Constant-round Blind Classical Verification of Quantum Sampling

Kai-Min Chung[1][0000−0002−3356−369X], Yi Lee[2][0000−0003−3742−3296], Han-Hsuan Lin[3][0000−0002−5126−0174], and Xiaodi Wu[2,4][0000−0001−8877−9802]

[1] Institute of Information Science, Academia Sinica, Taiwan
kmchung@iis.sinica.edu.tw
[2] Department of Computer Science, University of Maryland, USA
ylee1228@umd.edu, xwu@cs.umd.edu
[3] Department of Computer Science, National Tsing Hua University, Taiwan
linhh@cs.nthu.edu.tw
[4] Joint Center for Quantum Information and Computer Science, University of Maryland, USA

Abstract. In a recent breakthrough, Mahadev constructed a classical verification of quantum computation (CVQC) protocol for a classical client to delegate decision problems in BQP to an untrusted quantum prover under computational assumptions. In this work, we explore further the feasibility of CVQC with the more general *sampling* problems in BQP and with the desirable *blindness* property. We contribute affirmative solutions to both as follows.

- Motivated by the sampling nature of many quantum applications (e.g., quantum algorithms for machine learning and quantum supremacy tasks), we initiate the study of CVQC for *quantum sampling problems* (denoted by SampBQP). More precisely, in a CVQC protocol for a SampBQP problem, the prover and the verifier are given an input $x \in \{0,1\}^n$ and a quantum circuit C, and the goal of the classical client is to learn a sample from the output $z \leftarrow C(x)$ up to a small error, from its interaction with an untrusted prover. We demonstrate its feasibility by constructing a four-message CVQC protocol for SampBQP based on the quantum *Learning With Errors* assumption.
- The *blindness* of CVQC protocols refers to a property of the protocol where the prover learns nothing, and hence is blind, about the client's input. It is a highly desirable property that has been intensively studied for the delegation of quantum computation. We provide a simple yet powerful *generic* compiler that transforms any CVQC protocol to a blind one while preserving its completeness and soundness errors as well as the number of rounds.

Applying our compiler to (a parallel repetition of) Mahadev's CVQC protocol for BQP and our CVQC protocol for SampBQP yields the first *constant-round* blind CVQC protocol for BQP and SampBQP respectively, with negligible and inverse polynomial soundness errors respectively, and negligible completeness errors.

Keywords: classical delegation of quantum computation· blind quantum computation· quantum sampling problems

1 Introduction

Can quantum computation, with potential computational advantages that are intractable for classical computers, be efficiently verified by classical means? This problem has been a major open problems in quantum complexity theory and delegation of quantum computation [1]. A complexity-theoretic formulation of this problem by Gottesman in 2004 [1] asks about the possibility for an efficient classical verifier (a BPP machine) to verify the output of an efficient quantum prover (a BQP machine). In the absence of techniques for directly tackling this question, earlier feasibility results on this problem have been focusing on two weaker formulations. The first type of feasibility results (e.g., [4, 13, 21, 22]) considers the case where the verifier is equipped with limited quantum power. The second type of feasibility results (e.g, [18,23,25,35]) considers a BPP verifier interacting with at least two entangled, non-communicating quantum provers.

Recently, the problem is resolved by a breakthrough result of Mahadev [30], who constructed the first Classical Verification of Quantum Computation (CVQC) protocol for BQP, where an efficient classical (BPP) verifier can interact with an efficient quantum (BQP) prover to verify any BQP language. Soundness of Mahadev's protocol is based on a widely recognized computational assumption that the learning with errors (LWE) problem [34] is hard for BQP machines. The technique invented therein has inspired many subsequent developments of CVQC protocols with improved parameters and functionality. For example, Mahadev's protocol has a large constant soundness error. The works of [7, 15] use parallel repetition to achieve a negligible soundness error. As another example, the work of [24] extends Mahadev's techniques in an involved way to obtain a CVQC protocol with an additional blindness property.

In this work, we make two more contributions to this exciting line of research. First, we observe that the literature has mostly restricted the attention to delegation of *decision* problems (i.e., BQP). Motivated by the intrinsic randomness of quantum computation and the sampling nature of many quantum applications, we initiate the study of CVQC for quantum *sampling* problems. Second, we further investigate the desirable *blindness* property and construct the first *constant-round* blind CVQC protocols. We elaborate on our contributions in Section 1.1 and 1.2, respectively.

1.1 CVQC for Quantum Sampling Problems

We initiate the study of CVQC for quantum sampling problem, which we believe is highly desirable and natural for delegation of quantum computation. Due to the intrinsic randomness of quantum mechanics, the output from a quantum computation is randomized and described by a distribution. Thus, if a classical verifier want to utilize the full power of a quantum machine, the ability to get a verifiable sample from the quantum circuit's output distribution is desirable. On a more concrete level, quantum algorithms like Shor's algorithm [37] has a significant quantum sampling component, and the recent quantum supremacy

proposals (e.g., [3,8,36]) are built around sampling tasks, suggesting the importance of sampling in quantum computation.

It is worth noting that the difficulty of extending the delegation of decision problem to the delegation of sampling problems is quantum-specific. This is because there is a simple reduction from the delegation of *classical* sampling problems to decision ones: the verifier can sample and fix the random seed of the computation, which makes the computation deterministic. Then, the verifier can delegate the output of the computation bit-by-bit as decision problems. However, this derandomization trick does not work in the quantum setting due to its intrinsic randomness.

Our Contribution. As the first step to formalize CVQC for quantum sampling problems, we consider the complexity class $\mathsf{SampBQP}$ introduced by Aaronson [2] as a natural class to capture efficiently computable quantum sampling problems. $\mathsf{SampBQP}$ consists of sampling problems $(D_x)_{x\in\{0,1\}^*}$ that can be approximately sampled by a BQP machine with a desired inverse polynomial error (See Section 2 for the formal definition). We consider CVQC for a $\mathsf{SampBQP}$ problem $(D_x)_{x\in\{0,1\}^*}$ where a classical BPP verifier delegates the computation of a sample $z \leftarrow D_x$ for some input x to a quantum BQP prover. Completeness requires that when the prover is honest, the verifier should accept with high probability and learn a correct sample $z \leftarrow D_x$. For soundness, intuitively, the verifier should not accept and output a sample with incorrect distribution when interacting with a malicious prover. We formalize the soundness by a strong *simulation-based* definition, (Definition 3), where we require that the joint distribution (d, z) of the decision bit $d \in \{\mathsf{Acc}, \mathsf{Rej}\}$ and the output z (which is $\bot$ when $d = \mathsf{Rej}$) is ϵ-close (in either statistical or computational sense) to an "ideal distribution" (d, z_{ideal}), where z_{ideal} is sampled from the desired distribution D_x when $d = \mathsf{Acc}$ and set to $\bot$ when $d = \mathsf{Rej}$.[5]

As our main result, we construct a constant-round CVQC protocol for $\mathsf{SampBQP}$, based on the quantum LWE (QLWE) assumption that the learning-with-errors problem is hard for BQP machines.

Theorem 1 (informal). *Assuming the QLWE assumption, there exists a four-message CVQC protocol for all sampling problems in* $\mathsf{SampBQP}$ *with computational soundness and negligible completeness error.*

We note that since the definition of $\mathsf{SampBQP}$ allows an inverse polynomial error, our CVQC protocol also implicitly allows an arbitrary small inverse polynomial error in soundness (see Section 2 for the formal definition). Achieving negligible soundness error for delegating sampling problems is an intriguing open question; see Section 1.3 for further discussions.

The construction of our CVQC protocol follows the blueprint of Mahadev's construction [30]. However, there are several obstacles we need to overcome along the way. To explains the obstacles and our ideas, we first present a high-level overview of Mahadev's protocol.

[5] This simulation-based formulation is analogous to the standard composable security definition for QKD.

Overview of Mahadev's Protocol. Following [30], we define QPIP_τ as classes of interactive proof systems between an (almost) classical verifier and a quantum prover, where the classical verifier has limited quantum computational capability, formalized as possessing τ-qubit quantum memory. A formal definition is given in our full version [17].

At a high-level, the heart of Mahadev's protocol is a measurement protocol Π_{Measure} that can compile an one-round QPIP_1 protocol (with special properties) to a QPIP_0 protocol. Note that in a QPIP_1 protocol, the verifier with one-qubit memory can only measure the prover's quantum message qubit by qubit. Informally, the measurement protocol Π_{Measure} allows a BQP prover to "commit to" a quantum state ρ and a classical verifier to choose an X or Z measurement to apply to each qubit of ρ such that the verifier can learn the resulting measurement outcome.

Thus, if an (one-round) QPIP_1 verifier only applies X or Z measurement to the prover's quantum message, we can use the measurement protocol Π_{Measure} to turn the QPIP_1 protocol into a QPIP_0 protocol in a natural way. One additional requirement here is that the verifier's measurement choices need to be determined at the beginning (i.e., cannot depend adaptively on the intermediate measurement outcome).

Furthermore, in Π_{Measure}, the verifier chooses to run a "*testing*" round or a "*Hadamard*" round with $1/2$ probability, respectively. Informally, the testing round is used to "test" the commitment of ρ, and the Hadamard round is used to learn the measurement outcome. (See Protocol 1 for further details about the measurement protocol Π_{Measure}.) Another limitation here is that in the testing round, the verifier only "test" the commitment without learning any measurement result.

In [30], Mahadev's CVQC protocol for BQP is constructed by applying her measurement protocol to the one-round QPIP_1 protocol of [21, 32], which has the desired properties that the verifier only performs non-adaptive X/Z measurement to the prover's quantum message. The fact that the verifier does not learn the measurement outcome in the testing round is not an issue here since the verifier can simply accept when the test is passed (at the cost of a constant soundness error).

Overview of Our Construction. Following the blueprint of Mahadev's construction, our construction proceeds in the following two steps: 1. construct a QPIP_1 protocol for $\mathsf{SampBQP}$ with required special property, and 2. compile the QPIP_1 protocol using Π_{Measure} to get the desired QPIP_0 protocol. The first step can be done by combining existing techniques from different contexts, whereas the second step is the main technical challenge. At a high-level, the reason is the above-mentioned issue that the verifier does not learn the measurement outcome in the testing round. While this is not a problem for decision problems, for sampling problems, the verifier needs to produce an output sample when accepts, but there seems to be no way to produce the output for the verifier without learning the measurement outcome. We discuss both steps in turn as follows.

$\diamond$ *Construct a* QPIP_1 *protocol for* $\mathsf{SampBQP}$ *with required special property*: Interestingly, while the notion of delegation for quantum sampling problem is not explicitly formalized in their work, Hayashi and Morimae [26] constructed an one-round QPIP_1 protocol that can delegate quantum sampling problem and achieve our notion of completeness and soundness[6]. Furthermore, their protocol has information-theoretic security and additionally achieve the blindness property. However, in their protocol, the computation is performed by the verifier using measurement-based quantum computation (MBQC)[7], and hence the verifier needs to perform adaptive measurement choices. Therefore, we cannot rely on their QPIP_1 protocol for $\mathsf{SampBQP}$.

Instead, we construct the desired QPIP_1 protocol for $\mathsf{SampBQP}$ by generalizing the approach of local Hamiltonian reduction used in [21, 32] to verify $\mathsf{SampBQP}$. Doing so requires the combination of several existing techniques from different context with some new ideas. For example, to handle $\mathsf{SampBQP}$, we need to prove lower bound on the spectral gap of the reduced local Hamiltonian instance, which is reminiscent to the simulation of quantum circuits by adiabatic quantum computation [5]. To achieve soundness, we use cut-and-choose and analyze it using de Finetti theorem in a way similar to [26, 38]. See Section **??** for detailed discussions.

$\diamond$ *Compile the* QPIP_1 *protocol using* Π_{Measure}: We now discuss how to use Mahadev's measurement protocol to compile the above QPIP_1 protocol for $\mathsf{SampBQP}$ to a QPIP_0 protocol. As mentioned, a major issue we need to address in Mahadev's original construction is that when the verifier V chooses to run a testing round, V does not learn an output sample when it accepts.

Specifically, let Π_{int} be an "intermediate" QPIP_0 protocol obtained by applying Mahadev's compilation to the above QPIP_1 protocol. In such a protocol, when the verifier V chooses to run the Hadamard round, it could learn a measurement outcome from the measurement protocol and be able to run the QPIP_1 verifier to generate a decision and an output sample when accepts. However, when V chooses to run the testing round, it only decides to accept/reject without being able to output a sample.

A natural idea to fix the issue is to execute multiple copies of Π_{int} in parallel[8], and to choose a random copy to run the Hadamard round to generate an output sample and use all the remaining copies to run the testing round. The verifier accepts only when all executions accept and outputs the sample from the Hadamard round. We call this protocol Π_{Final}.

Clearly from the construction, the verifier now can output a sample when it decides to accept, and output a correct sample when interacting with an honest

[6] They did not prove our notion of soundness for their construction, but it is not hard to prove its soundness based on their analysis.

[7] In more detail, the prover of their protocol is required to send multiple copies of the graph states to the verifier (qubit by qubit). The verifier tests the received supposedly graph states using cut-and-choose and perform the computation using MBQC.

[8] It is also reasonable to consider sequential repetition, but we consider parallel repetition for its advantage of preserving the round complexity.

prover (completeness). The challenge is to show that Π_{Final} is computationally sound. Since we are now in the computational setting, we cannot use the quantum de Finetti theorem as above which only holds in the information-theoretical setting. Furthermore, parallel repetition for computationally sound protocols are typically difficult to analyze, and known to not always work for protocols with four or more messages even in the classical setting [10, 33].

Parallel repetition of Mahadev's protocol for BQP has been analyzed before in [7, 15]. However, the situation here is different. For BQP, the verifier simply chooses to run the Hadamard and testing rounds independently for each repetition. In contrast, our Π_{Final} runs the Hadamard round in one repetition and runs the testing rounds in the rest. The reason is that in $\mathsf{SampBQP}$, as well as generically in sampling problems, there is no known approach to combine multiple samples to generate one sample with reduced error, i.e., there is no generic error reduction method for the sampling problem. In contrast, the error reduction for decision problems can be done with the majority vote. As a result, while the soundness error decreases exponentially for BQP, as we see below (and also in the above QPIP_1 protocols), for $\mathsf{SampBQP}$, m-fold repetition only decreases the error to $\mathrm{poly}(1/m)$.

To analyze the soundness of Π_{Final}, we use the *partition lemma* developed in [15] to analyze the prover's behavior while executing copies of Π_{Measure}.[9] Intuitively, the partition lemma says that for any cheating prover and for each copy $i \in [m]$, there exist two efficient "projectors" [10] $G_{0,i}$ and $G_{1,i}$ in the prover's internal space with $G_{0,i} + G_{1,i} \approx Id$. $G_{0,i}$ and $G_{1,i}$ splits up the prover's residual internal state after sending back his first message. $G_{0,i}$ intuitively represents the subspace where the prover does not knows the answer to the testing round on the i-th copy, while $G_{1,i}$ represents the subspace where the prover does. Note that the prover is using a single internal space for all copies, and every $G_{0,i}$ and every $G_{1,i}$ is acting on this single internal space. By using this partition lemma iteratively, we can decompose the prover's internal state $|\psi\rangle$ into sum of subnormalized states. First we apply it to the first copy, writing $|\psi\rangle = G_{0,1}|\psi\rangle + G_{1,1}|\psi\rangle \equiv |\psi_0\rangle + |\psi_1\rangle$. The component $|\psi_0\rangle$ would then get rejected as long as the first copy is chosen as a testing round, which occurs with pretty high probability. More precisely, the output corresponding to $|\psi_0\rangle$ is $1/m$-close to the ideal distribution that just rejects all the time. On the other hand, $|\psi_1\rangle$ is now binding on the first copy; we now similarly apply the partition lemma of the second copy to $|\psi_1\rangle$. We write $|\psi_1\rangle = G_{0,2}|\psi_1\rangle + G_{1,2}|\psi_1\rangle \equiv |\psi_{10}\rangle + |\psi_{11}\rangle$, and apply the same argument about $|\psi_{10}\rangle$ and $|\psi_{11}\rangle$. We then continue to decompose $|\psi_{11}\rangle = |\psi_{110}\rangle + |\psi_{111}\rangle$ and so on, until we reach the last copy and obtain $|\psi_{1^m}\rangle$. Intuitively, all the $|\psi_{1\ldots10}\rangle$ terms will be rejected with high probability, while the $|\psi_{1^m}\rangle$ term represents the "good" component where the prover knows the answer to every testing round and therefore has high accept probabil-

[9] The analysis of [7] is more tailored to the decision problems setting, and it is unclear how to extend it to sampling problems where there are multiple bits of output.

[10] Actually they are not projectors, but for the simplicity of this discussion let's assume they are.

ity. Therefore, $|\psi_{1^m}\rangle$ also satisfies some binding property, so the verifier should obtain a measurement result of some state on the Hadamard round copy, and the soundness of the QPIP_1 protocol Π_{Samp} follows.

However, the intuition that $|\psi_{1^m}\rangle$ is binding to every Hadamard round is incorrect. As $G_{1,i}$ does not commute with $G_{1,j}$, $|\psi_{1^m}\rangle$ is unfortunately only binding for the m-th copy. To solve this problem, we start with a pointwise argument and fix the Hadamard round on the i-th copy where $|\psi_{1^i}\rangle$ is binding, and show that the corresponding output is $O(\||\psi_{1^{i-1}0}\rangle\|)$-close to ideal. We can later average out this error over the different choices of i, since not all $\||\psi_{1^{i-1}0}\rangle\|$ can be large at the same time. Another way to see this issue is to notice that we are partitioning a quantum state, not probability events, so there are some inconsistencies between our intuition and calculation. Indeed, the error we get in the end is $O(\sqrt{1/m})$ instead of the $O(1/m)$ we expected.

The intuitive analysis outlined above glosses over many technical details, and we substantiate this outline with full details in Section 4.

1.2 Blind CVQC Protocols

Another desirable property of CVQC protocols is *blindness*, which means that the prover does not learn any information about the private input for the delegated computation.[11] In the relaxed setting where the verifier has a limited quantum capability, Hayashi and Morimae [26] constructed a blind QPIP_1 protocol for delegating quantum computation with information-theoretic security that also handles sampling problems. However, for purely classical verifiers, blind CVQC protocols seem much more difficult to construct. This goal is recently achieved by the seminal work of Gheorghiu and Vidick [24], who constructed the first blind CVQC protocol for BQP by constructing a composable remote state preparation protocol and combining it with the verifiable blind quantum computation protocol of Fitzsimons and Kashefi [22]. However, their protocol has polynomially many rounds and requires a rather involved analysis. Before our work, it is an open question whether constant-round blind CVQC protocol for BQP is achievable.

Our Contribution. Somewhat surprisingly, we provide a simple yet powerful *generic* compiler that transforms any CVQC protocol to a blind one while preserving completeness, soundness, as well as its round complexity. Our compiler relies on quantum fully homomorphic encryption (QFHE) schemes with certain "classical-friendly" properties, which is satisfied by both constructions of Mahadev [29] and Brakerski [12].

Theorem 2 (informal). *Assuming the QLWE assumption*[12]*, there exists a protocol compiler that transforms any CVQC protocol Π to a CVQC protocol*

[11] In literature, the definition of blindness may also require to additionally hide the computation. We note the two notions are equivalent from a feasibility point of view by a standard transformation (see our full version [17]).

[12] By using Brakerski's QFHE, we only need to rely on the QLWE assumption with polynomial modulus in this theorem.

Π_{blind} that achieves blindness while preserves its round complexity, completeness, and soundness.

Applying our blindness compiler to the parallel repetition of Mahadev's protocol from [7, 15], we obtain the first constant-round blind CVQC protocol for BQP with negligible completeness and soundness error, resolving the aforementioned open question.

Theorem 3 (informal). *Under the QLWE assumption, there exists a blind, four-message CVQC protocol for all languages in* BQP *with negligible completeness and soundness errors.*

We can also apply our compier to our CVQC protocol for SampBQP to additionally achieve blindness.

Theorem 4 (informal). *Under the QLWE assumption, there exists a blind, four-message CVQC protocol for all sampling problems in* SampBQP *with computational soundness and negligible completeness error.*

Techniques. At a high-level, the idea is simple: we run the original protocol under a QFHE with the QFHE key generated by the verifier. Intuitively, this allows the prover to compute his next message under encryption without learning verifier's message, and hence achieves blindness while preserving the properties of the original protocol. One subtlety with this approach is the fact that the verifier is classical while the QFHE cipher text could contain quantum data. In order to make the classical verifier work in this construction, the ciphertext and the encryption/decryption algorithm need to be classical when the underlying message is classical. Fortunately, such "classical-friendly" property is satisfied by the construction of [12, 29].

A more subtle issue is to preserve the soundness. In particular, compiled protocols with a single application of QFHE might (1) leak information about the circuit evaluated by the verifier through its outputted QFHE ciphertexts (i.e., no *circuit privacy*); or (2) fail to simulate original protocols upon receiving invalid ciphertexts from the prover. We address these issues by letting the verifier switch to a fresh new key for each round of the protocol. Details are given in Section 5.

1.3 Related and Followup Works and Discussions

As mentioned, while we are the first to explicitly investigate delegation of quantum sampling problems, Hayashi and Morimae [26] constructed an one-round blind QPIP_1 protocol that can be used to delegate SampBQP and achieve our notion of information-theoretical security. Like our SampBQP protocol, their protocol has an arbitrarily small inverse polynomial soundness error instead of negligible soundness error. Also as mentioned, Gheorghiu and Vidick [24] constructed the first blind CVQC protocol for BQP by constructing a composable remote state preparation protocol and combining it with the verifiable blind

quantum computation protocol of Fitzsimons and Kashefi [22]. However, their protocol has polynomially many rounds and requires a rather involved analysis.

It is also worth noting that several existing constructions in the relaxed models (e.g., verifiable blind computation [22]) can be generalized to delegate SampBQP in a natural way, but it seems challenging to analyze the soundness of the generalized protocol. Furthermore, it is unlikely that these generalized protocols can achieve negligible soundness error for SampBQP. The reason is that in all these constructions, some form of cut and choose are used to achieve soundness. For sampling problems, as mentioned, there seems to be no generic way to combine multiple samples for error reduction, so the verifier needs to choose one sample to output in the cut and choose. In this case, an adversarial prover may choose to cheat on a random copy in the cut and choose and succeed in cheating with an inverse polynomial probability.

On the other hand, while the definition of SampBQP in [2,3] allows an inverse polynomial error, there seems to be no fundamental barriers to achieve negligible error. It is conceivable that negligible error can be achieved using quantum error correction. Negligible security error is also achievable in the related settings of secure multi-party quantum computation [19,20] and verifiable quantum FHE [6] based on verifiable quantum secret sharing or quantum authentication codes[13]. However, both primitives require computing and communicating quantum encodings and are not applicable in the context of CVQC and QPIP_1. An intriguing open problem is whether it is possible to achieve negligible soundness error with classical communication while delegating a quantum sampling problem.

In a recent work, Bartusek [9] used the technique we developed for delegation of SampBQP to construct secure quantum computation protocols with classical communication for pseudo-deterministic quantum functionalities.

Organization For preliminary technical background, see our full version [17]. Our simulation-based definition of CVQC for SampBQP is discussed in Section 2. Our main technical contributions are explained in Section 3 (a construction of QPIP_1 protocol for SampBQP), Section 4 (the construction of QPIP_0 protocol for SampBQP based on the above QPIP_1 protocol), and Section 5 (a generic compiler to upgrade QPIP_0 protocols with blindness).

2 Delegation of Quantum Sampling Problems

In this section, we formally introduce the task of delegation for quantum sampling problems. We start by recalling the complexity class SampBQP defined by Aaronson [2,3], which captures the class of sampling problems that are approximately solvable by polynomial-time quantum algorithms.

Definition 1 (Sampling Problem). *A* sampling problem *is a collection of probability distributions $(D_x)_{x\in\{0,1\}^*}$, one for each input string $x \in \{0,1\}^n$, where D_x is a distribution over $\{0,1\}^{m(n)}$ for some fixed polynomial m.*

[13] The security definitions are not comparable, but it seems plausible that the techniques can be used to achieve negligible soundness error for sampling problems.

Definition 2 (SampBQP)**.** SampBQP *is the class of sampling problems* $(D_x)_{x\in\{0,1\}^*}$ *that can be (approximately) sampled by polynomial-size uniform quantum circuits. Namely, there exists a Turing machine* M *such that for every* $n \in \mathbb{N}$ *and* $\epsilon \in (0,1)$*,* $M(1^n, 1^{1/\epsilon})$ *outputs a quantum circuit* C *in* $\mathrm{poly}(n, 1/\epsilon)$ *time such that for every* $x \in \{0,1\}^n$*, the output of* $C(x)$ *(measured in standard basis) is* ϵ*-close to* D_x *in the total variation distance.*

Note that in the above definition, there is an accuracy parameter ϵ and the quantum sampling algorithm only requires to output a sample that is ϵ-close to the correct distribution in time $\mathrm{poly}(n, 1/\epsilon)$. [2,3] discussed multiple reasons for allowing the inverse polynomial error, such as to take into account the inherent noise in conceivable physical realizations of quantum computer. On the other hand, it is also meaningful to require negligible error. As discussed, it is an intriguing open question to delegate quantum sampling problem with negligible error.

We next define what it means for a QPIP_τ protocol[14] to solve a SampBQP problem $(D_x)_{x\in\{0,1\}^*}$. Since sampling problems come with an accuracy parameter ϵ, we let the prover P and the verifier V receive the input x and $1^{1/\epsilon}$ as common inputs. Completeness is straightforward to define, which requires that when the prover P is honest, the verifier V should accept with high probability and output a sample z distributed close to D_x on input x. Defining soundness is more subtle. Intuitively, it requires that the verifier V should never be "cheated" to accept and output an incorrect sample even when interacting with a malicious prover. We formalize this by a strong simulation-based definition, where we require that the joint distribution of the decision bit $d \in \{\mathsf{Acc}, \mathsf{Rej}\}$ and the output z (which is $\perp$ when $d = \mathsf{Rej}$) is ϵ-close (in either statistical or computational sense) to an "ideal distribution" (d, z_{ideal}), where z_{ideal} is sampled from D_x when $d = \mathsf{Acc}$ and set to $\perp$ when $d = \mathsf{Rej}$. Since the protocol receives the accuracy parameter $1^{1/\epsilon}$ as input to specify the allowed error, we do not need to introduce an additional soundness error parameter in the definition.

Definition 3. *Let* $\Pi = (P, V)$ *be a* QPIP_τ *protocol. We say it is a protocol for the* SampBQP *instance* $(D_x)_{x\in\{0,1\}^*}$ *with completeness error* $c(\cdot)$ *and statistical (resp., computational) soundness if the following holds:*

- *On public inputs* 1^λ*,* $1^{1/\epsilon}$*, and* $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$*,* V *outputs* (d, z) *where* $d \in \{\mathsf{Acc}, \mathsf{Rej}\}$*. If* $d = \mathsf{Acc}$ *then* $z \in \{0,1\}^{m(|x|)}$ *where* m *is given in Definition 1, otherwise* $z = \perp$*.*
- *(Completeness): For all accuracy parameters* $\epsilon(\lambda) = \frac{1}{\mathrm{poly}(\lambda)}$*, security parameters* $\lambda \in \mathbb{N}$*, and* $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$*, let* $(d, z) \leftarrow (P, V)(1^\lambda, 1^{1/\epsilon}, x)$*, then* $d = \mathsf{Rej}$ *with probability at most* $c(\lambda)$*.*
- *(Statistical soundness): For all cheating provers* P^**, accuracy parameters* $\epsilon(\lambda) = \frac{1}{\mathrm{poly}(\lambda)}$*, sufficiently large* $\lambda \in \mathbb{N}$*, and* $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$*, consider the following experiment:*

[14] See our full version [17] for a formal definition of QPIP_τ.

- *Let* $(d, z) \leftarrow (P^*, V)(1^\lambda, 1^{1/\epsilon}, x)$.
- *Define* z_{ideal} *by*

$$\begin{cases} z_{ideal} = \perp & \text{if } d = \mathsf{Rej} \\ z_{ideal} \leftarrow D_x & \text{if } d = \mathsf{Acc} \end{cases}$$

.

It holds that $\|(d, z) - (d, z_{ideal})\|_{\mathrm{TV}} \leq \epsilon$.

– *(Computational soundness): For all cheating* BQP *provers* P^*, BQP *distinguishers* D, *accuracy parameters* $\epsilon(\lambda) = \frac{1}{\mathrm{poly}(\lambda)}$, *sufficiently large* $\lambda \in \mathbb{N}$, *and all* $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$, *let us define* d, z, z_{ideal} *by the same experiment as above. It holds that* (d, z) *is* ϵ*-computationally indistinguishable to* (d, z_{ideal}) *over* λ.

As in the case of BQP, we are particularly interested in the case that $\tau = 0$, i.e., when the verifier V is classical. In this case, we say that Π is a CVQC protocol for the $\mathsf{SampBQP}$ problem $(D_x)_{x\in\{0,1\}^*}$.

3 Construction of the $\mathbf{QPIP_1}$ Protocol for $\mathbf{SampBQP}$

As we mentioned in this introduction, we will employ the circuit *history* state in the original construction of the Local Hamiltonian problem [28] to encode the circuit information for $\mathsf{SampBQP}$. However, there are distinct requirements between certifying the computation for BQP and $\mathsf{SampBQP}$ based on the history state. For any quantum circuit C on input x, the original construction for certifying BQP[15] consists of local Hamiltonian $H_{\mathrm{in}}, H_{\mathrm{clock}}, H_{\mathrm{prop}}, H_{\mathrm{out}}$ where H_{in} is used to certify the initial input x, H_{clock} to certify the validness of the clock register, H_{prop} to certify the gate-by-gate evolution according to the circuit description, and H_{out} to certify the final output. In particular, the corresponding history state is in the ground space of H_{in}, H_{clock}, and H_{prop}. Note that BQP is a decision problem and its outcome (0/1) can be easily encoded into the energy H_{out} on the single output qubit. As a result, the outcome of BQP can simply be encoded by the *ground energy* of $H_{\mathrm{in}} + H_{\mathrm{clock}} + H_{\mathrm{prop}} + H_{\mathrm{out}}$.

To deal with $\mathsf{SampBQP}$, we will still employ $H_{\mathrm{in}}, H_{\mathrm{clock}}$, and H_{prop} to certify the circuit's input, the clock register, and gate-by-gate evolution. However, in $\mathsf{SampBQP}$, we care about the entire final state of the circuit, rather than the energy on the output qubit. Our approach to certify the entire final state (which is encoded inside the history state) is to make sure that the history state is the unique ground state of $H_{\mathrm{in}} + H_{\mathrm{clock}} + H_{\mathrm{prop}}$ and all other orthogonal states will have much higher energies. Namely, we need to construct some $H'_{\mathrm{in}} + H'_{\mathrm{clock}} + H'_{\mathrm{prop}}$ with the history state as the unique ground state and with a large *spectral* gap between the ground energy and excited energies. It is hence guaranteed that any state with close-to-ground energy must also be close to the history state. We remark that this is a different requirement from most local

[15] The original construction is for the purpose of certifying problems in QMA. We consider its simple restriction to problems inside BQP.

Hamiltonian constructions that focus on the ground energy. We achieve so by using the *perturbation* technique developed in [27] for reducing the locality of Hamiltonian. Another example of local Hamiltonian construction with a focus on the spectral gap can be found in [5], where the purpose is to simulate quantum circuits by adiabatic quantum computation.

We need two more twists for our purpose. First, as we will eventually measure the final state in order to obtain classical samples, we need that the final state occupies a large fraction of the history state. We can simply add dummy identity gates. Second, as we are only able to perform X or Z measurement by techniques from [30], we need to construct X-Z only local Hamiltonians. Indeed, this has been shown possible in, e.g., [11], which serves as the starting point of our construction.

We present the formal construction of our QPIP_1 protocol Π_{Samp} for $\mathsf{SampBQP}$ in our full version [17]. The soundness and completeness of **??** is stated in the following theorem, whose proof is also deferred to [17].

Theorem 5. *Π_{Samp} is a QPIP_1 protocol for the $\mathsf{SampBQP}$ problem $(D_x)_{x\in\{0,1\}^*}$ with negligible completeness error and is statistically sound*[16] *where the verifier only needs to do non-adaptive X/Z measurements.*

4 SampBQP Delegation Protocol for Fully Classical Client

In this section, we create a delegation protocol for $\mathsf{SampBQP}$ with fully classical clients by adapting the approach taken in [30]. In [30], Mahadev designed a protocol Π_{Measure} (Protocol 1) that allows a BQP prover to "commit a state" for a classical verifier to choose a X or Z measurement and obtain corresponding measurement results. Composing it with the QPIP_1 protocol for BQP from [21] results in a QPIP_0 protocol for BQP. In this work, we will compose Π_{Measure} with our QPIP_1 protocol Π_{Samp} (**??**) for $\mathsf{SampBQP}$ in order to obtain a QPIP_0 protocol for $\mathsf{SampBQP}$.

A direct composition of Π_{Samp} and Π_{Measure}, however, results in Π_{int} (Protocol 2) which does not provide reasonable completeness or accuracy guarantees. As we will see, this is due to Π_{Measure} itself having peculiar and weak guarantees: the client doesn't always obtain measurement outcomes even if the server were honest. When that happens under the BQP context, the verifier can simply accept the prover at the cost of some soundness error; under our $\mathsf{SampBQP}$ context, however, we must run many copies of Π_{int} in parallel so the verifier can generate its outputs from some copy. We will spend the majority of this section analyzing the soundness of this parallel repetition.

4.1 Mahadev's measurement protocol

Π_{Measure} is a 4-round protocol between a verifier (which corresponds to our client) and a prover (which corresponds to our server). The verifier (secretly) chooses

[16] The soundness and completeness of a $\mathsf{SampBQP}$ protocol is defined in Definition 3.

a string h specifying the measurements he wants to make, and generates keys pk, sk from h. It sends pk to the prover. The prover "commits" to a state ρ of its choice using pk and replies with its commitment y. The verifier must then choose between two options: do a *testing round* or a *Hadamard round*. In a testing round the verifier can catch cheating provers, and in a Hadamard round the verifier receives some measurement outcome. He sends his choice to the prover, and the prover replies accordingly. If the verifier chose testing round, he checks the prover's reply against the previous commitment, and rejects if he sees an inconsistency. If the verifier chose Hadamard round, he calculates $M_{XZ}(\rho, h)$ based on the reply. We now formally describe the interface of Π_{Measure} while omitting the implementation details.

Protocol 1 Mahadev's measurement protocol $\Pi_{\mathsf{Measure}} = (P_{\mathsf{Measure}}, V_{\mathsf{Measure}})$

Inputs:

- Common input: Security parameter 1^λ where $\lambda \in \mathbb{N}$.
- Prover's input: a state $\rho \in \mathcal{B}^{\otimes n}$ for the verifier to measure.
- Verifier's input: the measurement basis choice $h \in \{0,1\}^n$

Protocol:

1. The verifier generates a public and secret key pair $(pk, sk) \leftarrow \mathcal{V}_{\mathsf{Measure},1}(1^\lambda, h)$. It sends pk to the prover.
2. The prover generates $(y, \sigma) \leftarrow \mathcal{P}_{\mathsf{Measure},2}(pk, \rho)$. y is a classical "commitment", and σ is some internal state. He sends y to the verifier.
3. The verifier samples $c \xleftarrow{\$} \{0,1\}$ uniformly at random and sends it to the prover. $c = 0$ indicates a *testing round*, while $c = 1$ indicates a *Hadamard round*.
4. The prover generates a classical string $a \leftarrow \mathcal{P}_{\mathsf{Measure},4}(pk, c, \sigma)$ and sends it back to the verifier.
5. If it is a testing round ($c = 0$), then the verifier generates and outputs $o \leftarrow \mathcal{V}_{\mathsf{Measure},T}(pk, y, a)$ where $o \in \{\mathsf{Acc}, \mathsf{Rej}\}$. If it is a Hadamard round ($c = 1$), then the verifier generates and outputs $v \leftarrow \mathcal{V}_{\mathsf{Measure},H}(sk, h, y, a)$.

Π_{Measure} has negligible completeness errors, i.e. if both the prover and verifier are honest, the verifier accepts with overwhelming probability and his output on Hadamard round is computationally indistinguishable from $M_{XZ}(\rho, h)$. As for soundness, it gives the following *binding property* against cheating provers: if a prover would always succeed on the testing round, then there exists some ρ so that for any h the verifier obtains $M_{XZ}(\rho, h)$ if he had chosen the Hadamard round.

Lemma 1 (binding property of Π_{Measure}; special case of Claim 7.1 in [30]). *Let P^*_{Measure} be a BQP cheating prover for Π_{Measure} and λ be the security parameter. Let $1 - p_{h,T}$ be the probability that the verifier accepts P^*_{Measure} in*

the testing round on basis choice h.[17] *Under the QLWE assumption, there exists some ρ^* so that for all verifier's input $h \in \{0,1\}^n$, the verifier's outputs on the Hadamard round is $\sqrt{p_{h,T}} + \mathrm{negl}(n)$-computationally indistinguishable from $M_{XZ}(\rho^*, h)$.*

We now combine Π_{Measure} with our QPIP_1 Protocol for $\mathsf{SampBQP}$, $\Pi_{\mathsf{Samp}} = (P_{\mathsf{Samp}}, V_{\mathsf{Samp}})$ (**??**), to get a corresponding QPIP_0 protocol Π_{int}. Recall that in Π_{Samp} the verifier takes X and Z measurements on the prover's message. In Π_{int} we let the verifier use Π_{Measure} to learn those measurement outcomes instead.

Protocol 2 Intermediate QPIP_0 protocol Π_{int} for the $\mathsf{SampBQP}$ problem $(D_x)_{x \in \{0,1\}^*}$

Inputs:

- Security parameter 1^λ where $\lambda \in \mathbb{N}$
- Error parameter $\epsilon \in (0,1)$
- Classical input $x \in \{0,1\}^n$ to the $\mathsf{SampBQP}$ instance

Protocol:

1. The verifier chooses a XZ-measurement h from the distribution specified in Step **??** of Π_{Samp}.
2. The prover prepares ρ by running Step **??** of Π_{Samp}.
3. The verifier and prover run $(P_{\mathsf{Measure}}(\rho), V_{\mathsf{Measure}}(h))(1^\lambda)$.
   (a) The verifier samples $(pk, sk) \leftarrow \mathcal{V}_{\mathsf{int},1}(1^\lambda, h)$ and sends pk to the prover, where $\mathcal{V}_{\mathsf{int},1}$ is the same as $\mathcal{V}_{\mathsf{Measure},1}$ of Protocol 1.
   (b) The prover runs $(y, \sigma) \leftarrow \mathcal{P}_{\mathsf{int},2}(pk, \rho)$ and sends y to the verifier, where $\mathcal{P}_{\mathsf{int},2}$ is the same as $\mathcal{P}_{\mathsf{Measure},2}$. Here we allow the prover to abort by sending $y = \perp$, which does not benefit cheating provers but simplifies our analysis of parallel repetition later.
   (c) The verifier samples $c \xleftarrow{\$} \{0,1\}$ and sends it to the prover.
   (d) The prover replies $a \leftarrow \mathcal{P}_{\mathsf{int},4}(pk, c, \sigma)$.
   (e) If it is a testing round, the verifier accepts or rejects based on the outcome of Π_{Measure}. If it is a Hadamard round, the verifier obtains v.
4. If it's a Hadamard round, the verifier finishes the verification step of Protocol **??** by generating and outputting (d, z)

There are several problems with using Π_{int} as a $\mathsf{SampBQP}$ protocol. First, since the verifier doesn't get a sample if he had chosen the testing round in Step 3c, the protocol has completeness error at least $1/2$. Moreover, since Π_{Measure} does not check anything on the Hadamard round, a cheating prover can give up passing the testing round and breaks the commitment on the Hadamard round,

[17] Compared to Claim 7.1 of [30], we don't have a $p_{h,H}$ term here. This is because on rejecting a Hadamard round, the verifier can output a uniformly random string, and that is same as the result of measuring h on the totally mixed state.

with only a constant 1/2 probability of being caught. However, we can show that Π_{int} has a binding property similar to Π_{Measure}: if a cheating prover P^*_{int} passes the testing round with overwhelming probability whenever it doesn't abort on the second message, then the corresponding output $(d, z) \leftarrow (P^*_{\mathsf{int}}, V_{\mathsf{int}})$ is close to (d, z_{ideal}). Recall the ideal output is

$$\begin{cases} z_{ideal} = \bot & \text{if } d = \mathsf{Rej} \\ z_{ideal} \leftarrow D_x & \text{if } d = \mathsf{Acc}. \end{cases}$$

This binding property is formalized in Theorem 6. Intuitively, the proof of Theorem 6 combines the binding property of Protocol 2 (Lemma 1) and Π_{Samp}'s soundness (Theorem 5). There is a technical issue that Protocol 2 allows the prover to abort while Protocol 1 does not. This issue is solved by constructing another BQP prover P^* for every cheating prover P^*_{int}. Specifically, P^* uses P^*_{int}'s strategy when it doesn't abort, otherwise honestly chooses the totally mixed state for the verifier to measure.

Theorem 6 (binding property of Π_{int}). *Let P^*_{int} be a cheating BQP prover for Π_{int} and λ be the security parameter. Suppose that $\Pr[d = \mathsf{Acc} \mid y \neq \bot, c = 0]$ is overwhelming, under the QLWE assumption, then the verifier's output in the Hadamard round is $O(\epsilon)$-computationally indistinguishable from (d, z_{ideal}).*

Proof (Theorem 6). We first introduce the *dummy strategy* for Π_{Measure}, where the prover chooses ρ as the maximally mixed state and executes the rest of the protocol honestly. It is straightforward to verify that this prover would be accepted in the testing round with probability $1 - \mathrm{negl}(\lambda)$, but has negligible probability passing the verification after the Hadamard round.

Now we construct a cheating BQP prover for Protocol 2, P^*, that does the same thing as P^*_{int} except at Step 3, where the prover and verifier runs Protocol 1. P^* does the following in Step 3: for the second message, run $(y, \sigma) \leftarrow \mathcal{P}^*_{\mathsf{int},2}(pk, \rho)$. If $y \neq \bot$, then reply y; else, run the corresponding step of the dummy strategy and reply with its results. For the fourth message, if $y \neq \bot$, run and reply with $a \leftarrow \mathcal{P}^*_{\mathsf{int},4}(pk, c, \sigma)$; else, continue the dummy strategy.

In the following we fix an x. Let the distribution on h specified in Step 1 of the protocol be $p_x(h)$. Define $P^*_{\mathsf{sub}}(x)$ as P^*'s response in Step 3. Note that we can view $P^*_{\mathsf{sub}}(x)$ as a prover strategy for Protocol 1. By construction $P^*_{\mathsf{sub}}(x)$ passes testing round with overwhelming probability over $p_x(h)$, i.e. $\sum_h p_x(h) p_{h,T} = \mathrm{negl}(\lambda)$, where $p_{h,T}$ is P^*'s probability of getting accepted by the prover on the testing round on basis choice h. By Lemma 1 and Cauchy's inequality, there exists some ρ such that $\sum_h p_x(h) \left\| v_h - M_{XZ}(\rho, h) \right\|_c = \mathrm{negl}(\lambda)$, where we use $\left\| A - B \right\|_c = \alpha$ to denote that A is α-computational indistinguishable to B. Therefore $v = \sum_h p_x(h) v_h$ is computationally indistinguishable to $\sum_h p_x(h) M_{XZ}(\rho, h)$. Combining it with Π_{Samp}'s soundness (Theorem 5), we see that $(d', z') \leftarrow (P^*, V_{\mathsf{int}})(1^\lambda, 1^{1/\epsilon}, x)$ is ϵ-computationally indistinguishable to (d', z'_{ideal}).

Now we relate (d', z') back to (d, z). First, conditioned on that P^*_{int} aborts, since dummy strategy will be rejected with overwhelming probability in Hadamard

round, we have (d', z') is computationally indistinguishable to $(\mathsf{Rej}, \perp) = (d, z)$. On the other hand, conditioned on P^*_{int} not aborting, clearly $(d, z) = (d', z')$. So (d, z) is computationally indistinguishable to (d', z'), which in turn is $O(\epsilon)$-computationally indistinguishable to (d', z'_{ideal}). Since $\|d - d'\|_{tr} = O(\epsilon)$, (d, z_{ideal}) is $O(\epsilon)$-computationally indistinguishable to (d', z'_{ideal}). Combining everything, we conclude that (d, z) is $O(\epsilon)$-computationally indistinguishable to (d, z_{ideal}).

4.2 QPIP_0 protocol for SampBQP

We now introduce our QPIP_0 protocol Π_{Final} for SampBQP. It is essentially a m-fold parallel repetition of Π_{int}, from which we uniformly randomly pick one copy to run Hadamard round to get our samples and run testing round on all other $m-1$ copies. Intuitively, if the server wants to cheat by sending something not binding on some copy, he will be caught when that copy is a testing round, which is with probability $1-1/m$. This over-simplified analysis does not take into account that the server might create entanglement between the copies. Therefore, a more technically involved analysis is required.

In the description of our protocol below, we describe Π_{int} and Π_{Measure} in details in order to introduce notations that we need in our analysis.

Protocol 3 QPIP_0 protocol Π_{Final} for the SampBQP problem $(D_x)_{x\in\{0,1\}^*}$

Inputs:

- Security parameter 1^λ for $\lambda \in \mathbb{N}$.
- Accuracy parameter $1^{1/\epsilon}$ for the SampBQP problem.
- Input $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$ for the SampBQP instance.

Ingredient: Let $m = O(1/\epsilon^2)$ be the number of parallel repetitions to run.

Protocol:

1. The verifier generates m independent copies of basis choices $\vec{h} = (h_1, \ldots, h_m)$, where each copy is generated as in Step 1 of Π_{int}.
2. The prover prepares $\rho^{\otimes m}$; each copy of ρ is prepared as in Step 2 of Π_{int}.
3. The verifier generates m key pairs for Π_{Measure}, $\vec{pk} = (pk_1, \ldots, pk_m)$ and $\vec{sk} = (sk_1, \ldots, sk_m)$, as in Step 1 of Π_{Measure}. It sends $\vec{pk}$ to the prover.
4. The prover generates $\vec{y} = (y_1, \ldots, y_m)$ and σ as in Step 2 of Π_{Measure}. It sends $\vec{y}$ to the verifier.
5. The verifier samples $r \xleftarrow{\$} [m]$ which is the copy to run Hadamard round for. For $1 \le i \le m$, if $i \neq r$ then set $c_i \leftarrow 0$, else set $c_i \leftarrow 1$. It sends $\vec{c} = (c_1, \ldots, c_m)$ to the prover.
6. The prover generates $\vec{a}$ as in Step 4 of Π_{Measure}, and sends it back to the verifier.

7. The verifier computes the outcome for each round as in Step 4 of Π_{int}. If any of the testing round copies are rejected, the verifier outputs $(\mathsf{Rej}, \perp)$. Else, it outputs the result from the Hadamard round copy.

By inspection, Π_{Final} is a QPIP_0 protocol for $\mathsf{SampBQP}$ with negligible completeness error. To show that it is computationally sound, we first use the partition lemma from [15].

Intuitively, the partition lemma says that for any cheating prover and for each copy $i \in [m]$, there exist two efficient "projectors" [18] $G_{0,i}$ and $G_{1,i}$ in the prover's internal space with $G_{0,i} + G_{1,i} \approx Id$. $G_{0,i}$ and $G_{1,i}$ splits up the prover's residual internal state after sending back his first message. $G_{0,i}$ intuitively represents the subspace where the prover does not knows the answer to the testing round on the i-th copy, while $G_{1,i}$ represents the subspace where the prover does. Note that the prover is using a single internal space for all copies, and every $G_{0,i}$ and every $G_{1,i}$ is acting on this single internal space. By using this partition lemma iteratively, we can decompose the prover's internal state $|\psi\rangle$ into sum of subnormalized states. First we apply it to the first copy, writing $|\psi\rangle = G_{0,1}|\psi\rangle + G_{1,1}|\psi\rangle \equiv |\psi_0\rangle + |\psi_1\rangle$. The component $|\psi_0\rangle$ would then get rejected as long as the first copy is chosen as a testing round, which occurs with pretty high probability. More precisely, the output corresponding to $|\psi_0\rangle$ is $1/m$-close to the ideal distribution that just rejects all the time. On the other hand, $|\psi_1\rangle$ is now binding on the first copy; we now similarly apply the partition lemma of the second copy to $|\psi_1\rangle$. We write $|\psi_1\rangle = G_{0,2}|\psi_1\rangle + G_{1,2}|\psi_1\rangle \equiv |\psi_{10}\rangle + |\psi_{11}\rangle$, and apply the same argument about $|\psi_{10}\rangle$ and $|\psi_{11}\rangle$. We then continue to decompose $|\psi_{11}\rangle = |\psi_{110}\rangle + |\psi_{111}\rangle$ and so on, until we reach the last copy and obtain $|\psi_{1^m}\rangle$. Intuitively, the $|\psi_{1^m}\rangle$ term represents the "good" component where the prover knows the answer to every testing round and therefore has high accept probability. Therefore, $|\psi_{1^m}\rangle$ also satisfies some binding property, so the verifier should obtain a measurement result of some state on the Hadamard round copy, and the analysis from the QPIP_1 protocol Π_{Samp} follows.

However, the intuition that $|\psi_{1^m}\rangle$ is binding to every Hadamard round is incorrect. As $G_{1,i}$ does not commute with $G_{1,j}$, $|\psi_{1^m}\rangle$ is unfortunately only binding for the m-th copy. To solve this problem, we start with a pointwise argument and fix the Hadamard round on the i-th copy where $|\psi_{1^i}\rangle$ is binding, and show that the corresponding output is $O(\||\psi_{1^{i-1}0}\rangle\|)$-close to ideal. We can later average out this error over the different choices of i, since not all $\||\psi_{1^{i-1}0}\rangle\|$ can be large at the same time. Another way to see this issue is to notice that we are partitioning a quantum state, not probability events, so there are some inconsistencies between our intuition and calculation. Indeed, the error we get in the end is $O(\sqrt{1/m})$ instead of the $O(1/m)$ we expected.

Also a careful reader might have noticed that the prover's space don't always decompose cleanly into parts that the verifier either rejects or accepts with high probability, as there might be some states that is accepted with mediocre

[18] Actually they are not projectors, but for the simplicity of this discussion let's assume they are.

probability. As in [15], we solve this by splitting the space into parts that are accepted with probability higher or lower than a small threshold γ and applying Marriott-Watrous [31] amplification to boost the accept probability if it is bigger than γ, getting a corresponding amplified prover action Ext. However, states with accept probability really close to the threshold γ can not be classified, so we average over randomly chosen γ to have $G_{0,i} + G_{1,i} \approx Id$. Now we give a formal description of the partition lemma.

Lemma 2 (partition lemma; revision of Lemma 3.5 of [15][19]). *Let λ be the security parameter, and $\gamma_0 \in [0,1]$ and $T \in \mathbb{N}$ be parameters that will be related to the randomly-chosen threshold γ. Let (U_0, U) be a prover's strategy in a m-fold parallel repetition of Π_{Measure}[20], where U_0 is how the prover generates $\vec{y}$ on the second message, and U is how the prover generates $\vec{a}$ on the fourth message. Let $H_{\mathbf{X},\mathbf{Z}}$ be the Hilbert space of the prover's internal calculation. Denote the string $0^{i-1}10^{m-i} \in \{0,1\}^m$ as e_i, which corresponds to doing Hadamard round on the i-th copy and testing round on all others.*

For all $i \in [m]$, $\gamma \in \left\{\frac{\gamma_0}{T}, \frac{2\gamma_0}{T}, \ldots, \frac{T\gamma_0}{T}\right\}$, there exist two $\mathrm{poly}(1/\gamma_0, T, \lambda)$-time quantum circuit with post selection[21] $G_{0,i,\gamma}$ and $G_{1,i,\gamma}$ such that for all (possibly sub-normalized) quantum states $|\psi\rangle_{\mathbf{X},\mathbf{Z}} \in H_{\mathbf{X},\mathbf{Z}}$, properties 1 2 3 4, to be described later, are satisfied. Before we describe the properties, we introduce the following notations:

$$|\psi_{0,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}} := G_{0,i,\gamma} |\psi\rangle_{\mathbf{X},\mathbf{Z}}, \tag{4.1}$$

$$|\psi_{1,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}} := G_{1,i,\gamma} |\psi\rangle_{\mathbf{X},\mathbf{Z}}, \tag{4.2}$$

$$|\psi_{err,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}} := |\psi\rangle_{\mathbf{X},\mathbf{Z}} - |\psi_{0,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}} - |\psi_{1,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}}. \tag{4.3}$$

Note that $G_{0,i,\gamma}$ and $G_{1,i,\gamma}$ has failure probabilities, and this is reflected by the fact that $|\psi_{0,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}}$ and $|\psi_{1,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}}$ are sub-normalized. $G_{0,i,\gamma}$ and $G_{1,i,\gamma}$ depend on (U_0, U) and $\vec{pk}, \vec{y}$.

The following properties are satisfied for all $i \in [m]$:

[19] G_0 and G_1 of this version are created from doing G of [15] and post-selecting on the ph, th, in register being 0^t01 or 0^t11 then discard ph, th, in. Property 1 corresponds to Property 1. Property 2 corresponds to Property 4, with 2^{m-1} changes to $m-1$ because we only have m possible choices of $\vec{c}$. Property 3 corresponds to Property 5. Property 4 comes from the fact that G_0 and G_1 are post-selections of orthogonal results of the same G.

[20] A m-fold parallel repetition of Π_{Measure} is running step 3 4 5 6 of Protocol 3 with verifier input $\vec{h}$ and prover input $\rho^{\otimes n}$, followed by an output step where the verifier rejects if any of the $m-1$ testing round copies is rejected, otherwise outputs the result of the Hadamard round copy.

[21] A quantum circuit with post selection is composed of unitary gates followed by a post selection on some measurement outcome on ancilla qubits, so it produces a sub-normalized state, where the amplitude square of the output state is the probability of post selection.

1.
$$\mathop{\mathbb{E}}_{\gamma} \| \, |\psi_{err,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}} \, \|^2 \leq \frac{6}{T} + \mathrm{negl}(\lambda),$$
where the averaged is over uniformly sampled γ. This also implies
$$\mathop{\mathbb{E}}_{\gamma} \| \, |\psi_{err,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}} \, \| \leq \sqrt{\frac{6}{T}} + \mathrm{negl}(\lambda) \tag{4.4}$$
by Cauchy's inequality.
2. *For all $\vec{pk}$, $\vec{y}$, γ, and $j \neq i$, we have*
$$\left\| P_{acc,i} \circ U \frac{|e_j\rangle_{\mathbf{C}} \, |\psi_{0,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}}}{\| \, |\psi_{0,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}} \, \|} \right\|^2 \leq (m-1)\gamma_0 + \mathrm{negl}(\lambda), \tag{4.5}$$
where $P_{acc,i}$ are projector to the states that i-th testing round accepts with pk_i, y_i, including the last measurement the prover did before sending $\vec{a}$. This means that $|\psi_{0,i,\gamma}\rangle$ is rejected by the i-th testing round with high probability.
3. *For all $\vec{pk}$, $\vec{y}$, γ, and $j \neq i$, there exists an efficient quantum algorithm Ext_i such that*
$$\left\| P_{acc,i} \circ \mathsf{Ext}_i \left(\frac{|e_j\rangle_{\mathbf{C}} \, |\psi_{1,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}}}{\| \, |\psi_{1,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}} \, \|} \right) \right\|^2 = 1 - \mathrm{negl}(\lambda). \tag{4.6}$$
This will imply that $|\psi_{1,i,\gamma}\rangle$ is binding to the i-th Hadamard round.
4. *For all γ,*
$$\left\| |\psi_{0,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}} \right\|^2 + \left\| |\psi_{1,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}} \right\|^2 \leq \left\| |\psi\rangle_{\mathbf{X},\mathbf{Z}} \right\|^2. \tag{4.7}$$

Note that in property 3, we are using Ext_i instead of U because we use amplitude amplification to boost the success probability.

We now decompose the prover's internal state by using Lemma 2 iteratively. Let $|\psi\rangle$ be the state the prover holds before he receives $\vec{c}$; we denote the corresponding Hilbert space as $H_{\mathbf{X},\mathbf{Z}}$. For all $k \in [m]$, $d \in \{0,1\}^k$, $\gamma = (\gamma_1, \ldots, \gamma_k)$ where each $\gamma_j \in \{ \frac{\gamma_0}{T}, \frac{2\gamma_0}{T}, \ldots, \frac{T\gamma_0}{T} \}$, and $|\psi\rangle \in H_{\mathbf{X},\mathbf{Z}}$, define
$$|\psi_{d,\gamma}\rangle := G_{d_k,k,\gamma_k} \ldots G_{d_2,2,\gamma_2} G_{d_1,1,\gamma_1} \, |\psi\rangle \, .$$
For all $i \in [m]$, we then decompose $|\psi\rangle$ into
$$|\psi\rangle = \sum_{j=0}^{i-1} |\psi_{1^j 0,\gamma}\rangle + |\psi_{1^i,\gamma}\rangle + \sum_{j=1}^{i} |\psi_{err,j,\gamma}\rangle \tag{4.8}$$
by using Equations (4.1) to (4.3) repeatedly, where $|\psi_{err,i,\gamma}\rangle$ denotes the error state from decomposing $|\psi_{1^{i-1},\gamma}\rangle$.

We denote the projector in $H_{\mathbf{X},\mathbf{Z}}$ corresponding to outputting string z when doing Hadamard on i-th copy as $P_{acc,-i,z}$. Note that $P_{acc,-i,z}$ also depends

on $\vec{pk}, \vec{y}$, and (sk_i, h_i) since it includes the measurement the prover did before sending $\vec{a}$, verifier's checking on $(m-1)$ copies of testing rounds, and the verifier's final computation from (sk_i, h_i, y_i, a_i). $P_{acc,-i,z}$ is a projector because it only involves the standard basis measurements to get a and classical post-processing of the verifiers. Also note that $P_{acc,-i,z}P_{acc,-i,z'} = 0$ for all $z \neq z'$, and $\sum_z P_{acc,-i,z} = \Pi_{j\neq i}P_{acc,j} \leq Id$.

We denote the string $0^{i-1}10^{m-i} \in \{0,1\}^m$ as e_i. The output string corresponding to $|\psi\rangle \in H_{\mathbf{X},\mathbf{Z}}$ when $c = e_i$ is then

$$z_i := \mathop{\mathbb{E}}_{pk,y} \sum_z \left\| P_{acc,-i,z} U \left| e_i, \psi \right\rangle \right\|^2 |z\rangle\langle z|, \tag{4.9}$$

where $|e_i, \psi\rangle = |e_i\rangle_{\mathbf{C}} |\psi\rangle_{\mathbf{X},\mathbf{Z}}$ and U is the unitary the prover applies on the last round. Note that we have averaged over $\vec{pk}, \vec{y}$ where as previously everything has fixed $\vec{pk}$ and $\vec{y}$.

By Property 2 of Lemma 2, it clearly follows that

Corollary 1 *For all $\gamma \in \{ \frac{\gamma_0}{T}, \frac{2\gamma_0}{T}, \ldots, \frac{T\gamma_0}{T} \}$, and all $i, j \in [m]$ such that $j < i-1$, we have*

$$\left\| \sum_z P_{acc,-i,z} U \left| e_i, \psi_{1^j 0, \gamma} \right\rangle \right\|^2 \leq (m-1)\gamma_0 + \mathrm{negl}(n).$$

Now we define

$$z_{good,i} = \mathop{\mathbb{E}}_{\gamma,pk,y} \sum_z \left\| P_{acc,-i,z} U \left| e_i, \psi_{1^{i-1}1,\gamma} \right\rangle \right\|^2 |z\rangle\langle z| \tag{4.10}$$

as the output corresponding to a component that would pass the i-th testing rounds. We will show that it is $O(\||\psi_{1^{i-1}0}\rangle\|)$-close to z_i. Before doing so, we present a technical lemma.

Lemma 3. *For any state $|\psi\rangle$, $|\phi\rangle$ and projectors $\{P_z\}$ such that $P_z P_{z'} = 0$ for all $z \neq z'$, we have*

$$\sum_z |\langle\psi|P_z|\phi\rangle| \leq \sqrt{\left\| \sum_z P_z |\psi\rangle \right\|^2} \sqrt{\left\| \sum_z P_z |\phi\rangle \right\|^2}.$$

Proof.

$$\begin{aligned}
\sum_z |\langle\psi|P_z|\phi\rangle| = \sum_z |\langle\psi|P_z P_z|\phi\rangle| &\leq \sum_z \|\langle\psi| P_z\| \, \|P_z |\phi\rangle\| \\
&\leq \sqrt{\sum_z \|P_z |\psi\rangle\|^2} \sqrt{\sum_z \|P_z |\phi\rangle\|^2} \\
&\leq \sqrt{\left\| \sum_z P_z |\psi\rangle \right\|^2} \sqrt{\left\| \sum_z P_z |\phi\rangle \right\|^2},
\end{aligned}$$

where we used Cauchy's inequality on the first two inequalities and $P_z P_{z'} = 0$ on the last one.

Corollary 2 *For any state $|\psi\rangle$, $|\phi\rangle$ and projectors $\{P_z\}$ such that $\sum_z P_z \leq Id$ and $P_z P_{z'} = 0$ for all $z \neq z'$, we have*

$$\sum_z |\langle\psi|P_z|\phi\rangle| \leq \|\psi\| \, \|\phi\| .$$

Now we can estimate z_i using $z_{good,i}$, with errors on the orders of $\||\psi_{1^{i-1}0}\rangle\|$. This error might not be small in general, but we can average it out later by considering uniformly random $i \in [m]$. The analysis is tedious but straightforward; we simply expand z_i and bound the terms that are not $z_{good,i}$.

Lemma 4.

$$\begin{aligned}\mathrm{tr}|z_i - z_{good,i}| \leq \mathop{\mathbb{E}}_{pk,y,\gamma} \left[\left\| |\psi_{1^{i-1}0,\gamma}\rangle \right\|^2 + 2 \left\| |\psi_{1^{i-1}0,\gamma}\rangle \right\| \right] \\ + O\left(\frac{m^2}{\sqrt{T}} + m\sqrt{(m-1)\gamma_0} \right).\end{aligned}$$

Proof (Lemma 4). We take expectation of Equation (4.8) over γ

$$|\psi\rangle = \mathop{\mathbb{E}}_{\gamma} \left[\sum_{j=0}^{i-1} |\psi_{1^j 0,\gamma}\rangle + |\psi_{1^i,\gamma}\rangle + \sum_{j=1}^{i} |\psi_{err,j,\gamma}\rangle \right],$$

and expand z_i from Equation (4.9) as

$$\begin{aligned}z_i = z_{good,i} + \mathop{\mathbb{E}}_{pk,y,\gamma} \sum_z \Bigg[& \sum_{k=0}^{i-1} \langle\psi_{1^k 0,\gamma}| U^\dagger P_{acc,-i,z} U \sum_{j=0}^{i-1} |\psi_{1^j 0,\gamma}\rangle \\ & + \sum_{k=0}^{i-1} \langle\psi_{1^k 0,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{1^i,\gamma}\rangle + \sum_{k=0}^{i-1} \langle\psi_{1^k 0,\gamma}| U^\dagger P_{acc,-i,z} U \sum_{j=1}^{i} |\psi_{err,j,\gamma}\rangle \\ & + \langle\psi_{1^i,\gamma}| U^\dagger P_{acc,-i,z} U \sum_{j=0}^{i-1} |\psi_{1^j 0,\gamma}\rangle + \langle\psi_{1^i,\gamma}| U^\dagger P_{acc,-i,z} U \sum_{j=1}^{i} |\psi_{err,j,\gamma}\rangle \\ & + \sum_{k=1}^{i} \langle\psi_{err,k,\gamma}| U^\dagger P_{acc,-i,z} U \sum_{j=0}^{i-1} |\psi_{1^j 0,\gamma}\rangle + \sum_{k=1}^{i} \langle\psi_{err,k,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{1^i,\gamma}\rangle \\ & + \sum_{k=1}^{i} \langle\psi_{err,k,\gamma}| U^\dagger P_{acc,-i,z} U \sum_{j=1}^{i} |\psi_{err,j,\gamma}\rangle \Bigg] |z\rangle\langle z|,\end{aligned}$$

where we omitted writing out e_i. Therefore we have

$$\begin{aligned}\operatorname{tr}|z_i - z_{good,i}| \leq \mathop{\mathbb{E}}_{pk,y,\gamma} \sum_z \Bigg[&\sum_{k=0}^{i-1}\sum_{j=0}^{i-1} \left|\langle\psi_{1^k 0,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{1^j 0,\gamma}\rangle\right| \\ &+ 2\sum_{k=0}^{i-1} \left|\langle\psi_{1^k 0,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{1^i,\gamma}\rangle\right| + 2\sum_{k=0}^{i-1}\sum_{j=1}^{i} \left|\langle\psi_{1^k 0,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{err,j,\gamma}\rangle\right| \\ &+ 2\sum_{j=1}^{i} \left|\langle\psi_{1^i,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{err,j,\gamma}\rangle\right| + \sum_{k=1}^{i}\sum_{j=1}^{i} \left|\langle\psi_{err,k,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{err,j,\gamma}\rangle\right| \Bigg]\end{aligned}$$

by the triangle inequality. The last three error terms sum to $O\left(\frac{m^2}{\sqrt{T}}\right)$ by Corollary 2 and property 1 of Lemma 2. As for the first two terms, by Lemma 3 and Corollary 1, we see that

$$\begin{aligned}&\sum_z \sum_{k=0}^{i-1}\sum_{j=0}^{i-1} |\langle\psi_{1^k 0,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{1^j 0,\gamma}\rangle| \\ &\qquad \leq \sum_z |\langle\psi_{1^{i-1} 0,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{1^{i-1} 0,\gamma}\rangle| + O\left(m^2(m-1)\gamma_0\right) \\ &\qquad \leq \left\||\psi_{1^{i-1} 0,\gamma}\rangle\right\|^2 + O\left(m^2(m-1)\gamma_0\right)\end{aligned}$$

and similarly

$$\begin{aligned}&\sum_z \sum_{k=0}^{i-1} |\langle\psi_{1^k 0,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{1^i,\gamma}\rangle| \\ &\qquad \leq \sum_z |\langle\psi_{1^{i-1} 0,\gamma}| U^\dagger P_{acc,-i,z} U |\psi_{1^i,\gamma}\rangle| + O\left(m\sqrt{(m-1)\gamma_0}\right) \\ &\qquad \leq \left\||\psi_{1^i,\gamma}\rangle\right\| + O\left(m\sqrt{(m-1)\gamma_0}\right).\end{aligned}$$

Now let z_{true}, as a mixed state, be the correct sample of the $\mathsf{SampBQP}$ instance D_x, and let $z_{ideal,i} = \operatorname{tr}(z_{good,i}) z_{true}$. We show that $z_{ideal,i}$ is close to $z_{good,i}$.

Lemma 5. *$z_{good,i}$ is $O(\epsilon)$-computationally indistinguishable to $z_{ideal,i}$, where $\epsilon \in \mathbb{R}$ is the accuracy parameter picked earlier in Π_{Final}.*

Proof (Lemma 5). For every $i \in [m]$ and every prover strategy (U_0, U) for Π_{Final}, consider the following composite strategy, $\Pi_{\mathsf{comp,i}}$, as a prover for Π_{int}. Note that a prover only interacts with the verifier in Step 3 of Π_{int} where Π_{Measure} is run, so we describe a prover's action in terms of the four rounds of communication in Π_{Measure}.

$\Pi_{\mathsf{comp,i}}$ tries to run U_0 by taking the verifier's input as the input to the i-th copy of Π_{Measure} in Π_{Final} and simulating other $m-1$ copies by himself. The prover then picks a uniformly random γ and tries to generate $|\psi_{1^{i-1}1,\gamma}\rangle$ by applying $G_{i,1,\gamma}G_{i-1,1,\gamma}\cdots G_{2,1,\gamma}G_{1,1,\gamma}$. This can be efficiently done because of Lemma 2 and our choice of γ_0 and T in Theorem 7. If the prover fails to generate $|\psi_{1^{i-1}1,\gamma}\rangle$, he throws out everything and aborts by sending $\perp$ back. On the fourth round, If it's a testing round the prover reply with the i-th register of $\mathsf{Ext}_i\left(\frac{|e_j\rangle_{\mathbf{C}}|\psi_{1,i,\gamma}\rangle_{\mathbf{X},\mathbf{Z}}}{\||\psi_1\rangle_{\mathbf{X},\mathbf{Z}}\|}\right)$, where Ext_i is specified in property 3 of Lemma 2. If it's the Hadamard round the prover runs U and checks whether every copy except the i-th copy would be accepted. If all $m-1$ copies are accepted, he replies with the i-th copy, otherwise reply $\perp$.

Denote the result we would get in the Hadamard round by $z_{composite,i}$. By construction, when $G_{i,1,\gamma}\ldots G_{1,1,\gamma}$ succeeded, the corresponding output would be $z_{good,i}$. Also note that this is the only case where the verifier won't reject, so $z_{composite,i}=z_{good,i}$.

In the testing round, by property 3 of Lemma 2, the above strategy is accepted with probability $1-\mathrm{negl}(n)$ when the prover didn't abort. Since the prover's strategy is also efficient, by Theorem 6, $z_{composite,i}$ is $O(\epsilon)$-computationally indistinguishable to $z_{ideal,i}$.

Now we try to put together all $i\in[m]$. First let

$$z=\frac{1}{m}\sum_i z_i=\frac{1}{m}\sum_i\sum_z |z\rangle\langle z|\cdot\langle e_i,\psi|U^\dagger P_{acc,-i,z}U|e_i,\psi\rangle,$$

which is the output distribution of Π_{Final}. We also define the following accordingly:

$$z_{good}:=\frac{1}{m}\sum_i z_{good,i},$$

$$z_{ideal}:=\frac{1}{m}\sum_i z_{ideal,i}.$$

Notice that z_{ideal} is some ideal output distribution, which might not have the same accept probability as z.

Theorem 7. *Under the QLWE assumption, Π_{Final} is a protocol for the* SampBQP *problem $(D_x)_{x\in\{0,1\}^*}$ with negligible completeness error and is computationally sound.*[22]

Proof. Completeness is trivial. In the following we prove the soundness.

By Property 4 of Lemma 2, we have

$$\begin{aligned}\||\psi\rangle\|^2 &\geq \||\psi_{0,\gamma}\rangle\|^2+\||\psi_{1,\gamma}\rangle\|^2\\ &\geq \||\psi_{0,\gamma}\rangle\|^2+\||\psi_{10,\gamma}\rangle\|^2+\||\psi_{11,\gamma}\rangle\|^2\\ &\geq \||\psi_{0,\gamma}\rangle\|^2+\||\psi_{10,\gamma}\rangle\|^2+\cdots+\left\||\psi_{1^{m-1}0,\gamma}\rangle\right\|^2+\||\psi_{1^m,\gamma}\rangle\|^2. \qquad (4.11)\end{aligned}$$

[22] The soundness and completeness of a SampBQP protocol is defined in Definition 3

We have

$$\begin{aligned}
\operatorname{tr}|z - z_{good}| &= \operatorname{tr}\left|\frac{1}{m}\sum_i (z_i - z_{good,i})\right| \leq \frac{1}{m}\sum_i \operatorname{tr}|(z_i - z_{good,i})| \\
&\leq \frac{1}{m}\sum_i \left[\mathop{\mathbb{E}}_{pk,y,\gamma}\left[\left\||\psi_{1^{i-1}0,\gamma}\rangle\right\|^2 + 2\left\||\psi_{1^{i-1}0,\gamma}\rangle\right\|\right]\right. \\
&\quad \left. + O\left(\frac{m^2}{\sqrt{T}} + m\sqrt{(m-1)\gamma_0}\right)\right] \\
&\leq \frac{1}{m} + 2\frac{1}{\sqrt{m}} + O\left(\frac{m^2}{\sqrt{T}} + m\sqrt{(m-1)\gamma_0}\right) \\
&= O\left(\frac{1}{\sqrt{m}} + \frac{m^2}{\sqrt{T}} + m\sqrt{(m-1)\gamma_0}\right), \qquad (4.12)
\end{aligned}$$

where we used triangle inequality on the first inequality, Lemma 4 on the next one, Equation 4.11 and Cauchy's inequality on the last one. Set $m = O(1/\epsilon^2), T = O(1/\epsilon^2), \gamma_0 = \epsilon^8$. Combining Lemma 5 and Equation (4.12) by triangle inequality, we have z is $O(\epsilon)$-computationally indistinguishable to z_{ideal}. Therefore, (d, z) $O(\epsilon)$-computationally indistinguishable to (d, z_{ideal}).

Theorem 1 follows as a corollary.

5 Generic Blindness Protocol Compiler for QPIP_0

In this section, we present a generic protocol compiler that compiles any QPIP_0 protocol $\Pi = (P, V)$ (with an arbitrary number of rounds) to a protocol $\Pi_{\mathsf{blind}} = (P_{\mathsf{blind}}, V_{\mathsf{blind}})$ that achieve blindness while preserving the completeness, soundness, and round complexity. At a high-level, the idea is simple: we simply run the original protocol under a quantum homomorphic encryption QHE with the verifier's key. Intuitively, this allows the prover to compute his next message under encryption without learning the underlying verifier's message, and hence achieves blindness while preserving the properties of the original protocol.

However, several issues need to be taking care to make the idea work. First, since the verifier is classical, we need the quantum homomorphic encryption scheme QHE to be *classical friendly* as defined in Definition **??**. Namely, the key generation algorithm and the encryption algorithm for classical messages should be classical, and when the underlying message is classical, the ciphertext (potentially from homomorphic evaluation) and the decryption algorithm should be classical as well. Fortunately, the quantum homomorphic encryption scheme of Mahadev [29] and Brakerski [12] are classical friendly. Moreover, Brakerski's scheme requires a weaker QLWE assumption, where the modulus is polynomial instead of super-polynomial.

A more subtle issue is to preserve the soundness. Intuitively, the soundness holds since the execution of Π_{blind} simulates the execution of Π, and hence the soundness of Π implies the soundness of Π_{blind}. However, to see the subtle issue,

let us consider the following naive compiler that uses a single key: In Π_{blind}, the verifier V initially generates a pair QHE key (pk, sk), sends pk and encrypted input $\mathsf{QHE.Enc}(pk, x)$ to P. Then they run Π under encryption with this key, where both of them use homomorphic evaluation to compute their next message.

There are two reasons that the compiled protocol Π_{blind} may not be sound (or even not blind). First, in general, the QHE scheme may not have *circuit privacy*; namely, the homomorphic evaluation may leak information about the circuit being evaluated. Since the verifier computes his next message using homomorphic evaluation, a cheating prover P^*_{blind} seeing the homomorphically evaluated ciphertext of the verifier's message may learn information about the verifier's next message circuit, which may contain information about the secret input x or help P^*_{blind} to break the soundness. Second, P^*_{blind} may send invalid ciphertexts to V, so the execution of Π_{blind} may not simulate a valid execution of Π.

To resolve the issue, we let the verifier switch to a fresh new key for each round of the protocol.[23] For example, when the prover P_{blind} returns the ciphertext of his first message, the verifier V_{blind} decrypts the ciphertext, computes his next message (in the clear), and then encrypt it using a fresh key pk' and sends it to P_{blind}. Note that a fresh key pair is necessary here to ensure blindness, as decrypting uses information from the secret key. Since the verifier V_{blind} only sends fresh ciphertexts to P_{blind}, this avoids the issue of circuit privacy. Additionally, to allow P_{blind} to homomorphically evaluate its next message, V_{blind} needs to encrypt the previous secret key sk under the new public key pk' and send it along with pk' to P_{blind}. This allows the prover to homomorphically convert ciphertexts under key pk to ciphertexts under key pk'. By doing so, we show that for any cheating prover P^*_{blind}, the interaction $(P^*_{\mathsf{blind}}, V_{\mathsf{blind}})$ indeed simulates a valid interaction of (P^*, V) for some cheating P^*, and hence the soundness of Π implies the soundness of the compiled protocol. Finally, for the issue of the prover sending invalid ciphertexts, we note that this is not an issue if the decryption never fails, which can be achieved by simply let the decryption algorithm output a default dummy message (e.g., 0) when it fails.

We note that the idea of running the protocol under homomorphic encryptions is used in [16] in a classical setting, but for a different purpose of making the protocol "computationally simulatable" in their context.

We proceed to present our compiler. We start by introducing the notation of a QPIP_0 protocol Π as follows.

Protocol 4 QPIP_0 protocol $\Pi = (P, V)(x)$ where only the verifier receives outputs

---

[23] An alternative strategy is to assume circuit privacy of QHE. This seems to require many additional properties such as *malicious* circuit privacy with efficient simulation and extraction when $\mathsf{QHE.Keygen}$ is honest and secret key is available, multi-hop evaluation, and classical $\mathsf{QHE.Eval}$ on classical ciphertexts and circuits. While existing constructions such as [14] achieves some of these properties, we are unsure if any construction satisfies all of these requirements.

Common inputs[24]:

- Security parameter 1^λ where $\lambda \in \mathbb{N}$
- A classical input $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$

Protocol:

1. V generates $(v_1, st_{V,1}) \leftarrow \mathcal{V}_1(1^\lambda, x)$ and sends v_1 to the prover.
2. P generates $(p_1, st_{P,1}) \leftarrow \mathcal{P}_1(1^\lambda, v_1, x)$ and sends p_1 to the verifier.
3. for $t = 2, \ldots, T$:
   (a) V generates $(v_t, st_{V,t}) \leftarrow \mathcal{V}_t(p_{t-1}, st_{V,t-1})$ and sends v_t to the prover.
   (b) P generates $(p_t, st_{P,t}) \leftarrow \mathcal{P}_t(v_t, st_{P,t-1})$ and sends p_t to the verifier.
4. V computes its output $o \leftarrow \mathcal{V}_{out}(p_T, st_{V,T})$.

We compile the above protocol to achieve blindness as follows. For notation, when there are many sets of QHE keys in play at the same time, we use $\widehat{x}^{(i)}$ to denote x encrypted under pk_i.

Protocol 5 Blind QPIP_0 protocol $\Pi_{\mathsf{blind}} = (P_{\mathsf{blind}}, V_{\mathsf{blind}}(x))$ corresponding to Π_0

Inputs:

- Common input: Security parameter 1^λ where $\lambda \in \mathbb{N}$
- Verifier's input: $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$

Ingredients:

- Let L be the maximum circuit depth of $\mathcal{P}_t$.

Protocol:

1. V_{blind} generates $(v_1, st_{V,1}) \leftarrow \mathcal{V}_1(1^\lambda, x)$. Then it generates $(pk_1, sk_1) \leftarrow \mathsf{QHE.Keygen}(1^\lambda, 1^L)$, and encrypts $\widehat{x}^{(1)} \leftarrow \mathsf{QHE.Enc}(pk_1, x)$ and $\widehat{v}_1^{(1)} \leftarrow \mathsf{QHE.Enc}(pk_1, v_1)$. It sends pk_1, $\widehat{x}^{(1)}$, and $\widehat{v}_1^{(1)}$ to the prover.
2. P_{blind} generates $(\widehat{p}_1^{(1)}, \widehat{st}_{P,1}^{(1)}) \leftarrow \mathcal{P}_{\mathsf{blind},1}(1^\lambda, \widehat{v}_1^{(1)}, \widehat{x}^{(1)})$ by evaluating $(\widehat{p}_1^{(1)}, \widehat{st}_{P,1}^{(1)}) \leftarrow \mathsf{QHE.Eval}(pk, \mathcal{P}_1, \ \mathsf{QHE.Enc}(pk_1, 1^\lambda), \widehat{v}_1^{(1)}, \widehat{x}^{(1)})$. It sends $\widehat{p}_1^{(1)}$ to the verifier.
3. for $t = 2, \ldots, T$:
   (a) V_{blind} decrypts the prover's last message by $p_{t-1} \leftarrow \mathsf{QHE.Dec}(sk_{t-1}, \widehat{p}_{t-1}^{(t-1)})$, then generates $(v_t, st_{V,t}) \leftarrow \mathcal{V}_t(p_{t-1}, st_{V,t-1})$. Then it generates $(pk_t, sk_t) \leftarrow \mathsf{QHE.Keygen}(1^\lambda, 1^L)$, and produces encryptions $\widehat{v}_t^{(t)} \leftarrow \mathsf{QHE.Enc}(pk_t, v_t)$ and $\widehat{sk}_{t-1}^{(t)} \leftarrow \mathsf{QHE.Enc}(pk_t, sk_{t-1})$. It sends pk_t, $\widehat{v}_t^{(t)}$, and $\widehat{sk}_{t-1}^{(t)}$ to the prover.

[24] For the sake of simplicity, we omit accuracy parameter ϵ where it exists

(b) P_{blind} generates $(\widehat{p}_t^{(t)}, \widehat{st}_{P,t}^{(t)}) \leftarrow \mathcal{P}_{\mathsf{blind},t}(\widehat{v}_t^{(t)}, \widehat{sk}_{t-1}^{(t)}, \widehat{st}_{P,t-1}^{(t-1)})$ by first switching its encryption key; that is, it encrypts its state under the new key by $\widehat{st}_{P,t-1}^{(t-1,t)} \leftarrow \mathsf{QHE.Enc}(pk_t, \widehat{st}_{P,t-1}^{(t-1)})$), then homomorphically decrypts the old encryption by $\widehat{st}_{P,t-1}^{(t)} \leftarrow \mathsf{QHE.Eval}(pk_t, \mathsf{QHE.Dec}, \widehat{sk}_{t-1}^{(t)}, \widehat{st}_{P,t-1}^{(t-1,t)})$. Then it applies the next-message function homomorphically, generating $(\widehat{p}_t^{(t)}, \widehat{st}_{P,t}^{(t)}) \leftarrow \mathsf{QHE.Eval}(pk_t, \mathcal{P}_t, \widehat{v}_t^{(t)}, \widehat{st}_{P,t-1}^{(t)})$. It sends $\widehat{p}_t^{(t)}$ back to the verifier.

4. V_{blind} decrypts the prover's final message by $p_T \leftarrow \mathsf{QHE.Dec}(sk_T, \widehat{p}_T^{(T)})$. It then computes its output $o \leftarrow \mathcal{V}_{out}(p_T, st_{V,T})$.

By the correctness of QHE, the completeness error of Π_{blind} is negligibly close to that of Π. In particular, note that the level parameter L is sufficient for the honest prover which has a bounded complexity. For the soundness property, we show the following lemma, which implies that Π_{blind} preserves the soundness of Π_0.

Theorem 8. *For all cheating* BQP *provers* P^*_{blind}*, there exists a cheating* BQP *prover* P^* *s.t. for all* λ *and inputs* $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$*, the output distributions of* $(P^*_{\mathsf{blind}}, V_{\mathsf{blind}}(x))$ *and* $(P^*, V)(x)$ *are identical.*

Proof. We define P^* as follows.

For the first rounds, it generates $(pk_1, sk_1) \leftarrow \mathsf{QHE.Keygen}(1^\lambda, 1^L)$, then produces the encryptions $\widehat{x}^{(1)} \leftarrow \mathsf{QHE.Enc}(pk_1, x)$ and $\widehat{v}_1^{(1)} \leftarrow \mathsf{QHE.Enc}(pk_1, v_1)$. It then runs $(\widehat{p}_1^{(1)}, \widehat{st}_{P,1}^{(1)}) \leftarrow \mathcal{P}_{\mathsf{blind},1}(1^\lambda, \widehat{v}_1^{(1)}, \widehat{x}^{(1)})$. Finally, it decrypts $p_1 \leftarrow \mathsf{QHE.Dec}(sk_1, \widehat{p}_1^{(1)})$ and sends it back to the verifier, and keeps $\widehat{st}_{P,1}^{(1)}$ and sk_1.

For the other rounds, it generates $(pk_t, sk_t) \leftarrow \mathsf{QHE.Keygen}(1^\lambda, 1^L)$, and produces ciphertexts $\widehat{v}_t^{(t)} \leftarrow \mathsf{QHE.Enc}(pk_t, v_t))$ and $\widehat{sk}_{t-1}^{(t)} \leftarrow \mathsf{QHE.Enc}(pk_t, sk_{t-1})$. It then runs $(\widehat{p}_t^{(t)}, \widehat{st}_{P,t}^{(t)}) \leftarrow \mathcal{P}_{\mathsf{blind},t}(\widehat{v}_t^{(t)}, \widehat{sk}_{t-1}^{(t)}, \widehat{st}_{P,t-1}^{(t-1)})$. Finally, it decrypts $p_t \leftarrow \mathsf{QHE.Dec}(sk_t, \widehat{p}_t^{(1)})$ and sends it back to the verifier, and keeps $\widehat{st}_{P,t}^{(t)}$ and sk_t.

By construction, the experiments $(P^*_{\mathsf{blind}}, V_{\mathsf{blind}}(x))$ and $(P^*, V)(x)$ are identical.

Finally, we show the blindness of Π_{blind} through a standard hybrid argument where the sk_i's are "erased" one by one, starting from sk_T. Once sk_1 is eventually erased, $\mathsf{QHE.Enc}(pk_1, x)$ and $\mathsf{QHE.Enc}(pk_1, 0)$ become indistinguishable due to the IND-CPA security of QHE.Enc. We now fill in the details.

Theorem 9. *Under the QLWE assumption with polynomial modulus,* Π_{blind} *is blind.*

Proof. We show that for all cheating BQP provers P^*, $\lambda \in \mathbb{N}$, $x \in \{0,1\}^n$, P^* cannot distinguish $(P^*, V_{\mathsf{blind}}(x))(1^\lambda)$ from $(P^*, V_{\mathsf{blind}}(0^n))(1^\lambda)$ with noticeable probability in λ. We use a hybrid argument; let $\mathrm{Hyb}^x_{T+1} = (P^*, V_{\mathsf{blind}}(x))(1^\lambda)$

and $\mathrm{Hyb}^0_{T+1} = (P^*, V_{\mathsf{blind}}(0^n))(1^\lambda)$. For $2 \le t < T+1$, define Hyb^x_t to be the same as Hyb^x_{t+1}, except when V_{blind} should send $\widehat{v}^{(t)}_t$ and $\widehat{sk}^{(t)}_{t-1}$, it instead sends encryptions of 0 under pk_t. We define Hyb^x_1 to be the same as Hyb^x_2 except the verifier sends encryptions of 0 under pk_1 in place of $\widehat{x}^{(1)}$ and $\widehat{v}^{(1)}_1$. We define Hyb^0_t similarly. Note that Hyb^x_1 and Hyb^0_1 are identical.

For all t, from the perspective of the prover, as it receives no information on sk_t, Hyb^x_{t+1} is computationally indistinguishable from Hyb^x_t due to the CPA security of QHE under pk_t. By a standard hybrid argument, we observe that Hyb^x_1 is computationally indistinguishable with Hyb^x_{T+1}. We use the same argument for the computational indistinguishability between Hyb^0_1 and Hyb^0_{T+1}. We conclude that P^* cannot distinguish between Hyb^x_{T+1} and Hyb^0_{T+1}, therefore Π_{blind} is blind.

Applying our compiler to the parallel repetition of Mahadev's protocol for BQP from [7, 15] and our QPIP_0 protocol Π_{Final} from Protocol 3 for $\mathsf{SampBQP}$ yields the first constant-round blind QPIP_0 protocol for BQP and $\mathsf{SampBQP}$, respectively.

Theorem 10. *Under the QLWE assumption, there exists a blind, four-message* QPIP_0 *protocol for all languages in* BQP *with negligible completeness and soundness errors.*

Theorem 11. *Under the QLWE assumption, there exists a blind, four-message* QPIP_0 *protocol for all sampling problems in* $\mathsf{SampBQP}$ *with negligible completeness error and computational soundness.*

Acknowledgments The authors would like to thank Tomoyuki Morimae for his valuable feedback that helped improve the paper and for pointing out the related works [26, 38]. We are also thankful to anonymous reviewers for various useful comments.

Kai-Min Chung is partially supported by the 2019 Academia Sinica Career Development Award under Grant no. 23-17, and MOST QC project under Grant no. MOST 108-2627-E-002-001. This work was done while Yi Lee was affiliated to Academia Sinica and to National Taiwan University. Part of this work was done while Han-Hsuan Lin was supported by Scott Aaronson's Vannevar Bush Faculty Fellowship from the US Department of Defense. Partially funded by MOST Grant no. 110-2222-E-007-002-MY3. Xiaodi Wu is partially supported by the U.S. National Science Foundation grant CCF-1755800, CCF-1816695, and CCF-1942837 (CAREER).

References

1. Scott Aaronson. The aaronson $25.00 prize. *http://www.scottaaronson.com/blog/?p=284*.
2. Scott Aaronson. The equivalence of sampling and searching. *Theory of Computing Systems*, 55(2):281–298, 2013.
3. Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 333–342, New York, NY, USA, 2011. Association for Computing Machinery.

4. Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations. *arXiv*, 1704.04487, 2017.
5. Dorit Aharonov, Wim Van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Review*, 50(4):755–787, 2008.
6. G. Alagic, Y. Dulek, C. Schaffner, and F. Speelman. Quantum fully homomorphic encryption with verification. 2017.
7. Gorjan Alagic, Andrew M Childs, Alex B Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In *Theory of Cryptography Conference*, pages 153–180. Springer, 2020.
8. Frank Arute, Kunal Arya, Ryan Babbush, et al., and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.
9. James Bartusek. Secure quantum computation with classical communication. 2021.
10. Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does parallel repetition lower the error in computationally sound protocols? In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 374–383, 1997.
11. Jacob D. Biamonte and Peter J. Love. Realizable hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A*, 78:012352, Jul 2008.
12. Zvika Brakerski. Quantum fhe (almost) as secure as classical. *Lecture Notes in Computer Science Advances in Cryptology – CRYPTO 2018*, page 67–95, 2018.
13. A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 517–526, 2009.
14. Orestis Chardouvelis, Nico Döttling, and Giulio Malavolta. Rate-1 quantum fully homomorphic encryption. In *Theory of Cryptography Conference*, pages 149–176. Springer, 2021.
15. Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In *Theory of Cryptography Conference*, pages 181–206. Springer, 2020.
16. Kai-Min Chung. *Efficient Parallel Repetition Theorems with Applications to Security Amplification*. PhD thesis, Harvard University, 2011.
17. Kai-Min Chung, Yi Lee, Han-Hsuan Lin, and Xiaodi Wu. Constant-round blind classical verification of quantum sampling, 2021.
18. Andrea Coladangelo, Alex B. Grilo, Stacey Jeffery, and Thomas Vidick. Verifier-on-a-leash: New schemes for verifiable delegated quantum computation, with quasilinear resources. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 247–277, Cham, 2019. Springer International Publishing.
19. Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing - STOC 02*. ACM Press, 2002.
20. Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *Annual Cryptology Conference*, pages 794–811. Springer, 2012.
21. Joseph F. Fitzsimons, Michal Hajdušek, and Tomoyuki Morimae. Post hoc verification of quantum computation. *Phys. Rev. Lett.*, 120:040501, Jan 2018.
22. Joseph F. Fitzsimons and Elham Kashefi. Unconditionally verifiable blind quantum computation. *Phys. Rev. A*, 96:012303, Jul 2017.

23. Alexandru Gheorghiu, Elham Kashefi, and Petros Wallden. Robustness and device independence of verifiable blind quantum computing. *New Journal of Physics*, 17(8):083040, aug 2015.
24. Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. In *FOCS*, pages 1024–1033, 2019.
25. Michal Hajdušek, Carlos A. Pérez-Delgado, and Joseph F. Fitzsimons. Device-Independent Verifiable Blind Quantum Computation. *arXiv e-prints*, page arXiv:1502.02563, February 2015.
26. Masahito Hayashi and Tomoyuki Morimae. Verifiable measurement-only blind quantum computing with stabilizer testing. *Physical review letters*, 115(22):220502, 2015.
27. Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.
28. A.Y. Kitaev, A. Shen, and M.N. Vyalyi. *Classical and Quantum Computation*. Graduate studies in mathematics. American Mathematical Society, 2002.
29. Urmila Mahadev. Classical homomorphic encryption for quantum circuits. *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, 2018.
30. Urmila Mahadev. Classical verification of quantum computations. *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, 2018.
31. Chris Marriott and John Watrous. Quantum arthur–merlin games. *computational complexity*, 14(2):122–152, 2005.
32. Tomoyuki Morimae, Daniel Nagaj, and Norbert Schuch. Quantum proofs can be verified using only single-qubit measurements. *Phys. Rev. A*, 93:022326, Feb 2016.
33. Krzysztof Pietrzak and Douglas Wikström. Parallel repetition of computationally sound protocols revisited. *J. Cryptol.*, 25(1):116–135, 2012.
34. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.
35. Ben W Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems. *Nature*, 496(7746):456, 2013.
36. Dan Shepherd and Michael J. Bremner. Temporally unstructured quantum computation. In *Proc. R. Soc. A.*, volume 465, pages 1413–1439, 2009.
37. P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.
38. Yuki Takeuchi and Tomoyuki Morimae. Verification of many-qubit states. *Physical Review X*, 8(2):021060, 2018.