Probabilistically Checking Quantum Proofs, with Interaction

Baocheng Sun
Weizmann Institute of Science, Rehovot, Israel

Thomas Vidick
School of Computer and Communication Sciences and Center for Quantum Science and Engineering, École Polytechnique Fédérale de Lausanne, Switzerland
Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel

Abstract

The model of interactive oracle proofs (IOP) generalizes the notion of probabilistically checkable proof (PCP), in which a static proof is verified probabilistically by querying a small number of bits, to the interactive setting: a polynomial-time verifier interacts with an unbounded prover, but is restricted to only reading a small number of bits, in total, from the messages sent by the prover. IOPs provide a relaxed setting in which to study local probabilistic verification. They have proved instrumental in devising efficient methods for verification through subsequent compilation into non-interactive or succinct protoocls.

We study a quantum analogue of interactive oracle proofs (qIOP) in which the verifier and communication are both allowed to be quantum; yet the verifier is restricted to perform measurements only on a small number of qubits received from the prover. Our main result is a qIOP for any language in QMA, in which the total communication is polynomial but the verifier only reads a polylogarithmic number of qubits in total. The protocol has completeness parameter exponentially close to 1 and soundness bounded away from 1 by a constant. In the absence of a quantum PCP theorem, this provides the first information-theoretically sound local and robust characterization of QMA, albeit interactive. Previous works in the information-theoretic setting either considered two isolated but entangled quantum provers or quantum verifiers whose effort in a single round is small but remains polynomial when aggregated across all rounds of the protocol.

Our protocol combines the use of a quantum locally testable code (LTC) with classical techniques, notably probabilistically checkable proofs of proximity (PCPP). We avoid the necessity for complex multi-qubit tests employed in other settings by leveraging the local indistinguishability property of the quantum LTC.

2012 ACM Subject Classification Theory of computation → Interactive proof systems; Theory of computation → Quantum complexity theory

Keywords and phrases quantum complexity theory, quantum probabilistically checkable proofs, interactive oracle proofs, quantum locally testable codes, QMA

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.4

Acknowledgements Both authors were supported by AFOSR Grant No. FA9550-22-1-0391 and ERC Consolidator Grant VerNisQDevS (101086733).

1 Introduction

The PCP theorem shows that any problem which has efficiently verifiable proofs, has proofs that in addition can be verified with high confidence by probing only a small, in fact constant, number of bits. Based on the standard conjecture that QMA ≠ NP it is widely believed that there exist problems which admit efficiently verifiable *quantum* proofs, but no classical proofs. Do all such problems have (quantum) local verifiers, which only probe a constant number of qubits from the proof? This is the content of the *quantum PCP conjecture*, which

has been open for more than two decades. The main difficulty, colloquially, is that due to the presence of entanglement information in a quantum proof can be encoded globally and is not necessarily accessible to local probes.

In this work we give a new characterization of QMA in terms of (quantum) efficient *interactive* and local verification procedures. We study a recently introduced [33] quantum analogue of *interactive oracle proofs* (IOP) and show that every language in QMA has an efficient quantum interactive oracle proof (qIOP) where the verifier runs in quantum polynomial time and "reads" only a small (polylogarithmic, or even constant) number of qubits from the proof.[1] In addition, in our proof system the honest prover is relatively efficient and can be executed in quantum polynomial time given access to the right witness for the QMA language instance.

1.1 Results

To introduce our results in more detail we first need to sketch the model of quantum interactive oracle proof (qIOP). Informally, a qIOP (for a language L) is an interactive protocol between a quantum polynomial-time verifier and a quantum unbounded-time verifier in which the verifier "reads only a small number of qubits, in total, that were sent by the prover." Here and henceforth, we use "small" to refer to a quantity that is constant or at most polylogarithmic in the instance size. The protocol should satisfy the usual completeness and soundness conditions: for any instance that is in the language, there should exist a prover that is accepted with probability at least $2/3$;[2] and for any instance that is not in the language, any prover, even computationally unbounded, should be accepted with probability at most $1/3$.

Of course, the subtlety in the definition lies in the precise formalization of the intuitive condition "reads only a small number of qubits." Even for the case of a quantum probablistically checkable proof (qPCP), which corresponds to the non-interactive variant of a qIOP, the notion is not trivial to formalize and some work needs to be done to show that different natural formulations are equivalent (when thay are); see e.g.[1, 9] or even [28].

A formal definition of qIOP appears in [33]. We give an adaptation of the same definition in Section 4, removing variations that are not needed for this paper; we refer the reader to that section for the formalities. For the purpose of the introduction we proceed by giving a semi-formal definition that is tailored to our protocol construction and yet can be seen to fit the formal definitions from Section 4 in a straightforward manner.

Given a language L and functions (of the input length) ℓ, q, r, an (ℓ, q, r)-qIOP for L is given by the description of a quantum polynomial-time verifier which interacts with a quantum prover through a $(2r+1)$-message interaction. Without loss of generality, the first message comes from the prover and consists of an ℓ_0-qubit register W, which we think of as containing a suitably encoded quantum proof. Each subsequent round $i = 1, \ldots, r$ is a 2-message interaction which proceeds as follows. The verifier may perform three actions: measure up to q_i qubits from the prover; send M_i, a subset of qubits from W that is deterministically chosen before the protocol begins, to the prover; and send a classical message C_i, possibly depending on some private coin tosses, to the prover. The prover applies an arbitrary unitary on all his registers. He performs a measurement and returns a classical register C'_i to the

[1] Here, we put quotes because the notion of "reading" a qubit can take different formalizations; we adopt a natural one which is described later on.

[2] For this case, we may in addition require the prover to be efficient given a witness. Whenever this holds, it should be stated explicitly.

verifier.[3] (Let ℓ_i be the combined length, in qubits, of M_i, C_i and C'_i; the parameters ℓ for the qIOP is defined as $\sum \ell_i$.) At the last stage, i.e. after having received C'_r from the prover, the verifier flips some private coins and reads at most $q - \sum q_i$ bits from all classical registers he has in his possession (i.e. private classical registers, and all the C'_i registers). Based on the bits read, and any measurement outcomes and classical random bits generated by the verifier earlier in the protocol, the verifier makes his decision.

We note that this formulation precludes some adaptations that would be natural. For example, one could allow the verifier to adaptively determine which qubits to send back in each round. The verifier could also send qubits that were not directly obtained from the prover. The general model of qIOP naturally allows such operations; because our protocol does not require them we omit them from the preceding discussion, for simplicity.

For the protocol to be considered a qIOP, it is required that q is small; whereas ℓ is allowed to be an arbitrary polynomial. While in principle r may also be polynomial, for us it is sufficient to take r to be small as well (polylogarithmic). Using this formalism our main result can be informally stated as follows.

▶ **Theorem 1.1** (Main result, informal). *Every language in* QMA *has a* (poly, polylog, polylog)*-qIOP. In addition, the qIOP has completeness exponentially close to* 1 *and the honest prover is efficient, i.e. it can be executed in quantum polynomial time provided it is initially given access to the right quantum witness.*

We note a few desirable features of our protocol. Firstly, the verifier's additional classical bits sent to the prover are restricted to a single bit for all but the last message, which is polynomial (but could be made logarithmic using a stronger QMA-complete problem as starting point, see Section 1.2.1 below).

Secondly, the polylogarithmic query and round complexity are due to the sub-optimal parameters known for the best quantum locally testable code (qLTC) construction known [13], a key ingredient in our protocol. If there was a good qLTC construction, with positive rate, constant relative distance and soundness, and constant-weight parity checks, then the query and round complexity of our protocol would also both be improved to constant.

Thirdly, the total communication in the protocol is essentially quasi-linear, *except* for the length of the various (classical) PCPP proofs that are sent by the prover to the verifier, which to the best of our knowledge may be super-linear, e.g. quadratic in length. If linear PCPPs were obtained then the total communication in our protocol would be linear, up to polylogarithmic factors. Moreover, we note that the only quantum communication consists of (i) the quantum witness sent from prover to verifier in the first round, and (ii) successive blocks of the same witness sent back from the verifier to the prover in each of the r rounds of the protocol.

Nevertheless, we remark that the parameters as stated in the theorem may appear quite lose to a reader familiar with the literature on classical IOPs. Indeed, in the latter case there is a focus on the proof length (the parameter ℓ here) which is usually desired to be quasi-linear. This is because one already has access to the PCP theorem, which provides a polynomial-size proof, and the focus for IOPs is on strong efficiency. For us, there is no known quantum PCP and so any polynomial length proof is interesting. Furthermore, the size of the prover's first quantum message is actually linear in the QMA proof length (for

[3] The prover may be asked to return the register M_i, but in our protocol this is not needed by the verifier. Moreover, note that in general it cannot be enforced what qubits the prover sends back; only how many qubits are sent.

the specific language that we choose), while subsequent message registers W_i sent from the (honest) prover to the verifier can be longer but are "classical" in the sense that they are obtained by performing a classical computation in superposition on the register M_i.

1.2 Proof ideas

We now discuss the proof of our result, namely the construction of our quantum IOP for QMA and the completeness and soundness arguments.

1.2.1 Amplified XZ Hamiltonians

Our starting point is the QMA-complete local Hamiltonian problem [23]. For reasons that will become clear later on, it will be convenient for us to start with a local Hamiltonian H that decomposes as an average of two local Hamiltonians $H = \frac{1}{2}(H^Z + H^X)$, one of which, H^Z, is diagonal in the computational basis and the other, H^X, in the Hadamard basis. The QMA-completeness of such "XZ Hamiltonians" was already shown in [7].[4]

The associated completeness and soundness parameters (expressed as the smallest eigenvalue of H for YES and NO instances respectively), however, are only inverse polynomially separated. To obtain a qIOP with constant completeness-soundness gap this needs to be amplified before any protocol is attempted.[5] The usual tensoring amplification as used in e.g. [27] leads to mixed terms (with components in both bases), which is inconvenient for us. Instead, we need to make two modifications.

First, we need to start with a complete problem that has the desired structure, in terms of separation of the local terms by X or Z basis, but such that in addition each local term is a projection, the completeness is exponentially close to 0 (since we are talking about the minimum eigenvalue), and the soundness is inverse polynomially large. This will be needed for the subsequent amplification step. The result of [7] does not immediately give this. A close result is the QMA-completeness proof of the Clifford Hamiltonian problem from [8], which has the required parameters but is based on a richer set of bases than X and Z. Inspired by their work, we developed a bespoke reduction from QMA to a local Hamiltonian $H = \frac{1}{2}(\Pi^Z + \Pi^X)$ where Π^Z and Π^X are normalized averages of local projections diagonal in the computational and Hadamard basis respectively. The same reduction has been independently been obtained by Natarajan and Ma [24], with similar motivation. Since their work appeared before ours, we directly use their result.[6]

We then amplify the completeness-soundness gap of H while keeping the two-bases structure intact, e.g. we obtain $H^{amp} = \frac{1}{2}(\Pi^{Z,amp} + \Pi^{X,amp})$ where $\Pi^{Z,amp}$ and $\Pi^{X,amp}$ are normalized averages of non-local (but still efficiently described as tensor products of local terms, up to a subtraction of the identity) projections diagonal in the Z and X basis respectively. Furthermore, the smallest eigenvalue of H is exponentially close to 0 in the case of a positive instance, and larger than a universal constant in the case of a negative instance. This amplification is achieved using a simple method due to Anshu and explained in Section 5. We note that the amplification leads to $\Pi^{Z,amp}$ and $\Pi^{X,amp}$ that are averages of an exponential number of non-local terms; yet a given non-local term can still be efficiently

[4] In that reference such Hamiltonians, when in addition they are 2-local, are called "$ZZXX$ Hamiltonians." We use the terminology that was subsequently used in e.g. [18, 27].

[5] This is because e.g. sequential amplification of the protocol itself would require polynomially many repetitions, ruining any chances at keeping the query complexity small.

[6] And, obviously, do not claim credit for it.

sampled and described. This leads directly to the length of the last classical message sent by the verifier in our protocol to be polynomial. It is possible to make that message logarithmic by using the more advanced, derandomized amplification technique in [6]. Since at the moment we do not know if the length of that message is important (see Section 1.4 on open questions), we do not include details on this modification.

1.2.2 Trusted measurement provers

Given an amplified XZ Hamiltonian $H = \frac{1}{2}(\Pi^Z + \Pi^X)$ on B_q qubits provided as input, we warm up by describing a simple verification protocol in the case of a "trusted measurement prover." This is a fictional prover (considered only for the purpose of this introduction) who holds an arbitrary quantum state $|\varphi\rangle_{\mathsf{W}}$ of M qubits and returns M-bit measurement outcomes of $|\varphi\rangle_{\mathsf{W}}$ in the computational (Z) or Hadamard (X) basis whenever asked to do so.

Then the following simple, classical protocol would work. Since the verifier is only allowed to access a small number of bits of any outcome reported by the prover, and we do not assume any promise on the degree or locality of H (so a small number of bit or phase flips can potentially affect the energy by a lot), it is sensible to require $|\varphi\rangle_{\mathsf{W}}$ to be an encoding of a ground state $|\Gamma\rangle$ of H using a suitable quantum error correcting code. Thus let C_q (q for "quantum") be an $[M, B_q, d_q]$ quantum error correcting code with linear distance $d_q \propto M$, such that C_q is furthermore a CSS code with parity check matrices (H_X, H_Z). Let $\mathtt{E}_q$ the unitary encoding map of C_q, and $|\varphi\rangle_W = \mathtt{E}_q(|\Gamma\rangle_{\mathsf{W}} |0\rangle_{\mathsf{S}})$ where $|0\rangle_{\mathsf{S}}$ are ancilla qubits required for the encoding process.

Now suppose the verifier chooses a basis $W \in \{X, Z\}$ at random and requests measurement outcomes obtained from measuring $|\varphi\rangle_W$ in the basis W. The trusted measurement prover obliges, sending outcomes $w \in \{0,1\}^M$. The verifier then samples a projection Π_j^W in such a way that $\Pi^W = \mathbb{E}_j \Pi_j^W$. Since it is diagonal in basis W, the constraint Π_j^W can be represented by a boolean function $f_j : \{0,1\}^{B_q} \to \{0,1\}$ (we suppress the dependence on W for clarity) such that $\Pi_j^W = \sum_x f_j(x) S_W |x\rangle\langle x| S_W^\dagger$, where $S_W = \mathsf{I}^{\otimes B_q}$ if $W = Z$ and $S_W = H^{\otimes B_q}$ if $W = X$. Furthermore, we can represent the function f_j as a "logical" function $f'_j : \{0,1\}^M \to \{0,1\}$ by using logical operators for the code C_q.[7] For reasons that will soon become clear, define $\widetilde{f}_j = f'_j \circ \mathtt{E}_W \circ \mathtt{D}_W$, where $\mathtt{E}_W$ and $\mathtt{D}_W$ are the encoding and decoding maps associated with the linear code $C_W = \ker H_W$ respectively.[8]

The goal of the verifier is to check that $f'_j(w) = 0$. If $w \in C_W$, as it would be in case $|\varphi\rangle_{\mathsf{W}}$ is the intended witness, this is equivalent to checking that $\widetilde{f}_j(w) = 0$. So, it is enough for the verifier to check that $w \in C_W$ AND $\widetilde{f}_j(w) = 0$. Since it cannot do either task by itself (in particular, $\widetilde{f}_j$ is nonlocal) it will enlist the prover's help.

To explain this first recall that a *PCP of proximity* (PCPP) for an (efficiently computable) statement of the form "$g(y) = 0$" is a proof π that guarantees that y is *close* to a string y' such that $g(y') = 0$, and such that verifying π only requires reading a constant number of bits of y and π.

---

[7] Somewhat informally, we map a string $x \in \{0,1\}^M$ to $x' \in \{0,1\}^{B_q}$ by evaluating the logical operators, i.e. $x' = Kx$ where the rows of K represent the support of logical operators for C_q in basis W. We refer to the definition of $\widetilde{f}$ from f in Section 3 for the precise methodology.

[8] Note that C_W does not necessarily have distance. The only property we will need is that the decoder $\mathtt{D}_W$ decodes a small syndrome to an error whose size is not much bigger. This property is satisfied by the decoder in basis W for the code we use, see Section 1.2.5 and Section 2.4.

The prover is then asked to provide a PCPP for the statement: "$\widetilde{f_j}(w) = 0$ AND $H_W w = 0$". Here, the role of y is played by w, and the role of g is played by the function $(1 \oplus \widetilde{f_j}(\cdot)) \wedge 1_{H_W(\cdot)=0}$.[9] By the PCPP guarantee, if the PCPP proof is accepted with high probability then w is *close* to a string w' such that $H_W w' = 0$ and $\widetilde{f_j}(w') = 0$. We claim that it must be that $\widetilde{f_j}(w) = 0$ as well. This is because by the soundness property of the PCPP, we must have $w = w' + a$ where a is a small error. Using that $\mathtt{D}_W$ is a local decoder (see footnote 8), $a' = \mathtt{E}_W \circ \mathtt{D}_W(a)$ is a small element of $\mathrm{Im}(H_{W'}^T)$ (where W' is the complementary basis to W). Thus for every logical operator, $K_i \cdot a' = 0$ where K_i is the support of the ith logical operator in basis W (see Lemma 6.5). It follows from the definition that $\widetilde{f_j}(w) = \widetilde{f_j}(w')$, as desired. This concludes the analysis of the situation where the prover is a "trusted measurement prover."

1.2.3 Verifying measurement outcomes

Now of course the main difficulty remains: we need to devise a query-efficient procedure by which the verifier can determine that the prover correctly reports a string of measurement outcomes in the X or Z basis, where in either case the measurement has been performed on the same quantum state $|\varphi\rangle_{\mathsf{W}}$ (which need not be pure).

Interactive procedures for verifying measurement outcomes in these bases have been developed in other settings, and form the cornerstone of delegated computation protocols. For example, in the two-prover setting it is by now well-understood that building on non-local games such as the CHSH or Magic Square games, and combining with techniques from classical locally testable codes, one can devise games that force the prover to measure in the required bases; see e.g. [22, 10]. Similarly, Mahadev's delegated computation protocol [25], with a single quantum prover and a classical verifier, makes use of computational assumptions to essentially force the prover to make measurements in two mutually unbiased bases; see e.g. [20] for more on this. We also note that the two-prover multi-qubit tests have been ported to the single-prover, classical-verifier setting through compilation, based on quantum fully homomorphic encryption; see e.g. [31, 27].

In our setting, we have access to a single quantum prover and no computational assumptions. However, we may leverage quantum communication. A recent paper [33] introduces a direct analogue of the two-prover multi-qubit tests by using interaction and hiding the verifier's choice of test in entanglement between the verifier and prover. However, in that paper they were only able to achieve a qIOP with exponential communication. Our model and theirs are closely related, yet not entirely comparable.

Here we use a different idea, which leverages the properties of quantum error correcting codes and quantum locally testable codes.

1.2.4 Leveraging local indistinguishability

We first observe that there is a very classic single-qubit measurement test that leverages quantum communication. The verifier prepares an EPR pair and sends one qubit to the prover. The verifier asks the prover to measure their qubit in a randomly chosen basis $W \in \{X, Z\}$ and report the outcome x. The verifier measures their own half-EPR pair to

[9] Note that in the actual protocol, the verification that w is a codeword of the code C_W is performed by running the local testing algorithm of C_W, which is locally testable (see Definition 2.20). In the protocol, we also need to perform a codeword test for a classical ECC; however, this part is unrelated to the current section. The underlying intuition for that part is explained in Section 1.2.5.

obtain an outcome y, and accepts if and only if $x = y$. It is easily verified that the only possibility for winning perfectly in this game is for the prover to measure the qubit he received from the verifier in the correct basis.

The main limitation of this protocol is that the qubit measured is a half-EPR pair; whereas what we really want is a measurement of the quantum witness. A similar difficulty in the two-prover setting was resolved by the use of teleportation in [18]. Here, we employ a different idea, which is inspired by the local indistinguishability property of quantum error-correcting codes. Recall that in Section 1.2.2 we already required the witness to be encoded using a quantum code C_q. If the code has distance d_q, then it is well-known that the reduced density matrix ρ_S of any codeword on a subset S of $|S| < d_q$ qubits must be independent of the specific codeword. Suppose for simplicity that ρ_S is totally mixed. Then this means that, up to unitary rotations, the qubits in S and the qubits in $\overline{S}$ are equivalent to a number of EPR pairs! This suggests that a suitable adaptation of the simple test exposed in the previous paragraph could lead to certifying the prover's measurements on any sufficiently small portion of the code.

Concretely, we consider the following procedure. Divide the M qubits of an encoded state into r blocks such that each block has less than d_q qubits. For any $i \in \{1, \ldots, r\}$, the verifier sends to the prover the qubits in the ith block only. The verifier requests a measurement in basis W of those qubits, leading to outcomes x. The prover reports x. The verifier measures all other qubits in basis W, leading to an outcome x'. The verifier checks that together, (x, x') satisfy the code stabilizers in basis W, i.e. $H_W(x, x') = 0$.

It is not hard to show that a prover succeeding with probability 1 in this test must indeed perform honest measurements on the qubits of the codeword that it is given. This idea forms the basis for our measurement extraction procedure, described in full in Section 8. However, we need to perform a number of modifications that introduce additional technical difficulties.

Firstly, the description we gave requires the verifier to perform a large measurement on the witness, which is not allowed by the qIOP model. To alleviate this we rely on a quantum code that not only has low-weight parity checks but furthermore is locally testable. This ensures that the verifier (together with the prover's outcomes) will be able to assess the distance of the quantum witness from the quantum code space by measuring only a small number of qubits. Concretely we use the quantum locally testable (qLTC) codes based on cubical complexes from [13]. Such codes have good but not quite near-optimal parameters, i.e. for code length M they have linear rate, distance $\Omega(M/\mathrm{poly}\log M)$, soundness $\Omega(1/\mathrm{poly}\log M)$ and constant-weight parity checks.[10]

Secondly, we of course need to certify (and obtain) measurements on the entire witness, not only a small block of qubits. This is necessary because by the same principle of local indistinguishability, no information about the witness could be extracted by measuring only a small number of qubits. To address this we introduce an interactive procedure in which measurements are performed by the prover and only read and certified by the verifier when needed. We describe both aspects next.

1.2.5 Interactive measurement extraction using locally testable codes

We sketch the procedure by which the verifier is able to iteratively extract trusted measurement outcomes in basis W from the prover. The precise procedure and analysis are given in Section 8. The quantum witness sent by the prover to the verifier in the initial message is

[10] Some tradeoffs are possible between these parameters [34]; for clarity we do not explore these and stay with the "basic" construction.

divided into $r = M/m$ blocks of qubits, where $m \ll d_q$. The protocol then proceeds in r rounds (note that eventually we will have $r = \text{poly}\log(M)$). At the ith round, the m qubits in the ith block are sent to the prover. The prover is asked to measure them in basis W to obtain a string $x \in \{0,1\}^m$. In addition, the prover is asked to report an encoding $e = \mathtt{E}(x)$ of x according to an $[N, m, \Delta]$ linear error-correcting code C. It also provides PCPP proofs $b_1, \ldots, b_m$ where b_j is a proof for the claim that e is a valid codeword such that $\mathtt{D}(e)_j = x_j$, with $\mathtt{D}$ the decoding map of C (which is required to be efficient, up to some fraction of the distance).

In a randomly chosen round, the verifier samples a random basis-W stabilizer generator $g_W \in \{I, W\}^{\otimes M}$ for the quantum code C and verifies it as follows. If a qubit in the support of g_W has never been sent back to the prover, then the verifier measures it directly. If however the qubit has already been sent back to the prover, and thus measured in basis W and used in the preparation of PCPP proofs, then the verifier checks the associated PCPP and uses the measured value x_j returned by the prover instead.

To analyze this procedure we need to take into account that in later rounds, most of the qubits of the codeword have previously been acted on by the prover, and thus the local indistinguishability principle no longer applies. Nevertheless we are able to leverage the LTC property, which holds separately for both classical codes that underlie the CSS quantum LTC code, together with soundness of the PCPP to argue that any deviation from the prover in the complementary basis W' (i.e. X errors for an extraction in basis Z; these are the only errors that would affect a measurement in basis W) must have bounded weight. Using the linear distance property of the quantum code C, bounded weight errors do not affect logical operations. This guarantees that measurement outcomes are reported correctly, as desired.

To conclude it remains to bring together the two components, the constraint verification procedure described in Section 1.2.2 and the measurement extraction described here.

1.2.6 Protocol summary

We now summarize the structure of our protocol, from which it will be evident that the protocol has the parameters announced in Section 1.1. We describe the protocol in the case of a YES instance.

Let $H = \mathbb{E}_{W \in \{X,Z\}} \mathbb{E}_j \Pi_j^W$ be an instance of the amplified XZ Hamiltonian problem described in Section 1.2.1, let $|\Gamma\rangle$ be a ground state of H and $|\varphi\rangle_{\mathsf{W}} = E_q(|\Gamma\rangle\,|0\rangle_{\mathsf{S}})$, where E_q is the encoding unitary for a CSS code $C = (H_X, H_Z)$ with distance d_q that is locally testable. Let M be the code length. Divide M into r chunks of m qubits where $m \ll d_q$.

The protocol has two phases. The first phase is called the *measurement extraction* phase, and the second phase is the *constraint verification* phase. We first describe the measurement extraction phase. It consists of $1 + 2r$ messages, as follows:

1. The prover sends an M-qubit register W to the verifier. This register is supposed to contain $|\varphi\rangle_W$.
2. The verifier samples a basis $W \in \{X, Z\}$ uniformly at random.
3. For $i = 1, \ldots, r = M/m$ do the following:
   a. The verifier sends qubits in the ith block of W to the prover. If $i = 1$, the verifier also sends the basis W to the prover.
   b. The prover measures this block in basis W to obtain outcomes $x \in \{0,1\}^m$. He encodes x using a classical code C_c to obtain e. The prover returns the ith block of W together with x, e, and PCPP proofs b_j, for $j \in \{1, \ldots, m\}$, that e is a valid encoding and decodes to x_j.

c. In one of the rounds, chosen at random, the verifier executes the qLTC local tester for basis W. Whenever a query lands on a qubit that has previously been measured by the prover, the verifier uses x_j as the measurement outcome, checking the corresponding PCPP proof for correctness. Whenever the qubit has never been sent to the prover, the verifier measures it directly.

At the end of this phase, the verifier has in his possession a string x that is supposed to contain measurement outcomes corresponding to measuring the encoded witness $|\varphi\rangle_{\mathsf{W}}$ in basis W. Our analysis (see Lemma 8.11) shows that, indeed, any (efficiently computable) "logical" function $\widetilde{f}$ of x computed at the end of the phase is consistent with the evaluation of the "honest" observable associated with $\widetilde{f}$ on $|\varphi\rangle_{\mathsf{W}}$ at the beginning of the phase.

In the second and last phase, the *constraint verification* phase, the verifier interacts with the prover one more time to evaluate a non-local Hamiltonian constraint Π_j^W on x. This phase consists of only 2 messages, as follows:

1. The verifier selects a random constraint Π_j^W from the Hamiltonian H. Let $f_j : \{0,1\}^{B_q} \to \{0,1\}$ be the associated function, as in Section 1.2.2. The verifier sends j to the prover.
2. The prover sends the value $b = f(KE_W\mathtt{D}_W(x))$ together with a PCPP proof that $e_1, \ldots, e_r$ are valid codewords such that $f(KE_W\mathtt{D}_W(\mathtt{D}(e_1), \ldots, \mathtt{D}(e_r))) = b$. Here, $\mathtt{D}_W$ is the W-basis decoder for the qLTC, E_W is the encoder for the code $C_W = \ker H_W$, and K is a matrix of logical operators (in rows) for basis W. $\mathtt{D}$ is the decoder for the classical code. (See Section 1.2.2 for why this representation is used.)
3. The verifier accepts if and only if the PCPP proof is accepted and $b = 0$.

The analysis of this phase, given the results of the previous phase, is fairly straightforward and follows the sketch given in Section 1.2.2.

1.3 Related work

Previous works have proved or conjectured characterizations of QMA in terms of efficient, "local" verification in other settings. The *quantum games PCP* conjecture [28] considers the case of a classical verifier who exchanges $O(\log n)$-length messages with two provers sharing entanglement, where n is the instance size. While this conjecture remains open, in contrast to the quantum PCP conjecture mentioned earlier there has been notable progress on it in the past years and its resolution appears close (at least to the present authors). In particular, the development of so-called multi-qubit tests as e.g. the Pauli Braiding test of [29, 30, 21], which emphasize the role of tensors product observables involving separate X and Z observables only, influenced this work.

In a different direction, previous works have considered the case of a classical verifier running in polylogarithmic time (given the right input access model) and shown that languages in QMA have interactive verification protocols of this sort, under computational assumptions [35, 4, 31, 27]. We only consider the information-theoretic setting (but see the open problems subsection below).

Our work is arguably more related to early information-theoretic protocols for quantum verification with a small-size quantum verifier [2, 16, 15]. The main difference with our work is that, in all these results, the total quantum effort of the verifier, aggregated across rounds, remains polynomial. In our paper, the total quantum effort of the verifier is polylogarithmic. The difference is substantial; and indeed, while the cited works do not rely on further structural results than the quantum circuit model and Kitaev's reduction showing that the local Hamiltonian problem is QMA-hard, our results require a tailored, amplified QMA-hard problem to start with; classical probabilistic proof verification technique (in the form of PCPPs); as well as recent advances in quantum codes and in particular the design of quantum locally testable codes.

1.4 Open problems

In the classical literature on delegated computation, interactive oracle proofs are a step towards more "practical" primitives. The next step is achieved by removing interaction and/or achieving succinctness by replacing the long, mostly unused messages by efficient commitment schemes such as a Merkle tree. It is thus natural that the next step in our work would be to seek methods for "compiling" our quantum interactive PCP into a succinct and/or non-interactive scheme, based e.g. on the quantum random oracle model. A few features of the protocol make this step, in particular the non-interactive part, non-trivial. Notably, our protocol is private-coin (for example, qLTC stabilizer checks are done privately), and moreover involves sending quantum messages from the verifier to the prover; such messages are not easily removed by the standard Fiat-Shamir heuristic. Some approaches exist to Merkle tree-like constructions in the quantum setting [11, 19] and it would be natural to try to use those first in order to achieve succinctness. We leave such attempts for future work.

Another potential direction would aim to connect our work to "compiled" protocols as in [27], which are interactive protocols with a classical succinct verifier, obtained at the cost of relatively heavy computational assumptions such as quantum fully homomorphic encryption. It is possible that our protocol could be more directly compiled into one with a classical verifier, ideally while using weaker computational assumptions.

More immediate directions aim to reduce the complexity of our interactive protocol along different axes. It is possible that the protocol can be made public-coin, as the most important random choices of the verifier, the basis in which to extract measurement outcomes and the term of the Hamiltonian that is estimated in the energy check, are already done publicly. Other random choices, such as which round to perform a test in or which code stabilizer is measured, are made privately in our formulation; yet it appears that it would be straightforward to make them publicly. A more difficult direction is to remove quantum communication from the verifier to the prover, which could aid compilation. One could also focus on the total communication length. At the moment the most important components, such as the qLTC-encoded witness, are quasi-linear, but other components, such as the PCPPs, remain super-linear, to the best of the authors' knowledge.

Finally, while our work focuses on establishing a lower bound on the power of qIOPs, i.e. on designing a concrete protocol, it is of course interesting to study potential limitations on the model. As one puts restrictions on the communication, the locality, etc. when does one cross the boundary between QMA and NP (or, say, QCMA)?

2 Preliminaries

We collect mostly standard material on classical and quantum error correcting codes (ECCs), locally testable codes (LTCs), and probabilistically checkable proofs of proximity (PCPPs).

2.1 Error correction codes

We begin by recalling the existence of good families of error-correcting codes.

▶ **Theorem 2.1** (Error correction codes). *There exist families of linear error-correcting codes $(C_k)_{k\geq 1}$ that encode k bits in $n = n(k)$ bits and have polynomial rate, constant relative distance, and deterministic polynomial-time encoding and decoding algorithms.*

▶ **Definition 2.2.** *Fix a family of codes $(C_k)_{k\geq 1}$ as in Theorem 2.1. Here, C_k encodes k bits into $n(k)$ bits with distance $d(k)$, where $n(k) = \mathrm{poly}(k)$ and $d(k) = \Omega(n)$. Let $\mathtt{E}$ and $\mathtt{D}$ be the associated encoding and decoding algorithms respectively (which implicitly depend on k). Further, let $\mathtt{T}$ be the associated code tester, which evaluates to 1 on the range of $\mathtt{E}$ and 0 elsewhere.*[11]

Next, we introduce a quantum operator corresponding to the classical encoding process.

▶ **Definition 2.3** (Encoding operator)**.** *Let $\mathtt{E}$ be the encoding algorithm from Definition 2.2. For $k \in \mathbb{Z}_+$, $a \in \mathbb{F}_2^{n(k)}$, and $w \in \mathbb{F}_2^k$, define the encoding operator*

$$E(|a\rangle\,|w\rangle) = |a \oplus \mathtt{E}(w)\rangle\,|w\rangle\,.$$

We now formally define classical locally testable codes, which guarantee that proximity to the codespace can be efficiently checked using a small number of queries.

▶ **Definition 2.4** (Classical locally testable codes)**.** *Let $C = \ker(H)$ be a classical code, where $H \in \mathbb{F}_2^{m\times n}$ is a parity-check matrix. We say that C is* locally testable *with soundness ρ if for all $x \in \mathbb{F}_2^n$, it holds that*

$$\frac{1}{m}|Hx| \geq \rho\frac{d_H(x,C)}{n}\,,$$

where $|x|$ denotes the Hamming weight (number of nonzero coordinates) of x and $d_H(x,C)$ denotes the Hamming distance between x and the code C.

We define the local tester associated with a classical locally testable code. This tester checks whether a given string is a valid codeword by examining a small number of randomly selected parity constraints. Note that the tester depends on the choice of a parity check matrix for the code.

▶ **Definition 2.5** (Local tester for a classical LTC)**.** *Let $c \in \mathbb{Z}_+$. Let $C = \ker(H)$ be a classical locally testable code, where $H \in \mathbb{F}_2^{m\times n}$ is a parity-check matrix. Define the local tester $\mathtt{T}$ for the code C as the following procedure:*

1. *Given input $w \in \mathbb{F}_2^n$,*
2. *Randomly sample $i_1, \ldots, i_c \in_R [m]$,*
3. *Compute*

$$r = \begin{bmatrix} H_{i_1} \\ \vdots \\ H_{i_c} \end{bmatrix} \cdot w \in \mathbb{F}_2^c,$$

4. *Reject if $\bigvee_{j\in[c]} r_j = 1$ (i.e., any parity constraint is violated).*

We now present a method to amplify the tester of any locally testable code, for ease of application:

▶ **Lemma 2.6.** *Let $C = \ker(H)$ be a locally testable code with soundness ρ, where each row of H has weight at most R. Let $\kappa > \rho$.*

[11] `T` can be implemented by e.g. executing `D` and then `E` and returning 1 if the result is unchanged, 0 otherwise.

Then, there exists a code tester for C with query complexity Q such that for all $x \in \mathbb{F}_2^n$,

$$\Pr[\textit{verifier rejects } x] \geq 1 - \left(1 - \rho \cdot \frac{d_H(x,C)}{n}\right)^{Q/R} .$$

Furthermore, there exists a code tester for C with query complexity

$$Q = R \cdot \frac{\kappa}{\rho} \cdot \frac{\log(1/\varepsilon)}{1-\varepsilon}$$

such that for all $x \in \mathbb{F}_2^n$, it holds that

$$\Pr[\textit{verifier rejects } x] \geq \min\left\{\kappa \cdot \frac{d_H(x,C)}{n}, 1-\varepsilon\right\} .$$

Proof. By independently sampling a row of H a total of $N = Q/R$ times and checking whether the codeword satisfies the corresponding parity check, we obtain a code tester that rejects a near-codeword with the following probability:

$$\begin{aligned}\Pr[\text{verifier rejects}] &= 1 - \left(1 - \frac{1}{m}|Hx|\right)^N \\ &\geq 1 - \left(1 - \rho\frac{d_H(x,C)}{n}\right)^N .\end{aligned}$$

By taking $N = \frac{\kappa}{\rho} \cdot \frac{\log(1/\varepsilon)}{1-\varepsilon}$, we have

$$1 - \left(1 - \rho\frac{d_H(x,C)}{n}\right)^N \geq 1 - \exp\left(-\rho\frac{d_H(x,C)}{n}N\right) .$$

If $\rho\frac{d_H(x,C)}{n}N \geq \log(1/\varepsilon)$, then

$$1 - \exp\left(-\rho\frac{d_H(x,C)}{n}N\right) \geq 1 - \varepsilon .$$

If $\rho\frac{d_H(x,C)}{n}N \leq \log(1/\varepsilon)$, note that for $x \in [0,a]$,

$$\exp(-x) \leq 1 - \frac{1-\exp(-a)}{a}x.$$

Then,

$$\begin{aligned}1 - \exp\left(-\rho\frac{d_H(x,C)}{n}N\right) &\geq 1 - \left(1 - \frac{1-\varepsilon}{\log(1/\varepsilon)} \cdot \rho\frac{d_H(x,C)}{n}N\right) \\ &= \kappa\frac{d_H(x,C)}{n}.\end{aligned}$$

Therefore, in all cases,

$$\Pr[\text{verifier rejects}] \geq \min\left\{\kappa\frac{d_H(x,C)}{n}, 1-\varepsilon\right\}. \qquad \blacktriangleleft$$

We introduce a quantum analogue of the local tester. This operator allows the verifier to coherently verify proximity of a string to the code space using a classical local test.

▶ **Definition 2.7** (Local tester operator). *Let T be the local tester algorithm for a classical LTC, as defined in Definition 2.5. Let l_r denote the number of random bits used by T, and let q denote the number of bits of q that are queried by the tester. For $r \in \mathbb{F}_2^{l_r}$, let $\mathsf{T}(w;r)$ denote the output of T when the randomness is r and the query results corresponding to that randomness are given by w.*

Then, for $a \in \mathbb{F}_2$, $w \in \mathbb{F}_2^q$, and $r \in \mathbb{F}_2^{l_r}$, define the local tester operator as

$$L_r(|a\rangle |w\rangle) = |a \oplus \mathsf{T}(w;r)\rangle |w\rangle \ .$$

2.2 Quantum CSS codes

We first introduce standard notation for Pauli operators.

▶ **Definition 2.8** (Pauli Operators). *The Pauli operators are defined as*

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

We introduce a convenient notation for applying a single-qubit operator selectively across multiple qubits, according to a binary string.

▶ **Definition 2.9** (Tensor product of single-qubit operators). *Let $e \in \mathbb{F}_2^n$ and let W be a single-qubit operator. Define*

$$W(e) = W^{e_1} \otimes W^{e_2} \otimes \cdots \otimes W^{e_n},$$

that is, $W(e)$ applies W at each position i where $e_i = 1$ and applies the identity operator otherwise.

Next, we define CSS codes.

▶ **Definition 2.10** (CSS Code). *An $[n, k, d]$ quantum CSS code on n qubits is a subspace $C \subseteq \mathcal{H} = (\mathbb{C}^2)^{\otimes n}$. It is defined by a pair of parity-check matrices H_X, H_Z such that $\mathrm{Im}(H_X^\top) \perp \mathrm{Im}(H_Z^\top)$. We denote the code as $C = CSS(H_X, H_Z)$. Explicitly, the code subspace is given by*

$$CSS(H_X, H_Z) = \left\{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : X(x)|\psi\rangle = |\psi\rangle \ \forall x \in H_X,\ Z(z)|\psi\rangle = |\psi\rangle \ \forall z \in H_Z \right\},$$

which can also be written as

$$CSS(H_X, H_Z) = Span\left\{ \frac{1}{\sqrt{|H_X|}} \sum_{x \in H_X} |z + x\rangle : z \in \ker(H_Z) \right\}.$$

The code encodes $k = \log_2 \left(|\ker(H_X)/\mathrm{Im}(H_Z^T)| \right)$ logical qubits and has distance

$$d = \min \left\{ |w| : w \in (\ker(H_X) \setminus \mathrm{Im}(H_Z^T)) \cup (\ker(H_Z) \setminus \mathrm{Im}(H_X^T)) \right\}.$$

Finally, we define the distance of the W-code modulo the image of the parity-check matrix of the complementary code.

▶ **Definition 2.11.** *Let C be a quantum CSS code and H_X, H_Z the associated parity check matrices, as in Definition 2.10. Define*

$$d_{q,X} = \min \left\{ |w| : w \in \ker(H_X) \setminus \mathrm{Im}(H_Z^T) \right\},$$

$$d_{q,Z} = \min \left\{ |w| : w \in \ker(H_Z) \setminus \mathrm{Im}(H_X^T) \right\}.$$

We now state a lemma that guarantees the existence of a natural basis of logical Pauli operators in CSS codes.

▶ **Lemma 2.12.** *Let C be an $[M, B_q, d_q]$ CSS code. Then there exist $F, G \in \mathbb{F}_2^{B_q \times M}$ such that*

$$X(F_1), \ldots, X(F_{B_q})$$

are B_q linearly independent logical operators composed only of X and identity operators, and

$$Z(G_1), \ldots, Z(G_{B_q})$$

are B_q linearly independent logical operators composed only of Z and identity operators. Moreover, for each $i, j \in \{1, \ldots, B_q\}$, these operators satisfy

$$X(F_i)Z(G_j) = (-1)^{1(i=j)} Z(G_j)X(F_i).$$

Finally, it holds that

$$FH_Z^T = GH_X^T = 0\,. \tag{1}$$

Proof. Eq. (1) follows since logical operators commute with the stabilizers. ◀

Finally we define appropriate unitary encoding and decoding maps. Regarding encoding, we provide a unitary encoding function based on the logical operators defined in Lemma 2.12.

▶ **Definition 2.13** (Encoding of a CSS code)**.** *Let C be an $[M, B_q, d_q]$ CSS code. Let H_X, H_Z be defined as in Definition 2.10. Let G be defined as in Lemma 2.12.*

For each $a \in \mathbb{F}_2^{B_q}$, let

$$|\varphi_a\rangle = \prod_{j \in [k]} \frac{I + (-1)^{a_j} Z(G_j)}{2} \cdot \prod_j \frac{I + X((H_X)_j)}{2} \cdot \prod_j \frac{I + Z((H_Z)_j)}{2} |+\rangle^{\otimes M}\,.$$

Let the following unitary E_q be the encoding unitary of C:

$$E_q(|a\rangle\,|0\rangle^{\otimes(M-B_q)}) = \frac{|\varphi_a\rangle}{\| \, |\varphi_a\rangle \, \|}\,.$$

For convenience, we also write $E_q(|\varphi\rangle)$ to denote $E_q(|\varphi\rangle\,|0\rangle^{\otimes(M-B_q)})$.

▶ Remark 2.14. Note that the $|\varphi_a\rangle$ are orthogonal for different a, and $\| \, |\varphi_a\rangle \, \| > 0$. Thus E_q is a well-defined unitary.

The following decomposition of arbitrary states as linear combinations of codewords with Pauli errors will be convenient for our analysis.

▶ **Lemma 2.15** (Stabilizer decomposition, [17])**.** *Let $C = CSS(H_X, H_Z)$ be a stabilizer code on n qubits. Let*

$$\mathcal{S} = \Big\{ (a,b) \,\Big|\, a, b \in \mathbb{F}_2^n,\ \exists r, s \text{ s.t. } |a \vee b| \text{ is minimized, and } (a,b) \text{ is lex smallest among all } (a',b') \text{ with } H_X b' = r \text{ and } H_Z a' = s \Big\}\,.$$

Then any n-qubit state $|\psi\rangle$ has a decomposition of the form

$$|\psi\rangle = \sum_{(a,b) \in \mathcal{S}} \alpha_{a,b} X(a)Z(b)\,|\eta_{a,b}\rangle\ ,$$

where $\| \, |\eta_{a,b}\rangle \, \| = 1$, $|\eta_{a,b}\rangle \in C$, the terms $X(a)Z(b)\,|\eta_{a,b}\rangle$ are pairwise othogonal, and $\sum_{a,b} |\alpha_{a,b}|^2 = 1$.

Proof. Any n-qubit state $|\psi\rangle$ decomposes as

$$|\psi\rangle = \sum_{r,s} \alpha_{r,s} |\psi_{r,s}\rangle \ ,$$

where $|\psi_{r,s}\rangle$ is a joint eigenvector of all $X(x)$, $x \in H_X$ and $Z(z)$, $z \in H_Z$ whose associated eigenvalues are given by the vectors r, s. This is because these eigenspaces form a direct sum decomposition of the entire space. For any such r, s, if we let (a, b) be the pair associated to it as in the definition of $\mathcal{S}$, then $|\psi_{r,s}\rangle = X(a)Z(b)|\eta_{a,b}\rangle$ for some codeword $|\eta_{a,b}\rangle$. ◀

We then define the weight of a Pauli error.

▶ **Definition 2.16** (Weight of an error in stabilizer codes)**.** *Let $C = CSS(H_X, H_Z)$ be a stabilizer code on n qubits. For $a, b \in \mathbb{F}_2^n$, let $E = X(a)Z(b)$. Define the weight of E modulo the centralizer of the code generating set as:*

$$wt_{cent}(E) = \min_{\substack{a',b' \in \mathbb{F}_2^n, \\ H_X b' = 0, H_Z a' = 0}} \left\{ \ wt(X(a')Z(b')E) \ \right\} .$$

▶ Remark 2.17. Note that, for $(a, b) \in \mathcal{S}$, $wt_{\text{cent}}(X(a)Z(b)) = |a \vee b|$.

We end with the decoding operator for CSS codes, which will be used to map an encoded quantum state back to its logical components and syndrome information. Note that this decoding map, as defined, is not necessarily efficient; and we will only use it for purposes of analysis. In Section 2.4, we discuss efficient maps associated with more restricted forms of decoding.

▶ **Definition 2.18** (Decoding qLTC code)**.** *Let H_X, H_Z be defined as in Definition 2.20. Let $\mathcal{S}$ be defined as in Lemma 2.15. Any n-qubit quantum state $|\psi\rangle$ decomposes as*

$$|\psi\rangle = \sum_{(a,b)\in\mathcal{S}} \alpha_{a,b} X(a)Z(b)|\eta_{a,b}\rangle \, ,$$

where $\| \, |\eta_{a,b}\rangle \, \| = 1$ and $|\eta_{a,b}\rangle \in C$. Let D_q be the following unitary:

$$D_q(|\psi\rangle) = \sum_{(a,b)\in\mathcal{S}} \alpha_{a,b} \left|\eta'_{a,b}\right\rangle |H_Z a\rangle \, |H_X b\rangle \, ,$$

where $\eta_{a,b} = E(\eta'_{a,b})$ and E is the qLTC encoding of $\eta'_{a,b}$.

▶ Remark 2.19. By Lemma 2.15, D_q is well-defined and is indeed a unitary. Note that D_q does not necessarily need to be efficient.

2.3 Quantum locally testable codes

We then introduce the standard definition of quantum locally testable codes.

▶ **Definition 2.20** (Local testability for quantum CSS codes, [13])**.** *Let $C = \text{CSS}(H_X, H_Z)$ be a quantum CSS code. We say that C is* locally testable *with soundness ρ if both $C_X = \ker(H_X)$ and $C_Z = \ker(H_Z)$ are classical locally testable codes with soundness at least ρ.*

Next, we present a known construction of good quantum locally testable codes.

▶ **Theorem 2.21** (Quantum locally testable codes, [13])**.** *There exists a polynomial-time algorithm that, given an integer N (in unary), returns explicit descriptions of parity-check matrices H_X and H_Z on $\Theta(N)$ qubits such that the following properties hold:*

1. *The weight of each row of H_X and H_Z is $\Theta(1)$;*
2. *The code $C = \text{CSS}(H_X, H_Z)$ has dimension $\Theta(N)$ and distance $\Theta(\frac{N}{(\log N)^3})$;*
3. *C is a quantum locally testable code (qLTC) with soundness parameter $\rho = \Omega(\frac{1}{(\log N)^3})$.*

2.4 Decoding the X- and Z-codes of a CSS code

As stated in Theorem 2.21, we use the quantum LTC code based on cubical complexes from [13]. This code has an efficient decoder that generalizes the small-set bit-flip decoder, as adapted to quantum LDPC code construction in [12] and extended to the qLTC construction from [13] in [32]. As stated in [32, Theorem 3.3] the decoder runs in linear time and correctly decodes errors up to weight $\Theta(N/\mathrm{poly}\log(N))$, where N is the blocklength of the code.

For this work we need an additional property of the decoder, which is that if $e \in \{0,1\}^M$ is any "error" such that the associated syndrome in basis W, for $W \in \{X, Z\}$, is small, i.e. $\sigma = H_W e$ has small Hamming weight, then the decoder on input σ (deterministically) returns e' such that

$$|e + e'| \leq \mathrm{poly}\log(M) \cdot |\sigma| \quad \text{and} \quad H_W e' = \sigma \,. \tag{2}$$

Note that in general this property is not automatic, as an arbitrary decoder only needs to guarantee that e' has the correct syndrome σ; but e' may differ from e by an arbitrary element of $\mathrm{Im}(H_{W'}^T)$, where W' is the complementary basis to W. Such elements may a priori have arbitrary weight.

However, the decoder from [32] is a local decoder that progressively constructs an error e' such that $H_W e' = \sigma$ by finding $e_1, \ldots$, such that $|\sigma + H_W(e_1)| < \sigma$, etc. until no progress is found and one sets $e' = e_1 + \cdots$. Therefore, the number of iterations of the decoder is at most $|\sigma|$ and furthermore at each step only a "local" error is added, i.e. $|e_i| = O(\mathrm{poly}\log M)$ for each i. This is evident from the decoder description in Algorithm 3 and Algorithm 4 in [32] and establishes the desired property (2). We record it in the following fact:

▶ **Fact 2.22.** *Let C_q be the qLTC family from Theorem 2.21. Then C_q admits an efficient (in fact, linear time) decoder $\mathtt{D}_q$ for errors of weight up to $\eta_q d_q$, where d_q is the quantum distance of the code and η_q is a constant. Furthermore, let $\mathtt{D}_W$ be the decoder in basis $W \in \{X, Z\}$. We take this to be the map $\mathtt{D}_W : \mathbb{F}_2^M \to \mathbb{F}_2^{B_W}$ that to a noisy vector returns the "closest" element of the code H_W.* [12]

Then $\mathtt{D}_W$ is local in the sense that there exists $\eta_W = \Omega(1/\mathrm{poly}\log M)$ such that for any $e \in \{0,1\}^M$ such that $|e| < \eta_W d_{q,W}$ and any $z \in \{0,1\}^{B_W}$, then $|\mathtt{E}_W z + \mathtt{E}_W \mathtt{D}_W(E_W z + e)| \leq (1/\eta_W)|e|$.

For later use we define a partial composite decoding operator, which combines decoding results from different stages of the encoding process.

▶ **Definition 2.23** (Partial composite decoding operator)**.** *Let $M, m \in \mathbb{Z}_+$ such that m divides M. Let $\mathtt{D}$ denote the decoding algorithm from Theorem 2.1, and let $n(\cdot)$ be the length of the encoded message, as defined in Theorem 2.1. Let $W \in \{X, Z\}$. Let C_W be the linear code with parity-check matrix H_W from Theorem 2.21, with codeword length M, dimension B, and distance d. Let $\mathtt{D}_W$ denote its decoding algorithm.*

Let $e = (e_i)_{i \in [M/m]}$ be a tuple of encoded blocks, and let $x = (x_i)_{i \in [M/m]}$ be a tuple of plaintexts. Define the partial composite decoding function *as:*

$$\mathtt{D}_{p,W,i}(e, x) = \mathtt{D}_W(\mathtt{D}(e_1), \ldots, \mathtt{D}(e_i), x_{i+1}, \ldots, x_{M/m}).$$

[12] Sometimes, the decoder is taken to be a map $\mathtt{D}'_W : \mathbb{F}_2^{c_W} \to \mathbb{F}_2^M$ that to a syndrome returns an associated error vector, where c_W would be the number of parity checks. With our notation, $\mathtt{D}_W(x) = (E_W)^{-1}(x + \mathtt{D}'_W(H_W x))$ where (provided the decoder succeeds) $x + \mathtt{D}'_W(H_W x) \in \ker H_W$ so it is indeed possible to efficiently find a unique preimage of it under the generator matrix E_W.

For $i \in \{0, \ldots, M/m\}$, $w \in \mathbb{F}_2^{M/m \cdot n(m)}$, $x \in \mathbb{F}_2^M$, and $r \in \mathbb{F}_2^B$, define the partial composite decoding operator as:

$$D_{p,W,i}(|r\rangle |w\rangle |x\rangle) = |r \oplus \mathsf{D}_{p,W,i}(w,x)\rangle |w\rangle |x\rangle .$$

Moreover, for $x \in \mathbb{F}_2^M$ and $r \in \mathbb{F}_2^B$, define the one-sided decoding operator:

$$D_W(|r\rangle |x\rangle) = |r \oplus \mathsf{D}_{p,W,0}(0,x)\rangle |x\rangle .$$

For $w \in \mathbb{F}_2^{M/m \cdot n(m)}$ and $r \in \mathbb{F}_2^B$, define the composite decoding operator:

$$D_{c,W}(|r\rangle |w\rangle) = \left|r \oplus \mathsf{D}_{p,W,M/m}(w,0)\right\rangle |w\rangle .$$

2.5 PCP of Proximity

In this section, we recall the notion of probabilistically checkable proofs of proximity (PCPPs), introduced in [5], and their application to the canonical $\mathcal{P}$-complete problem Circuit Value. These notions play an important role in our protocols.

▶ **Definition 2.24** ([5]). *We define the $\mathcal{P}$-complete language* Circuit Value *as* CktVal $= \{(C,w) : C(w) = 1\}$, *where C is a Boolean circuit and w is an input string. For a pair language $L \subseteq \{0,1\}^* \times \{0,1\}^*$, we define $L(x) = \{y : (x,y) \in L\}$.*

The following definition formalizes a proof system in which the verifier can efficiently check whether a given input is close to being valid, without reading the entire input.

▶ **Definition 2.25** (PCPs of Proximity (PCPPs), [5]). *Let $s, \delta : \mathbb{Z}^+ \to [0,1]$ be functions. A verifier V is said to be a probabilistically checkable proof of proximity (PCPP) system for a pair language L with proximity parameter δ and soundness error s if, for every pair of strings (x,y), the following conditions hold:*

- Completeness: *If $(x,y) \in L$, then there exists a proof string π such that $V(x)$ accepts the oracle $y \circ \pi$ with probability 1.*
- Soundness: *If y is $\delta(|x|)$-far from $L(x)$, then for every π, the verifier $V(x)$ accepts the oracle $y \circ \pi$ with probability strictly less than $s(|x|)$.*

If s and δ are not explicitly specified, they are assumed to be constants in the interval $(0,1)$.

The following theorem, due to [5], establishes that the Circuit Value problem admits a PCPP.

▶ **Theorem 2.26** ([5]). *There exists a constant $\delta_0 \in (0,1)$ such that for all $n \in \mathbb{Z}^+$ and all $\delta < \delta_0$, the language* Circuit Value *has a PCP of proximity (for circuits of size n) with the following parameters:*

- *randomness complexity $O(\log n)$,*
- *query complexity $O(1)$,*
- *perfect completeness, and*
- *soundness error $1 - \Omega(\delta)$ for proximity parameter δ.*

We now present an amplified version of the above theorem for ease of application:

▶ **Lemma 2.27.** *There exists a constant $\delta_0 \in (0,1)$ such that for all $n \in \mathbb{Z}^+$ and all $\delta < \delta_0$, the language* Circuit Value *has a PCP of proximity (for circuits of size n) with the following parameters:*

- *randomness complexity* $O(\log n)$,
- *query complexity* $O(\log(s)/\log(1-\Omega(\delta)))$,
- *perfect completeness, and*
- *soundness error* s *for proximity parameter* δ.

Proof. The lemma follows by repeatedly applying the PCPP verifier from Theorem 2.26 independently at least $N = \log(s)/\log(1-\Omega(\delta))$ times, and accepting if and only if all verifiers accept.

Note that for a no-instance, the acceptance probability satisfies

$$\Pr[\text{verifier accepts}] \leq (1-\Omega(\delta))^N \leq s.$$ ◀

We now define the quantum operator associated with generating a proof in a probabilistically checkable proof of proximity (PCPP).

▶ **Definition 2.28** (PCPP generating operator)**.** *Let* $s, l_x, l_y \in \mathbb{Z}_+$. *Let* $f : \mathbb{F}_2^{l_x} \times \mathbb{F}_2^{l_y} \to \mathbb{F}_2$ *be classical circuits of size* s. *For each* $x \in \mathbb{F}_2^{l_x}, y \in \mathbb{F}_2^{l_y}$, *let* $f_x(y) = f(x,y)$. *Let* $\mathtt{N}$ *be the classical algorithm that generates the proof for inputs to* CktVal *from Lemma 2.27. Suppose the output length of* $\mathtt{N}$ *on input of size* $\Theta(s)$ *is bounded by* $l_w(s)$. *For* $a \in \mathbb{F}_2^{l_w(s)}$, $x \in \mathbb{F}_2^{l_x}$, $y \in \mathbb{F}_2^{l_y}$, *define the operator*

$$N_f(|a\rangle|x\rangle|y\rangle) = |a \oplus \mathtt{N}(f_x, y)\rangle|x\rangle|y\rangle.$$

▶ Remark 2.29. In our protocol, x typically denotes the output of a circuit C on input y. We define f_x as the circuit $1(C(\cdot) = x)$. Thus, $(f_x, y) \in$ CktVal if and only if $C(y) = x$.

Finally, we define the quantum operator corresponding to verifying the PCPP proof.

▶ **Definition 2.30** (PCPP checking operator)**.** *Use the same notation as Definition 2.28. Let* $\mathtt{T}$ *be the verifier's algorithm from Lemma 2.27. Let the randomness complexity of* $\mathtt{T}$ *on instance size of* $\Theta(n)$ *be bounded by* $l_r(n)$.

For $a \in \mathbb{F}_2$, $x \in \mathbb{F}_2^{l_x}$, $y \in \mathbb{F}_2^{l_y}$, $w \in \mathbb{F}_2^{l_w(n)}$, *and* $r \in \mathbb{F}_2^{l_r(n)}$, *define*

$$T_f(|a\rangle|x\rangle|y\rangle|w\rangle|r\rangle) = |a \oplus \mathtt{T}(f_x, y, w; r)\rangle|x\rangle|y\rangle|w\rangle|r\rangle.$$

▶ Remark 2.31. In our protocol, r will be provided as classical randomness and thus does not consume quantum resources. However, for the sake of analysis, we prefer all quantum states to be represented as pure states.

3 Notational Conventions

In this section we summarize, for ease of reference, notation that is commonly used across the paper. It is advisable to skip the section at first read but refer to it whenever a notation used in a statement or proof is unclear.

3.1 General notation

For ℓ an integer we write $[\ell]$ for the set $[\ell] = \{1, \ldots, \ell\}$. For S a finite set, we write $s \in_R S$ for a uniformly random element s of S.

3.2 Codes

Our protocols rely on two different codes, a classical error-correcting code C_C and a quantum locally testable code C. In addition, we frequently consider the two codes C_X and C_Z that constitute C as a CSS code. We summarize notation used to discuss each code.

▶ **Notation 3.1** (Codes).

Conventions for classical code.

- *Let C_C be the error-correcting code defined in Theorem 2.1. Then C_C has codewords of length $n(k)$ on messages of length k, and it has relative distance Δ.*
- *Let $\mathtt{T}$ be the code tester of C_C.*
- *Let $\mathtt{D}$ be the decoding algorithm of C_C, which can decode any error of weight less than $\eta_C\Delta$; that is, for any $e \in \mathbb{F}_2^{n(k)}$ and $w \in C_C$ satisfying $|e| < \eta_C\Delta$, we have $\mathtt{D}(w+e) = \mathtt{D}(w)$.*

Convention for qLTC code.

- *C is the quantum LTC code from Theorem 2.21, instantiated such that the code has codeword length M. Let the code dimension be B_q, the distance d_q, and the locality q. C is locally testable with soundness ρ.*
- *Let E_q be the encoding unitary for C as defined in Definition 2.13. Let D_q be the decoding unitary as defined in Definition 2.18.*
- *Let $\mathcal{S}$ be defined as in Lemma 2.15.*

Convention for X/Z codes associated with the qLTC.

- *$W \in \{X, Z\}$ denotes a single-qubit observable. We often denote by W' the complementary observable, i.e. $W' = X$ if $W = Z$ and vice-versa.*
- *If $W = X$, let $S_W = H^{\otimes M}$. If $W = Z$, let $S_W = I$. When there is no ambiguity, we also use S to denote S_W.*
- *C_W is the classical linear code corresponding to the X-check matrix H_X when $W = X$, or the Z-check matrix H_Z when $W = Z$, as in Theorem 2.21. The code has word length M, dimension B_W. Let $d_{q,X}$ and $d_{q,Z}$ be as defined in Definition 2.11.*
- *Let $E_W \in \mathbb{F}_2^{M \times B_W}$ denote the generator matrix of the code C_W.*
- *Let $\mathtt{L}_W$ be the LTC code tester for C_W from Lemma 2.6, where the parameter ε is set to s. The tester has query complexity Q_W and soundness parameter κ_W. Define $Q = \max(Q_X, Q_Z)$ and $\kappa = \min(\kappa_X, \kappa_Z)$.*
- *Let $\mathtt{D}_W$ be the decoding algorithm for C_W. Note that $\mathtt{D}_W$ is efficient and a local decoder with locality parameter η_W in the sense of Fact 2.22. Let $\eta_q = \min(\eta_X, \eta_Z)$.*
- *Let D_W be defined as in Definition 2.23.*
- *Let F, G be defined as in Definition 2.12. Let $K_W = F$ if $W = X$, $K_W = G$ if $W = Z$. When there is no ambiguity, we also use K to denote K_W.*

3.3 PCPP

Regarding probabilistically chechable proofs of proximity, we use the following.

▶ **Notation 3.2** (PCPP).

- *$\mathtt{N}$ is the classical algorithm that generates the PCPP proof for inputs to* CktVal*, and N_f denotes its coherent application as in Definition 2.28 (on input a classical circuit $f : \mathbb{F}_2^{l_x} \times \mathbb{F}_2^{l_y} \to \mathbb{F}_2$).*
- *$\mathtt{T}$ is the PCPP verification algorithm from Lemma 2.27 and T_f denotes its coherent application as in Definition 2.30.*
- *Let δ and s denote the proximity parameter and soundness error from Lemma 2.27, respectively.*

3.4 Registers

Since the protocols we design can involve relatively large numbers of quantum registers, we strive to label them consistently using sans serif font as e.g. M, V, etc. In the protocols, the following registers will be commonly used.

▶ **Notation 3.3** (Registers).

- *$M, m \in \mathbb{Z}_+$ such that m divides M. The parameter m represents the size of a local block, and the witness is partitioned into $r = M/m$ blocks.*
- *P is a register of arbitrary dimension, initially held by the prover; it is the prover's private register.*
- *Let $\mathsf{W} = (\mathsf{W}_{i,j})_{i\in[M/m],\, j\in[m]}$ denote registers initially held by the verifier, containing the witness at the beginning of the protocol. Each $\mathsf{W}_{i,j}$ is a single-qubit register.*
- *Let $\mathsf{B} = (\mathsf{B}_{i,j})_{i\in[M/m],\, j\in[m]}$, where each $\mathsf{B}_{i,j}$ is an $l_w(\mathrm{poly}(m))$-qubit PCPP proof register. These registers hold proofs that the content of register E_i correctly decodes to the value stored in register X_i; each bit of X_i has a separate proof.*
- *Let $\mathsf{E} = (\mathsf{E}_i)_{i\in[M/m]}$ and $\mathsf{E}' = (\mathsf{E}'_i)_{i\in[M/m]}$, where each E_i, resp. E'_i, is an $n(m)$-qubit register intended to contain a codeword. Specifically, these registers store the classical ECC encoding of the content in W_i. The prover holds E'_i and returns E_i to the verifier.*
- *Let $\mathsf{X} = (\mathsf{X}_{i,j})_{i\in[M/m],\, j\in[m]}$ and $\mathsf{X}' = (\mathsf{X}'_{i,j})_{i\in[M/m],\, j\in[m]}$, where each $\mathsf{X}_{i,j}$, resp. $\mathsf{X}'_{i,j}$, is a single-qubit register. The registers X_i and X'_i are used to store copies (in the W basis) of the content in W_i during the global measurement extraction phase. The prover holds X'_i and returns X_i to the verifier.*

We often write W_i for the m-qubit register $(\mathsf{W}_{i,j})_{j\in[m]}$, and similarly for X_i and X'_i.

3.5 Operators

The following notation specifies unitary operators that play an important role in the proofs.

▶ **Notation 3.4** (Operators). *Fix a Boolean function $f : \{0,1\}^{B_q} \to \{0,1\}$. For $W \in \{X, Z\}$ let $W_f = \sum_x (-1)^{f(x)} S_W \left|x\right\rangle\!\left\langle x\right| S_W$.*

- *We label the register on which D_q acts as $\mathsf{W} = (\mathsf{L}, \mathsf{S})$, where L denotes the register holding the decoded quantum state, and S denotes the register holding the syndrome. Let*

$$W'_{f,W} = D^\dagger_{q,\mathsf{W}} (W_f)_{\mathsf{L}} D_{q,\mathsf{W}}.$$

When there is no ambiguity, we also use W'_f to denote $W'_{f,W}$.

- *Let T be a quantum register of B_W qubits. Let $\widetilde{f} : \{0,1\}^{B_W} \to \{0,1\}$ be defined by $\widetilde{f}(x) = f(KE_W x)$. Let*

$$W''_{f,W} = S^\dagger_{\mathsf{W}} D^\dagger_{W,\mathsf{TW}} (Z_{\widetilde{f}})_{\mathsf{T}} D_{W,\mathsf{TW}} S_{\mathsf{W}}.$$

When there is no ambiguity, we also use W''_f to denote $W''_{f,W}$.

- *Let T be a quantum register of B_W qubits. Let*

$$Q_{i,W} = S^\dagger_{\mathsf{W}} D^\dagger_{p,W,i,\mathsf{TEW}} (Z_{\widetilde{f}})_{\mathsf{T}} D_{p,W,i,\mathsf{TEW}} S_{\mathsf{W}}\ . \tag{3}$$

When there is no ambiguity, we also use Q_i to denote $Q_{i,W}$.

4 Formulation

We proceed with a formal definition of our model, which is a restricted variant of the qIOPs introduced in [33].

In the prior work, a general framework for quantum interactive oracle proofs (qIOPs) was proposed. Our result also fits within that framework, but our protocol exhibits a more structured and constrained form. We therefore define a restricted model, which we call a *Quantum Interactive Probabilistically Checkable Proof* (qIPCP) system. This name reflects the key characteristics of the model: the verifier begins the protocol holding a quantum witness and verifies it through limited interaction with a prover, making only a small number of queries to the prover's messages. Hence, the proof is checked in an interactive and probabilistic way.

More specifically: For YES instances, there exists a witness that is initially unentangled with the prover's private register, such that the verifier accepts with probability at least $2/3$. For NO instances, no matter what witness is used, even if it is initially entangled with the prover's private register, the verifier accepts with probability at most $1/3$.

Note that this model is equivalent to the one described in Section 1.1. In our formulation, for the malicious prover, the witness initially held by the verifier may be an arbitrary quantum state, potentially entangled with the prover's private register. Thus, this is equivalent to the prover sending a witness to the verifier at the start of the protocol.

We now describe the model more precisely. In each round, the verifier may send the prover a classical message and a subset of the quantum witness register. Importantly, once the verifier sends part of the quantum witness to the prover, he can no longer access it. This is because a malicious prover is not required to return the quantum state at a later stage (even though an honest prover would do so). Furthermore, we explicitly forbid the verifier from returning any of the prover's previous messages. Therefore, the only quantum messages the verifier can send are parts of the initial quantum witness.

Despite these strong restrictions, we demonstrate that the verifier can still verify QMA-complete languages within this model.

▶ **Definition 4.1** (Quantum interactive probabilistically checkable proof system)**.** *Let $n \in \mathbb{Z}_+$ denote the instance size, and let $m : \mathbb{Z}_+ \to \mathbb{Z}_+$ be a non-decreasing, even-valued function representing the total number of messages as a function of the instance size. Define $h(n) := m(n)/2$.*

An m-message quantum interactive probabilistically checkable proof system *(qIPCP) is specified by a tuple of quantum algorithms $\mathsf{V} = (\mathsf{V}_1, \mathsf{V}_2, \ldots, \mathsf{V}_{h+1})$.*

For each $i \in [h+1]$, the verifier's algorithm V_i takes x as input and acts on registers as follows:

$$\big((\mathsf{R}_{i-1}), (\mathsf{R}_1, \ldots, \mathsf{R}_{i-2}, \mathsf{W}_i, \ldots, \mathsf{W}_h)\big) \longrightarrow \big((\mathsf{W}_i, \mathsf{J}_i), (\mathsf{R}_1, \ldots, \mathsf{R}_{i-1}, \mathsf{W}_{i+1}, \ldots, \mathsf{W}_h)\big),$$

where

- R_i *is the quantum prover's message register, for each* $i \in \{-1, 0, \ldots, h\}$,
- W_i *is the quantum witness register, for each* $i \in \{1, \ldots, h+1\}$,
- J_i *is the verifier's classical message register, for each* $i \in \{1, \ldots, h+1\}$.
- R_{-1}, R_0, *and* W_{h+1} *are all empty registers.*
- *The register* J *is initialized to the all-zero classical string.*
- *The register* $\mathsf{W} = (\mathsf{W}_i)_{i \in [h]}$ *is initialized to a quantum witness for the input* x.

Each verifier algorithm V_i must be implementable in polynomial time.

Let $I = \{1, \ldots, h\}$. A prover strategy *is defined by a collection of quantum channels $\{\mathbb{P}_i\}_{i\in I}$ (possibly depending on x) and a private quantum register P initialized with an quantum state, where each $\mathbb{P}_i$ acts as:*

$$((\mathsf{W}_i, \mathsf{J}_i), \mathsf{P}) \longrightarrow (\mathsf{R}_i, \mathsf{P}).$$

The first (classical) bit of the register J_{h+1} in the output of V_{h+1} represents the verifier's final decision to accept or reject. Let $\omega(\mathsf{V}, |\varphi\rangle)$ denote the value of the interactive game defined by V_x when the initial state of the entire system is $|\varphi\rangle$. That is, $\omega(\mathsf{V}, |\varphi\rangle)$ is the maximum acceptance probability of V over all prover strategies $\{\mathbb{P}_i\}_{i\in I}$, where $|\varphi\rangle$ is defined on the joint quantum register WP.

Resource bounds. *We define the resource complexity of the system as follows:*

- ***Communication complexity:*** *The verifier V has communication complexity $l : \mathbb{Z}_+ \to \mathbb{Z}_+$ if*

$$l(n) \geq \sum_{i\in[h(n)]} l_i(n),$$

where $l_i(n)$ is the total size (in qubits) of the registers $\mathsf{R}_i, \mathsf{J}_i, \mathsf{W}_i$.

- ***Query complexity:*** *The verifier V has query complexity $q : \mathbb{Z}_+ \to \mathbb{Z}_+$ if*

$$q(n) \geq \sum_{i\in[h(n)+1]} q_i(n),$$

where $q_i(n)$ is the number of qubits accessed by V_i in the registers

$$(\mathsf{R}_1, \ldots, \mathsf{R}_{i-1}, \mathsf{W}_i, \ldots, \mathsf{W}_h).$$

We now formalize the notions of completeness and soundness for a QIPCP system verifying a promise problem.

▶ **Definition 4.2** (Quantum interactive probabilistically checkable proof systems for a promise problem)**.** *A quantum interactive probabilistically checkable proof system for a promise problem (S_{yes}, S_{no}) satisfies the following:*

Let $\omega(\mathsf{V}, |\varphi\rangle_{\mathsf{WP}})$ be defined as in Definition 4.1. The system must satisfy:

- Completeness: *There exists a constant $c \in (0, 1]$ such that for all $x \in S_{yes}$, the maximum acceptance probability satisfies $\max_{|\psi\rangle} \omega(\mathsf{V}, |\psi\rangle_{\mathsf{W}} |0\rangle_{\mathsf{P}}) \geq c$. That is, there exists a quantum witness $|\psi\rangle$ and an honest prover strategy under which the verifier accepts with probability at least c.*
- Soundness: *There exists a constant $s \in [0, c)$ such that for all $x \in S_{no}$, the maximum acceptance probability satisfies $\max_{|\varphi\rangle} \omega(\mathsf{V}, |\varphi\rangle_{\mathsf{WP}}) \leq s$. That is, no combination of prover strategy, prover's initial state, and jointly entangled quantum witness can cause the verifier to accept a no-instance with probability greater than s.*

We define the main complexity class studied in this paper.

▶ **Definition 4.3** (Quantum interactive probabilistically checkable proof)**.** *Let $r, l, q : \mathbb{Z}_+ \to \mathbb{Z}_+$. We define $\mathcal{QIPCP}(r, l, q)$ as the set of promise problems (S_{yes}, S_{no}) that have an r-message quantum interactive probabilistically checkable proof system with communication complexity l and query complexity q.*

5 QMA-Complete Problems

We introduce a variant of the local Hamiltonian problem where each term is diagonal in the X or Z basis respectively, and is furthermore not necessarily local but efficiently sampleable and computable. The QMA-completeness of this variant essentially follows from the results of [24] and [6].

We first introduce the Hamiltonian problem and then state its QMA-completeness. Before that, we define a basic term used in the Hamiltonian.

▶ **Definition 5.1** (X/Z projection). *For $W \in \{X, Z\}$ we say that a Hamiltonian (hermitian matrix) H is a W projection if and only if H acts non-trivially on a quantum register S of k qubits as $I + W_f/2$, where $f : \mathbb{F}_2^k \to \mathbb{F}_2$ is a boolean function.*[13] *More specifically,*

$$H = \left(\frac{I + W_f}{2}\right)_{\mathsf{S}} .$$

Next, we present the formal formulation of the Hamiltonian and define the corresponding promise problem associated with it.

▶ **Definition 5.2** (Dual-basis projector Hamiltonian). *A Hamiltonian $H \in \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}$ is a dual-basis projector Hamiltonian if it is the uniform average of a polynomial number $m = \mathrm{poly}(n)$ of X projection or Z projection terms H_i. That is,*

$$H = \frac{1}{m} \sum_{i \in [m]} H_i .$$

Moreover, H is k-local if each H_i acts non-trivially on at most k qubits.

▶ **Definition 5.3** (Local dual-basis projector Hamiltonian problem). *Let $k \in \mathbb{Z}^+$ and let $a, b : \mathbb{Z}^+ \to \mathbb{R}^+$ be non-decreasing functions such that $\forall n \in \mathbb{Z}^+$, $a(n) < b(n)$. The local dual-basis projector Hamiltonian problem $\mathcal{DLH}(k, a, b)$ is the promise problem (S_{yes}, S_{no}) where*

- *$S_{yes}(n)$ is the set of k-local dual-basis projector Hamiltonians $H \in \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}$ such that $\lambda_{\min}(H) \leq a(n)$,*
- *$S_{no}(n)$ is the set of k-local dual-basis projector Hamiltonians $H \in \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}$ such that $\lambda_{\min}(H) \geq b(n)$.*

Finally, we state the QMA-completeness of the Hamiltonian problem. Note that this refers to the local Hamiltonian problem before amplification, where the promise gap is inverse-polynomial. The following follows immediately from [24].

▶ **Theorem 5.4.** *There exist polynomials p and q such that the problem*

$$\mathcal{DLH}(5, 2^{-p(n)}, 1/q(n))$$

is QMA-complete.

Proof. In [24], a somewhat incomparable result is shown, where $2^{-p(n)}$ above is replaced by 0, and the class characterized is QMA_1 (the variant of QMA with perfect completeness, meaning the verifier always accepts a yes-instance with probability 1). The present result follows using the same proof, noting that QMA is well-known to have verifiers with completeness exponentially close to 1 [26]. ◀

[13] Recall the notation $W_f = \sum_x (-1)^{f(x)} S_W |x\rangle\langle x| S_W$, with S_W a tensor product of I ($W = Z$) or H ($W = X$).

Then, we introduce the Hamiltonian amplification method from [6]. We first present their construction, followed by a statement of their result.

Essentially, the idea is to divide the Hamiltonian into commuting layers. In our case, there are two: the Z-layer, where each term is diagonal in the computational basis, and the X-layer, where each term is diagonal in the Hadamard basis. Each layer is then amplified separately by performing all tests corresponding to that layer simultaneously, and repeating the layer test sequentially t times.

▶ **Definition 5.5** (Amplified Hamiltonian, [6])**.** *Let H be a k-local Hamiltonian on n qubits expressed as an expectation over m projection terms,*

$$H = \frac{1}{m} \sum_{i \in [m]} \Pi_i.$$

where the local terms of H can be efficiently and equitably partitioned into $g = O(1)$ commuting layers, indexed by $\chi \in [g]$. We denote

$$H_\chi = \mathbb{E}_{i \in [m_\chi]} [\Pi_i^\chi], \quad w_\chi = \frac{m_\chi}{m} = \Omega(g^{-1}),$$

where w_χ represents the relative weight of clauses assigned to layer χ, $H = \mathbb{E}_\chi H_\chi$.

For any integer parameter $t \geq 1$, we define the amplified Hamiltonian *as*

$$H^{(t)} = I - \mathbb{E}_\chi \bigotimes_{j \in [t]} \Big(\prod_{i \in [m_\chi]} (I - \Pi_i^\chi) \Big), \tag{4}$$

where, $\mathbb{E}_\chi$ indicates sampling χ from the distribution $(w_1, w_2, \ldots, w_g)$ over $[g]$.

By applying this procedure, the minimum energy of the no-instance can be amplified until it exceeds a certain threshold.

▶ **Theorem 5.6** (Amplification from the DL, [6])**.** *Assume that the minimum eigenvalue of H satisfies $\lambda_{\min}(H) \geq \gamma$. Then, the minimum eigenvalue of the amplified Hamiltonian $H^{(t)}$ satisfies*

$$\lambda_{\min}\big(H^{(t)}\big) \geq \min \Big\{ \Theta(g^{-2}),\ \Omega\Big(\frac{t \cdot m \cdot \gamma}{g^4} \Big) \Big\}.$$

Finally, we apply the gap amplification theorem from [6] to the dual-basis projector Hamiltonian problem.

▶ **Corollary 5.7.** *There exists a polynomial p and a constant $c_{\mathrm{LH}} > 0$ such that the problem*

$$\mathcal{DLH}(\mathrm{poly}(n), 2^{-p(n)}, c_{\mathrm{LH}})$$

is QMA-complete.

Proof. The completeness follows from the fact that, within each layer, all projectors commute. The maximum eigenvalue $\lambda_{\max}(\prod_{i \in [m_\chi]} (I - \Pi_i^\chi))$ is lower bounded by $1 - \min_{|\varphi\rangle} \sum_i \langle\varphi| \Pi_i^\chi |\varphi\rangle$. In the yes-instance, there exists a state $|\varphi\rangle$ such that each $\langle\varphi| \Pi_i^\chi |\varphi\rangle \leq 1/\exp(n)$, since there are at most $\mathrm{poly}(n)$ such terms. Moreover, because each layer contains at most $\mathrm{poly}(n)$ terms, the completeness follows.

The soundness follows from Theorem 5.4 and Theorem 5.6. ◀

6 Evaluating Observables on Corrupted Codewords of a CSS Code

This section establishes some utility lemma on the structure of corrupted codewords of a quantum CSS code. The main lemma of the section is Lemma 6.6, which compares two different methods of evaluating a "logical" operation on a corrupted codeword.

We first formalize the notion of local testing for a quantum locally testable code (qLTC).

▶ **Definition 6.1** (Local testing of a qLTC code). *Let $C = CSS(H_X, H_Z)$ be a stabilizer code on n qubits, and let $|\varphi\rangle$ be a quantum state. The verifier performs the following steps c times:*

- *The verifier randomly samples $W \in \{X, Z\}$.*
- *The verifier randomly samples a row r_W from H_W.*
- *The verifier measures the observable $W(r_W)$ on the state $|\varphi\rangle$ and rejects if the measurement outcome is -1.*

If none of the tests reject, the verifier accepts.

▶ **Lemma 6.2.** *Let $C = CSS(H_X, H_Z)$ be a stabilizer code on n qubits such that $H_X \in \mathbb{F}_2^{c_X \times M}$, $H_Z \in \mathbb{F}_2^{c_Z \times M}$, are parity check matrices associated with classical locally testable codes with soundness ρ. Let $\mathcal{S}$ be defined as in Lemma 2.15. Let*

$$|\psi\rangle = \sum_{(a,b)\in\mathcal{S}} \alpha_{a,b} X(a)Z(b) |\eta_{a,b}\rangle$$

be a quantum state, where $\{|\eta_{a,b}\rangle\}$ are a family of codewords. Then the verifier in Definition 6.1 accepts $|\psi\rangle$ with probability at most

$$\Pr[\textit{verifier accepts}] \leq \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \cdot \left(1 - \frac{\rho}{2M} \cdot wt_{cent}(X(a)Z(b))\right)^c .$$

▶ Remark 6.3. A similar statement was first proposed in [3]. The result was further strengthened in [14], although no proof was provided. We strengthen the result even further.

Proof. Let the verifier in Definition 6.1 be denoted V. Suppose the sequence of observables measured by V is $(W_i(r_i))_{i\in[c]}$. Note that these observables pairwise commute. Thus

$$\begin{aligned}
\Pr\left[V \text{ accepts } \mid (W_i(r_i))_{i\in[c]}\right] &= \left\| \prod_{i\in[c]} \frac{I + W_i(r_i)}{2} |\varphi\rangle \right\|^2 \\
&= \langle\varphi| \prod_{i\in[c]} \frac{I + W_i(r_i)}{2} |\varphi\rangle \\
&= \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \langle\eta_{a,b}| Z(b)X(a) \\
&\qquad \cdot \Big(\prod_{i\in[c]} \frac{I + W_i(r_i)}{2} \Big) X(a)Z(b) |\eta_{a,b}\rangle \ .
\end{aligned}$$

Here, the second step follows from $(\frac{I+W_i(r_i)}{2})^2 = \frac{I+W_i(r_i)}{2}$, and the third step follows from the fact that the $X(a)Z(b)|\eta_{a,b}\rangle$ are orthonormal eigenvectors of the $W_i(r_i)$. Continuing, since the $|\eta_{a,b}\rangle$ are codewords,

$$\langle\eta_{a,b}| Z(b)X(a) \Big(\prod_{i\in[c]} \frac{I + W_i(r_i)}{2} \Big) X(a)Z(b) |\eta_{a,b}\rangle = \prod_{i\in[c]} 1_{[X(a)Z(b), W_i(r_i)]=0} \ .$$

Averaging over the verifier's choices of W and r_i,

$$\Pr[V \text{ accepts}] = \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \cdot \left(\begin{array}{l} \frac{1}{2} \cdot \mathop{\mathbb{E}}_{i\in_R[c_X]} 1_{[X(a)Z(b),X((H_X)_i)]=0} \\ + \frac{1}{2} \cdot \mathop{\mathbb{E}}_{i\in_R[c_Z]} 1_{[X(a)Z(b),Z((H_Z)_i)]=0} \end{array} \right)^c$$

$$= \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \cdot \left(\frac{c_X - wt(H_X b)}{2c_X} + \frac{c_Z - wt(H_Z a)}{2c_Z} \right)^c . \tag{5}$$

Let C_X be the locally testable code corresponding to the X-check matrix H_X. Let C_Z be the locally testable code corresponding to the Z-check matrix H_Z. By Definition 2.4,

$$wt(H_Z a) \geq \rho \cdot \frac{c_Z}{M} \cdot d_H(a, C_Z) ,$$

$$wt(H_X b) \geq \rho \cdot \frac{c_X}{M} \cdot d_H(b, C_X) .$$

Moreover, for any $W \in \{X, Z\}$ and $s \in \mathbb{F}_2^n$, let W' be the complementary basis of W, then

$$\begin{aligned} d_H(s, C_W) &= \min_{r\in\mathbb{F}_2^M, H_W\cdot r=0} \{wt(r+s)\} \\ &\geq \min_{a',b'\in\mathbb{F}_2^n, H_X b'=0, H_Z a'=0} \{wt(X(a')Z(b')(W')(s))\} \\ &= wt_{\text{cent}}(W'(s)) , \end{aligned}$$

and

$$wt_{\text{cent}}(X(a)) + wt_{\text{cent}}(Z(b)) \geq wt_{\text{cent}}(X(a)Z(b)) .$$

Plugging back into (5),

$$\begin{aligned} \Pr[V \text{ accepts}] &\leq \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \cdot \left(1 - \frac{\rho}{2M} \cdot d_H(b, C_X) - \frac{\rho}{2M} d_H(a, C_Z)\right)^c \\ &\leq \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \cdot \left(1 - \frac{\rho}{2M} \cdot wt_{\text{cent}}(X(a)Z(b))\right)^c . \end{aligned}$$ ◀

The next two lemmas, albeit simple, are used repeatedly throughout the proofs. In particular, they appear in the proof of the lemma that follows them.

▶ **Lemma 6.4.** *Recall the notation introduced in Section 3. For any $u, v \in C_W$, if $|u - v| < d_{q,W}$, then*

$$Ku = Kv.$$

Proof. Let W' be the complementary basis of W.

Since $u, v \in C_W$, it follows that $u - v \in C_W$. Moreover,

$$|u - v| < d_{q,W} = \min\{\, |w| : w \in \ker(H_W) \setminus \mathrm{Im}(H_{W'}^T) \,\},$$

implies that $u - v \in \mathrm{Im}(H_{W'}^T)$.

Because $KH_{W'}^T = 0$ by (1), we conclude that $K(u - v) = 0$, and hence $Ku = Kv$. ◀

▶ **Lemma 6.5.** *Recall the notation introduced in Section 3. Then, for any $p \in \mathbb{F}_2^M$ such that $|p| < \eta_W d_{q,W}$, then for any $v \in \mathbb{F}_2^{B_W}$, it holds that*

$$KE_W \, \mathtt{D}_W(p + E_W v) = KE_W v.$$

Proof. By Fact 2.22,

$$\left| \mathtt{E}_W v + \mathtt{E}_W \mathtt{D}_W (p + E_W v) \right| \leq \frac{1}{\eta_W} \cdot |p| < d_{q,W}.$$

Then, by Lemma 6.4,

$$K E_W \, \mathtt{D}_W (p + E_W v) = K E_W v. \qquad \blacktriangleleft$$

The following important lemma relates measurements obtained on the decoded quantum word (the observable W'), to measurements obtained directly on the encoded word but subsequently decoded according to the classical code (the observable W''). The lemma will be a key workhorse of soundness arguments in the following sections.

▶ **Lemma 6.6.** *Recall the notations introduced in Section 3. Let*

$$|\beta\rangle = \sum_{(a,b)\in\mathcal{S}} 1(wt_{cent}(X(a)Z(b)) < \eta_q d_q) \cdot \alpha_{a,b} X(a)Z(b) \, |\eta_{a,b}\rangle$$

be a quantum state, where $\{|\eta_{a,b}\rangle\}$ are codewords. Then, for any Boolean function $f : \{0,1\}^{B_q} \to \{0,1\}$,

$$W''_{\widetilde{f},W} \, |\beta\rangle = W'_{f,W} \, |\beta\rangle \, .$$

▶ Remark 6.7. Note that, for $v \in \mathbb{F}_2^{B_W}$, $\mathtt{D}_W(E_W v) = v$.

Proof.

Part I: analysis of the behavior of W'_f. Let $|\theta_{a,b}\rangle$ be such that $|\eta_{a,b}\rangle = E_q(|\theta_{a,b}\rangle)$. For $x \in \mathbb{F}_2^{B_q}$ let $|\varphi_x\rangle = E_q(|x\rangle)$ be as in Definition 2.13. Let $|\psi_x\rangle = \sum_{i\in\mathbb{F}_2^{B_q}} (-1)^{i\cdot x} |\varphi_i\rangle$ Let $f'_W : \{0,1\}^M \to \{0,1\}$ be $f'_W(x) = f(K_W x)$.

Note that in the definition of $|\varphi_x\rangle$ from Definition 2.13, the projection

$$\Pi_x = \prod_{j\in[k]} \frac{I + (-1)^{x_j} Z(G_j)}{2}$$

is diagonal in the computational basis. For any $y \in \mathbb{F}_2^M$ in the support of Π_x, we have $G_j y = x_j$. Therefore, $f'(y) = f(Gy) = f(x)$, and

$$\begin{aligned} &Z_{f'} E_q(|x\rangle) \\ &= (-1)^{f(x)} \cdot E_q(|x\rangle), \end{aligned}$$

and similarly,

$$\begin{aligned} &X_{f'} E_q(H\,|x\rangle) \\ &= (-1)^{f(x)} \cdot E_q(H\,|x\rangle). \end{aligned}$$

Using the definition of F and G,

$$Z_{f'} \, |\varphi_x\rangle = (-1)^{f(x)} \cdot E_q(|x\rangle) = E_q(Z_f \, |x\rangle) \, ,$$

and

$$X_{f'} \, |\psi_x\rangle = (-1)^{f(x)} \cdot E_q(H \, |x\rangle) = E_q(X_f H \, |x\rangle) \, ,$$

Expanding $|\eta_{a,b}\rangle$ in the codeword basis given by the $|\varphi_x\rangle$ or $|\psi_x\rangle$ it follows by linearity that

$$W_{f'}|\eta_{a,b}\rangle = E_q(W_f|\theta_{a,b}\rangle)\,.$$

By the definitions of d_q, W'_f, and D_q in Notation 3.1, we have, for $(a,b)\in\mathcal{S}$,

$$\begin{aligned} W'_f X(a)Z(b)|\eta_{a,b}\rangle &= D_q^\dagger W_f D_q X(a)Z(b)|\eta_{a,b}\rangle \\ &= D_q^\dagger W_f|\theta_{a,b}\rangle|H_Z a\rangle|H_X b\rangle \\ &= D_q^\dagger D_q(W_{f'}|\eta_{a,b}\rangle)|H_Z a\rangle|H_X b\rangle \\ &= X(a)Z(b)W_{f'}|\eta_{a,b}\rangle\,. \end{aligned} \tag{6}$$

Part II: analysis of the behavior of $W''_{w^T K E_W}$. Let $v\in\mathbb{F}_2^B$,

$$W''_{w^T K E_W} S|E_W v\rangle = (-1)^{w^T K E_W v} S|E_W v\rangle = W(w^T K)S|E_W v\rangle\,.$$

Therefore,

$$W''_{w^T K E_W}|\eta_{a,b}\rangle = W(w^T K)|\eta_{a,b}\rangle\,.$$

Moreover, we have that

$$\begin{aligned} &D^\dagger_{Z,\mathsf{TW}} Z_{\widetilde{f},\mathsf{T}} D_{Z,\mathsf{TW}} X(p)Z(q)|E_Z v\rangle \\ &= (-1)^{p^T q} D^\dagger_{Z,\mathsf{TW}} Z_{\widetilde{f},\mathsf{T}} D_{Z,\mathsf{TW}} Z(q)X(p)|E_Z v\rangle \\ &= (-1)^{p^T q} D^\dagger_{Z,\mathsf{TW}} Z_{\widetilde{f},\mathsf{T}} D_{Z,\mathsf{TW}} Z(q)|p+E_Z v\rangle \\ &= (-1)^{p^T q} Z(q) D^\dagger_{Z,\mathsf{TW}} Z_{\widetilde{f},\mathsf{T}} D_{Z,\mathsf{TW}}|p+E_Z v\rangle \\ &= (-1)^{p^T q + f(KE_Z \mathtt{D}_Z(p+E_Z v))} Z(q)|p+E_Z v\rangle \\ &= (-1)^{p^T q + f(KE_Z \mathtt{D}_Z(p+E_Z v))} Z(q)X(p)|E_Z v\rangle \\ &= (-1)^{f(KE_Z \mathtt{D}_Z(p+E_Z v))} X(p)Z(q)|E_Z v\rangle\,. \end{aligned}$$

Similarly, we have that

$$D^\dagger_{X,\mathsf{TW}} X_{\widetilde{f},\mathsf{T}} D_{X,\mathsf{TW}} X(p)Z(q)H|E_X v\rangle = (-1)^{f(KE_X \mathtt{D}_X(q+E_X v))} X(p)Z(q)H|E_X v\rangle\,.$$

Moreover,

$$\begin{aligned} &X(p)Z(q)Z_{f'}|E_Z v\rangle \\ &= (-1)^{f(KE_Z v)} X(p)Z(q)|E_Z v\rangle\,, \end{aligned}$$

and

$$\begin{aligned} &X(p)Z(q)X_{f'}H|E_X v\rangle \\ &= (-1)^{f(KE_X v)} X(p)Z(q)H|E_X v\rangle\,. \end{aligned}$$

Since $wt_{\mathrm{cent}}(X(a)Z(b)) < \eta_q d_q$, we have that $|a|,|b| < \eta_q d_q$. By Lemma 6.5,

$$GE_Z \mathtt{D}_Z(a+E_Z v) = GE_Z v\,, \qquad FE_X \mathtt{D}_X(b+E_X v) = FE_X v\,.$$

Therefore,

$$W''_{\widetilde{f}} X(a)Z(b)|\eta_{a,b}\rangle = X(a)Z(b)W_{f'}|\eta_{a,b}\rangle\,. \tag{7}$$

Part III: Putting things together. Combining (6) and (7),

$$W'_f X(a)Z(b)\,|\eta_{a,b}\rangle = W''_{\widetilde{f}} X(a)Z(b)\,|\eta_{a,b}\rangle\,,$$

and

$$W'_f\,|\beta\rangle = W''_{\widetilde{f}}\,|\beta\rangle\,. \qquad \blacktriangleleft$$

▶ **Lemma 6.8.** *Recall the notations introduced in Section 3. Let*

$$|\psi\rangle = \sum_{(a,b)\in\mathcal{S}} \alpha_{a,b} X(a)Z(b)\,|\eta_{a,b}\rangle$$

be a quantum state, where $\{|\eta_{a,b}\rangle\}$ *are a family of codewords. Then, for any boolean function* $f : \mathbb{F}_2^{B_q} \to \mathbb{F}_2$,

$$\|W''_{\widetilde{f}}\,|\psi\rangle - W'_f\,|\psi\rangle\,\|^2 \leq \Theta(1)\cdot \sum_{(a,b)\in\mathcal{S}} 1(wt_{cent}(X(a)Z(b)) \geq \eta_q d_q)\cdot |\alpha_{a,b}|^2\,.$$

Proof. This follows directly from Lemma 6.6. ◀

7 PCP of Proximity Applied to Quantum States

In this section we introduce and analyze a simple interaction between prover and verifier in which the prover is asked to compute a PCPP for the validity of a certain computation (abstracted through the function f below) when given an encoded input to the computation and a target output. This will be used as a building block later on.

7.1 Actions of the honest prover

In the following definition the prover is given registers U, B, and E. The register U represents the (target) output of a computation. The register B represents the witness for the verification of the computation, which the prover is to compute: initially, the register B is all $|0\rangle$. Finally, E is prepared as the encoded, according to a classical error correcting code, input of the computation.

Note that, to maintain the flexibility and generality of the results in this section, we do not adopt a specific classical code (e.g., the one defined in Notation 3.1); instead, we state the results for any classical error-correcting code C_P with appropriate parameters.

▶ Remark 7.1. In this section as in subsequent ones we describe the actions of a honest prover in a coherent form, i.e. replacing measurements by copying information, etc. This is to preserve the protocol's flexibility, e.g., the ability to generalize it to handle more complex observables. Any prover can be written in such a purified form, even though a dishonest prover may of course apply any unitaries they like.

▶ **Definition 7.2** (PCP of proximity for verifying a computation, honest prover)**.** *Recall the notations introduced in Section 3.*

Let C_P *be a classical error-correcting code with dimension* k*, codeword length* $n_P(k)$*, decoder* D_P*, code tester* T_P*, relative distance* Δ_P*, and* $\eta_P\Delta_P$ *the weight of errors that* D_P *can correct. That is, for any* $e \in \mathbb{F}_2^{n_P(k)}$ *and* $w \in C_P$ *satisfying* $|e| < \eta_P\Delta_P$*, we have* $D_P(w+e) = D_P(w)$*.*

Let U *be an* l_x*-qubit register denoting the output,* E *an* l_y*-qubit register denoting the encoded input, let* $f : \mathbb{F}_2^{n_P^{-1}(l_y)} \to \mathbb{F}_2^{l_x}$ *represent the verification of the computation. Let* $g : \mathbb{F}_2^{l_x} \times \mathbb{F}_2^{l_y} \to \mathbb{F}_2$ *be defined as*

$$g(x, y) = 1(f(D_P(y)) = x) \wedge T_P(y) .$$

Let s *be the size of the circuit* g*. Let* B *be a quantum register of* $l_w(s)$ *qubits. The honest prover performs the following steps:*

- *The prover computes an* $l_w(s)$*-bit long PCPP proof for a statement regarding the content of register* E*. This proof is stored in register* B*. Formally, the prover applies the following unitary:*

  $$N_{g,\mathsf{BUE}} .$$

- *The prover sends the verifier the quantum registers* B*.*

▶ Remark 7.3. Throughout the paper, C denotes either C_C itself or the code $C_C^{\oplus n}$, where each block corresponds to an independent codeword from the base code C_C.

7.2 Actions of the verifier

Before running the following protocol, the honest prover has already sent U, the register storing the computation output, to the verifier.

▶ **Definition 7.4** (PCP of proximity for verifying a computation, verifier)**.** *Use the notation in Definition 7.2. The verifier performs the following steps:*

- *The verifier executes the PCPP verifier on input* (g_x, y) *and stores the verification result in a single qubit register* C*, where* x *is the contents of register* U*,* y *is the contents of register* E *and the proof is in register* B*. The randomness is stored in register* R*. Formally, the verifier initialize* C *to* $|0\rangle$ *and* R *to* $l_r(s)$ *classical randomness bits, then the verifier applies*

  $$T_{g,\mathsf{CUEBR}}.$$

- *The verifier rejects if the PCPP verifier rejects. Formally, the verifier measures* C *and rejects if the outcome is* 0*.*

7.3 Main results

We now state the main lemma in this section. Informally, the lemma states that as long as the decoder of the classical ECC can decode errors smaller than the proximity parameter of the PCPP, the verifier's acceptance probability is at most the sum of the PCPP soundness parameter and the probability that a measurement of the whole quantum system yields a classical string that is a good instance. Note that we can obtain a PCPP verifier with improved soundness (while preserving the proximity parameter) by repeatedly applying the PCPP verifier, at the cost of increased query complexity.

▶ **Lemma 7.5** (Implication of the PCPP test)**.** *Let* s, δ *be defined as in Notation 3.2. Use the notation introduced in Definition 7.4.*

Let O *be a quantum register representing all registers in the system except for* U*,* E*, and* B*. Let* $|\varphi\rangle_{\mathsf{OUEB}}$ *be the quantum state of the system before the prover returns the quantum registers* B*. Let the state be written as*

$$|\varphi\rangle = \sum_{o,u,e,b} \alpha_{o,u,e,b} \, |o\rangle_{\mathsf{O}} \, |u\rangle_{\mathsf{U}} \, |e\rangle_{\mathsf{E}} \, |b\rangle_{\mathsf{B}} .$$

If $\eta_P \Delta_P \geq \delta$, then

- *The acceptance probability of the verifier is upper bounded by*

$$\Pr[\textit{PCPP test accepts}] \leq \max\left\{s, \sum |\alpha_{o,u,e,b}|^2 \cdot 1\Big(f(\mathtt{D}_P(e)) = u\Big)\right\}.$$

- *The rejection probability of the verifier is lower bounded by*

$$\Pr[\textit{PCPP test rejects}] \geq -s + \sum |\alpha_{o,u,e,b}|^2 \cdot 1\Big(f(\mathtt{D}_P(e)) \neq u\Big).$$

Proof. For $u \in \mathbb{F}_2^{l_x}$, define

$$S_u = \{e \in \mathbb{F}_2^{l_y} \mid 1(f(\mathtt{D}_P(e)) = u) \wedge \mathtt{T}_P(e) = 1\}.$$

We analyze the acceptance probability:

$$\Pr[\text{verifier accepts}] = \sum |\alpha_{o,u,e,b}|^2 \cdot \Pr[\text{verifier accepts } |o,u,e,b\rangle].$$

- If $\operatorname{dist}(e, S_u) > \delta$, then

$$\Pr[\text{PCPP accepts } u,e,b] \leq s.$$

- If $\operatorname{dist}(e, S_u) \leq \delta$, then

$$\Pr[\text{PCPP accepts } u,e,b] \leq 1 = 1\Big(f(\mathtt{D}_P(e)) = u\Big),$$

where the last step follows from

$$\delta < \eta_P \Delta_P.$$

As a result,

$$\begin{aligned}\Pr[\text{verifier accepts}] &\leq \sum |\alpha_{o,u,e,b}|^2 \cdot \max\left\{s, 1\Big(f(\mathtt{D}_P(e)) = u\Big)\right\} \\ &\leq \max\left\{s, \sum |\alpha_{o,u,e,b}|^2 \cdot 1\Big(f(\mathtt{D}_P(e)) = u\Big)\right\},\end{aligned}$$

and

$$\Pr[\text{verifier rejects}] \geq -s + \sum |\alpha_{o,u,e,b}|^2 \cdot 1\Big(f(\mathtt{D}_P(e)) \neq u\Big).$$ ◀

8 Global Measurement Extraction

In our protocol, the quantum proof is given as a qLTC encoding of a witness. The verifier first checks the correctness of the qLTC encoding. Ultimately, the witness used will be an amplified version of a witness for the QMA language instance provided as input. In the following, we present the general procedure for verification of the encoding, when applied to an arbitrary witness.

In the end, our goal is to measure observables of the form X_f or Z_f on the physical qubits. Since the procedures are very similar, we use the case of measuring Z_f as an example. To perform this measurement, the verifier must extract measurement outcomes in the Z basis. This overall process is referred to as *Global Measurement Extraction*. The Global Measurement Extraction procedure consists of repeatedly applying the *Single-Round Measurement Extraction* subroutine to successive portions of the qLTC-encoded state until all qubits are measured.

The Single-Round Measurement Extraction process works as follows: the verifier sends a small portion of the encoded witness, consisting of m qubits, to the prover. The only requirement is that this portion must be much smaller than the recoverable distance of the code, i.e. an error of weight at most m can be successfully corrected by the code decoder.

For concreteness, consider the case where the verifier sends the first m qubits. The prover is then asked to measure these m qubits in the Z basis, and to reply with:

- the measurement results themselves,
- a classical ECC encoding of the purified measurement results, and
- m different PCPPs, each proving that the corresponding bit of the ECC decoding matches the claimed measurement result.

The prover also retains a copy of the measurement outcomes.

Later in the protocol, the ECC encoding will be used as a proxy for the measurement outcomes on this portion of the qubits (as opposed to the measurement results themselves, also reported). Ideally, whenever the verifier wishes to access information about this portion, he would decode the ECC and compute based on the decoded information. However, decoding the ECC is not local. Fortunately, when trying to evaluate Z_f the verifier only needs access to information related to Z basis measurements. Therefore, instead of decoding the ECC, the verifier leverages classical PCPPs: he asks the prover to prove, via PCPPs, that any computation on the decoded results would yield the claimed outcome.

After receiving the prover's message, the verifier checks the Z part of the qLTC code. Specifically, the verifier randomly samples a Z-type stabilizer. Since stabilizers are local, they act on only a small number of qubits. For qubits that the verifier holds, he can measure them directly. For qubits that the prover already measured in the Z-basis, the verifier reads the reported bits and checks their correctness using the corresponding PCPP proofs. In particular, the prover must prove that the ECC decoding yields the correct value at the qubits involved in the stabilizer.

Note that the prover could introduce additional errors into the ECC-encoded portion during the Single-Round Measurement Extraction. Because Z (phase) errors do not affect Z basis measurement outcomes, we only need to worry about X (bit-flip) errors. The Z-LTC test performed after receiving the prover's message ensures that all X errors remain bounded to a small level.

The verifier then applies the Single-Round Measurement Extraction process to the next portion of the remaining qLTC-encoded witness. This process is repeated until all physical qubits have been measured in the Z-basis.

8.1 Actions of the honest prover

We now describe formally the basic procedure through which the prover extracts measurement results from a small portion of the quantum state. This process, which we refer to as *Single Round Measurement Extraction*, enables the prover to verifiably encode measurement outcomes for a small block of qubits. Even though in the protocol itself the prover is required to send back classical information, for purposes of modeling and analysis we will consider the actions of a purified prover, i.e. such that measurement outcomes are stored coherently in an outcome register. This register is later to be measured, as it is sent from prover to verifier.

Before we start, we introduce some notation that will be useful for presenting the prover's protocol.

▶ **Definition 8.1** (Controlled unitary). *Let* P *be a register and* U_{P} *a unitary acting on it. Let* R *be a one-qubit register. The controlled unitary* $C(U_{\mathsf{P}}, \mathsf{R})$ *applies* U_{P} *if the qubit in* R *is* $|1\rangle$*:*

$$C(U_{\mathsf{P}}, \mathsf{R}) = \begin{pmatrix} I_{\mathsf{P}} & 0 \\ 0 & U_{\mathsf{P}} \end{pmatrix} .$$

▶ **Definition 8.2** (Purified measurement). *Let* P, R *be registers and* O_{P} *an observable decomposed as* $O = \sum_{i=1}^{r} a_i P_i$*, where each* P_i *is an orthogonal projection and* $I = \sum_i P_i$*. The unitary that performs the purified measurement and stores the outcome in* R *is:*

$$P(O_{\mathsf{P}}, \mathsf{R}) = \sum_{i=1}^{r} P_{i,\mathsf{P}} \cdot (X(a_i))_{\mathsf{R}} \ .$$

The honest prover's actions in this procedure are defined below.

▶ **Definition 8.3** (Single round measurement extraction, honest prover). *Let* W *be a single-qubit observable. The honest prover performs the following steps:*

- *The verifier sends the prover a quantum register* W *consisting of* m *qubits.*
- *The prover measures the observable* W *m times, on each of the qubits of* W*, and stores two copies of each result, in single-qubit registers* X_i *and* X'_i *respectively.*
  Formally, the prover applies the following unitary:

$$\prod_{i\in[m]} P(W_{\mathsf{W}}(e_i), \mathsf{X}_i) P(W_{\mathsf{W}}(e_i), \mathsf{X}'_i) \ .$$

- *The prover encodes the* m*-qubit registers* X *and* X' *into* $n(m)$*-qubit registers* E *and* E' *respectively using the classical code from Theorem 2.1.*
  Formally, the prover applies the unitaries E_{EX}*,* $E_{\mathsf{E'X}}$ *where* E *is the encoding operator from Definition 2.3.*
- *For each* $i \in [m]$ *the prover computes a PCPP proof for the statement that the content of register* E *is a valid codeword such that the* i*-th bit of the encoded message matches the contents of register* X_i*. This proof is stored in register* B_i*.*
  Formally, for each $i \in [m]$*, the prover runs the procedure in Definition 7.2 with* $C = C_C, f(x) = (x)_i, \mathsf{U} = \mathsf{X}_i, \mathsf{E} = \mathsf{E}, \mathsf{B} = \mathsf{B}_i$*.*
- *Finally, the prover sends the verifier the quantum registers* WXE*.*

▶ Remark 8.4. Note that the register B is sent to the verifier during the procedure in Definition 7.2.

We extend the single round measurement extraction procedure to a global measurement extraction procedure. The global measurement extraction process involves repeating the single round measurement extraction across multiple disjoint portions of the qLTC encoded state. We formally define the honest prover's actions for this procedure below.

▶ **Definition 8.5** (Global measurement extraction, honest prover). *Recall the notations introduced in Section 3. The honest prover performs the following steps, for* $i = 1, \ldots, M/m$*:*

- *The prover executes the actions described in Definition 8.3, using* $W = W, \mathsf{B} = \mathsf{B}_i, \mathsf{E} = \mathsf{E}_i, \mathsf{E}' = \mathsf{E}'_i, \mathsf{X} = \mathsf{X}_i, \mathsf{X}' = \mathsf{X}'_i$*.*

8.2 Actions of the verifier

We now describe the verifier's actions in the single round measurement extraction process, corresponding to the honest prover's strategy previously defined.

▶ **Definition 8.6** (Single round measurement extraction, verifier). *Let* W *be a single-qubit observable. The verifier performs the following steps:*

- *The verifier sends the prover a quantum register* W *consisting of* m *qubits.*
- *The prover sends back the quantum registers* WXEB*, where* $\mathsf{B} = (\mathsf{B}_1, \ldots, \mathsf{B}_m)$*.*

We now describe the subroutine used by the verifier to check the validity of a partially extracted quantum state against a qLTC code. This procedure ensures that the measurement results encoded in the prover's messages are consistent with the structure of the code.

▶ **Definition 8.7** (Validity check, verifier). *Recall the notations introduced in Section 3. In addition, let $I \subseteq [M]$ such that, for every $i \in [M/m]$, either $((i-1)\cdot m, i\cdot m] \subseteq I$ or $((i-1)\cdot m, i\cdot m] \subseteq [M]\setminus I$ holds.*

The verifier performs the following steps:

- *The verifier samples queries $(j_1, \ldots, j_Q) \in [M]^Q$ for the code tester L_W, where Q is the query complexity.*
- *The verifier initializes a single-qubit register T' and a Q-qubit register T in state $|0\rangle$.*
- *For each $k \in [Q]$, the verifier performs the following.*
  If $j_k \in I$, the verifier directly copies register W_{j_k} in register T_k. Formally, the verifier applies $C(\mathsf{T}_k, \mathsf{W}_{j_k})$.
  If $j_k \in [M]\setminus I$, the verifier "extracts" W_{j_k} from registers provided by the prover, as follows:
  - *Let (p, q) be such that j_k is the q-th element of the p-th block of m qubits among $\mathsf{W}_1, \ldots, \mathsf{W}_M$.*
  - *The verifier runs the procedure in Definition 7.4 with $C = C_C, f(x) = (x)_q, \mathsf{U} = \mathsf{X}_{p,q}, \mathsf{E} = \mathsf{E}_p, \mathsf{B} = \mathsf{B}_{p,q}$. The verifier rejects if this procedure rejects.*
  - *The verifier copies register $\mathsf{X}_{p,q}$ in register T_k. Formally, the verifier applies $C(\mathsf{T}_k, \mathsf{X}_{p,q})$.*
- *The verifier rejects if the bits $(\mathsf{T}_1, \ldots, \mathsf{T}_Q)$ are rejected by the code tester L. Otherwise, it accepts.*
  Formally, the verifier applies the unitary $L_{\mathsf{T}'\mathsf{T},r}$, where L is the local tester operator as in Definition 2.7, measures T' and rejects if the outcome is 0.

Before defining the full global measurement extraction procedure for the verifier, we recall that this process allows the verifier to sequentially extract measurement outcomes from different parts of the qLTC-encoded state, while enforcing local consistency conditions verified through PCPP checks. We now describe the detailed steps of the verifier's actions.

▶ **Definition 8.8** (Global measurement extraction, verifier). *Recall the notations introduced in Section 3. Let $s \in \mathbb{F}_2^{n+1}$.*

- *If $s_0 = 1$, the verifier first runs the local tester of the qLTC on the quantum state in register W and rejects if the test fails. More specifically, the verifier runs the procedure from Definition 8.7 with parameters $I = [M]$, $W = W$, $\mathsf{W} = \mathsf{W}$, $\mathsf{B} = \mathsf{B}$, $\mathsf{E} = \mathsf{E}$, and $\mathsf{X} = \mathsf{X}$, and rejects if this procedure rejects.*
- *The verifier performs the following steps, repeating the actions described in Definition 8.6 a total of $r = M/m$ times. In the i-th iteration:*
  - *The verifier applies the procedure from Definition 8.6 using $W = W, \mathsf{W} = \mathsf{W}_i$, $\mathsf{B} = \mathsf{B}_i$, $\mathsf{E} = \mathsf{E}_i$, and $\mathsf{X} = \mathsf{X}_i$.*
  - *If $s_i = 1$, the verifier runs the procedure from Definition 8.7 with parameters $I = (i\cdot m, M]$, $W = W$, $\mathsf{W} = \mathsf{W}$, $\mathsf{B} = \mathsf{B}$, $\mathsf{E} = \mathsf{E}$, and $\mathsf{X} = \mathsf{X}$, and rejects if this procedure rejects.*

8.3 Main results

▶ **Lemma 8.9** (Implication of the validity test). *Fix $W \in \{X, W\}$. Recall the notations introduced in Section 3. Fix notation as in Definition 8.7. Let*

$$|\varphi\rangle_{\mathsf{WBEXP}} = \sum_{p,x,e,b,w} \alpha_{p,x,e,b,w} \, |p\rangle_{\mathsf{P}} \, |x\rangle_{\mathsf{X}} \, |e\rangle_{\mathsf{E}} \, |b\rangle_{\mathsf{B}} \, (S \, |w\rangle_{\mathsf{W}})$$

be the quantum state of the system before the test defined in Definition 8.7.

For $x = (x_i)_{i\in[M/m]}$, $e = (e_i)_{i\in[M/m]}$, $b = (b_i)_{i\in[M/m]}$, and $w = (w_i)_{i\in[M/m]}$, define for each $i \in [M/m]$:

- *If $((i-1)\cdot m, i\cdot m] \subseteq I$, let $r_i = w_i$;*
- *If $((i-1)\cdot m, i\cdot m] \subseteq [M]\setminus I$, let $r_i = \mathsf{D}(e_i)$.*

Suppose that $\delta < \eta_C\Delta$, then the rejection probability of the verifier from Definition 8.7 satisfies

$$\Pr[\textit{verifier rejects in the validity test}] \geq \kappa \cdot \sum |\alpha_{p,x,e,b,w}|^2 \cdot \frac{dist(r, C_W)}{M} - \Theta(s) \ .$$

Proof. Define for each $i \in [M/m]$:

- If $((i-1)\cdot m, i\cdot m] \subseteq I$, set $o_i = w_i$;
- If $((i-1)\cdot m, i\cdot m] \subseteq [M]\setminus I$, set $o_i = x_i$.

Fix a query $j_k \in [M]\setminus I$. Let $p = \lceil j_k/m\rceil$ and $q = j_k - m(p-1)$. By applying Lemma 7.5 with $C_P = C_C$ and $\delta < \eta_C\Delta$, we have that

$$\Pr[\text{PCPP accepts } x_{p,q}, e_p, b_{p,q}] \leq \max\Big\{1\big((r_p)_q = (o_p)_q\big),\ s\Big\}.$$

Then,

$$\begin{aligned}\Pr[\text{each PCPP accepts} \mid j_1,\dots,j_Q] &\leq \prod_{k=1}^{Q} \max\Big\{1(r_{j_k} = o_{j_k}),\ s\Big\} \\ &\leq \prod_{k=1}^{Q} 1(r_{j_k} = o_{j_k}) + s \ .\end{aligned}$$

Writing $j = (j_1,\dots,j_Q)$ for succinctness,

$$\begin{aligned}&\Pr[\text{verifier accepts } |p,x,e,b\rangle\, S\, |w\rangle] \\ &= \sum_j \Pr[j]\cdot\Pr[\text{local tester of } C_W \text{ accepts } o \mid j]\cdot\Pr[\text{each PCPP accepts} \mid j] \\ &\leq \sum_j \Pr[j]\cdot\Pr[\text{local tester of } C_W \text{ accepts } o \mid j]\cdot\Big(\prod_k 1(o_{j_k} = r_{j_k}) + s\Big) \\ &\leq \sum_j \Pr[j]\cdot\Pr[\text{local tester of } C_W \text{ accepts } r \mid j] + s \\ &\leq \Pr[\text{local tester of } C_W \text{ accepts } r] + s \\ &\leq 1 - \kappa\cdot\mathrm{dist}(r, C_W) + \Theta(s) \ .\end{aligned}$$

Therefore,

$$\begin{aligned}\Pr[\text{verifier rejects}] &= \sum |\alpha_{p,x,e,b,w}|^2 \cdot \Pr[\text{verifier rejects } |p,x,e,b\rangle\, S\, |w\rangle] \\ &\geq \sum |\alpha_{p,x,e,b,w}|^2 \cdot \Big(\kappa\cdot\frac{\mathrm{dist}(r,C_W)}{M} - \Theta(s)\Big) \\ &\geq \kappa\cdot\sum |\alpha_{p,x,e,b,w}|^2\cdot\frac{\mathrm{dist}(r,C_W)}{M} - \Theta(s) \ .\end{aligned}$$

◀

▶ **Lemma 8.10** (Implication of the single round measurement extraction)**.** *Recall the notations introduced in Section 3. Fix notation as in Definition 8.8. Let*

$$|\varphi\rangle = \sum_{p,x,e,b,w} \alpha_{p,x,e,b,w}\, |p\rangle_{\mathsf{P}}\, |x\rangle_{\mathsf{X}}\, |e\rangle_{\mathsf{E}}\, |b\rangle_{\mathsf{B}}\, S\, |w\rangle_{\mathsf{W}} \ ,$$

$$|\varphi'\rangle = \sum_{p,x,e,b,w} \alpha'_{p,x,e,b,w}\, |p\rangle_{\mathsf{P}}\, |x\rangle_{\mathsf{X}}\, |e\rangle_{\mathsf{E}}\, |b\rangle_{\mathsf{B}}\, S\, |w\rangle_{\mathsf{W}}$$

be the quantum state of the system respectively before and after the ith iteration of the single round measurement extraction in the global measurement extraction process introduced in Definition 8.8.

Let U_i be the global unitary that corresponds to the verifier and the prover's joint actions during the ith round of the interaction. For $x = (x_i)_{i\in[M/m]}$, $e = (e_i)_{i\in[M/m]}$, $b = (b_i)_{i\in[M/m]}$, and $w = (w_i)_{i\in[M/m]}$, define for each $j \in [M/m]$:

- *If $j \geq i$, let $r_j = w_j$;*
- *If $j < i$, let $r_j = D(e_j)$.*

Then,

$$\begin{aligned}&\|U_i Q_{i-1}(|0\rangle_{\mathsf{T}} |\varphi\rangle) - Q_i U_i(|0\rangle_{\mathsf{T}} |\varphi\rangle)\|^2 \\ &\leq \Theta(1) \cdot \Big(\sum_{dist(r,C_W)\geq \eta_q d_{q,W} - m} |\alpha_{p,x,e,b,w}|^2 + \sum_{dist(r,C_W)\geq \eta_q d_{q,W} - m} |\alpha'_{p,x,e,b,w}|^2 \Big) .\end{aligned}$$

Proof. Let

$$|\beta\rangle = \sum_{\mathrm{dist}(r,C_W)<\eta_q d_{q,W} - m} \alpha_{p,x,e,b,w} |p\rangle_{\mathsf{P}} |x\rangle_{\mathsf{X}} |e\rangle_{\mathsf{E}} |b\rangle_{\mathsf{B}} (S|w\rangle_{\mathsf{W}})$$

and

$$|\beta'\rangle = \sum_{\mathrm{dist}(r,C_W)<\eta_q d_{q,W} - m} \alpha'_{p,x,e,b,w} |p\rangle_{\mathsf{P}} |x\rangle_{\mathsf{X}} |e\rangle_{\mathsf{E}} |b\rangle_{\mathsf{B}} (S|w\rangle_{\mathsf{W}}) .$$

Let $\mathsf{E} = (\mathsf{E}_i)_{i\in[M/m]}$ and $\mathsf{W} = (\mathsf{W}_i)_{i\in[M/m]}$. Let W_0 be a quantum register of m qubits. Let E_0 be a quantum register of $n(m)$ qubits. Let $\mathsf{E}' = (\mathsf{E}_1, \ldots, \mathsf{E}_{i-1}, \mathsf{E}_0, \mathsf{E}_{i+1}, \ldots, \mathsf{E}_{M/m})$, $\mathsf{W}' = (\mathsf{W}_1, \ldots, \mathsf{W}_{i-1}, \mathsf{W}_0, \mathsf{W}_{i+1}, \ldots, \mathsf{W}_{M/m})$. Recall the partial composite decoding operator $D_{p,W,i}$ from Definition 2.23, and let

$$Q'_i = S^\dagger_{\mathsf{W}'} D^\dagger_{p,W,i,\mathsf{TE}'\mathsf{W}'} Z_{\widetilde{f},\mathsf{T}} D_{p,W,i,\mathsf{TE}'\mathsf{W}'} S_{\mathsf{W}'} ,$$

and

$$Q'_{i-1} = S^\dagger_{\mathsf{W}'} D^\dagger_{p,W,i-1,\mathsf{TE}'\mathsf{W}'} Z_{\widetilde{f},\mathsf{T}} D_{p,W,i-1,\mathsf{TE}'\mathsf{W}'} S_{\mathsf{W}'} .$$

By definition, for any quantum state $|\psi\rangle$ on registers $\mathsf{E}'_{-i}\mathsf{W}'_{-i}$:

$$Q'_{i-1}(|0\rangle_{\mathsf{TE}_0\mathsf{W}_0} |\psi\rangle) = Q'_i(|0\rangle_{\mathsf{TE}_0\mathsf{W}_0} |\psi\rangle) .$$

This is because $D_{p,W,i-1,\mathsf{TE}'\mathsf{W}'}$ in Q'_{i-1} uses the decoding of the first $i-1$ blocks of E' and the last $M/m - i + 1$ blocks of W', while $D_{p,W,i,\mathsf{TE}'\mathsf{W}'}$ in Q'_i uses the decoding of the first i blocks of E' and the last $M/m - i$ blocks of W'. However, both the ith block of E' and the ith block of W' are set to $|0\rangle$, so the two operators Q'_{i-1} and Q'_i act identically.

Note that

$$\begin{aligned}&Q_{i-1}(|0\rangle_{\mathsf{TE}_0\mathsf{W}_0} |\beta\rangle) \\ &= Q'_{i-1}(|0\rangle_{\mathsf{TE}_0\mathsf{W}_0} |\beta\rangle).\end{aligned}$$

This follows since $|\beta\rangle$ is supported on r such that $\mathrm{dist}(r, C_W) < \eta_q d_{q,W} - m$ and Lemma 6.5. More precisely, the difference between Q_{i-1} and Q'_{i-1} is that the former uses the ith block of W, while the latter replaces that block with $|0\rangle$. Decompose the state in the computational basis; then, applying the decoding to the first $i-1$ blocks of E (respectively E') and combining

the last $M/m - i + 1$ blocks of W (respectively W') gives us a string r (respectively r'). For an honest prover, r is a codeword of C_W, and r' is obtained from r by setting the ith block to 0. As long as the two decoded strings from r and r' are equal, or their respective differences from the same codeword both lie within the tolerance required by Lemma 6.5, the actions of Q_{i-1} and Q'_{i-1} will be the same (since we only consider $\widetilde{f}(x)$ of the form $f(KE_W x)$). Indeed, $\mathrm{dist}(r, C_W) < \eta_q d_{q,W} - m$ implies the existence of $s \in C_W$ such that $\mathrm{dist}(r, s) < \eta_q d_{q,W} - m \leq \eta_q d_{q,W}$, and hence $\mathrm{dist}(r', s) < \eta_q d_{q,W}$.

Therefore,

$$\begin{aligned}
&U_i Q_{i-1}(|0\rangle_{\mathsf{TE_0W_0}} |\beta\rangle) \\
&= U_i Q'_{i-1}(|0\rangle_{\mathsf{TE_0W_0}} |\beta\rangle) \\
&= U_i Q'_i(|0\rangle_{\mathsf{TE_0W_0}} |\beta\rangle) \\
&= Q'_i U_i(|0\rangle_{\mathsf{TE_0W_0}} |\beta\rangle) \ ,
\end{aligned}$$

Furthermore,

$$Q'_i(|0\rangle_{\mathsf{TE_0W_0}} |\beta'\rangle) = Q_i(|0\rangle_{\mathsf{TE_0W_0}} |\beta'\rangle) \ ,$$

Moreover,

$$\| \, |\beta\rangle - |\varphi\rangle \, \|^2 \leq \sum_{\mathrm{dist}(r, C_W) \geq \eta_q d_{q,W} - m} |\alpha_{p,x,e,b,w}|^2 \ ,$$

and

$$\| \, |\beta'\rangle - |\varphi'\rangle \, \|^2 \leq \sum_{\mathrm{dist}(r, C_W) \geq \eta_q d_{q,W} - m} |\alpha'_{p,x,e,b,w}|^2 \ .$$

Note that

$$\begin{aligned}
&\|U_i \, |\beta\rangle - |\beta'\rangle \, \|^2 \\
&\leq \Theta(1) \cdot (\|U_i \, |\beta\rangle - U_i \, |\varphi\rangle \, \|^2 + \| \, |\varphi'\rangle - |\beta'\rangle \, \|^2) \\
&= \Theta(1) \cdot (\| \, |\beta\rangle - |\varphi\rangle \, \|^2 + \| \, |\varphi'\rangle - |\beta'\rangle \, \|^2) \ .
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\|U_i Q_{i-1}(|0\rangle_{\mathsf{TE_0W_0}} |\varphi\rangle) - Q_i U_i(|0\rangle_{\mathsf{TE_0W_0}} |\varphi\rangle)\|^2 \\
&\leq \Theta(1) \cdot (\| \, |\beta\rangle - |\varphi\rangle \, \|^2 + \| \, |\varphi'\rangle - |\beta'\rangle \, \|^2) \\
&\leq \Theta(1) \cdot \Big(\sum_{\mathrm{dist}(r, C_W) \geq \eta_q d_{q,W} - m} |\alpha_{p,x,e,b,w}|^2 + \sum_{\mathrm{dist}(r, C_W) \geq \eta_q d_{q,W} - m} |\alpha'_{p,x,e,b,w}|^2 \Big) \ .
\end{aligned}$$

◀

▶ **Lemma 8.11** (Implication of the global measurement extraction)**.** *Use the same notation as in Lemma 8.10, except that now*

$$|\varphi\rangle_{\mathsf{WBEXP}} = \sum_{p,x,e,b,w} \alpha_{p,x,e,b,w} \, |p\rangle_{\mathsf{P}} \, |x\rangle_{\mathsf{X}} \, |e\rangle_{\mathsf{E}} \, |b\rangle_{\mathsf{B}} \, (S \, |w\rangle_{\mathsf{W}})$$

is the quantum state of the system before the global measurement extraction process. In particular, let Q_i, for $i \in \{0, \ldots, n\}$, be as defined in (3), for some $W \in \{X, Z\}$. Suppose that for each i,

$$\Pr[\textit{verifier rejects in the validity test in the } i\textit{th iteration}] \leq \varepsilon_i \ .$$

Assume that $\delta < \eta_C \Delta$. Suppose that during the interaction, the verifier and prover together apply the unitary U_b to the entire system. Then:

$$\|U_b Q_0(|0\rangle_{\mathsf{T}} |\varphi\rangle) - Q_n U_b(|0\rangle_{\mathsf{T}} |\varphi\rangle)\|^2 \leq \Theta\Big(\frac{r^2 \cdot (s + \mathbb{E}_i\, \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)}\Big)\,,$$

where $r = M/m$.

Proof. Let $|\varphi_k\rangle_{\mathsf{WBEXP}}$ be the quantum state of the system after the kth iteration of the single round measurement extraction in the global measurement extraction process. Let the state be written as

$$|\varphi_k\rangle = \sum_{p,x,e,b,w} \alpha_{k,p,x,e,b,w} |p\rangle_{\mathsf{P}} |x\rangle_{\mathsf{X}} |e\rangle_{\mathsf{E}} |b\rangle_{\mathsf{B}} (S|w\rangle_{\mathsf{W}})\,.$$

For $x = (x_i)_{i\in[M/m]}$, $e = (e_i)_{i\in[M/m]}$, $b = (b_i)_{i\in[M/m]}$, and $w = (w_i)_{i\in[M/m]}$, define for each $j \in [M/m]$:

- If $j \geq i$, let $r_j = w_j$;
- If $j < i$, let $r_j = \mathtt{D}(e_j)$.

By Lemma 8.9,

$$\begin{aligned}
&\Pr[\text{verifier rejects in the validity test in the } k\text{th iteration}] \\
&\geq \kappa \cdot \sum |\alpha_{k-1,p,x,e,b,w}|^2 \cdot \frac{\mathrm{dist}(r, C_W)}{M} - \Theta(s) \\
&\geq \kappa \cdot \sum_{\mathrm{dist}(r,C_W) \geq \eta_q d_{q,W} - m} |\alpha_{k-1,p,x,e,b,w}|^2 \cdot \frac{\eta_q d_{q,W} - m}{M} - \Theta(s).
\end{aligned}$$

Therefore,

$$\sum_{\mathrm{dist}(r,C_W) \geq \eta_q d_{q,W} - m} |\alpha_{k-1,p,x,e,b,w}|^2 \leq \Theta\Big(\frac{(s + \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)}\Big)\,. \tag{8}$$

By Lemma 8.10,

$$\begin{aligned}
&\|U_k Q_{k-1}(|0\rangle_{\mathsf{T}} |\varphi_{k-1}\rangle) - Q_k U_k(|0\rangle_{\mathsf{T}} |\varphi_{k-1}\rangle)\|^2 \\
&\leq \Theta(1) \cdot \Big(\sum_{\mathrm{dist}(r,C_W) \geq \eta_q d_{q,W} - m} |\alpha_{k-1,p,x,e,b,w}|^2 + \sum_{\mathrm{dist}(r_k,C_W) \geq \eta_q d_{q,W} - m} |\alpha_{k,p,x,e,b,w}|^2 \Big)\,.
\end{aligned}$$

Using the triangle inequality and plugging in (8),

$$\begin{aligned}
&\|U_b Q_0(|0\rangle_{\mathsf{T}} |\varphi\rangle) - Q_n U_b(|0\rangle_{\mathsf{T}} |\varphi\rangle)\| \\
&\leq \sum_{k\in[M/m]} \|U_k Q_{k-1}(|0\rangle_{\mathsf{T}} |\varphi_{k-1}\rangle) - Q_k U_k(|0\rangle_{\mathsf{T}} |\varphi_{k-1}\rangle)\| \\
&\leq r \cdot \sqrt{\Theta\Big(\frac{(s + \mathbb{E}_i\, \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)}\Big)}.
\end{aligned}$$

Squaring gives the desired result:

$$\|U_b Q_0(|0\rangle_{\mathsf{T}} |\varphi\rangle) - Q_n U_b(|0\rangle_{\mathsf{T}} |\varphi\rangle)\|^2 \leq \Theta(\frac{r^2 \cdot (s + \mathbb{E}_i\, \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)}).$$ ◀

9 Measurement in the Computational or Hadamard Basis

To measure observables of the form X_f or Z_f on the physical qubits, the procedures are very similar. We continue with the Z_f case as an example. After completing the Global Measurement Extraction process described in the previous section, the verifier has all the necessary information to evaluate any measurement in the Z basis. To measure Z_f, the verifier sends the description of f to the prover. The prover then uses the stored measurement outcomes from each portion and provides a PCPP proof demonstrating that the ECC-decoded outcomes, when evaluated by the function f, yield the claimed value.

9.1 Actions of the honest prover

We introduce a basic quantum operator that computes a function on the string stored in a quantum register.

▶ **Definition 9.1** (Function evaluation operator). *Let $M \in \mathbb{Z}_+$ and $f : \{0,1\}^M \to \{0,1\}$. For $a \in \mathbb{F}_2$ and $w \in \mathbb{F}_2^M$, the evaluation operator R_f acts as:*

$$R_f(|a\rangle |w\rangle) = |a \oplus f(w)\rangle |w\rangle ,$$

Before proceeding, we introduce a notation for a blockwise extension of the classical code C_C. This construction will be useful when describing the encoding structure of the prover's registers and the corresponding decoding procedures.

▶ **Definition 9.2.** *Using the notation from Notation 3.1, define code $C_{C,r} = \bigoplus_{i\in[r]} C_C$.*

We now describe how the honest prover performs measurements in either the computational basis (Z basis) or the Hadamard basis (X basis), depending on the verifier's instruction. This procedure builds on the previously defined global measurement extraction process.

▶ **Definition 9.3** (Measurement in the computational or Hadamard basis, honest prover). *Recall the notations introduced in Section 3.*

The honest prover performs the following steps:

- *The verifier sends the prover the description of a classical function $h : \{0,1\}^M \to \{0,1\}$.*
- *The prover applies the unitary $R_{h,\mathsf{IX}'}$.*
- *The prover runs the procedure in Definition 7.2 with $C = C_{C,r}, f = h, \mathsf{U} = \mathsf{I}, \mathsf{E} = \mathsf{E}', \mathsf{B} = \mathsf{A}$, where $C_{C,r}$ is defined as in Definition 9.2.*
- *The prover sends the verifier the quantum register I.*

▶ Remark 9.4. The function h will be specified as $f(KE_W \mathtt{D}_W(\cdot))$ in a later section, but we present a general protocol here for clarity.

9.2 Actions of the verifier

In the following, we describe the verifier's procedure for performing a global measurement in either the computational (Z) basis or the Hadamard (X) basis. Since the measurement outcomes have already been extracted, the protocol remains the same for both bases. However, the evaluation function h differs between the two cases, even for the same function f. This process can be viewed as a verification of function evaluation using a PCPP.

▶ **Definition 9.5** (Measurement in the computational or Hadamard basis, verifier). *Recall the notations introduced in Section 3 and Definition 9.3. Let $s \in \mathbb{F}_2$.*

The verifier performs the following steps:

- *The verifier sends the prover the description of a classical function* $h : \{0,1\}^M \to \{0,1\}$.
- *The prover sends back to the verifier the quantum registers* IA.
- *If* $s = 1$*, the verifier runs the procedure in Definition 7.4 with* $C = C_{C,n}, f = h, \mathsf{U} = \mathsf{I}, \mathsf{E} = \mathsf{E}', \mathsf{B} = \mathsf{A}$*, where* $C_{C,n}$ *is defined as in Definition 9.2. The verifier accepts if and only if this procedure accepts.*

9.3 Main results

▶ **Lemma 9.6** (Implication of the PCPP check in the measurement procedure)**.** *Recall the notations introduced in Section 3 and Definition 9.3. Let* $\mathsf{O} = (\mathsf{P}, \mathsf{W}, \mathsf{B}, \mathsf{X})$ *be a quantum register. Let* $|\varphi\rangle_{\mathsf{OIEA}}$ *be the quantum state of the system after the measurement in the computational or Hadamard basis process. Let the state be written as*

$$|\varphi\rangle = \sum_{o,i,e,a} \alpha_{o,i,e,a} |o\rangle_{\mathsf{O}} |i\rangle_{\mathsf{I}} |e\rangle_{\mathsf{E}} |a\rangle_{\mathsf{A}} .$$

Assume that $r\delta < \eta_C \Delta$*. Then the rejection probability of the verifier is lower bounded by*

$$\Pr[\textit{verifier rejects}] \geq -s + \sum |\alpha_{o,i,e,a}|^2 \cdot 1\Big(h(\mathtt{D}(e_1), \ldots, \mathtt{D}(e_n)) \neq i\Big).$$

Proof. Let $C_{C,n}$ be defined as in Definition 9.2. Let $\mathtt{D}'_C$ denote the decoder of $C_{C,n}$, $\mathtt{T}'_C$ the code tester, Δ'_C its relative distance, and $\eta'_C \Delta'_C$ the weight of errors that $\mathtt{D}'_C$ can correct. Note that $\mathtt{D}'_C$ can be implemented by applying $\mathtt{D}$ independently to each block of the code, $\mathtt{T}'_C$ can be implemented by applying $\mathtt{T}$ independently to each block of the code. In this case, $\Delta'_C = \Delta/n$ and $\eta'_C = \eta_C$.

Note that the assumption $n\delta < \eta_C \Delta$ implies that $\delta < \eta'_C \Delta'_C$.

The proof follows from Lemma 7.5 with $\eta_P = \eta'_C, \Delta_P = \Delta'_C, \mathtt{D}_P = \mathtt{D}'_C, \mathtt{T}_P = \mathtt{T}'_C$ and $\delta < \eta'_C \Delta'_C$. ◀

▶ **Lemma 9.7** (Implication of the measurement in the computational or Hadamard basis)**.** *Recall the notations introduced in Section 3 and Definition 9.3. Let* $\mathsf{O} = (\mathsf{P}, \mathsf{W}, \mathsf{B}, \mathsf{X})$ *be a quantum register. Let* $|\varphi\rangle_{\mathsf{OIEA}}$ *be the quantum state of the system after the measurement in the computational or Hadamard basis process. Let the state be written as*

$$|\varphi\rangle = \sum_{o,i,e,a} \alpha_{o,i,e,a} |o\rangle_{\mathsf{O}} |i\rangle_{\mathsf{I}} |e\rangle_{\mathsf{E}} |a\rangle_{\mathsf{A}} .$$

If $h(x) = f(KE_W \mathcal{D}_W(x))$*, then:*

$$\| Z_{\mathsf{I}} |\varphi\rangle - Q_r |\varphi\rangle \|^2 \leq \Theta(1) \cdot \sum |\alpha_{o,i,e,a}|^2 \cdot 1\Big(h(\mathcal{D}(e_1), \ldots, \mathcal{D}(e_r)) \neq i\Big).$$

Proof. Let

$$|\beta\rangle = \sum_{o,i,e,a} 1\Big(h(\mathtt{D}(e_1), \ldots, \mathtt{D}(e_r)) = i\Big) \cdot \alpha_{o,i,e,a} |o\rangle_{\mathsf{O}} |i\rangle_{\mathsf{I}} |e\rangle_{\mathsf{E}} |a\rangle_{\mathsf{A}} .$$

Note that

$$Z_{\mathsf{I}} |\beta\rangle = Q_r |\beta\rangle .$$

Therefore,

$$\begin{aligned} &\| Z_{\mathsf{I}} |\varphi\rangle - Q_r |\varphi\rangle \|^2 \\ &\leq \Theta(1) \cdot (\| |\varphi\rangle - |\beta\rangle \|^2 + \| Z_{\mathsf{I}} |\beta\rangle - Q_r |\beta\rangle \|^2) \\ &\leq \Theta(1) \cdot \sum |\alpha_{o,i,e,a}|^2 \cdot 1\Big(h(\mathtt{D}(e_1), \ldots, \mathtt{D}(e_r)) \neq i\Big). \end{aligned}$$

◀

10 QIPCP protocol for QMA

In this section, we describe our protocol for estimating the energy of a dual-basis projector Hamiltonian, and thereby distinguishing between the yes and no instances of the corresponding QMA-complete problem. This will establish our main result, restated as Theorem 10.3 below.

To estimate the energy, we sample a term according to its coefficient in the Hamiltonian (each term being a projector) and then project the witness state according to that projector by implementing the corresponding measurement. The measurement procedures required for this have already been developed in Sections 8 and 9. Here we combine the results from these two sections and apply them to the specific case where the function f arises from a projector in the support of a dual-basis projector Hamiltonian (Definition 5.2).

10.1 Setup

We summarize the main notation, parameters and assumptions that are made throughout this section.

Firstly, $H = \mathbb{E}_i H_i$ s a dual-basis projector Hamiltonian, as introduced in Definition 5.2. According to Theorem 5.4, it is QMA-hard to distinguish between the case where H has smallest eigenvalue exponentially close to 0, or bounded away from 0 by a constant. We let B_q be the number of qubits on which H acts.

Secondly, two codes are chosen. A quantum LTC code C that encodes B_q qubits into M qubits, has distance d_q and locality q. We will use that C_q admits an efficient decoder $\mathtt{D}_q$ for errors of weight up to $\eta_q d_q$, where d_q is the quantum distance of the code. These parameters satisfy the conditions of Theorem 2.21 and Fact 2.22.

The M qubits are divided into r groups of m qubits, where r is chosen sufficiently large that the condition (10) on m holds; this can be satisfied with $r = \operatorname{poly}\log(B_q)$.

A classical code C_C encodes m bits into $n(m)$ bits. We let Δ be the distance of C_C and $\eta_C\Delta$ the (relative) decoding radius of a given efficient decoder. We can take $\Delta = \Theta(n(m))$ and $\eta_C = \Theta(1)$ by Theorem 2.1.

Further notation and parameters associated with the codes is summarized in Section 3.2. The quantum LTC code is used to encode the witness; while the classical code is used to encode measurement outcomes obtained from each of the r blocks of m qubits.

Thirdly, we fix a PCPP verifier and corresponding proof generating algorithm for the CktVal problem. The PCPP has proximity parameter δ and soundness error s. (Further notation associated with the PCPP is recalled in Section 3.3.)

The main assumption relating these parameters are the following.

- The PCPP proximity parameter δ is chosen in such a way that

$$r\delta < \eta_C\Delta\,. \tag{9}$$

  Intuitively, this relation is needed for the PCPP verifier to interact well with the classical code; it first arises from the use of Lemma 7.5. Given that Δ and η_C are both $\Theta(1)$ and $r = M/m = \Theta(\operatorname{poly}\log B_q)$, this assumption can be satisfied by choosing a sufficiently small δ such that $\delta^{-1} = \Theta(\operatorname{poly}\log B_q)$.
- The number m of qubits of the qLTC-encoded witness exchanged in each round is chosen such that

$$\eta_q d_q \geq 2m\,. \tag{10}$$

This can be achieved by choosing r to be sufficiently large, $r = \text{polylog}(B_q)$ will suffice given the quantum LTC code parameters (which are all a multiplicative polylog away from optimal). Intuitively this condition ensures that the prover manipulates too few qubits at a time to perform an undetected attack (in the basis currently being verified).

- The soundness error s of the PCPP is a small enough constant. This is because this soundness error appears as an additive term in the final soundness error.

10.2 Actions of the honest prover

We define the behavior of an honest prover during the energy test, which consists of two phases: the global measurement extraction phase and the measurement phase.

▶ **Definition 10.1** (Energy test, honest prover). *The honest prover performs the following steps:*

- *The verifier sends the basis $W \in \{X, Z\}$ to the prover.*
- *The honest prover performs the actions described in Definition 8.5.*
- *The honest prover performs the actions described in Definition 9.3.*

10.3 Actions of the verifier

We define the verifier's strategy in the energy test, which is designed to verify that the witness state has low energy. In what follows, the case $s_c = 3$ corresponds to the energy test, whose success probability is proportional to the energy of the Hamiltonian. The case $s_c = 1$ is intended to enforce honest behavior from the prover during the global measurement extraction phase, while the case $s_c = 2$ ensures honest behavior during the measurement phase.

▶ **Definition 10.2** (Energy test, verifier). *Recall the notations introduced in Section 3. Let $H = \mathbb{E}_i H_i$ be as in Definition 5.2, where $H_i = I + W_{i,f_i}/2$, W_i is either X or Z, $f_i : \mathbb{F}_2^{B_q} \to \mathbb{F}_2$ is a boolean function.*

The verifier performs the following steps:

- *The verifier samples the phase of the protocol to test by choosing $s_c \in_R [3]$, and selects a round to test in the global measurement extraction phase by choosing $s'_g \in_R [n]$ with probability $1/2$, and setting $s'_g = 0$ with probability $1/2$. If the verifier tests the global measurement extraction phase ($s_c = 1$), it sets $s_g = e_{s'_g}$, where $e_{s'_g} \in \{0,1\}^{n+1}$ is the bit string with all entries 0 except the s'_g-th bit, which is 1. If the verifier does not test during the global measurement extraction phase ($s_c \neq 1$), it sets $s_g = 0^{n+1}$.*
- *The verifier samples a term H_i by selecting the index i uniformly at random. The verifier sends the basis W_i to the prover.*
- *The verifier performs the actions described in Definition 8.8 with $W = W_i$, $s = s_g$.*
- *If $s_c = 1$, the verifier accepts if and only if the preceding procedure accepts.*
- *If $s_c = 2$, the verifier performs the actions described in Definition 9.5 with $h(x) = f_i(KE_W D_W(x))$ and $s = 1$. The verifier accepts if and only if the procedure accept.*
- *If $s_c = 3$, the verifier performs the actions described in Definition 9.5 with $h(x) = f_i(KE_W D_W(x))$ and $s = 0$. The verifier measures the register I and accept if and only if the measurement result is 0.*

10.4 Main results

▶ **Theorem 10.3.** $\mathrm{QMA} \subseteq \mathrm{QIPCP}(O(\mathrm{poly}), O(\mathrm{poly}\log), O(\mathrm{poly}\log))$.

Proof. Let $\mathtt{V}$ be the verifier in Definition 10.2. Assume that the different codes used for the protocol have been chosen in such a way that $r\delta < \eta_C \Delta$, where $r = M/m$ is the number of rounds, δ the proximity error of the PCPP and $\eta_C \Delta$ the (relative) decoding radius of the classical code C_C, as in Notation 3.1. This can be achieved by choosing good enough PCPP and classical code.

We will show that $\mathtt{V}$ is a poly log-message quantum interactive probabilistically checkable proof system for $\mathcal{DLH}(\mathrm{poly}, 2^{-p(n)}, c_{\mathrm{LH}})$ (see Definition 5.3), where $c_{\mathrm{LH}} > 0$ is the constant from Corollary 5.7. It follows that

$$\mathcal{DLH}(\mathrm{poly}, 2^{-p(n)}, c_{\mathrm{LH}}) \in \mathrm{QIPCP}(O(\mathrm{poly}), O(\mathrm{poly}\log), O(\mathrm{poly}\log)) \ .$$

By Corollary 5.7, $\mathcal{DLH}(\mathrm{poly}, 2^{-p(n)}, c_{\mathrm{LH}})$ is QMA-complete. Therefore,

$$\mathrm{QMA} \subseteq \mathrm{QIPCP}(O(\mathrm{poly}), O(\mathrm{poly}\log), O(\mathrm{poly}\log)) \ .$$

We now show the required properties of $\mathtt{V}$. Let n be the instance size, i.e., the description length of a $\mathcal{DLH}$ instance H. In particular, the number of qubits B_q on which H acts satisfies $B_q \leq n$. For the construction of the QIPCP, we use parameters as introduced in Section 10.1.

First, by inspecting Definitions 10.2 and 4.1, it is clear that $\mathtt{V}$ has polynomial-time complexity.

Rounds. By Definitions 10.2 and 4.1, the verifier and the prover exchange a total of $2M/m + 2$ messages.

Recall that M is the length of the qLTC codeword, and m is chosen to be smaller than $\eta_q d_q$ by at least a constant factor. By Theorem 2.21, $d_q = M/\mathrm{poly}\log(M)$, and by Fact 2.22, $\eta_q = 1/\mathrm{poly}\log(M)$. Moreover, the qLTC code has a linear code length $M = O(n)$. Therefore,

$$2M/m + 2 = \mathrm{poly}\log(n) \ .$$

Communication complexity. We compute the communication complexity according to Definitions 4.1 and 10.2. In each of the first M/m rounds, the verifier sends a quantum message of length m (plus 1 classical bit for the basis choice). The prover's message consists of two parts:

- an $n(m)$-bit classical ECC encoding of (measurements from) the verifier's message of length m;
- a PCPP proof of length $l_w(\mathrm{poly}(n(m)))$ for each bit measured from the verifier's message, where the $\mathrm{poly}(n(m))$ term arises from the size of the classical ECC decoding circuit.

The communication in the last round is $l_w(\mathrm{poly}(M/m \cdot n(m)))$. Here, the outer l_w comes from the PCPP proof, while $M/m \cdot n(m)$ corresponds to the size of the implicit input (i.e., the input to the circuit being verified). This circuit evaluates the logical Boolean function corresponding to the sampled projector after decoding the classical ECC code. Since decoding the classical ECC, transforming the Boolean function into its logical form, and evaluating the Boolean function itself are all polynomial-time operations, the PCPP also has an explicit input (the circuit being verified) of size $\mathrm{poly}(M/m \cdot n(m))$. Therefore, the total communication complexity is $\mathrm{poly}(n)$.

Query complexity. By Definitions 4.1 and 10.2, the query complexity in each of the first M/m rounds arises from the local tester for the W-code (in the qLTC) and the PCPP used to verify the queries of the local tester.

We first analyze the query complexity of a single invocation of the local tester for the W-code. Let $\varepsilon = \Theta(1)$ be a small constant that will be specified later, and let $s = \Theta(\varepsilon)$ be the soundness error of the PCPP. Note that the soundness parameter of the LTC is $\rho = \Theta(1/\operatorname{poly}\log(n))$. Define

$$\kappa = \Theta\left(\frac{r^2(s+\varepsilon)M}{\eta_q d_{q,W} - m}\right) = \operatorname{poly}\log(n)\ , \tag{11}$$

since $\eta_q d_{q,W} \geq \eta_q d_q \geq 2m$ by assumption (10) and $M/m = r = \operatorname{poly}\log(n)$. By setting $R = \Theta(1)$ and $\varepsilon = s$ in Lemma 2.6, we obtain the query complexity of the local tester for the W codes of the qLTC as

$$Q = \Theta(1) \cdot \frac{\kappa}{\rho} \cdot \frac{\log(1/s)}{1-s} = \operatorname{poly}\log(n)\ .$$

Next we analyze the query complexity of each invocation of the PCPP verifier. By applying Lemma 2.27, we obtain that the query complexity of the PCPP verifier is

$$O\left(\frac{\log(s)}{\log(1-\Omega(\delta))}\right) = \operatorname{poly}\log(n)\ .$$

Therefore, in each of the first $r = M/m$ rounds, the total query complexity is upper bounded by the product of the query complexity of the local tester and that of the PCPP verifier:

$$\operatorname{poly}\log(n) \cdot \operatorname{poly}\log(n)\ .$$

The query complexity in the final round is due to the PCPP verifier and is thus at most

$$O\left(\frac{\log(s)}{\log(1-\Omega(\delta))}\right) = \operatorname{poly}\log(n)\ ,$$

which again follows from the application of Lemma 2.27. Thus, the overall query complexity of the verifier is $\operatorname{poly}\log(n)$.

Completeness. By Definition 10.1, note that the honest prover uses a qLTC encoding with no error, so the local tester always accepts. Moreover, the PCPP statements in both the global measurement extraction and the measurement phases verify valid computations; hence, the corresponding instances are always yes-instances. By the completeness of the PCPP verifier, the PCPP verifier during these phases succeeds with probability 1. Therefore, all tests except the energy test pass with probability 1.

Let $|\psi\rangle_{\mathsf{OIEA}}$ denote the quantum state of the system after the measurement in the computational or Hadamard basis, as defined in Lemma 9.7, and let $|\varphi\rangle_{\mathsf{WBEXP}}$ be the quantum state of the system before the global measurement extraction, as defined in Lemma 8.11.

If $s_c = 3$, then by Definition 10.2, the energy test passes if and only if the measurement result of the register I is 0 with the corresponding probability:

$$\begin{aligned}
\mathop{\mathbb{E}}_i \left\| \frac{I + Z_{\mathsf{I}}}{2} |\psi\rangle \right\|^2 &= \mathop{\mathbb{E}}_i \left\| \frac{I + Q_r}{2} |\psi\rangle \right\|^2 \\
&= \mathop{\mathbb{E}}_i \left\| \frac{I + Q_0}{2} |\varphi\rangle \right\|^2 \\
&= \mathop{\mathbb{E}}_i \left\| (I - W_{i,f_i}) |\varphi\rangle \right\|^2 \\
&= \mathop{\mathbb{E}}_i \left\| (I - H_i) |\varphi\rangle \right\|^2 \\
&= \langle\varphi| (I - H) |\varphi\rangle\ ,
\end{aligned}$$

where the first step follows from Lemma 9.6, Lemma 9.7, and the fact that the verifier never rejects in the measurement phase; the second step follows from Lemma 8.11 and the verifier never rejecting in the global measurement extraction phase; and the third step follows from Lemma 6.8 and the fact that the quantum witness is a qLTC encoding. As a result,

$$\begin{aligned}\Pr[\mathtt{V}\text{ accepts}] =\ & \Pr[\mathtt{V}\text{ accepts} \mid s_c = 1]\cdot\Pr[s_c=1] + \Pr[\mathtt{V}\text{ accepts} \mid s_c = 2]\cdot\Pr[s_c=2] \\ & + \Pr[\mathtt{V}\text{ accepts} \mid s_c = 3]\cdot\Pr[s_c=3] \\ =\ & \frac{1}{3}+\frac{1}{3}+\frac{1}{3}\langle\varphi|(I-H)|\varphi\rangle \\ \geq\ & 1-\Theta(2^{-p(n)})\,.\end{aligned}$$

Soundness. Let $|\psi\rangle_{\mathsf{OIEA}}$ denote the quantum state of the system after the measurement in the computational or Hadamard basis, as defined in Lemma 9.7. This corresponds to the state of the quantum system after the verifier performs the procedure described in Definition 9.5 within Definition 10.2, in both cases $s_c = 2$ and $s_c = 3$. Let $|\varphi\rangle_{\mathsf{WBEXP}}$ denote the quantum state of the system before the global measurement extraction procedure, as defined in Lemma 8.11, i.e., the state of the quantum system before the verifier executes the procedure from Definition 8.8 in Definition 10.2. Then,

$$\Pr[\text{energy test accepts} \mid s_c = 3] = \mathop{\mathbb{E}}_i \left\| \frac{I+Z_{\mathsf{I}}}{2}|\psi\rangle \right\|^2 .$$

Note that for any vectors a, b and $k > 0$,

$$\|a+b\|^2 \leq \frac{1+k^2}{k^2}\|a\|^2 + (1+k^2)\|b\|^2\,.$$

Using this with $k = 10$,

$$\left\|\frac{I+Z_{\mathsf{I}}}{2}|\psi\rangle\right\|^2 \leq 1.01\left\|\frac{I+Q_r}{2}|\psi\rangle\right\|^2 + 101\left\|\frac{Z_{\mathsf{I}}|\psi\rangle - Q_r|\psi\rangle}{2}\right\|^2 .$$

Using the same reasoning,

$$\left\|\frac{I+Q_r}{2}|\psi\rangle\right\|^2 \leq 1.01\left\|U_b\frac{I+Q_0}{2}(|0\rangle_{\mathsf{T}}|\varphi\rangle)\right\|^2 + 101\left\|\frac{(U_bQ_0 - Q_rU_b)(|0\rangle_{\mathsf{T}}|\varphi\rangle)}{2}\right\|^2 .$$

Finally,

$$\begin{aligned}\left\|\frac{I+Q_0}{2}(|0\rangle_{\mathsf{T}}|\varphi\rangle)\right\|^2 \leq\ & 1.01\|(I-W_{f_i})(|0\rangle_{\mathsf{T}}|\varphi\rangle)\|^2 \\ & + 101\left\|\frac{(Q_0-(I-2W_{f_i}))(|0\rangle_{\mathsf{T}}|\varphi\rangle)}{2}\right\|^2 .\end{aligned}$$

Summarizing the preceding equations,

$$\begin{aligned}&\Pr[\text{energy test accepts} \mid s_c = 3] \\ &\quad\leq 101\mathop{\mathbb{E}}_i\left\|\frac{Z_{\mathsf{I}}|\psi\rangle - Q_r|\psi\rangle}{2}\right\|^2 \\ &\qquad + 1.01\cdot 101\left\|\frac{(U_bQ_0-Q_rU_b)(|0\rangle_{\mathsf{T}}|\varphi\rangle)}{2}\right\|^2 \\ &\qquad + 1.01^2\Big(1.01\|(I-W_{f_i})(|0\rangle_{\mathsf{T}}|\varphi\rangle)\|^2 \\ &\qquad\quad + 101\left\|\frac{(Q_0-(I-2W_{f_i}))(|0\rangle_{\mathsf{T}}|\varphi\rangle)}{2}\right\|^2\Big)\,. \qquad (12)\end{aligned}$$

Assume that

$$\begin{aligned}\Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g = 0] \geq 1 - \varepsilon \ , \\ \Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g \neq 0] \geq 1 - \varepsilon \ , \\ \Pr[\text{measurement phase accepts} \mid s_c = 2] \geq 1 - \varepsilon \ ,\end{aligned} \tag{13}$$

where ε is the previously introduced small constant that will be specified later. By Lemma 9.6 and Lemma 9.7,

$$\| Z_{\mathsf{I}} |\psi\rangle - Q_r |\psi\rangle \|^2 \leq \Theta(s + \varepsilon) \ . \tag{14}$$

Moreover, suppose $\Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g = i] = 1 - \varepsilon_i$ for some ε_i, then by Lemma 8.11,

$$\| U_b Q_0 (|0\rangle_{\mathsf{T}} |\varphi\rangle) - Q_r U_b (|0\rangle_{\mathsf{T}} |\varphi\rangle) \|^2 \leq \Theta\Big(\frac{r^2 \cdot (s + \mathbb{E}_i \, \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)} \Big) \ . \tag{15}$$

Let

$$|\varphi\rangle = \sum_{(a,b) \in \mathcal{S}} \alpha_{a,b} X(a) Z(b) |\eta_{a,b}\rangle \ .$$

By Lemma 6.8,

$$\begin{aligned}\| Q_0 (|0\rangle_{\mathsf{T}} |\varphi\rangle) - (I - 2W_{f_i})(|0\rangle_{\mathsf{T}} |\varphi\rangle) \|^2 \leq \Theta(1) \cdot \sum_{(a,b) \in \mathcal{S}} 1(wt_{\text{cent}}(X(a)Z(b)) \geq \eta_q d_q) \\ \cdot |\alpha_{a,b}|^2 \ .\end{aligned}$$

Let c be the number of repetitions of the local testing verifier from Definition 6.1, which will be chosen below in (18). By Lemma 6.2,

$$\begin{aligned}1 - \varepsilon &\leq \sum_{(a,b) \in \mathcal{S}} |\alpha_{a,b}|^2 \cdot \Big(1 - \frac{\rho}{2M} \cdot wt_{\text{cent}}(X(a)Z(b)) \Big)^c \\ &\leq \sum_{(a,b) \in \mathcal{S}} 1(wt_{\text{cent}}(X(a)Z(b)) \geq \eta_q d_q) \cdot |\alpha_{a,b}|^2 \cdot \Big(1 - \frac{\rho}{2M} \cdot \eta_q d_q \Big)^c \\ &\quad + \sum_{(a,b) \in \mathcal{S}} 1(wt_{\text{cent}}(X(a)Z(b)) < \eta_q d_q) \cdot |\alpha_{a,b}|^2 \ .\end{aligned}$$

Therefore,

$$\| Q_0 (|0\rangle_{\mathsf{T}} |\varphi\rangle) - (I - 2W_{f_i})(|0\rangle_{\mathsf{T}} |\varphi\rangle) \|^2 \leq \Theta\Big(\varepsilon \Big(1 - \frac{\rho}{2M} \cdot \eta_q d_q \Big)^c \Big)^{-1} \Big) \ . \tag{16}$$

Plugging the bounds (14), (15) and (16) in (12) and using the assumptions (13) we have obtained that

$$\begin{aligned}\Pr[\text{energy test accepts}] \leq 1.1 \cdot \langle \varphi | \, (I - H) \, |\varphi\rangle + \Theta(s + \varepsilon) + \Theta\Big(\frac{r^2 \cdot (s + \mathbb{E}_i \, \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)} \Big) \\ + \Theta\Big(\varepsilon \Big(1 - \Big(1 - \frac{\rho}{2M} \cdot \eta_q d_q \Big)^c \Big)^{-1} \Big) \ .\end{aligned} \tag{17}$$

Note that

$$\Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g \neq 0] = 1 - \mathop{\mathbb{E}}_i \varepsilon_i \ .$$

Choose

$$c := \frac{\Theta(1)}{-\log(1-\frac{\rho}{2M}\cdot\eta_q d_q)} = \operatorname{poly}\log(n)\ . \tag{18}$$

Plugging back into (17) and using also the choice of κ in (11),

$$\begin{aligned}\Pr[\text{energy test accepts}] &\leq 1.1\cdot\langle\varphi|\,(I-H)\,|\varphi\rangle + \Theta(\varepsilon)\\ &\leq 1.1 - 1.1c_{\mathrm{LH}} + \Theta(\varepsilon)\ .\end{aligned}$$

If any of the following fails:

$$\begin{aligned}\Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g = 0] &\geq 1-\varepsilon\ ,\\ \Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g \neq 0] &\geq 1-\varepsilon\ ,\\ \Pr[\text{measurement phase accepts} \mid s_c = 2] &\geq 1-\varepsilon\ ,\end{aligned}$$

then

$$\Pr[\mathtt{V}\text{ accepts}] \leq 1-\Theta(\varepsilon)\ .$$

Note that we can take c_{LH} in Corollary 5.7 to be sufficiently large, and in particular strictly larger than 0.1. As a result, in all cases, we get that $\Pr[\mathtt{V}\text{ accepts}]$ is smaller than some constant less than 1, as desired.

By applying sequential repetition a constant number of times (since $c_{\mathrm{LH}} = \Theta(1), \varepsilon = \Theta(1)$), we obtain a protocol in which the verifier accepts yes-instances with probability greater than $2/3$ and accepts no-instances with probability less than $1/3$. ◀

References

1. Dorit Aharonov, Itai Arad, Zeph Landau, and Umesh Vazirani. The detectability lemma and quantum gap amplification. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 417–426, 2009. doi:10.1145/1536414.1536472.
2. Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations. *arXiv preprint*, 2017. arXiv:1704.04487.
3. Dorit Aharonov and Lior Eldar. Quantum locally testable codes. *SIAM J. Comput.*, 44(5):1230–1262, 2015. doi:10.1137/140975498.
4. James Bartusek, Yael Tauman Kalai, Alex Lombardi, Fermi Ma, Giulio Malavolta, Vinod Vaikuntanathan, Thomas Vidick, and Lisa Yang. Succinct classical verification of quantum computation. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 195–211. Springer, 2022. doi:10.1007/978-3-031-15979-4_7.
5. Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 1–10. ACM, 2004. doi:10.1145/1007352.1007361.
6. Thiago Bergamaschi, Tony Metger, Thomas Vidick, and Tina Zhang. Derandomised tensor product gap amplification for quantum Hamiltonians. *arXiv preprint arXiv:2510.01333*, 2025. doi:10.48550/arXiv.2510.01333.
7. Jacob D Biamonte and Peter J Love. Realizable hamiltonians for universal adiabatic quantum computers. *Physical Review A—Atomic, Molecular, and Optical Physics*, 78(1):012352, 2008.

8 Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for QMA. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 31–40. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.13.

9 Harry Buhrman, Jonas Helsen, and Jordi Weggemans. Quantum PCPs: on adaptivity, multiple provers and reductions to local hamiltonians. *Quantum*, 9:1791, 2025. doi:10.22331/Q-2025-07-11-1791.

10 Michael Chapman, Thomas Vidick, and Henry Yuen. Efficiently stable presentations from error-correcting codes. *arXiv preprint*, 2023. arXiv:2311.04681.

11 Lijie Chen and Ramis Movassagh. Quantum Merkle trees. *Quantum*, 8:1380, 2024. doi:10.22331/Q-2024-06-18-1380.

12 Irit Dinur, Min-Hsiu Hsieh, Ting-Chun Lin, and Thomas Vidick. Good quantum LDPC codes with linear time decoders. In *Proceedings of the 55th annual ACM symposium on theory of computing*, pages 905–918, 2023. doi:10.1145/3564246.3585101.

13 Irit Dinur, Ting-Chun Lin, and Thomas Vidick. Expansion of high-dimensional cubical complexes: with application to quantum locally testable codes. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 379–385. IEEE, 2024. doi:10.1109/FOCS61266.2024.00031.

14 Lior Eldar and Aram W. Harrow. Local hamiltonians whose ground states are hard to approximate. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 427–438. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.46.

15 Joseph F Fitzsimons, Michal Hajdušek, and Tomoyuki Morimae. Post hoc verification of quantum computation. *Physical review letters*, 120(4):040501, 2018.

16 Joseph F Fitzsimons and Elham Kashefi. Unconditionally verifiable blind quantum computation. *Physical Review A*, 96(1):012303, 2017.

17 Daniel Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, Thesis, eprint: quant-ph/9705052, 1997.

18 Alex Bredariol Grilo. A simple protocol for verifiable delegation of quantum computation in one round. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019*, volume 132, pages 28–1, 2019.

19 Sam Gunn, Nathan Ju, Fermi Ma, and Mark Zhandry. Commitments to quantum states. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1579–1588, 2023. doi:10.1145/3564246.3585198.

20 Sam Gunn, Yael Tauman Kalai, Anand Natarajan, and Agi Villanyi. Classical commitments to quantum states. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 234–244, 2025.

21 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP*=RE. *Communications of the ACM*, 64(11):131–138, 2021.

22 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Quantum soundness of testing tensor codes. *Discrete Analysis*, December 6 2022. doi:10.19086/da.55554.

23 Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47 in Graduate Studies in Mathematics. American Mathematical Soc., 2002.

24 Henry Ma and Anand Natarajan. Two bases suffice for QMA_1-completeness. *arXiv preprint arXiv:2509.24390*, 2025. doi:10.48550/arXiv.2509.24390.

25 Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018. doi:10.1109/FOCS.2018.00033.

26 Chris Marriott and John Watrous. Quantum Arthur–Merlin games. *computational complexity*, 14(2):122–152, 2005. doi:10.1007/S00037-005-0194-X.

27 Tony Metger, Anand Natarajan, and Tina Zhang. Succinct arguments for QMA from standard assumptions via compiled nonlocal games. *CoRR*, abs/2404.19754, 2024. doi:10.48550/arXiv.2404.19754.

28 Anand Natarajan and Chinmay Nirkhe. The status of the quantum PCP conjecture (games version). *CoRR*, abs/2403.13084, 2024. doi:10.48550/arXiv.2403.13084.

29 Anand Natarajan and Thomas Vidick. Robust self-testing of many-qubit states. *CoRR*, abs/1610.03574, 2016. arXiv:1610.03574.

30 Anand Natarajan and John Wright. NEEXP is contained in MIP*. In *2019 ieee 60th annual symposium on foundations of computer science (focs)*, pages 510–518. IEEE, 2019.

31 Anand Natarajan and Tina Zhang. Bounding the quantum value of compiled nonlocal games: From CHSH to BQP verification. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 1342–1348. IEEE, 2023. doi:10.1109/FOCS57990.2023.00081.

32 Quynh T Nguyen and Christopher A Pattison. Quantum fault tolerance with constant-space and logarithmic-time overheads. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 730–737, 2025. doi:10.1145/3717823.3718318.

33 Baocheng Sun and Thomas Vidick. Quantum interactive oracle proofs. In *Theory of Cryptography Conference (TCC)*. Springer-Verlag, 2025.

34 Adam Wills, Ting-Chun Lin, and Min-Hsiu Hsieh. Tradeoff constructions for quantum locally testable codes. *IEEE Transactions on Information Theory*, 2024.

35 Jiayu Zhang. Succinct blind quantum computation using a random oracle. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1370–1383, 2021. doi:10.1145/3406325.3451082.