list size* of a quantum list decoder with respect to input size n refers to the maximal size of any valid list produced by the algorithm on any input of size n. In certain applications, the list size of a single valid list plays a crucial role; for instance, when a quantum list decoder produces only a single valid list L (along all measured outcomes) with probability at least $\delta(n)$, certain "advice" of size $\lceil \log_{|\Sigma|} |L| \rceil$ over a message alphabet Σ may help specify a hidden message x *uniquely* with the same success probability.

A close connection between quantum list decoding and (strongly) quantum one-way functions was exhibited by Kawachi and Yamakami (2010). The rest of this subsection briefly discusses a further relationship between quantum list decoding and a restricted form of quantum one-way functions, called *quantum super one-way functions*, which can be seen as a natural extension of quantum one-way permutations.

Definition 5 (quantum super one-wayness). Let f be any function mapping Σ^* to Σ^* with length function $\ell : \mathbb{N} \to \mathbb{N}$, that is, $|f(x)| = \ell(|x|)$ for every x. This function f is called *quantum super one-way* if (i) there exists a polynomial-time quantum algorithm $\mathcal{A}$ such that, for every input x of length n, $\mathcal{A}|x\rangle|0^{\ell(n)}\rangle|0^{e(n)}\rangle = |x\rangle|f(x)\rangle|\phi_x\rangle$ holds for a certain unit-norm quantum state $|\phi_x\rangle$ of $e(n)$ qubits and (ii) for any positive

polynomial p and any polynomial-time quantum algorithm $\mathcal{B}$, the probability that $\mathcal{B}$ on input $|1^n\rangle|f(x)\rangle|\phi_x\rangle$ outputs x of length n is at most $1/p(n)$ for all but finitely many strings x.

In comparison with Definition 5, the quantum one-wayness formulated by Kawachi and Yamakami (2010) requires that $\mathcal{B}|f(x)\rangle$ outputs x only with negligible probability whereby the information $|\phi_x\rangle$ is *hidden* from the adversary $\mathcal{B}$ who tries to invert f. Definition 5, on the contrary, indicates that $\mathcal{B}$ cannot output x with non-negligible probability even though $|\phi_x\rangle$ is given to $\mathcal{B}$ besides $f(x)$ as supplemental information. In computational cryptography, this notion naturally arises. A typical example of super one-way function is a *quantum one-way permutation* obtained by replacing further the quantum state $|\phi_x\rangle$ in Definition 5 with $|0^m\rangle$, which is obtained, for example, by uncomputing a deterministic procedure that computes $f(x)$ from x.

In what follows, for any index i, the notation $(f(x))_i$ denotes the ith bit of the value $f(x)$ whenever $1 \leq i \leq |f(x)|$.

Lemma 6. *Let f be any quantum super one-way function with its* length function $m(n) \in n^{O(1)}$ *(i.e., $|f(x)| = m(|x|)$). Consider an $(m(n), n, d(n))_{q(n)}$-code C whose codeword $C_x(r)$ is $(f(x))_r$. For every positive polynomial p, this code C cannot be polynomial-time quantum list decodable with confidence* $1/p(n)$.

Proof. Let f be a quantum super one-way function with its length function $m(n)$, where $m(n)$ is polynomially bounded, and consider an $(m(n), n, d(n))_{q(n)}$-code C satisfying $C_x(r) = (f(x))_r$ for any x and r. Since a certain polynomial-time quantum algorithm must compute f exactly as stated in Definition 5, by modifying this algorithm slightly, we obtain another polynomial-time quantum algorithm, say, $\mathcal{A}$ that computes $C(x, r)$. Without loss of generality, we may assume that, for every $n \in \mathbb{N}^+$, every $x \in \Sigma^n$, and every $r \in [\ell(n)]$, $\mathcal{A}|x\rangle|r\rangle|0\rangle|0^{e(n)}\rangle = |x\rangle|r\rangle|C_x(r)\rangle|\phi_x\rangle$ holds for a certain quantum state $|\phi_x\rangle$ that depends only on x. Here, we fix x of length n arbitrarily and define $O_x|r\rangle|s\rangle|0^{e(n)}\rangle = |r\rangle|s \oplus C_x(r)\rangle|\phi_x\rangle$ for any strings r and s. Notice that $\mathrm{Pre}_{O_x}(C_x) = 1$ holds. Toward a contradiction, assume that C has a polynomial-time quantum list decoder $\mathcal{B}$ such that, since the presence of C_x in O_x is 1, $\mathcal{B}$ on input 1^n produces the hidden string x with probability at least $1/p(n)$ for a certain fixed positive polynomial p, where "1^n" indicates an input representing "n" in the definition of the ε-QLDP. We want to invert f in polynomial time. For this goal, we define a quantum algorithm $\mathcal{D}$ as follows.

> On input $|1^n\rangle|f(x)\rangle|\phi_x\rangle$, where $n = |x|$, we run the quantum list decoder $\mathcal{B}$ on input 1^n using O_x as an oracle. However, whenever $\mathcal{B}$ makes an oracle query $|r\rangle|s\rangle|t\rangle$ to the oracle O_x, we simulate the behavior of O_x as follows. We generate an oracle answer $|r\rangle|s \oplus (f(x))_r\rangle|\phi_x\rangle$ directly using the input information. Finally, we output an outcome of $\mathcal{B}$. Since $\mathrm{Pre}_{O_x}(C_x) = 1$, the outcome of $\mathcal{B}$ must be x itself.

The above algorithm $\mathcal{D}$ thus inverts f correctly with probability at least $1/p(n)$. This implies that f cannot be quantum super one-way, a contradiction against our assumption. Therefore, C is not polynomial-time quantum list decodable with confidence $1/p(n)$. □

3 Codes of Polynomially Small Rate

The proof of our main theorem (Theorem 1) requires a suitable code family of polynomially small code rate over a fixed code alphabet. Such a code family can be obtained by Forney's (1966) idea of concatenating two appropriate code families. In Section 3.1, we will claim that this concatenated code family is efficiently quantumly list decodable for a certain choice of code parameters. This claim—Theorem 7—then leads to the main theorem. Therefore, our primary goal is to conduct necessary ground work that leads to the proof of Theorem 7. For the sake of readability, we will split the proof into two claims—Proposition 8 and Lemma 11—and this section will prove only the proposition, leaving the lemma to Section 4. A key proof technique of this section in handling the concatenated code is a *quantum reduction* between two quantumly corrupted codewords, maintaining "similar" codeword presence values.

3.1 Concatenated Codes

A typical way to build a family of classical block codes that have desired code rate and desired code alphabet size is to compose two appropriate block codes with certain necessary code properties. This is Forney's (1966) novel method of creating so-called *concatenated codes*. In our case, concatenating an appropriate generalized Reed-Solomon code with its matching Hadamard code, we can build a code of polynomially small code rate and constant code alphabet size. For such a code family, we will prove its efficient quantum list decodability, provided that the generalized Reed-Solomon codes have efficient quantum list decoders.

More formally, let us consider two block codes C_1 and C_2 such that C_1 is an $(M_1, n_1, d_1)_{q^{n_2}}$-code and C_2 is an $(M_2, n_2, d_2)_q$-code. Let $x = (x_1, x_2, \ldots, x_{n_1})$ be any message of length n_1, where each entry x_i is taken from Σ^{n_2} over a q-letter alphabet Σ. Since x_i can be expressed as an n_2-letter string, x can be viewed as a string of total length $n_1 n_2$ over Σ. By taking the inner code C_2 concatenated with the outer code C_1, the concatenated code $C = C_2 \odot C_1$ is defined as $C(x, r, s) = C_2(C_1(x, r), s)$ for every triplet (x, r, s). This code C becomes an $(M_1 M_2, n_1 n_2, d)_q$-code with d satisfying $d \geq d_1 d_2$, where $d_1 d_2$ is called the *design distance*.

For our purpose of this section, we choose the concatenated code $C^{GRS\text{-}H}[n, q, \theta]$ given by Guruswami and Sudan (2000). This concatenated code is obtained from a certain generalized Reed-Solomon code used as an outer code together with an appropriate Hadamard code used as an inner code. Following Guruswami and Sudan (2000), here we choose three parameters (n, q, θ) with $n, q \in \mathbb{N}$ and $\theta \in [0, 1]$ that satisfy $n \geq 1$, $q \geq 2$, $n = mq^m\theta$, and $q^m\theta \in \mathbb{N}$ for a certain number $m \in \mathbb{N}$. In what follows, we freely identify elements in $(\mathbb{F}_q)^n$ with elements in $\mathbb{F}_{q^n}$ in the standard fashion.

Concatenated Code $C^{GRS\text{-}H}[n, q, \theta]$. The concatenated code $C^{GRS\text{-}H}[n, q, \theta]$ is defined by $C^{GRS\text{-}H}[n, q, \theta] = \mathrm{HAD}^{(q,m)} \odot \mathrm{GRS}^{(q^m, q^m\theta, q^m)}$. This is a $(q^{2m}, n, d)_q$-code, where $n = mq^m\theta$ and $d \geq (1-1/q)(1-\theta)q^{2m}$ (design distance). From $n = mq^m\theta$, we obtain $\log n = \log mq^m\theta$, from which $m = \frac{\log n - \log m + \log(1/\theta)}{\log q}$ follows. This implies $\frac{\log(1/\theta)}{\log q} \leq m \leq n$; thus, $q^m = \frac{n}{m\theta} \leq \frac{n \log q}{\theta \log(1/\theta)}$. As long as q is fixed and $\theta = \Omega(1/n^k)$

holds for a certain constant $k \in \mathbb{N}^+$, q^m is upper-bounded by $O(n^{k+1}/\log n)$. Hence, the code rate n/q^{2m} is lower-bounded by $\frac{c \log n}{n^k}$ for a certain constant $c > 0$.

This concatenated code family $C^{GRS\text{-}H} = \{C^{GRS\text{-}H}[n, q, \theta]\}_{n,q,\theta}$ is proven by Guruswami and Sudan (2000) to be efficiently classically list decodable; that is, there exists a polynomial-time probabilistic algorithm that produces, from any classically corrupted codeword (or conventionally, a *received word*) w, a list containing all messages x whose codewords are all at distance close to w. To prove Theorem 1, it therefore suffices to show that the code family $C^{GRS\text{-}H}$ is also quantumly list decodable in an efficient manner for appropriately chosen parameters. In a more general fashion, we intend to show the following statement. Let $\mathcal{T}$ denote the collection of all tuples (n, m, q, θ) such that $m, q, \in \mathbb{N}^+$, $q \geq 2$, $\theta \in [0, 1]$, $q^m\theta \in \mathbb{N}$, and $n = mq^m\theta$.

Theorem 7. *For each $n \in \mathbb{N}^+$, assume that a parameter tuple $(m, q, \theta, \varepsilon, \delta)$ satisfies the following conditions: $(n, m, q, \theta) \in \mathcal{T}$, $\varepsilon, \delta \in [0, 1]$, $2(1 - 1/q)^2(1/M + \varepsilon') < \varepsilon^2$, and $2(1 - 1/q)^2(1 - 1/M - \varepsilon')\sqrt{\frac{\Delta}{1+M\varepsilon'}} < \varepsilon^2$ for a certain $\varepsilon' \in (0, 1)$, where $M = q^m$ and $\Delta = 2(n - 1)\log(M^2/(1 - \delta))$. The concatenated code family $C^{GRS\text{-}H} = \{C^{GRS\text{-}H}[n, q, \theta]\}_{n,q,\theta}$ with the above conditions has a quantum list decoder with bias ε and confidence δ running in time polynomial in $(n, q, 1/\varepsilon, 1/\delta, 1/(1 - \delta))$.*

Here, we give the proof of Theorem 1 using Theorem 7. Although it is possible to relax the conditions stated in Theorem 7 further, they are sufficient to prove Theorem 1.

Proof of Theorem 1. Fix a prime number q, a constant $k \in \mathbb{N}^+$, and a confidence parameter δ. Here, we set $\varepsilon = 1/n^k$. We also choose other parameters $(n, m, q, \theta) \in \mathcal{T}$ and $M = q^m = O(n^\ell) \cap \Omega(n^{8k+4})$ for a certain fixed constant $\ell \geq 8k + 4$ and consider the code $C^{GRS\text{-}H}[n, q, \theta]$. In this case, it holds that $\theta = n/mM = \Omega(1/n^{\ell-1})$ since $m = \log M/\log q$. This guarantees the polynomially small code rate of C. Let us define $t(n) = M^2$. Note that the value $\Delta = 2(n - 1)\log(M^2/(1 - \delta))$ satisfies $\Delta = O(n \log n) = O(n^{1.4})$. For simplicity, set $\alpha = 1/M + \varepsilon'$. Now, defining $\varepsilon' = \sqrt{1/M}$, we obtain $M\alpha = 1 + \sqrt{M} = \Omega(n^{4k+2})$. It thus follows that $2(1-\alpha)\sqrt{\frac{\Delta}{M\alpha}} = O(n^{0.7}/n^{2k+1})$. Since $\varepsilon = 1/n^k$, we obtain $2(1 - \alpha)\sqrt{\frac{\Delta}{M\alpha}} < \varepsilon^2$ and $2\alpha < \varepsilon^2$ for any sufficiently large n. Since all premises of Theorem 7 are fulfilled, there must exist a quantum list decoder with bias ε and confidence δ. This quantum list decoder runs in time polynomial in n since the parameters (q, δ) are constants. This completes the proof. □

Let us return to Theorem 7. This theorem, in fact, follows from two technical claims: Proposition 8 and Lemma 11. We will prove in Proposition 8 that the concatenated code family $C^{GRS\text{-}H} = \{C^{GRS\text{-}H}[n, q, \theta]\}_{n,q,\theta}$ has a polynomial-time quantum list decoder for an appropriate choice of three parameters (n, q, θ), assuming that the generalized Reed-Solomon codes are quantum list decodable in polynomial time. This last assumption will be later eliminated, in Lemma 11, completing the proof of Theorem 7.

Proposition 8. *For each $n \in \mathbb{N}^+$, let $(q, \theta, m, \varepsilon, \varepsilon', \delta)$ satisfy the following conditions: $(n, m, q, \theta) \in \mathcal{T}$, $\varepsilon, \varepsilon', \delta \in [0, 1]$, and $\varepsilon^2 \geq (1 - 1/q)^2(1/M + \varepsilon')$. If the $(M, M\theta, (1 -$*

$\theta)M+1)_M$-generalized Reed-Solomon code has a quantum list decoder with bias ε' and confidence δ running in time polynomial in $(n, q, 1/\varepsilon', 1/\delta, 1/\theta)$, where $M = q^m$, then $C^{GRS\text{-}H}[n, q, \theta]$ has a quantum list decoder with bias ε and confidence δ running in time polynomial in $(n, q, 1/\varepsilon', 1/\delta, 1/\theta)$.

Note that the confidence δ for the GRS-code in Proposition 8 is carried over to the confidence for the concatenated code $C^{GRS\text{-}H}[n, q, \theta]$. The proposition is an important ingredient of Theorem 7 and its proof will be given in the subsequent subsection.

3.2 A Quantum Reduction Technique

Aiming at proving Proposition 8, we wish to construct a "quantum reduction" between two quantumly corrupted codewords. Such a reduction, say, from O to O' can be described as a quantum algorithm that, on input of the form $|r\rangle|s\rangle|t\rangle$, computes the outcome $O'|r\rangle|s\rangle|t\rangle$ by invoking a number of oracle calls to O as well as O^{-1}. This can be seen as a strong form of well-known *Turing reduction* between two languages.

Here, let C be any $(q^m, n/m)_{q^m}$-code, which is, as before, treated as a function $C(x, r)$ mapping from $(\mathbb{F}_q)^{\frac{n}{m}} \times \mathbb{F}_{q^m}$ to $\mathbb{F}_{q^m}$ whenever $n/m \in \mathbb{N}^+$. Recall that we freely identify $(\mathbb{F}_q)^m$ with $\mathbb{F}_{q^m}$. As a technical lemma essential for the proof of Proposition 8, we will show a general result concerning a concatenated code $D = \mathrm{HAD}^{(q,m)} \odot C$. Since the q-ary Hadamard code $\mathrm{HAD}^{(q,m)}$ is used as an inner code, we can rephrase D as

$$D(x, r, s) = C(x, r) \cdot s \mod q$$

for any $r, s \in \mathbb{F}_{q^m}$ and any $x \in (\mathbb{F}_{q^m})^n$.

In what follows, let us aim at constructing a quantum reduction between quantumly corrupted codewords O_C and O_D associated with the codes C and D, respectively. For convenience, we introduce new terminology. For any unitary transform U, we say that a quantum algorithm $\mathcal{A}$ *realizes* U if, for any *basis* quantum state $|r\rangle$, $\mathcal{A}$ on input $|r\rangle$ exactly produces the quantum state $U|r\rangle$. This notion will help describe a quantum reduction from O_C to O_D.

Lemma 9. *Let C and D be the codes given as above. For any quantumly corrupted codeword O_C for C, there exist a polynomial-time quantum algorithm $\mathcal{A}$ and a quantumly corrupted codeword O_D for D such that*

1. $\mathrm{Pre}_{O_D}(D_x) = 1/q + (1 - 1/q)\,\mathrm{Pre}_{O_C}(C_x)$*; and*
2. *$\mathcal{A}$ realizes O_D with one oracle access to O_C.*

For the proof of Proposition 8, we need to weaken the notions of "quantumly corrupted codeword" and "realization." A *generalized quantumly corrupted codeword* O is defined by Eq.(1) except that we require only the inequality $\sum_z |\alpha_{r,z}|^2 \leq 1$ among the amplitudes $\{\alpha_{r,z}\}_{r,z}$. The codeword presence of C_x in each of the operators O_k is defined as before. Let $\mathcal{O} = \{O_k\}_{k\in[q-1]}$ denote a series of generalized quantumly corrupted codewords. For this series $\mathcal{O}$, we also define the *average (codeword) presence* $av\mathrm{Pre}_{\mathcal{O}}(C_x)$ of C_x in $\mathcal{O}$ to be $(1/(q-1))\sum_{k\in[q-1]} \mathrm{Pre}_{O_k}(C_x)$. For a series $\mathcal{U} = \{U_k\}_{k\in[q-1]}$ of unitary operations, we say that a quantum algorithm $\mathcal{A}$ *weakly realizes* $\mathcal{U}$ if $\mathcal{A}$ on input $|k\rangle|r\rangle|0\rangle$ generates a certain quantum state and, after tracing out the third register by the observable $|0\rangle$, it becomes $|k\rangle \otimes U_k|r\rangle$.

Lemma 10. *Let C and D be the codes given as above. For any quantumly corrupted codeword O_D for D, then there exist a polynomial-time quantum algorithm $\mathcal{A}$ and a series $\mathcal{O} = \{O_k\}_{k\in[q-1]}$ of generalized quantumly corrupted codewords for C such that*

1. $av\mathrm{Pre}_{\mathcal{O}}(C_x) \geq (q/(q-1))^2(\mathrm{Pre}_{O_D}(D_x) - 1/q)^2$*; and*
2. *$\mathcal{A}$ weakly realizes $\mathcal{O}$ with one oracle access to each of O_D and O_D^{-1}.*

Lemma 10 gives a fast quantum reduction from O_D to O_C. From this lemma directly follows Proposition 8. Before proving Lemmas 9–10, we briefly describe the proof of the proposition.

Proof of Proposition 8. Let $n \in \mathbb{N}^+$ be any length parameter and assume that all other parameters $(m, q, \theta, \varepsilon, \varepsilon', \delta)$ satisfy the premise of the proposition. Hereafter, we set $M = q^m$ and $D = C^{GRS\text{-}H}[n, q, \theta]$ for brevity. Let us assume that the $(M, M\theta, (1-\theta)M+1)_M$-generalized Reed-Solomon code has a polynomial-time quantum list decoder, say, $\mathcal{A}$ with bias ε' and confidence δ. Take any quantumly corrupted codeword O for D. Our goal here is to find from O all messages x that satisfy the inequality $\mathrm{Pre}_O(C_x) \geq 1/q + \varepsilon$ in time polynomial in $(n, q, 1/\varepsilon', 1/\delta, 1/\theta)$ with confidence δ.

Since $D = \mathrm{HAD}^{(q,m)} \odot \mathrm{GRS}^{(M,M\theta,M)}$, Lemma 10 helps reduce O to a series $\mathcal{O}' = \{O'_k\}_{k\in[q-1]}$ of generalized quantumly corrupted codewords for the outer code $\mathrm{GRS}^{(M,M\theta,M)}$ so that $\mathcal{O}'$ can be weakly realized by a certain polynomial-time quantum algorithm, say, $\mathcal{B}$ with the following average presence condition:

$$\begin{aligned} av\mathrm{Pre}_{\mathcal{O}'}\left(\mathrm{GRS}_x^{(M,M\theta,M)}\right) &\geq \left(\frac{q}{q-1}\right)^2 \left(\mathrm{Pre}_O(D_x) - \frac{1}{q}\right)^2 \\ &\geq \left(\frac{q}{q-1}\right)^2 \varepsilon^2 \quad \geq \quad \frac{1}{M} + \varepsilon', \end{aligned}$$

where the last inequality follows directly from the bound $\varepsilon^2 \geq (1-1/q)^2(1/M+\varepsilon')$, which is given as a part of the premise of the proposition. In other words, the average value of $\mathrm{Pre}_{O'_k}(\mathrm{GRS}_x^{(M,M\theta,M)})$ over all $k \in [q-1]$ is lower-bounded by $1/M + \varepsilon'$. Thus, we can choose an index $k_0 \in [q-1]$ for which $\mathrm{Pre}_{O'_k}(\mathrm{GRS}_x^{(M,M\theta,M)}) \geq 1/M + \varepsilon'$. By our assumption, for this k_0, $\mathcal{A}$ correctly produces a list including all messages x satisfying $\mathrm{Pre}_{O'_k}(\mathrm{GRS}_x^{(M,M\theta,M)}) \geq 1/M + \varepsilon'$ with confidence δ.

Let us consider the following quantum algorithm, which uses $\mathcal{A}$ and $\mathcal{B}$ as subroutines.

> On input, we first set $k = 0$ and, by incrementing k by one, we inductively run the quantum list decoder $\mathcal{A}$ with O'_k as an oracle to produce a list of message candidates. During inductive steps, we always append new candidates to the existing list. Whenever a query is made, we run $\mathcal{B}$ to generate its oracle answer. This is possible because $\mathcal{B}$ weakly realizes $\mathcal{O}'$. Eventually, we reach k_0 and we then obtain a list containing of all messages x satisfying $\mathrm{Pre}_{O'}(\mathrm{GRS}_x^{(M,M\theta,M)}) \geq 1/M + \varepsilon'$ with probability at least δ.

This algorithm is obviously a quantum list decoder and it produces with confidence δ a list that contains all messages x satisfying $\mathrm{Pre}_O(D_x) \geq 1/q + \varepsilon$. This completes the proof. □

Next, we want to prove Lemmas 9–10. We begin with the proof of Lemma 10.

Proof of Lemma 10. Let C be any $(q^m, n/m)_{q^m}$-code. We denote by D the concatenated code $\mathrm{HAD}^{(q,m)} \odot C$ and assume that O_D satisfies $O_D|r,s\rangle|u\rangle|0^{\ell(n)}\rangle = \sum_{z\in\mathbb{F}_q} \alpha_{r,s,z}|r,s\rangle|u \oplus z\rangle|\phi_{r,s,z}\rangle$ for any $r, s \in \mathbb{F}_{q^m}$, where $\ell(n)$ indicates the size of garbage information $|\phi_{r,s,z}\rangle$. Note that $\mathrm{Pre}_{O_D}(D_x) = q^{-2m} \sum_{r,s\in\mathbb{F}_{q^m}} |\alpha_{r,s,D_x(r,s)}|^2$ and that $\sum_{z\in\mathbb{F}_q} |\alpha_{r,s,z}|^2 = 1$ for every pair (r,s).

We wish to define the desired quantum algorithm $\mathcal{A}$ and the desired series $\mathcal{O} = \{O_k\}_k$ of generalized quantumly corrupted codewords that can be weakly realized by $\mathcal{A}$ using O_D as an oracle. To describe the algorithm $\mathcal{A}$, we utilize a special unitary transform U over $[q-1]$ acting as $U|0\rangle = (1/\sqrt{q-1}) \sum_{k\in[q-1]} |k\rangle$ as well as a *quantum Fourier transform* F_q over $\mathbb{F}_q$ that acts as $F_q|s\rangle = q^{-1/2} \sum_{w\in\mathbb{F}_q} \omega_q^{s\cdot w}|w\rangle$ for any $s \in \mathbb{F}_q$. It was proven by van Dam, Hallgren, and Ip (2006) that F_q can be approximated to within error η on a quantum computer in time polynomial in $(\log q, \log(1/\eta))$.

QUANTUM ALGORITHM $\mathcal{A}$:

(1) Start with an initial quantum state $|\psi_1\rangle = |k\rangle|r\rangle|0^m\rangle|0\rangle|0^\ell\rangle$.

(2) By applying the quantum Fourier transform $(F_q)^m$ to the third register, we generate the quantum state $|\psi_2\rangle = q^{-m/2} \sum_{s\in(\mathbb{F}_q)^m} |k\rangle|r,s\rangle|0\rangle|0^\ell\rangle$, where $|r,s\rangle$ is a shorthand for $|r\rangle|s\rangle$.

(3) Apply O_D to the last three registers. This step transforms the quantum state $|\psi_2\rangle$ into $|\psi_3\rangle = q^{-m/2} \sum_{s\in(\mathbb{F}_q)^m} \sum_{z\in\mathbb{F}_q} \alpha_{r,s,z}|k\rangle|r,s\rangle|z\rangle|\phi_{r,s,z}\rangle$.

(4) Apply the *phase encoding* of Kawachi and Yamakami (2010); that is, encode the content of the fourth register into the "phase" together with the information on k to obtain $|\psi_4\rangle = q^{-m/2} \sum_{s\in(\mathbb{F}_q)^m} \sum_{z\in\mathbb{F}_q} \omega_q^{k\cdot z}\alpha_{r,s,z}|k\rangle|r,s\rangle|z\rangle|\phi_{r,s,z}\rangle$.

(5) Apply O_D^{-1}, the *inverse* of O_D, to the last four registers. The resulted state $|\psi_5\rangle$ can be expressed as $\sum_{s\in(\mathbb{F}-q)^m} \sum_{z\in\mathbb{F}_q} \beta_{k,r,s,z}|k\rangle|r,s\rangle|0\rangle|0^\ell\rangle + |k\rangle|\Delta_{k,r}\rangle$ with certain amplitudes $\beta_{k,r,s,z}$ and a certain vector $|\Delta_{k,r}\rangle$ whose last two registers does not contain the term $|0\rangle|0^\ell\rangle$. Each amplitude $\beta_{k,r,s,z}$ is calculated as

$$\beta_{k,r,s,z} = \langle k|\langle r|\langle s|\langle 0|\langle 0^\ell|I \otimes O_D^{-1}|\psi_k\rangle = \frac{1}{q^{m/2}}\omega_q^{k\cdot z}|\alpha_{r,s,z}|^2,$$

where I is the identity transform. The quantum state $|\psi_5\rangle$ is thus written in the form

$$\frac{1}{q^{m/2}} \sum_{s\in(\mathbb{F}_q)^m} \sum_{z\in\mathbb{F}_q} \omega_q^{k\cdot z}|\alpha_{r,s,z}|^2|k\rangle|r\rangle|s\rangle|0\rangle|0^\ell\rangle + |k\rangle|\Delta_{k,r}\rangle.$$

(6) Focusing on the last two registers, if they contain $|0\rangle|0^\ell\rangle$, then we multiply the content s of the third register by k to obtain $k \cdot s$ (seen as a scalar multiplication of a vector); otherwise, do nothing. Note that $k \cdot s$ is in $(\mathbb{F}_q)^m$ since $s \in (\mathbb{F}_q)^m$ and $k \in \mathbb{F}_q$. Let $|\psi_6\rangle$ denote the obtained quantum state.

(7) Similarly, whenever $|0\rangle|0^\ell\rangle$ appears in the last two registers, apply the inverse of the quantum Fourier transform $(F_q^{-1})^m$ to the third register. This transform produces the quantum state $|\psi_7\rangle = \sum_{w\in(\mathbb{F}_q)^m} \gamma_{k,r,w}|k\rangle|r\rangle|w\rangle|0\rangle|0^\ell\rangle + |k\rangle|\Delta_{k,r}\rangle$, where $\gamma_{k,r,w}$ is a complex number given as

$$\gamma_{k,r,w} = \langle r|\langle w|\langle 0|\langle 0^\ell|(F|\psi_6\rangle) = \frac{1}{q^m}\sum_{s\in(\mathbb{F}_q)^m}\sum_{z\in\mathbb{F}_q}\omega_q^{k(z-w\cdot s)}|\alpha_{r,s,z}|^2,$$

where $F = I\otimes I\otimes (F_q^{-1})^m\otimes I$.

(8) Observe the last register in state $|0\rangle|0^\ell\rangle$ and discard the term $|\Delta_{k,r}\rangle$. Finally, output the final quantum state $|\psi_8\rangle = \sum_{w\in(\mathbb{F}_q)^m}\gamma_{k,r,w}|k\rangle|r\rangle|w\rangle$. This finishes the description of $\mathcal{A}$.

We define $\mathcal{O} = \{O_k\}_{k\in[q-1]}$, where each O_k is a generalized quantumly corrupted codeword that is realized by $\mathcal{A}$ with $|k\rangle$ in the first register.

To complete the proof, we need to estimate the average presence $av\mathrm{Pre}_{\mathcal{O}}(C_x)$ of C_x in $\mathcal{O}$. For each index $k \in [q-1]$, the presence of C_x in O_k is exactly $\mathrm{Pre}_{O_k}(C_x) = q^{-m}\sum_{r\in\mathbb{F}_{q^m}}|\gamma_{k,r,C_x(r)}|^2$, which equals $q^{-m}\sum_{r\in\mathbb{F}_q}\left|q^{-m}\sum_s\sum_z\omega_q^{k(z-w\cdot s)}|\alpha_{r,s,z}|^2\right|^2$. It thus follows that

$$\begin{aligned}
av\mathrm{Pre}_{\mathcal{O}}(C_x) &= \frac{1}{q-1}\sum_{k\in[q-1]}\frac{1}{q^m}\sum_{r\in\mathbb{F}_{q^m}}\left|\frac{1}{q^m}\sum_{z\in\mathbb{F}_q}\sum_{s\in(\mathbb{F}_q)^m}\omega_q^{k(z-D_x(r,s))}|\alpha_{r,s,D_x(r,s)}|^2\right|^2\\
&= \frac{1}{q^m(q-1)}\sum_k\sum_r\left|\sum_{j\in\mathbb{F}_q}\omega_q^{k\cdot j}\left(\frac{1}{q^m}\sum_s|\alpha_{r,s,D_x(r,s)+j}|^2\right)\right|^2\\
&\geq \frac{1}{q^{2m}(q-1)^2}\left|\sum_k\sum_r\sum_{j\in\mathbb{F}_q}\omega_q^{k\cdot j}\left(\frac{1}{q^m}\sum_s|\alpha_{r,s,D_x(r,s)+j}|^2\right)\right|^2,
\end{aligned}$$

where the last inequality follows from $\sum_{i=1}^n a_i^2 \geq \frac{1}{n}\left(\sum_{i=1}^n a_i\right)^2$. We therefore obtain

$$av\mathrm{Pre}_{\mathcal{O}}(C_x) \geq \frac{1}{(q-1)^2}\left|\sum_{k\in[q-1]}\sum_{j\in\mathbb{F}_q}\omega_q^{k\cdot j}\left(\frac{1}{q^{2m}}\sum_{r\in\mathbb{F}_q}\sum_{s\in(\mathbb{F}_q)^m}|\alpha_{r,s,D_x(r,s)+j}|^2\right)\right|^2.$$

For each index $j\in\mathbb{F}_q$, we write β_j for $q^{-2m}\sum_r\sum_s|\alpha_{r,s,D_x(r,s)+j}|^2$. Similarly to the proof of Lemma 4.5 of Kawachi and Yamakami (2010), we can derive

$$\begin{aligned}
av\mathrm{Pre}_{\mathcal{O}}(C_x) &\geq \frac{1}{(q-1)^2}\left|\sum_{k\in[q-1]}\beta_0 + \sum_{1\leq j<q}\left(\sum_{k\in[q-1]}\omega_q^{k\cdot j}\right)\beta_j\right|^2\\
&= \frac{1}{(q-1)^2}\left|(q-1)\beta_0 - \sum_{1\leq j<q}\beta_j\right|^2 = \frac{1}{(q-1)^2}\left|q\beta_0 - 1\right|^2,
\end{aligned}$$

because $\sum_{j\in\mathbb{F}_q}\beta_j = 1$ and $\sum_{k\in\mathbb{F}_q}\omega_q^{k\cdot j} = 0$ for any $j \neq 0$. Since $\mathrm{Pre}_{O_D}(D_x) = q^{-2m}\sum_{r,s\in(\mathbb{F}_q)^m}|\alpha_{r,s,D_x(r,s)}|^2 = \beta_0$, it follows that

$$av\mathrm{Pre}_{\mathcal{O}}(C_x) \geq \frac{1}{(q-1)^2}\left|q\cdot\mathrm{Pre}_O(D_x) - 1\right|^2 = \left(\frac{q}{q-1}\right)^2\left(\mathrm{Pre}_{O_D}(D_x) - \frac{1}{q}\right)^2.$$

This completes the proof of Lemma 10. □

Next, we give the remaining proof of Lemma 9.

Proof of Lemma 9. Recall that $D = \mathrm{HAD}^{(q,m)} \odot C$ for a given $(q^m, n/m)_{q^m}$-code C, provided that $n/m \in \mathbb{N}^+$. Regarding this code C, a quantumly corrupted codeword O_C is assumed to act as $O_C|r\rangle|0\rangle|0^d\rangle = \sum_{z\in\mathbb{F}_{q^m}}\alpha_{r,z}|r\rangle|z\rangle|\phi_{r,z}\rangle$. Using this O_C as an oracle, let us consider a polynomial-time quantum algorithm $\mathcal{A}$ defined below. Let e be the size of garbage qubits produced in the description of $\mathcal{A}$.

QUANTUM ALGORITHM $\mathcal{A}$:

(1) Start with the quantum state $|\psi_1\rangle = |r\rangle|s\rangle|0\rangle|0^d\rangle|0^e\rangle$, where $r, s \in \mathbb{F}_{q^m}$.
(2) Change the register order to obtain the quantum state $|\psi_2\rangle = |r\rangle|0\rangle|0^d\rangle|s\rangle|0^e\rangle$.
(3) Invoke O_C using the first three registers and obtain the quantum state $|\psi_3\rangle = \sum_{z\in\mathbb{F}_{q^m}}\alpha_{r,z}|r\rangle|z\rangle|\phi_{r,z}\rangle|s\rangle|0^e\rangle$.
(4) Compute the value $u = z\cdot s \bmod q$ in a reversible fashion from (s,z). We then obtain the quantum state $|\psi_4\rangle = \sum_{z\in\mathbb{F}_{q^m}}\alpha_{r,z}|r\rangle|z\rangle|\phi_{r,z}\rangle|s\rangle|u\rangle|\phi'_{s,z}\rangle$, where $|\phi'_{s,z}\rangle$ indicates a certain garbage that might be produced while reversing the computation for u on a quantum computer.
(5) Again, change the register order so that we obtain the quantum state $|\psi_5\rangle = \sum_{z\in\mathbb{F}_{q^m}}\alpha_{r,z}|r\rangle|s\rangle|z\cdot s \bmod q\rangle|z\rangle|\phi_{r,z}\rangle|\phi'_{s,z}\rangle$. Finally, output $|\psi_5\rangle$.

The desired quantumly corrupted codeword O for D is defined as

$$\begin{aligned} O|r\rangle|s\rangle|0^l\rangle|0^d\rangle|0^e\rangle &= \sum_{z\in\mathbb{F}_{q^m}}\alpha_{r,z}|r\rangle|s\rangle|z\cdot s \bmod q\rangle|z\rangle|\phi_{r,z}\rangle|\phi'_{s,z}\rangle \\ &= \sum_{w\in\mathbb{F}_q}|r\rangle|s\rangle|w\rangle\otimes\left(\sum_{z\in A_s(w)}\alpha_{r,z}|z\rangle|\hat{\phi}_{r,s,z}\rangle\right), \end{aligned}$$

where $|\hat{\phi}_{r,s,z}\rangle = |\phi_{r,z}\rangle|\phi'_{s,z}\rangle$ and $A_s(a) = \{z \in \mathbb{F}_{q^m} \mid z\cdot s \equiv a \bmod q\}$ for any $a \in \mathbb{F}_q$. It is obvious that O can be realized by $\mathcal{A}$.

To end the proof, we want to show that $\mathrm{Pre}_O(D_x)$ equals $1/q + (1 - 1/q)\mathrm{Pre}_{O_C}(C_x)$. For convenience, let the notation T_r for each index $r \in \mathbb{F}_{q^m}$ express the value $\sum_{s\in\mathbb{F}_{q^m}}\|\sum_{z\in A_s(D_x(r,s))}\alpha_{r,z}|z\rangle|\hat{\phi}_{r,s,z}\rangle\|^2$. With this notation, the presence $\mathrm{Pre}_O(D_x)$ can be expressed as $q^{-m}\sum_{r\in\mathbb{F}_{q^m}}T_r$, which equals $\sum_{s\in\mathbb{F}_{q^m}}\sum_{z\in A_s(D_x(r,s))}|\alpha_{r,z}|^2$. Since the condition "$z\cdot s \equiv D_x(r,s) \bmod q$"

is equivalent to the condition "$z \cdot s \equiv C_x(r) \cdot s \mod q$," it follows that $T_r = \sum_{z\in\mathbb{F}_{q^m}} \sum_{s\in EQ_q(z,C_x(r))} |\alpha_{r,z}|^2$, where $EQ_q(a,b) = \{s \in \mathbb{F}_{q^m} \mid a\cdot s \equiv b\cdot s \bmod q\}$. We therefore derive

$$\begin{aligned} T_r &= |EQ_q(C_x(r), C_x(r))| \cdot |\alpha_{r,C_x(r)}|^2 + \sum_{z:z\neq C_x(r)} |EQ_q(z, C_x(r))| \cdot |\alpha_{r,z}|^2 \\ &= q^m |\alpha_{r,C_x(r)}|^2 + q^{m-1} \sum_{z:z\neq C_x(r)} |\alpha_{r,z}|^2 \\ &= q^m \left(\frac{1}{q} + \left(1-\frac{1}{q}\right)|\alpha_{r,C_x(r)}|^2\right), \end{aligned}$$

where the second equality follows from the fact that $|EQ_q(a,b)| = q^{m-1}$ if $a \neq b$. From the above relation, we obtain

$$\begin{aligned} \mathrm{Pre}_O(D_x) = \frac{1}{q^m}\sum_{r\in\mathbb{F}_{q^m}} T_r &= \frac{1}{q} + \frac{1}{q^m}\left(1-\frac{1}{q}\right)\sum_{r\in\mathbb{F}_{q^m}} |\alpha_{r,C_x(r)}|^2 \\ &= \frac{1}{q} + \left(1-\frac{1}{q}\right)\mathrm{Pre}_{O_C}(C_x). \end{aligned}$$

This completes the proof of Lemma 9. □

In the end, we have finished the proof of Proposition 8.

4 Complexity of Generalized Reed-Solomon Codes

We have shown in Proposition 8 that the concatenated code family $C^{GRS\text{-}H}$ has an efficient quantum list decoder if the generalized Reed-Solomon (GRS) codes are efficiently quantum list decodable. In order to verify Theorem 7, however, it remains to claim that the GRS-codes are efficiently quantum list decodable when the bias is relatively large. This claim will be proven as Lemma 11 in Section 4.1 in a more general fashion. For a much smaller bias, in contrast, there seems little hope in finding an efficient quantum list decoder, based on the common belief that NP-complete problems have no efficient quantum algorithms. In Sections 4.2–4.3, we will further show that the GRS-codes have natural connections to the *noisy polynomial interpolation problem* (NPIP) of Naor and Pinkas (1999) and a lattice problem, which we call the *bounded distance vector problem* (BDVP).

4.1 Polynomial Reconstruction

Proposition 8 requires the existence of efficient quantum list decodability of a family of GRS-codes. This assumption can be removed for an appropriate choice of parameters. Now, we claim, in the following technical lemma, that the family of GRS-codes is indeed quantumly list decodable.

Lemma 11. *For any number $n \in \mathbb{N}$, assume that a prime number q and real numbers $\varepsilon, \delta \in (0,1)$ satisfy the following conditions: $2 \le n \le q$ and $\varepsilon' + (1-1/q-\varepsilon')\sqrt{\frac{\Delta}{1+q\varepsilon'}} < \varepsilon \le 1-1/q$ for a certain number $\varepsilon' \in (0,1)$, where $\Delta = 2(n-1)\log(q^2/(1-\delta))$. There exists a quantum list decoder for a $(q,n,q-n+1)_q$-generalized Reed-Solomon code with bias ε and confidence δ running in time polynomial in $(n,q,1/\delta,1/(1-\delta))$.*

Combining Proposition 8 together with Lemma 11, Theorem 7 follows immediately. Before proving Lemma 11, we briefly present the proof of Theorem 7, which leads to Theorem 1.

Proof of Theorem 7. Consider the concatenated code $C^{GRS\text{-}H}[n,q,\theta] = \mathrm{HAD}^{(q,m)} \odot \mathrm{GRS}^{(M,M\theta,M)}$ with parameters n, q, θ as specified in the theorem, where $M = q^m$ and $n = mq^m\theta$. The premise of the theorem implies

$$\left(1-\frac{1}{q}\right)^2\left[\frac{1}{M}+\varepsilon''+\left(1-\frac{1}{M}-\varepsilon''\right)\sqrt{\frac{\Delta}{1+M\varepsilon''}}\right] < \frac{\varepsilon^2}{2}+\frac{\varepsilon^2}{2} = \varepsilon^2,$$

which further implies $\varepsilon'' + (1-1/M-\varepsilon'')\sqrt{\Delta/(1+M\varepsilon'')} < \frac{q^2\varepsilon^2}{(q-1)^2} - \frac{1}{M}$. Now, choose an appropriate real number $\varepsilon' \in (0,1)$ so that (1) $\varepsilon'' + (1-1/M-\varepsilon'')\sqrt{\Delta/(1+M\varepsilon'')} < \varepsilon'$ and (2) $\varepsilon' \le \frac{q^2\varepsilon^2}{(q-1)^2} - \frac{1}{M}$ (or equivalently, $\varepsilon^2 \ge (1-1/q)^2(1/M+\varepsilon')$).

From (1), Lemma 11 guarantees the existence of a quantum list decoder $\mathcal{A}$ for $\mathrm{GRS}^{(M,M\theta,M)}$ with bias ε' and confidence δ running in time polynomial in $(n,M,1/\delta,1/(1-\delta))$. With this quantum list decoder together with (2), Proposition 8 provides us with the desired quantum list decoder for $C^{GRS\text{-}H}$ with bias ε and confidence δ. □

To complete the proof of Theorem 7, what still remains to deal with is the proof of Lemma 11. A direct use of a polynomial reconstruction algorithm of Guruswami and Sudan (1999) works well to prove this lemma. We will apply this classical algorithm after collecting enough information on possible values of a target "polynomial" by a simple application of random sampling, that is, performing measurement on all oracle answers.

Proof of Lemma 11. Let n be an arbitrary message length and choose four parameters $q \in \mathbb{N}^+$ and $\varepsilon, \varepsilon', \delta \in (0,1)$ that satisfy the premise of the lemma. For simplicity, we write C for $\mathrm{GRS}^{(q,n,q)}$. Let O be any quantumly corrupted codeword for C_x, having the form $O|r\rangle|s\rangle|0^\ell\rangle = \sum_{z\in\mathbb{F}_q}\alpha_{r,z}|r\rangle|s\oplus z\rangle|\phi_{r,z}\rangle$ for certain complex numbers $\alpha_{r,z}$ and certain unit-norm quantum states $|\phi_{r,z}\rangle$. Recall that the presence of C_x in O is $(1/q)\sum_{r\in\mathbb{F}_q}|\alpha_{r,C_x(r)}|^2$. Here, we want to find all messages x satisfying the inequality $\mathrm{Pre}_O(C_x) \ge 1/q+\varepsilon$.

Fix a message x arbitrarily and omit script "x" in the following argument. Let us define two sets $A_{\varepsilon'} = \{r \in \mathbb{F}_q \mid |\alpha_{r,C_x(r)}|^2 \ge 1/q+\varepsilon'\}$ and $D_{\varepsilon'} = \{(r,y) \in \mathbb{F}_q^2 \mid$

$|\alpha_{r,y}|^2 \geq 1/q + \varepsilon'\}$. Note that C_x passes at least $|A_{\varepsilon'}|$ points in $D_{\varepsilon'}$. First, we note that $|D_{\varepsilon'}| \leq q^2/(1+q\varepsilon')$. This upper bound is easily obtained from

$$q^2 \geq \sum_r \sum_y |\alpha_{r,y}|^2 \geq \sum_{(r,y)\in D_{\varepsilon'}} |\alpha_{r,y}|^2 \geq |D_{\varepsilon'}| \left(\frac{1}{q} + \varepsilon'\right).$$

The assumption $\mathrm{Pre}_O(C_x) \geq 1/q + \varepsilon$ implies

$$\begin{aligned} \frac{1}{q} + \varepsilon \leq \mathrm{Pre}_O(C_x) &= \frac{1}{q} \sum_{r\in A_{\varepsilon'}} |\alpha_{r,C_x(r)}|^2 + \frac{1}{q} \sum_{r\in \mathbb{F}_q - A_{\varepsilon'}} |\alpha_{r,C_x(r)}|^2 \\ &\leq \frac{|A_{\varepsilon'}|}{q} + \frac{q-|A_{\varepsilon'}|}{q}\left(\frac{1}{q}+\varepsilon'\right). \end{aligned}$$

This concludes that $|A_{\varepsilon'}| \geq (1-\gamma_{\varepsilon,\varepsilon'})q$, where $\gamma_{\varepsilon,\varepsilon'} = \frac{1-1/q-\varepsilon}{1-1/q-\varepsilon'}$.

For a later use, we set $T' = \frac{q^2\log(q^2/(1-\delta))}{1+q\varepsilon'}$. Now, we claim that $(1-\gamma_{\varepsilon,\varepsilon'})^2 q^2 > 2(n-1)T'$. This inequality is equivalent to $(\varepsilon-\varepsilon')^2 > (1-1/q-\varepsilon')^2 \frac{\Delta}{1+q\varepsilon'}$, which directly follows from our assumption that $\varepsilon > \varepsilon' + (1-1/q-\varepsilon')\sqrt{\Delta/(1+q\varepsilon')}$, where $\Delta = 2(n-1)\log(q^2/(1-\delta))$. Let us consider the following quantum algorithm.

Initially, from the quantum state $|0\rangle|0\rangle|0\rangle$, we generate $|\psi_0\rangle = (1/\sqrt{q})\sum_{r\in\mathbb{F}_q}|r\rangle|0\rangle|0\rangle$. By making a query to oracle O, we generate $|\psi_1\rangle = (1/\sqrt{q})\sum_r\sum_y \alpha_{r,y}|r\rangle|y\rangle|\phi_y\rangle$. Next, we measure the first two registers and obtain (r,y) with probability $|\alpha_{r,y}|^2/q$. Let us repeat these steps exactly T times, where T is the minimal positive integer satisfying $2T' \geq T \geq T'$. Since $T \geq T'$, we obtain

$$T \geq \frac{q^2\log(q^2/(1-\delta))}{1+q\varepsilon'} \geq \frac{\log\left(1+q\varepsilon'\right)(1-\delta)/q^2}{\log\left(1-1/q^2-\varepsilon'/q\right)}, \tag{2}$$

where we use inequalities: $\log(1-z) < -z$ and $1+q\varepsilon' \geq 1$. After receiving each answer from O, we perform a measurement in the computational basis over $\mathbb{F}_q \times \mathbb{F}_q$ and store a point (r,y) that is a result of this measurement.

Let $S_{\varepsilon'}$ indicate the set of all the obtained points. Clearly, $|S_{\varepsilon'}| \leq T$ holds. Note that, with probability $(1-|\alpha_{r,y}|^2/q)^T$, each point (r,y) is never observed during the procedure. Hence, the probability P of obtaining all (r,y)'s in $D_{\varepsilon'}$ is lower-bounded by

$$P \geq 1 - \sum_{(r,y)\in D_{\varepsilon'}} \left(1-\frac{|\alpha_{r,y}|^2}{q}\right)^T \geq 1 - \frac{q^2}{1+q\varepsilon'}\cdot\left(1-\frac{1}{q^2}-\frac{\varepsilon'}{q}\right)^T \geq \delta,$$

where the last inequality follows from Eq.(2). Therefore, the probability that $S_{\varepsilon'}$ includes $D_{\varepsilon'}$ is at least δ.

Lastly, we wish to find all univariate polynomials p of degree at most $n-1$ that lie on at least $|A_{\varepsilon'}|$ points in $S_{\varepsilon'}$. For this purpose, we run the well-known Guruswami-Sudan polynomial reconstruction algorithm. Earlier, Guruswami and Sudan (1999) described a deterministic algorithm $\mathcal{A}$ that solves in time polynomial in $(m, \log q)$ the following *polynomial reconstruction problem*.

POLYNOMIAL RECONSTRUCTION PROBLEM

- INPUT: three positive integers m', n', t and m' points $\{(x_i, y_i)\}_{i\in[m']} \subseteq \mathbb{F}_q \times \mathbb{F}_q$.
- OUTPUT: all univariate polynomials p of degree at most n' that lie on at least t points, provided that $t > \sqrt{m'n'}$.

To apply the Guruswami-Sudan algorithm to our case, letting $n' = n-1$, $m' = |S_{\varepsilon'}|$, and $t = |A_{\varepsilon'}|$, we should demand the requirement that $|A_{\varepsilon'}| > \sqrt{(n-1)|S_{\varepsilon'}|}$. This requirement is met because the choice of our parameters ε and ε' implies that

$$|A_{\varepsilon'}| \geq (1-\gamma_{\varepsilon,\varepsilon'})q > \sqrt{2(n-1)T'} > \sqrt{(n-1)T} \geq \sqrt{(n-1)|S_{\varepsilon'}|}.$$

Therefore, the algorithm $\mathcal{A}$ correctly produces a list that includes all the polynomials p of degree at most $n-1$ satisfying $|\alpha_{r,p(r)}|^2 \geq 1/q + \varepsilon'$ for at least $|A_{\varepsilon'}|$ indices r. Concerning the efficiency of the algorithm, we note that the running time of $\mathcal{A}$ is bounded by a polynomial in (q, n). As a consequence, the list produced by $\mathcal{A}$ includes all messages x for which $\mathrm{Pre}_O(C_x) \geq 1/q + \varepsilon$.

Since $\mathcal{A}$ is deterministic, we can execute it quantumly as well. In the end, we produce the desired list with probability at least δ in time polynomial in $(n, q, 1/\delta, 1/(1-\delta))$. □

Due to the random sampling of quantum states necessary to apply for the Guruswami-Sudan algorithm, the total number of oracle queries made by the quantum algorithm described in the above proof of Lemma 11 is at most T, guaranteeing the confidence δ. An important open question is whether the same confidence δ can be achieved with a significantly *fewer* (e.g., a constant number of) queries.

To apply the Guruswami-Sudan algorithm, we have required the bias ε in the proof of Lemma 11 to be relatively large. One may wonder whether, even if the bias is relatively small, there is another way to list-decode the generalized Reed-Solomon codes from a quantumly corrupted codeword. In the next proposition, we will show that any *efficient* quantum list decoder for the generalized Reed-Solomon codes with small bias and high confidence can be used to solve all NP-problems *efficiently* on a quantum computer with high success probability, leading to $\mathrm{NP} \subseteq \mathrm{BQP}$.

Proposition 12. *Let $t(n)$ be any function from $\mathbb{N}$ to $\mathbb{N}$ with $t(n) \geq n$ for all $n \in \mathbb{N}$. If, for any arbitrary bias $\varepsilon(n)$, there exists a quantum list decoder $\mathcal{A}$ for the generalized Reed-Solomon codes with bias $\varepsilon(n)$ and confidence $2/3$ running in $t(n)$ time, then every NP-problem can be solved in $n^{O(1)}t(n)$ time by a certain quantum algorithm with success probability at least $2/3$. In particular, if $\mathcal{A}$ runs in polynomial time, then* $\mathrm{NP} \subseteq \mathrm{BQP}$ *holds.*

Proof. We want to give a polynomial-time reduction from a certain suitable NP-complete problem to an ε-QLDP for the GRS-code with respect to a specific quantumly corrupted codeword, where ε will be defined later. As a target NP-complete problem, we choose the following restricted form of the *interpolation problem* discussed by Goldreich, Rubinfeld, and Sudan (1995).

CONSTRAINED INTERPOLATION PROBLEM (CIP)

- INPUT: three numbers $d, e, m \in \mathbb{N}^+$, a prime number q, and a set $A = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq \mathbb{F}_q \times \mathbb{F}_q$ of m points, expressed appropriately in binary.

- REQUIREMENT: $d_A(x_i) = 2$ for any index $i \in [m]$, where $d_A(x) = |\{y \mid (x,y) \in A\}|$.
- QUESTION: is there any univariate polynomial p over $\mathbb{F}_q$ of degree at most d such that $p(x_i) = y_i$ for at least e different i's?

Note that, when $e = 1$, we always take a polynomial p satisfying $p(x_1) = y_1$. Therefore, in what follows, we assume that $e \geq 2$.

The problem CIP is clearly in NP and it can be proven to be NP-hard.[3] As a starting point, let $d, e, m \in \mathbb{N}^+$, let q be a prime number, and let $A = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq \mathbb{F}_q \times \mathbb{F}_q$ as an input to the CIP. Let $\ell = (m-1)/2$. Any polynomial $p(r) = \sum_{i=1}^{d+1} z_i r^{i-1}$ for any r can be viewed as a codeword $\mathrm{GRS}_z^{(\ell,d+1,q)}$, where $z = z_1 z_2 \cdots z_{d+1}$. For convenience, since $d_A(x_i) = 2$ for all i's, the set $D = \{x_1, \ldots, x_m\}$ of *code locators* has cardinality exactly ℓ. Without loss of generality, we assume that $\ell + 1 \leq q$.

Based on the set A, we wish to construct a quantumly corrupted codeword O. For any point (x,y) in $\mathbb{F}_q \times \mathbb{F}_q$, if $(x,y) \in A$, let $\alpha_{x,y} = 1/\sqrt{d_A(x)}$; otherwise, let $\alpha_{x,y} = 0$. The amplitude set $\{\alpha_{x,y}\}_{x,y \in \mathbb{F}_q}$ defines O as $O|x\rangle|s\rangle|t\rangle = \sum_{y \in \mathbb{F}_q} \alpha_{x,y}|x\rangle|y \oplus s\rangle|t\rangle$. Define $\varepsilon = 1/q - e/2\ell$.

It is not difficult to show that, for any polynomial p of degree d, p passes on at least e points in A if and only if the presence of p (seen as a codeword) in O satisfies the inequalities:

$$\mathrm{Pre}_O(\mathrm{GRS}_z^{(\ell,d+1,q)}) = \frac{1}{|D|} \sum_{x \in D} |\alpha_{x,p(x)}|^2 \geq \frac{1}{\ell} \sum_{i=1}^{e} \frac{1}{2} = \frac{e}{2\ell} = \frac{1}{q} + \varepsilon,$$

provided that p is identical with $\mathrm{GRS}_z^{(\ell,d+1,q)}$. Therefore, solving the CIP can be reduced to solving the ε-QLDP for the GRS-code $\{\mathrm{GRS}^{(\ell,d+1,q)}\}_{\ell,d,q}$ with respect to O. Moreover, it takes only quantum polynomial-time to *realize* O from the set A (which is given as an input). Applying a $t(n)$-time quantum list decoder for the ε-QLDP with confidence $2/3$, we can obtain a valid list of polynomials p. Obviously, the size of the obtained list is at most $t(n)$. Since the list may contain certain illegitimate polynomials, we need to check that every candidate p passes on at least e different points in A. If the list contains a legitimate polynomial, we output "YES"; otherwise, output "NO." This quantum algorithm solves the CIP with success probability at least $2/3$.

If this quantum algorithm runs in polynomial time, we can solve efficiently the CIP with high probability, leading to the inclusion NP $\subseteq$ BQP because the CIP is NP-complete. □

Despite the power of quantum computation, it seems unlikely that polynomial-time quantum algorithms can solve all the NP-problems with high success probability. Proposition 12 thus leaves little hope for finding a "polynomial-time" quantum list decoder for the GRS-codes with a *smaller bias*. However, it seems a challenging task to determine the exact threshold of such a bias for efficient quantum list decoders to exist.

[3] This fact is observed by examining the reduction constructed by Goldreich, Rubinfeld, and Sudan (1995) from the *subset sum problem*, which is known to be NP-complete.

4.2 Noisy Polynomial Interpolation Problem

As Proposition 12 indicates, for the generalized Reed-Solomon (GRS) codes, we may not be able to obtain a polynomial-time quantum list decoder having extremely small bias; however, it is still meaningful to study, for example, *subexponential-time* quantum list decoders with relatively small bias for the GRS-codes and thus to seek their applications to the field of computational cryptography. Here, we wish to propose one of those possible applications.

Earlier, Naor and Pinkas (1999) studied the *noisy polynomial interpolation problem* (NPIP) as an intractable assumption for a new cryptographic primitive, called *oblivious polynomial evaluation*. We restate their noisy interpolation problem as a promise problem of finding a unique polynomial passing through exactly one point from each given set.

NOISY POLYNOMIAL INTERPOLATION PROBLEM (NPIP)

- INPUT: three numbers $k, m, n \in \mathbb{N}^+$, a prime number q, n distinct points $\{x_1, x_2, \ldots, x_n\}$ in $\mathbb{F}_q$, and n sets $S_1, \ldots, S_n$, each of which consists of exactly m elements from $\mathbb{F}_q$, where $k+1 \leq n \leq q$.
- PROMISE: there exists a *unique* polynomial p of degree at most k such that, for each index $i \in [n]$, there exists exactly one element $y \in S_i$ satisfying $p(x_i) = y$.
- OUTPUT: the hidden polynomial p.

Disappointingly, no polynomial-time algorithm has been so far known to solve this promise problem NPIP. Apparent similarity exists between this problem and the GRS-codes (see, e.g., Roth (2006)) and, in the following proposition, this similarity helps us solve the NPIP using suitable quantum list decoders for the GRS-codes if such list decoders are actually built.

Proposition 13. *If, for any bias parameter $\varepsilon(n)$, there exists a quantum list-decoder for any $\mathrm{GRS}^{(n,k+1,q)}$-code with bias $\varepsilon(n)$ and confidence $2/3$, then there exists a quantum algorithm that solves the NPIP with probability at least $2/3$.*

Proof. Take n distinct elements $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{F}_q$ and n sets $S_1, \ldots, S_n$ of m elements each. Let us assume that the promise of the NPIP holds for a unique polynomial, say, p^* of degree at most k. Note that $k, m, n \leq q$. We set the bias parameter ε to be $1/m - 1/q$, and let S be $\bigcup_{i\in[n]} S_i$.

Here, we define the ε-QLDP for the $\mathrm{GRS}^{(n,k+1,q)}$-code with respect to a quantumly corrupted codeword O, which is defined by $O|x_i\rangle|0\rangle = \frac{1}{\sqrt{m}} \sum_{y\in S} \alpha_{x_i,y}|x_i\rangle|y\rangle$ for each index $i \in [n]$, where $\alpha_{x_i,y} = 1$ if $y \in S_i$ and 0 otherwise. We first claim that the unique polynomial p^* satisfies the condition $\mathrm{Pre}_O(p^*) \geq 1/q + \varepsilon$. Since $|X| = n$, it follows that

$$\mathrm{Pre}_O(p^*) = \frac{1}{|X|} \sum_{x\in X} |\alpha_{x,p^*(x)}|^2 = \frac{1}{n} \sum_{x\in X} \frac{1}{m} = \frac{1}{q} + \left(\frac{1}{m} - \frac{1}{q}\right) \geq \frac{1}{q} + \varepsilon.$$

Hence, p^* has codeword presence at least $1/q + \varepsilon$.

The assumption of the proposition guarantees the existence of a quantum list decoder $\mathcal{A}$ that solves the ε-QLDP with confidence $2/3$. To *realize* O from the given inputs $(x_1, \ldots, x_n, S_1, \ldots, S_n)$ of the NPIP, we generate the quantum state $O|x_i\rangle|s\rangle$ by choosing y in S uniformly at random and then generating the amplitude $\alpha_{x_i,y}/\sqrt{m}$. For the NPIP, let us consider the following quantum algorithm.

> Taking (k, m, n, q), $(x_1, \ldots, x_n)$ and $S_1, \ldots, S_n$ as input instance, run the quantum list decoder $\mathcal{A}$ using O as an oracle. We then obtain a list of polynomials p that satisfy $\mathrm{Pre}_O(p) \geq 1/q + \varepsilon$. Since the hidden polynomial p^* must be in the list, we *deterministically* check, through this list, whether each polynomial passes exactly one point from each set S_i. The uniqueness of p^* ensures that this algorithm eventually finds p^*.

It is not difficult to show that the above quantum algorithm solves the NPIP with success probability at least $2/3$ because $\mathcal{A}$ has confidence $2/3$. □

4.3 Bounded Distance Vector Problem

The previous section has sought out an application of a quantum list decoder for the generalized Reed-Solomon (GRS) codes. Here, we further intend to explore its relevant computational problems. Let us recall that codewords (viewed as functions) of the GRS codes can be identified with *polynomials*. Since polynomials are closely related to certain types of lattice problems, by exploiting this relationship, we will introduce a specific lattice problem, which we preferably call the *bounded distance vector problem* (BDVP). Next, we will show that any quantum algorithm solving this BDVP with high probability yields, for any bias ε, a quantum list decoder for GRS-codes with bias ε and relatively high confidence. The problem BDVP is formally described as follows.

BOUNDED DISTANCE VECTOR PROBLEM (BDVP)

- INPUT: a number $n \in \mathbb{N}^+$, m basis vectors $b_1, b_2, \ldots, b_m \in \mathbb{Z}^n$, and a radius $\xi \in \mathbb{Q}^{\geq 0}$.
- IMPLICIT INPUT: an oracle that, given a vector $v \in \mathbb{Z}^n$, returns the square of the weighted norm, $\|v\|^2 = \sum_{j\in[n]} \lambda_j^2 v_j^2$, where $\lambda = (\lambda_j)_j \in [0,1]^n$ is a predetermined (but hidden) weight vector and $v = (v_1, \ldots, v_n)$.
- OUTPUT: a list that contains all vectors v in the lattice L spanned by $\{b_1, b_2, \ldots, b_m\}$ for which $\|v\|^2 \leq \xi$ holds.

In the next proposition, we show the aforementioned relationship between the BDVP and quantum list decoding.

Proposition 14. *If there exists a quantum algorithm that solves the BDVP with probability at least $2/3$, then, for any positive bias ε, there exists a quantum list decoder for the family of generalized Reed-Solomon codes with bias ε and confidence $2/3$.*

Proof. A basic idea of using *Lagrange's interpolation formulas* in the following argument comes from Bleichenbacher and Nguyen (2000). To prove the proposition, it suffices to construct a quantum "reduction" to the BDVP from the ε-QLDP for the generalized Reed-Solomon code $\mathrm{GRS}^{(M,n,q)}$, where ε is any positive bias. Let us assume

that the BDVP with a hidden weight vector is quantumly solvable with success probability at least $2/3$. We start with an arbitrary input instance given to the ε-QLDP for $\mathrm{GRS}^{(M,n,q)}$.

Fix a message length n arbitrarily. Let ε be any positive bias and assume, without loss of generality, that ε is a rational number. Fix a set $D_M = \{x_1, x_2, \ldots, x_M\}$ of M distinct code locators in $\mathbb{F}_q$ and express the Cartesian product $D_M \times \mathbb{F}_q$ as $\{(x_i, z_j) \mid i \in [M], j \in [q]\}$. Let O denote any quantumly corrupted codeword O for $\mathrm{GRS}^{(M,n,q)}$ and assume that $O|x_i\rangle|s\rangle|0\rangle = \sum_{j\in[q]} \alpha_{i,j}|x_i\rangle|s \oplus z_j\rangle|\phi_{i,j}\rangle$, where $|\phi_{i,j}\rangle$ is a certain unit-norm quantum state. Let $a = (a_1, a_2, \ldots, a_n) \in (\mathbb{F}_q)^n$ be any hidden message and let $p_a(x) = \sum_{k\in[n]} a_k x^{k-1} \pmod q$ denote its codeword $\mathrm{GRS}_a^{(M,n,q)}$, which is a polynomial over $\mathbb{F}_q$ of degree at most $n-1$. Now, assume that $\mathrm{Pre}_O(p_a) = \frac{1}{M}\sum_{i\in[M]} |\alpha_{x_i,p_a(x_i)}|^2 \geq 1/q + \varepsilon$.

Next, we will define an instance to the BDVP. Firstly, we define our radius $\xi \in \mathbb{Q}$ as $M(1 - 1/q - \varepsilon)$. Secondly, we define a lattice L spanned by certain basis vectors $\{b_1, b_2, \ldots, b_m\}$ as follows. The *(special) Lagrange interpolation polynomials* corresponding to D_M are $L_i(x) = \prod_{j\in[M]-\{i\}} \frac{x-x_j}{x_i-x_j}$ in $\mathbb{F}_q[x]$, which are polynomials of degree $M-1$, for each index $i \in [M]$. Every polynomial $L_i(x)$ satisfies the following property: $L_i(x_i) = 1$ and $L_i(x_j) = 0$ if $j \neq i$. Here, we assume that $L_i(x)$ is of the form $\sum_{k\in[M]} c_{ik}x^{k-1}$ for certain constants c_{ik} in $\mathbb{F}_q$. Note that p_a satisfies the *Lagrange's interpolation formula*:

$$\begin{aligned} p_a(x) = \sum_{i\in[M]} p_a(x_i)L_i(x) &= \sum_{i\in[M]}\sum_{j\in[q]} \delta_{ij}^{(a)} z_j L_i(x) \\ &= \sum_{k\in[M]} \left(\sum_{i\in[M]}\sum_{j\in[q]} \delta_{ij}^{(a)} z_j c_{ik} \right) x^{k-1}, \end{aligned}$$

where $\delta_{ij}^{(a)} = 1$ if $p_a(x_i) = z_j$ and 0 otherwise. Obviously, for each fixed pair i and a, it holds that $\sum_{j\in[q]} \delta_{ij}^{(a)} = 1$. The vector $\delta^{(a)} = (\delta_{ij}^{(a)})_{ij} \in \mathbb{Z}^{qM}$ becomes our *target vector* in the desired lattice L (which will be defined below).

We consider only vectors $d = (d_{ij})_{ij} \in \mathbb{Z}^{qM}$ satisfying the condition $deg\left(\sum_{k\in[M]} \left(\sum_{i\in[M]}\sum_{j\in[q]} d_{ij}z_j c_{ik}\right) x^{k-1}\right) \leq n$, which is equivalent to $\sum_{i\in[M]}\sum_{j\in[q]} d_{ij}z_j c_{ik} = 0 \pmod q$ for every index $k \in [n+1, q]_{\mathbb{Z}}$. Moreover, d should satisfy that $\sum_{j=1}^{q} d_{ij} = \sum_{j=1}^{q} d_{i'j}$ for all pairs (i, i'). At last, the lattice L is defined as the collection of all vectors $d = (d_{ij})_{ij} \in \mathbb{Z}^{qM}$ such that

1. $\sum_{j\in[q]} d_{ij} = \sum_{j\in[q]} d_{i'j} \pmod q$ for all pairs $i, i' \in [M]$; and
2. $\sum_{i\in[M]}\sum_{j\in[q]} d_{ij}z_j c_{ik} = 0 \pmod q$ for all $k \in [n+1, M]_{\mathbb{Z}}$.

It is not difficult to show that L forms a lattice. It is important to note that the target vector $\delta^{(a)}$ belongs to L. From the definition of L, a suitable set of basis vectors $\{b_1, b_2, \ldots, b_m\}$ for L can be found easily (see, e.g., Bleichenbacher and Nguyen (2000)).

Finally, we introduce an oracle O' for the BDVP. To formulate this O', it suffices to define its associated weight vector $\lambda = (\lambda_{ij})_{ij} \in [0,1]^{qM}$. For each point $(x_i, z_j) \in$

$D_M \times \mathbb{F}_q$, let $\lambda_{i,j} = \sqrt{1 - |\alpha_{x_i,z_j}|^2}$. The *weighted norm* $\|d\|$ of a vector $d = (d_{ij})_{ij} \in L$ is thus calculated as $\|d\| = \sqrt{\sum_{i,j} d_{ij}^2 \lambda_{ij}^2} = \sqrt{\sum_{i,j} d_{ij}^2 (1 - |\alpha_{x_i,z_j}|^2)}$. Therefore, the square of the weighted norm of $\delta^{(a)}$ equals

$$\begin{aligned}\|\delta^{(a)}\|^2 &= \sum_{i\in[M]} \sum_{j\in[q]} \left(\delta_{ij}^{(a)}\right)^2 \left(1 - |\alpha_{x_i,p_a(x_i)}|^2\right) \;=\; M - \sum_{i\in[M]} |\alpha_{x_i,p_a(x_i)}|^2 \\ &= M\left(1 - \mathrm{Pre}_O(p_a)\right).\end{aligned}$$

Since $\xi = M(1 - 1/q - \varepsilon)$, it follows that $\|\delta^{(a)}\|^2 \le \xi$ iff $\mathrm{Pre}_O(p_a) \ge 1/q + \varepsilon$.

To solve the ε-QLDP for $\mathrm{GRS}^{(M,n,q)}$ with respect to O, we first compute the set of basis vectors $b_1, \ldots, b_m$ and the radius ξ as defined above. We then solve the BDVP using the weight vector (given by the oracle O') with success probability at least $2/3$. Let $v_1, \ldots, v_k$ be the resulted list of vectors in L. For each v_i, find $a_i \in (\mathbb{F}_q)^n$ such that $v_i = \delta^{(a_i)}$ by solving a set of linear equations. These a_i's form a list that contains all messages satisfying $\mathrm{Pre}_O(p_a) \ge 1/q + \varepsilon$. Moreover, this list can be obtained with probability at least $2/3$.

It is not difficult to show that the above-described quantum algorithm indeed solves the ε-QLDP for $\mathrm{GRS}^{(M,n,q)}$. This completes the proof of Proposition 14. □

5 An Application to Quantum Search Problems

Theorem 1 has given an efficiently quantumly list-decodable code family C over a fixed code alphabet that has polynomially small code rate; in addition, C is also efficiently classically list decodable. This fulfills our primary goal of this paper. As the next goal, we will seek an application of such an interesting code family to computational complexity theory. Of all possible applications, we will choose an issue on approximate solvability of quantum search problems. For ease of description, we use the notation $\mathrm{Prob}_M[M(x) = b]$ to denote the probability that observing the final configuration of a quantum algorithm M starting with input x results in b. Analogous to NP-search problems, an *NBQP-search problem* $\mathcal{P}$ is formally defined as a triplet (Σ^*, M, p), where M is a polynomial-time quantum algorithm taking inputs from $\Sigma^* \times \Sigma^*$ and p is a polynomial, together with the requirement that, for every $x \in \Sigma^*$ and every witness $y \in \Sigma^{p(|x|)}$, there exists a bit b such that $\mathrm{Prob}_M[M(x,y) = b] \ge 2/3$. For each $x \in \Sigma^*$, let $S_{x,M} = \{y \in \Sigma^{p(|x|)} \mid \mathrm{Prob}_M[M(x,y) = 1] \ge 2/3\}$ be the set of *solutions* of x. For simplicity, we fix our message alphabet Σ to be $\{0,1\}$ throughout this section.

NBQP-SEARCH PROBLEM

- INPUT: a (binary) string x of length n;
- OUTPUT: a solution $y \in S_{x,M}$ for x if $S_x \neq \emptyset$. Otherwise, output $\perp$ (a special symbol not in Σ).

Define $L_M = \{x \mid S_{x,M} \neq \emptyset\}$. A *solution function* f for the NBQP-search problem $\mathcal{P} = (\Sigma^*, M, p)$ satisfies that (i) for every $x \in L_M$, $f(x)$ belongs to $S_{x,M}$ and (ii)

for every $x \notin L_M$, $f(x) = \perp$. We also introduce a class NBQP of decision problems as follows: a language L belongs to NBQP if and only if there exist a polynomial-time quantum algorithm M and a polynomial p for which (Σ^*, M, p) is an NBQP-search problem and $L = L_M$.

We want to show that a certain NBQP-search problem cannot be solved even "approximately" if $\mathrm{BQP} \neq \mathrm{NBQP}$.

Proposition 15. *Assuming that* $\mathrm{BQP} \neq \mathrm{NBQP}$*, for every positive polynomial triplet* (p, p', p'') *with* $p'(n) > p(n)$ *for all numbers* $n \in \mathbb{N}$*, there exists an NBQP-search problem* $\mathcal{P} = (\Sigma^*, M, p)$ *that satisfies the following: for any solution function* f *for* $\mathcal{P}$*, no polynomial-time quantum algorithm* $\mathcal{B}$ *finds strings* y*, on each input* $x \in L_M$ *of length* n*, with probability at least* $1 - \frac{2p(n)}{p'(n)(p(n)+2)}$ *such that the relative distance* $\Delta(y, f(x))$ *is at most* $1/2 - 1/p(n)$*; on every input* $x \notin L_M$*,* $\mathcal{B}$ *outputs* $\perp$ *with probability at least* $1/2 + 1/p''(n)$.

This proposition roughly implies that solving NBQP-search problems on average leads to solving them in worst case. The proof of the proposition requires the following technical lemma, which gives a method of computing solution functions. Recall from Section 2.4 the notation $(f(x))_i$.

Lemma 16. *Let* s *be any positive polynomial with* $s(n) \geq 6$ *for every* $n \in \mathbb{N}$*. The following two statements are logically equivalent.*

1. *For every* NBQP*-search problem* $\mathcal{P} = (\Sigma^*, M, p)$*, there exist its solution function* g *and a polynomial-time quantum algorithm* $\mathcal{A}$ *such that (i) for every* $x \in L_M$*,* $\mathrm{Prob}_{\mathcal{A},i}[\mathcal{A}(x, 1^i) = (g(x))_i] \geq 1/2 + 1/s(|x|)$ *and (ii) for every* $x \notin L_M$*,* $\mathrm{Prob}_{\mathcal{A},i}[\mathcal{A}(x, 1^i) = 0] \geq 1/2 + 1/s(|x|)$*, where "*$i$*" is a random variable uniformly distributed over* $[p(n)]$.
2. *For every* NBQP*-search problem, there exist its solution function* f *and a polynomial-time quantum algorithm* $\mathcal{B}$ *such that, for every* $x \in \Sigma^*$*,* $\mathrm{Prob}_{\mathcal{B}}[\mathcal{B}(x) = f(x)] \geq 2/3$.

With the help of the above lemma, we give the proof of Proposition 15.

Proof of Proposition 15. We show the proposition by contradiction. First of all, we assume that $\mathrm{BQP} \neq \mathrm{NBQP}$. Toward a contradiction, we assume that there exist a positive polynomial triplet (p, p', p'') satisfying $p'(n) > p(n)$ for every $n \in \mathbb{N}$ that meet the following requirement: for any choice of NBQP-search problem $\mathcal{P} = (\Sigma^*, M, p)$, there are a solution function g for $\mathcal{P}$ and a polynomial-time quantum algorithm $\mathcal{B}$ for which (i) on each input $x \in L_M$, $\mathcal{B}$ finds with probability at least $1 - \frac{2p(n)}{p'(n)(p(n)+2)}$ a string y satisfying $\Delta(y, g(x)) \leq 1/2 - 1/p(n)$ and (ii) on every input $x \notin L_M$, $\mathcal{B}$ outputs $\perp$ with probability at least $1/2 + 1/p''(n)$. Let us fix a polynomial s satisfying that $s(n) \geq \max\{6, p''(n), p'(n)p(n)/(p'(n) - p(n))\}$ for all numbers $n \in \mathbb{N}$. Notice that L_M belongs to NBQP.

We wish to compute $(g(x))_i$ from $(x, 1^i)$ using $\mathcal{B}$ so that we obtain Lemma 16(1). Let us consider the following algorithm $\mathcal{A}$: on input $(x, 1^i)$, run the quantum algorithm $\mathcal{B}$ on input x and then output the ith bit of its outcome y if $y \neq \perp$, and output 0 otherwise.

Let x be an arbitrary string of length n. If $x \in L_M$, then the average probability of $\mathcal{A}$ producing $(g(x))_i$ correctly over all i's is lower-bounded by

$$\begin{aligned}\mathrm{Prob}_{\mathcal{A},i}[\mathcal{A}(x,1^i) = (g(x))_i] &\geq \left(1 - \frac{2p(n)}{p'(n)(p(n)+2)}\right)\left(1 - \max_y\{\Delta(y,g(x))\}\right) \\ &\geq \left(1 - \frac{2p(n)}{p'(n)(p(n)+2)}\right)\left(1 - \left(\frac{1}{2} - \frac{1}{p(n)}\right)\right) \\ &= \frac{1}{2} + \frac{1}{p(n)} - \frac{1}{p'(n)} \;\geq\; \frac{1}{2} + \frac{1}{s(n)},\end{aligned}$$

where the maximization is taken over all strings y produced by $\mathcal{B}$ that satisfy $\Delta(y, g(x)) \leq 1/2 - 1/p(n)$. If $x \notin L_M$, then it follows that $\mathrm{Prob}_{\mathcal{A},i}[\mathcal{A}(x,1^i) = 0] = \mathrm{Prob}_{\mathcal{B}}[\mathcal{B}(x) = \perp] \geq 1/2 + 1/p''(n) \geq 1/2 + 1/s(n)$. Since $\mathcal{P}$ is arbitrary, the statement of Lemma 16(1) holds. Lemma 16(2) then provides us with a polynomial-time quantum algorithm that computes a certain solution function f correctly with probability at least $2/3$. Since $L_M = \{x \mid f(x) \in \Sigma^*\}$ holds, L_M must be recognized with probability at least $2/3$ on a quantum computer in polynomial time; thus, L_M belongs to BQP. As a result, we conclude that NBQP is included in BQP, a contradiction against our assumption that $\mathrm{BQP} \neq \mathrm{NBQP}$. □

Finally, we present the proof of Lemma 16, in which we extensively utilize an efficiently quantumly and classically list-decodable code family given in Theorem 1.

Proof of Lemma 16. Let s be any positive polynomial with $s(n) \geq 6$ for every $n \in \mathbb{N}$. Because we consider only sufficiently large lengths n, we can assume without loss of generality that, for a certain fixed constant $k \geq 1$, $s(n) = n^k$ holds for all numbers $n \geq 6$. By Theorem 1, there are a polynomial-time computable function t and a $(t(n), n)_2$-code family C that has a polynomial-time quantum list decoder $\mathcal{D}$, with bias $1/s(n)$ and confidence $2/3$, producing a list of message candidates, where n is a message length. Let q denote a positive polynomial that bounds the sizes of any valid list produced by $\mathcal{D}$. For convenience, we also assume that $t(n) \geq n$ for all numbers $n \in \mathbb{N}$. Moreover, we write D for a polynomial-time classical list decoder for C. Note that, for each y, C_y denotes the codeword, to which y is encoded, of block length $t(|y|)$. For the sake of convenience, in this proof, we also identify this codeword C_y (defined as a function in Section 2.1) as a $t(n)$-letter string $C_y(0)C_y(1)\cdots C_y(t(n)-1)$.

The implication (2) ⇒ (1) in the lemma is trivial, since $1/2 + 1/s(n) \leq 2/3$ and, if we can computer $f(x)$ with high probability, then we can compute its ith bit $(f(x))_i$ or the symbol $\perp$ with success probability at least $2/3$. Hereafter, assuming (1), we intend to show (2). Let $\mathcal{P} = (\Sigma^*, M, p)$ be any NBQP-search problem. To make our proof simple, we assume that $p(n) \geq n$ for all numbers $n \in \mathbb{N}$. First, we reduce the error probability of the quantum algorithm M to be exponentially small (without changing the witness size). This step can be done by a standard technique of *majority voting* among polynomially many runs of the original quantum algorithm. To be more precise, there exists a polynomial-time quantum algorithm M', depending only on (p, r, M), that satisfies the following two conditions:

1. for every $x \in L_M$ and every $y \in S_{x,M}$, $\mathrm{Prob}_{M'}[M'(x,y) = 1] \geq 1 - 2^{-r(|x|)}$; and

2. for any other pair (x, y) with $y \in \Sigma^{p(|x|)}$, $\mathrm{Prob}_{M'}[M'(x, y) = 0] \geq 1 - 2^{-r(|x|)}$,

where $r(n) = q(p(n)) + 3$. Notice that $L_{M'}$ coincides with L_M.

Let us consider a quantum search problem $\mathcal{P}' = (\Sigma^*, N, p)$ defined by the following quantum algorithm N.

> On input (x, z) with $n = |x|$, if $|z| \neq t(p(n))$, then reject the input immediately. Otherwise, run the classical list decoder D in polynomial time using z as a classically corrupted codeword (or a received word) to produce with probability at least $5/6$ a list T of message candidates for C. Check deterministically whether $z = C_y$ holds for a certain string y in T. If there is no such y, reject the input. On the contrary, if $z = C_y$, then run M' on the input (x, y) and outputs its outcome.

First, we claim that $\mathcal{P}'$ is indeed an NBQP-search problem. Fix an arbitrary $n \in \mathbb{N}$, take any $x \in \Sigma^n$, and consider the case in which $x \in L_M$. Since there exists a witness $y \in \Sigma^{p(n)}$ for x, y should be included in the list T. Hence, its corresponding codeword $z = C_y$ forces N to accept (x, z) with probability at least $\frac{5}{6}\left(1 - 2^{-r(|x|)}\right) \geq 2/3$, because $r(n) \geq 3$. For the other case where $x \notin L_M$, let z be any string in $\Sigma^{t(p(n))}$. If $z \neq C_y$ for all $y \in T$, then N rejects (x, z) with probability at least $5/6$. By contrast, if $z = C_y$ holds for a certain $y \in T$, then N accepts (x, z) with probability $\leq \frac{5}{6} \cdot 2^{-r(|x|)} \leq 1/3$. Therefore, $\mathcal{P}'$ is an NBQP-search problem.

Again, applying the majority vote technique, we can reduce the error probability of N down to $2^{-r(n)}$. Abusing the notation, we use the same notation N to denote this new algorithm. For our NBQP-search problem $\mathcal{P}'$, the statement (1) gives a solution function g and a polynomial-time quantum algorithm $\mathcal{A}$ for which $\mathrm{Prob}_{\mathcal{A},i}[\mathcal{A}(x, 1^i) = (g(x))_i] \geq 1/2 + 1/s(n)$ for every $x \in L_M \cap \Sigma^n$, and $\mathrm{Prob}_{\mathcal{A},i}[\mathcal{A}(x, 1^i) = 0] \geq 1/2 + 1/s(n)$ for every $x \in \Sigma^n - L_M$. Now, assume that the final quantum state $\mathcal{A}|x, 1^i\rangle$ has the form

$$\mathcal{A}|x, 1^i\rangle = \alpha_{x,i,0}|i\rangle|0\rangle|\phi_{x,0}\rangle + \alpha_{x,i,1}|i\rangle|1\rangle|\phi_{x,1}\rangle$$

with certain amplitudes $\{\alpha_{x,i,b}\}_{x,i,b}$, where $\||\phi_{x,b}\rangle\| = 1$ for any bit b. It is obvious that $\mathrm{Prob}_{\mathcal{A},i}[\mathcal{A}(x, 1^i) = (g(x))_i] = (1/p(n)) \sum_{i \in [p(n)]} |\alpha_{x,i,(g(x))_i}|^2$ for $x \in L_M$ and $\mathrm{Prob}_{\mathcal{A},i}[\mathcal{A}(x, 1^i) = 0] = (1/p(n)) \sum_{i \in [p(n)]} |\alpha_{x,i,0}|^2$ for $x \notin L_M$.

We fix an arbitrary $x \in L_M$ of length n and, in the meantime, we omit script "x." Let us define an oracle O as

$$O|i\rangle|e\rangle|0\rangle = \alpha_{i,0}|i\rangle|e \oplus 0\rangle|\phi_{i,0}\rangle + \alpha_{i,1}|i\rangle|e \oplus 1\rangle|\phi_{i,1}\rangle$$

for any $e \in \{0, 1\}$ and any $i \in [p(n)]$. This oracle O is a quantumly corrupted codeword for C and obviously O can be realized by $\mathcal{A}$. If there exists a string y satisfying $C_y = g(x)$, then the presence of C_y in O is calculated as

$$\mathrm{Pre}_O(C_y) = \frac{1}{p(n)} \sum_i |\alpha_{i,C_y(i)}|^2 = \frac{1}{p(n)} \sum_i |\alpha_{i,(g(x))_i}|^2 \geq \frac{1}{2} + \frac{1}{s(n)}.$$

This makes us possible to run $\mathcal{D}$ using O to list-decode C.

Finally, we define a new quantum algorithm $\mathcal{B}$, based on the quantum list decoder $\mathcal{D}$ for C, that finds a witness of the problem $\mathcal{P}'$. Recall that $\mathcal{D}$ produces a list of size at most $q(n')$ for each message length n'. We assume the standard lexicographic order in $\Sigma^{p(n)}$. Let us consider the following quantum algorithm $\mathcal{B}$.

> On input x ($n = |x|$), run $\mathcal{D}$ using O as an oracle to produce a list T' of at most $q(p(n))$ message candidates (since the message size is $p(n)$), which includes the solution $g(x)$ (if $x \in L_M$) or the string $0^{p(n)}$ (if $x \notin L_M$), with probability at least $1 - 2^{-r(n)}$. Run N on the input (x, z) sequentially for all elements $z \in T'$ in order. Output the lexicographically smallest $z \in T'$ for which $N(x, z)$ outputs 1 if any. On the contrary, if there is no such z, output $\perp$.

Let $f(x)$ denote the minimal string z in T' such that (i) $\mathrm{Prob}_N[N(x, z) = 1] \geq 1 - 2^{-r(n)}$ and (ii) $\mathrm{Prob}_N[N(x, z') = 0] \geq 1 - 2^{-r(n)}$ for all $z' < z$ in T' if any; let $f(x) = \perp$ otherwise. Since $|T'| \leq q(p(n))$, the probability that $\mathcal{B}$ on input x of length n outputs $f(x)$ correctly is lower-bounded by

$$\left(1 - 2^{-r(n)}\right)^{q(p(n))} \geq 1 - 2^{-r(n)+q(p(n))-1} \geq 3/4.$$

This guarantees that the success probability of obtaining $f(x)$ is at least $3/4$. Since $\mathcal{P}$ is arbitrary, the statement (2) should hold. □

6 Local Quantum List Decoding

The previous sections have dealt with a specific computational model using implicit inputs and explicit outputs. When the running time of a quantum list decoder is limited to *sublinear*, however, it becomes impossible to produce a short list of messages *explicitly*. In such a case, it is better to allow the quantum list decoder to produces a list of short "descriptions" of *oracle quantum circuits*, each of which can generate every block symbol of a specific message by means of an appropriate oracle access to a given quantumly corrupted codeword. We call such a model an *implicit-input implicit-output model*. We will discuss a realm of quantum list decoding on this specific model and briefly state two results on the hardness amplification of quantum circuits.

Let us first introduce the notion of *local quantum list decoding*, analogous to the well-known notion of local list decoding.

Definition 17 (local quantum list decoding). Let C be any $(M(n), n, d(n))_{q(n)}$-code family with a message alphabet Σ (not depending on the choice of n). We say that C is *locally quantum list decodable* with bias ε and confidence δ if there exists a quantum algorithm $\mathcal{A}$ such that, for any message length $n \in \mathbb{N}$, any quantumly corrupted codeword O for C, and any message $x = x_1 x_2 \cdots x_n \in \Sigma^n$ satisfying $\mathrm{Pre}_O(C_x) \geq 1/q + \varepsilon(n)$, the following two conditions hold with probability at least $3/4$:

1. $\mathcal{A}(n)$ outputs a list of "descriptions" of ℓ oracle quantum circuits $D_1, D_2, \ldots, D_\ell$; and

2. there exists an index $j \in [\ell]$ such that, for every index $i \in [n]$ (expressed in binary), D_j^O on input i outputs x_i with probability at least $\delta(n)$

Similarly to the concatenated code family $C^{GRS\text{-}H}$, we can define another concatenated code family $C^{RM\text{-}H}$ using appropriate Reed-Müller codes instead of the generalized Reed-Solomon codes. Following an argument of Sudan, Trevisan, and Vadhan (2001), we can claim that the code $C^{RM\text{-}H}$ is efficiently locally quantumly list decodable with polynomially small bias and confidence $2/3$. For the proof of this claim, by Lemma 10, it suffices for us to construct an efficient quantum list decoder for the Reed-Müller codes by following Sudan, Trevisan, and Vadhan (2001). Such a quantum list decoder can be given by employing an argument similar to that of Lemma 11. Hence, we can conclude:

Proposition 18. *There exists a code family of polynomially small rate and constant codeword alphabet size that are efficiently locally quantum list decodable with polynomially small bias and confidence* $2/3$.

An immediate consequence of this proposition is the hardness amplification of quantum circuits, obtained by again following an argument of Sudan, Trevisan, and Vadhan (2001).

Corollary 19. *There exists a constant $d > 0$ for which the following is true. Let $\varepsilon \in (0,1)$ and let f be any Boolean function from $\{0,1\}^{k(n)}$ for a certain function $k(n)$. If no quantum circuit of size s computes f with success probability at least δ, then there exists a Boolean function g mapping $\{0,1\}^{\ell(k(n))}$ to $\{0,1\}$ with a certain function $\ell(n) \in n^{O(1)}$ such that no quantum circuit C of size $s' = (k(n)/\varepsilon)^d \cdot s$ satisfies $\mathrm{Prob}_{C,x}[C(x) = g(x)] \geq 1/2 + \varepsilon$, where $C(x)$ denotes the random variable indicating the observed outcome bit of C on input x.*

7 Concluding Remarks and Open Problems

The main theme of this paper is to show the existence of a quantumly list-decodable code family of polynomially small code rate over a fixed code alphabet and to seek its application to computational complexity theory. To achieve such goals, we have considered certain codes made up of generalized Reed-Solomon (GRS) codes, concatenated with the Hadamard codes, and we have proven that they are indeed efficiently quantum list decodable whenever the bias of their codeword presence is relatively large. Notice that a core part of the proof of this result heavily relies on a classical algorithm of Guruswami and Sudan (1999) and it therefore requires a relatively large number of queries. For certain types of applications, it may be desirable to make a fewer queries. At present, we have no answer to the question of whether there exists a quantum list decoder that makes a significantly fewer queries (say, less than the degree of a hidden polynomial).

Because of the different formulations of classical list decoding and quantum list decoding, we cannot verify that all classically list decodable codes are also quantumly list decodable. Among all codes of polynomially small rate, is there any quantum list decodable code that is not even classically list decodable?

When a bias becomes arbitrary small, in contrast, we have shown that the aforementioned concatenated code is unlikely to be efficiently quantumly list decodable, because the GRS codes are unlikely to have efficient quantum list decoders against arbitrary small bias. If we relax the running time of list decoders, can we build a subexponential-time quantum list decoder for the GRS code against arbitrary bias? Another important open problem is to find useful applications of quantum list decoding to a wide range of topics in quantum information processing.

Appendix

In this appendix, we will present the proofs of Propositions 2–3, which have left unproven in Section 2.3. The proof of Proposition 2 comes from an early result of Kawachi and Yamakami (2010) and the proof of Proposition 3 closely follows an argument of Guruswami, Håstad, Sudan, and Zuckerman (2002).

We begin with the proof of Proposition 2. Earlier, Kawachi and Yamakami (2010) presented a relatively good upper bound on the size of a message list in terms of the value of codeword presence by employing a geometric method of Guruswami and Sudan (2001), who gave a q-ary extension of the well-known *Johnson bound*. Let C be any $(M(n), n, d(n))_{q(n)}$-code family with a message space Σ_n and define $P_{q(n)}(M(n), d(n), \varepsilon(n))$ as $\sup_O \{|\{x \in \Sigma_n \mid \mathrm{Pre}_O(C_x) \geq \varepsilon(n)\}|\}$, where "sup" is taken over all quantumly corrupted codeword O for C. The following statement is a slight modification of Lemma 3.4 of Kawachi and Yamakami (2010) and we therefore omit its proof.

Lemma 20. *Let n be any message length. Let $(\varepsilon(n), q(n), d(n), M(n))$ satisfy the inequality $\varepsilon(n) > \ell(n)$, where $\ell(n)$ equals $1/q(n) + (1-1/q(n))\sqrt{1-(d(n)/M(n))\,(q(n)/(q(n)-1))}$. Assume that C is an $(M(n), n, d(n))_{q(n)}$-code family. The value $P_{q(n)}(M(n), d(n), \varepsilon(n))$ is upper-bounded by $\min\left\{M(n)(q(n)-1), \frac{d(n)(1-1/q(n))}{d(n)(1-1/q(n))+M(n)\varrho(n)}\right\}$, where $\varrho(n) = (\varepsilon(n) - 1/q(n))^2 - (1-1/q(n))^2$. In the case of $\varepsilon(n) = \ell(n)$, it holds that $P_{q(n)}(M(n), d(n), \varepsilon(n)) \leq 2M(n)(q(n)-1) - 1$.*

Proposition 2 easily follows from Lemma 20.

Proof of Proposition 2. Consider any $(M(n), n, d(n))_{q(n)}$-code family C. Since λ is the relative distance of C, it follows that $\lambda = d(n)/M(n)$. For readability, we will omit the parameter "n" in the following calculation. Let $c > 0$ be a constant and suppose that the upper bound of $P_q(M, d, \varepsilon)$ given in Lemma 20 does not exceed an^c; that is, assuming $\varepsilon > \ell$, it holds that

$$P_q(M, d, \varepsilon) \leq \frac{d\,(1-1/q)}{d\,(1-1/q) + M\varrho} \leq an^c.$$

From the last inequality, we immediately obtain

$$M\left(\varepsilon - \frac{1}{q}\right)^2 \geq \frac{d\,(1-1/q)}{an^c} - d\left(1-\frac{1}{q}\right) + M\left(1-\frac{1}{q}\right)^2,$$

since $\varrho = (\varepsilon - 1/q)^2 - (1-1/q)^2$. The absolute value $|\varepsilon - 1/q|$ is thus lower-bounded by

$$|\varepsilon - \frac{1}{q}| \geq \left(\frac{d\,(1-1/q)}{Man^c} - \frac{d\,(1-1/q)}{M} + \left(1 - \frac{1}{q}\right)^2 \right)^{1/2}.$$

Assuming that $\varepsilon \geq 1/q$, we therefore conclude

$$\varepsilon \geq \frac{1}{q} + \left(1 - \frac{1}{q}\right)\left(1 - \frac{d}{M\,(1-1/q)} + \frac{d}{Man^c\,(1-1/q)}\right)^{1/2}.$$

Moreover, in the case of $\varepsilon = \ell$, the definition of ℓ yields $\varepsilon = \frac{1}{q} + \left(1-\frac{1}{q}\right)\left(1 - \frac{d}{M(1-1/q)}\right)^{1/2}$. Since $QL_c^{poly}(\lambda) \geq \varepsilon$, the proposition follows immediately from the relation $\lambda = d/M$.

The second part of the proposition can be directly obtained by making c approach to the infinity. □

Next, we give the proof of Proposition 3.

Proof of Proposition 3. Let q be any odd prime number and fix $c \in \mathbb{N}^+$ and $R \in (0,1)$ arbitrarily to satisfy $c > 2(q-1)$. Let $n = \lfloor MR \rfloor$ and set $I_n = [0, M-1]_{\mathbb{Z}}$. In this proof, we consider only linear $(M,n)_q$-codes. Recall from Section 2.3 the notations V_n, W_n, and $E(w,\varepsilon)$. For brevity, let $\varepsilon = QU_c^{const}(R)$ and set $\mathbf{0} = 0^M$ and $t = q-1$.

Since V_n is composed of all vectors $v = (v_{r,z})_{r\in I_n, z\in\mathbb{F}_q}$ with $v_{r,z} \in [0,t]_{\mathbb{Z}}$ and $\sum_{z\in\mathbb{F}_q} v_{r,z} = t$ for every index $r \in I_n$, it follows that $|V_n| \geq q^M$. Note that, for every $v \in V_n$, the vector $\hat{v} = (v_{r,z}/t)_{r,z}$ belongs to W_n. Write $\hat{v}_{r,z}$ for $v_{r,z}/t$. Let the notation $V_{r,n}$ denote the rth block of V_n. Note that $V_{r,n}$ is related to Faulhaber's formula and it holds that

$$|V_{r,n}| = \sum_{j_{q-1}=0}^{t} \left(\cdots \sum_{j_2=0}^{j_3} \left(\sum_{j_1=0}^{j_2} 1 \right) \cdots \right) = \frac{t^{q-1}}{(q-1)!} + \Theta(t^{q-2}).$$

Hence, we obtain $|V_n| \leq \prod_{i=1}^{M} |V_{r,n}| \leq \left(\frac{t^{q-1}}{q^q}\right)^M$. Here, we want to introduce a new notion. For any subset $A \subseteq V_n$ and any function $f : W \to \mathbb{R}$, a *restricted expectation* $\check{E}_A[f(\hat{v})]$ is defined to be $\frac{1}{|V_n|}\sum_{v\in V_n} A(v) f(\hat{v})$, where $A(v)$ is the characteristic function for A (i.e., $A(v) = 1$ if $v \in A$ and $A(v) = 0$ otherwise). For simplicity, let α denote $(1-\varepsilon)^{\frac{(q-2)M}{2}} q^{\frac{(c-q)M}{2c}}$. In the rest of this proof, we assume that $q^{MR}\alpha < 1$. If we can find a linear $(M,n)_q$-code C such that, for every $v \in V_n$, there exists a vector $b \in C$ satisfying $\mathrm{Pre}_{\hat{v}}(b) < \varepsilon$, our assumption $q^{MR}\alpha < 1$ implies that

$$QU_c^{const}(R) \geq \varepsilon = 1 - \alpha^{\frac{2}{(q-2)M}} q^{-\frac{c-q}{c(q-2)}} \geq 1 - q^{-\frac{(1+2R)c-q}{(q-2)c}}.$$

Therefore, the remaining task is to show the existence of a linear code C that satisfies the following condition: for every $v \in V_n$, $|E(\hat{v},\varepsilon) \cap C| \leq c$ holds under the assumption of $q^{MR}\alpha < 1$.

Hereafter, we construct C by stages. The notation C_i expresses a code defined at Stage $i \in [0,n]_{\mathbb{Z}}$ and, in the end of our construction, we set the desired code C to be C_n. The key notion for this construction is the *potential function* S_i for C_i defined as $S_i = \check{E}_{V_n}[|V_n|^{\frac{1}{c}|E(\hat{v},\epsilon)\cap C_i|}]$ for each index $i \in [0,n]_{\mathbb{Z}}$. At Stage 0, we set $b_0 = \mathbf{0}$ and $C_0 = span\{b_0\}$. Clearly, for every $v \in V_n$, we have $|E(\hat{v},\epsilon)\cap C_0| \leq 1$. Since $\mathrm{Pre}_{\hat{v}}(\mathbf{0}) = \frac{1}{M}\langle v(\mathbf{0})|\hat{v}\rangle = \frac{1}{M}\sum_r \hat{v}_{r,0}$, it follows that $\mathrm{Pre}_{\hat{v}}(\mathbf{0}) \geq \epsilon$ iff $\sum_{r\in I_n}\hat{v}_{r,0} \geq \epsilon M$. Next, we consider the set $T_t^{(\varepsilon)} = \{v \in V_n \mid \sum_{r\in I_n}\hat{v}_{r,0} \geq \epsilon M t\}$. Here, we give a crude estimation to the size of $T_t^{(\varepsilon)}$ as follows. The average value of $\hat{v}_{r,0}$ over all $r \in I_n$ is $t\varepsilon$, and at most a half of them should be at least this value. Hence, each block indexed r contains at most $\frac{(q-1)(1-\varepsilon)^{q-2}}{t}|V_{r,n}|$ possible choices of vectors $v = (v_{r,z})_{z\in\mathbb{F}_q}$. Since there are at most $M!$ possible series $(v_{r,0})_{r\in I_n}$ and $M! \leq t^{(1/2-(q-1)/c)M}$, $|T_t^{(\varepsilon)}|$ is upper-bounded by

$$\begin{aligned}|T_t^{(\varepsilon)}| &\leq M!\cdot\left(\frac{(q-1)(1-\varepsilon)^{q-2}}{t}\cdot|V_{0,n}|\right)^{\frac{1}{2}M}|V_{0,n}|^{\frac{1}{2}M}\\ &\leq (1-\varepsilon)^{\frac{(q-2)M}{2}}q^{\frac{M}{2}}t^{-\frac{q-1}{c}M}|V_n|.\end{aligned}$$

Note that $v \in T_t^{(\varepsilon)}$ iff $\mathbf{0} \in E(\hat{v},\epsilon)$ iff $|E(\hat{v},\epsilon)\cap C_0| = 1$. Therefore, since $|V_n|^{1/c} \leq \left(\frac{t^{q-1}}{q^q}\right)^{M/c}$, we can calculate S_0 as

$$\begin{aligned}S_0 = \check{E}_{V_n - T_t^{(\varepsilon)}}[1] + \check{E}_{T_t^{(\varepsilon)}}[|V_n|^{1/c}] &= \frac{|V_n - T_t^{(\varepsilon)}|}{|V_n|} + \frac{|V_n|^{1/c}|T_t^{(\varepsilon)}|}{|V_n|}\\ \leq 1 + (1-\varepsilon)^{\frac{(q-2)M}{2}}q^{\frac{(c-q)M}{2c}} &= 1+\alpha.\end{aligned}$$

At Stage $i \geq 1$, we choose b_i uniformly at random from V_n so that b_i is linearly independent of $b_1,\ldots,b_{i-1}$. We define $C_i = span\{C_{i-1}\cup\{b_i\}\}$. Since b_i is a random variable, so is C_i. To complete the construction, we should claim that $|E(\hat{v},\epsilon)\cap C_n| \leq c$ for any $v \in V_n$. For this purpose, we will define a series $\{\hat{S}_i\}_{0\leq i\leq n}$ of "average" values of S_i's, starting with $\hat{S}_0 = S_0$. Let us consider the conditional expectation $E'_{b_{i+1}}[S_{i+1} \mid S_i = \hat{S}_i]$ over a random choice of b_{i+1} chosen uniformly at random from $V_n - span\{b_1,\ldots,b_i\}$. Now, we define $\hat{S}_{i+1}$ by $\hat{S}_{i+1} = E'_{b_{i+1}}[S_{i+1}|S_i = \hat{S}_i]$, and we want to show that $\hat{S}_n \leq 6$. Let the notation $E_{b_{i+1}}[S_{i+1} \mid S_i = \hat{S}_i]$ be defined similarly, except that b_{i+1} is taken uniformly at random from V_n. Similarly to an argument of Guruswami, Håstad, Sudan, and Zuckerman (2002), it holds that $E_{b_{i+1}}[S_{i+1} \mid S_i = \hat{S}_i] \leq (\hat{S}_i)^q$. Since $\frac{|V_n|-q^i}{|V_n|}\cdot E'_{b_{i+1}}[S_{i+1} \mid S_i = \hat{S}_i] \leq E_{b_{i+1}}[S_{i+1} \mid S_i = \hat{S}_i]$, we conclude that $E'_{b_{i+1}}[S_{i+1}|S_i = \hat{S}_i] \leq (1-q^{-M+i})^{-1}(\hat{S}_i)^q$ since $|V_n| \geq q^M$. Therefore, when $i = n$, we obtain

$$\hat{S}_n \leq \frac{(\hat{S}_{n-1})^q}{1-q^{-M+n-1}} \leq \frac{(\hat{S}_0)^{q^n}}{\prod_{i=0}^{n-1}(1-q^{-M+i})^{q^{n-i}}} \leq 2(\hat{S}_0)^{q^n},$$

where the last inequality follows from the lower bound $\prod_{i=0}^{n-1}(1-q^{-M+i})^{q^{n-i}} \geq \frac{1}{2}$. Since $q^{MR}\alpha < 1$, it follows that

$$\hat{S}_n \leq 2(\hat{S}_0)^{q^n} \leq 2\,(1+\alpha)^{q^n} \leq 2\left(1+2q^{MR}\alpha\right) \leq 2(1+2) = 6$$

since $n \leq MR$ and $(1+x)^m \leq 1+2mx$ for any and $m \in \mathbb{N}^+$ and any $x < 1/m$. By the definition of S_i, it follows that $\hat{S}_i \geq |V_n|^{-1}|V_n|^{\frac{1}{c}|E(\hat{v},\varepsilon)\cap C_i|}$ for every $v \in V_n$. In particular, we obtain $|V_n|^{-1}|V_n|^{\frac{1}{c}|E(\hat{v},\epsilon)\cap C_n|} \leq \hat{S}_n \leq 6$, and we therefore conclude that $|E(\hat{v},\epsilon)\cap C_n| \leq \left(1+\frac{\log 6}{\log |V_n|}\right)c < c+1$, as requested. □

Acknowledgments

The author thanks Akinori Kawachi for a discussion on quantum cryptography and Igor Shparlinski for a useful pointer to Reference (Bleichenbacher and Nguyen, 2000) when the author was preparing the preliminary version of this paper for the conference proceedings of CATS 2007.

References

Adcock, M., Cleve, R. (2002). A quantum Goldreich-Levin theorem with cryptographic applications. In the *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science* (STACS 2002). Lecture Notes in Computer Science, Springer, vol. 2285, pp. 323–334.

Bernstein, E., Vazirani, U. (1997). Quantum complexity theory. *SIAM J. Comput.* 26, 1411–1473.

Bleichenbacher, D., Nguyen, P. Q. (2000). Noisy polynomial interpolation and noisy Chinese remaindering. In the *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques—Advances in Cryptology* (EUROCRYPT 2000), Lecture Notes in Computer Science, Springer, vol. 1807, pp. 53–69.

Elias, P. List decoding for noisy channels. *WESCON Convention Record*, Part 2, Institute of Radio Engineers, pp.94–104.

Forney, G. D. (1966). *Concatenated Codes*, MIT Press, Cambridge, MA.

Goldreich, O., Levin, L. A. (1989). A hard-core predicate for all one-way functions. In the *Proceedings of the 21st Annual ACM symposium on Theory of computing* (STOC'89), pp. 25–32.

Goldreich, O., Rubinfeld, R., Sudan, M. (1995). Learning polynomials with queries: the highly noisy case. In the *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science* (FOCS'95), pp. 294–303.

Guruswami, V., Håstad, J., Sudan, M., Zuckerman, D. (2002). Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory* 48, 1021–1034.

Guruswami, V., Sudan, M. (1999). Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory* 45, 1757–1767.

Guruswami, V., Sudan, M. (2000). List decoding algorithms for certain concatenated codes. In the *Proceedings of the 32nd Annual ACM symposium on Theory of computing* (STOC 2000), pp.181–190.

Guruswami, V., Sudan, M. (2001). Extensions to the Johnson bound. Unpublished manuscript. Available at `http://madhu.seas.harvard.edu/papers/2001/johnson.pdf`.

Katz, J., Trevisan, L. (2000). On the efficiency of local decoding procedures for error-correcting codes. In the *Proceedings of the 32nd Annual ACM symposium on Theory of computing* (STOC 2000), pp. 80–86.

Kawachi, A., Yamakami, T. (2010). Quantum hardcore functions by complexity-theoretical quantum list decoding. *SIAM J. Comput.* 39, 2941–2969. An extended abstract appeared in the *Proc. of the 33rd International Colloquium on Automata, Languages and Programming* (ICALP 2006), Lecture Notes in Computer Science, Springer, vol. 4052, pp. 216–227, 2006.

Kumar, S. R., Sivakumar, D. (1999). Proofs, codes, and polynomial-time reducibilities. In the *Proceedings of the 14th Annual IEEE Conference on Computational Complexity*, pp. 46–53.

Naor, M., Pinkas, B. (1999). Oblivious transfer and polynomial evaluation. In the *Proceedings of the 31st Annual ACM symposium on Theory of computing* (STOC'99), pp. 245–254.

Nielsen, M. A., Chuang, I. L. (2000). *Quantum Computation and Quantum Information*, Cambridge University Press.

Reed, I. S., Solomon, G. (1960). Polynomial codes over certain finite fields. *J. SIAM* 8, 300–304.

Roth, R. M. (2006). *Introduction to Coding Theory*, Cambridge University Press.

Sudan, M. (1997). Decoding of Reed-Solomon codes beyond the error-correction bound. *J. Complexity* 13, 180–193.

Sudan, M. (2000). List decoding: Algorithms and applications. *SIGACT News*, vol. 31, pp. 16–27.

Sudan, M., Trevisan, L., Vadhan, S. (2001). Pseudorandom generators without the XOR lemma. *J. Comput. System Sci.* 62, 236–266.

Trevisan, L. (2004). Some applications of coding theory in computational complexity. Available at `https://arxiv.org/abs/cs/0409044`.

van Dam, W., Hallgren, S., Ip, L. (2006). Quantum algorithms for some hidden shift problems. *SIAM J. Comput.* 36, 763–778.

Wozencraft, J. M. (1958). List decoding. *Quarterly Progress Report.* Research Laboratory of Electronics, MIT, vol. 48, pp. 90–95.

Yamakami, T. (1999). A foundation of programming a multi-tape quantum Turing machine. In the *Proceedings of the 24th International Symposium on Mathematical Foundations of Computer Science* (MFCS'99), Lecture Notes in Computer Science, Springer, vol. 1672, pp. 430–441. Also available at `https://arxiv.org/abs/quant-ph/9906084`.

Yamakami, T. (2003). Analysis of quantum functions. *Int. J. Found. Comput. Sci.* 14, 815–852.

Yao, A. C. (1993). Quantum circuit complexity. In the *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science* (FOCS'93), pp. 352–361.

Received July 30, 2016 , accepted August 27, 2016