Electronic Colloquium on Computational Complexity, Report No. 107 (2023)

Quantum Logspace Computations are Verifiable

Uma Girish* Ran Raz† Wei Zhan‡

Abstract

In this note, we observe that quantum logspace computations are verifiable by classical logspace algorithms, with unconditional security. More precisely, every language in BQL has an (information-theoretically secure) streaming proof with a quantum logspace prover and a classical logspace verifier. The prover provides a polynomial-length proof that is streamed to the verifier. The verifier has a read-once one-way access to that proof and is able to verify that the computation was performed correctly. That is, if the input is in the language and the prover is honest, the verifier accepts with high probability, and, if the input is not in the language, the verifier rejects with high probability even if the prover is adversarial. Moreover, the verifier uses only $O(\log n)$ random bits.

1 Introduction

The problem of how to classically verify that a quantum computation was performed correctly, first suggested by Gottesman in 2004, has been studied in numerous recent works (see for example [BFK09, RUV12, FK17, ABEM17, Mah22, CBGJV19, CCY20, ACGH20, BKLMMVVY22]). Mahadev's breakthrough work presented the first protocol for classical verification of quantum computations [Mah22]. Her protocol is only secure against computationally bounded adversarial provers, under cryptographic assumptions. In this note we observe that for quantum logspace computations, there is a simple verification protocol, with a classical logspace verifier, such that the protocol is secure against adversarial provers with unlimited computational power. Moreover, the protocol is non-interactive. Our proof is similar to our recent proof that shows that randomized logspace computations are verifiable using only $O(\log n)$ random bits [GRZ23].

*Princeton University. E-mail: ugirish@cs.princeton.edu. Research supported by a Simons Investigator Award, by the National Science Foundation grants No. CCF-1714779, CCF-2007462 and by the IBM Phd Fellowship.

†Princeton University. E-mail: ranr@cs.princeton.edu. Research supported by a Simons Investigator Award and by the National Science Foundation grants No. CCF-1714779, CCF-2007462.

‡Princeton University. E-mail: weizhan@cs.princeton.edu. Research supported by a Simons Investigator Award and by the National Science Foundation grants No. CCF-1714779, CCF-2007462.

ISSN 1433-8092

1.1 Streaming Proofs

A streaming proof consists of a pair of (classical or quantum) randomized algorithms, a prover and a verifier, which share a common stream tape. In our work, the prover is a quantum logspace machine and the verifier is a classical randomized logspace machine. The prover doesn't have a separate output tape, instead, it has write-once access to the proof tape onto which it writes a classical bit string Π. The verifier has read-once access to the proof tape from which it can read Π. Both the verifier and the prover have read-many access to the input $x \in \{0,1\}^*$. We allow the prover and verifier to output a special symbol $\perp$. Upon outputting this symbol, the algorithm stops all further processing and we say that the algorithm aborts.

Definition 1.1 (Logspace Streaming Proofs)**.** *Let $\mathcal{F} = \{f_n : \{0,1\}^n \to \{0,1\}\}_{n\in\mathbb{N}}$ be a family of functions. Let $P : \mathbb{N} \to \mathbb{N}$ be a monotone computable function. We say that $\mathcal{F}$ has a logspace streaming proof of length P if there is a (possibly quantum) logspace prover $\mathcal{P}$ and a classical randomized logspace verifier $\mathcal{V}$, that uses a random string R, such that on input $x \in \mathrm{supp}(f_n)$,*

1. *The honest prover $\mathcal{P}$, with at least $\frac{3}{4}$ probability, outputs a (randomized) proof $\Pi \in \{0,1\}^{P(n)}$ such that*
$$\Pr_R[\mathcal{V}(x,\Pi) = f_n(x)] \geq \tfrac{3}{4}$$
(where the probability is over the uniform distribution over R.)

2. *For an arbitrary $\Pi \in \{0,1\}^{P(n)}$ (even adversarially chosen after seeing the input x),*
$$\Pr_R[\mathcal{V}(x,\Pi) \in \{f_n(x), \perp\}] \geq \tfrac{3}{4}$$
(where the probability is over the uniform distribution over R.)

Let $k : \mathbb{N} \to \mathbb{N}$ be a monotone computable function. If the verifier $\mathcal{V}$ never reads more than $k(n)$ random bits from R, we say that the verifier uses at most $k(n)$ random bits.

We sometimes omit the length of the proof and it is understood that P is at most the runtime of the prover, which is polynomial in n.

1.2 Our Main Result

Our main result is as follows.

Proposition 1.2. *A language is in* BQL *if and only if it has a streaming proof between a quantum logspace prover and a classical logspace verifier where the verifier uses $O(\log n)$ random bits.*

2 Preliminaries

Let $n \in \mathbb{N}$. We use $[n]$ to denote $\{1, 2, \ldots, n\}$. Let $v \in \mathbb{R}^n$. For $i \in [n]$ we use v_i to denote the i-th coordinate of v. Let $1 \leq k < \infty$. Let $\|v\|_k := \left(\sum_{i\in[n]} |v_i|^k\right)^{1/k}$ denote the ℓ_k-norm of v. This induces an operator norm on matrices $M \in \mathbb{R}^{n\times n}$ by $\|M\|_k := \max_{v\in\mathbb{R}^n\setminus\{\vec{0}\}} \frac{\|M(v)\|_k}{\|v\|_k}$. This norm is sub-multiplicative, i.e., $\|M \cdot N\|_k \leq \|M\|_k \cdot \|N\|_k$ for all $M, N \in \mathbb{R}^{n\times n}$. Let $\|v\|_\infty = \max_{i\in[n]} |v_i|$ denote the ℓ_∞-norm of v and let $\|M\|_{\max} = \max_{i,j\in[n]} |M_{i,j}|$ (this is not an induced operator norm). We have the following inequalities for all $M \in \mathbb{R}^{n\times n}, v \in \mathbb{R}^n$ and $1 \leq k, k' < \infty$.

$$\|M\|_{\max} \leq \|M\|_k \leq n \cdot \|M\|_{\max}$$

$$k \geq k' \implies \|v\|_k \leq \|v\|_{k'}$$

We use $M[i, j]$ to refer to the $(i, j)^{\text{th}}$ entry of the matrix M.

2.1 Our Model of Computation

In this work, a deterministic Turing machine consists of a read-only input tape, a work tape and a write-once output tape. A randomized Turing Machine has an additional read-once randomness tape consisting of random bits. Let $S, T, R : \mathbb{N} \to \mathbb{N}$ be any monotone computable functions. We typically use S to denote the space complexity and T to denote the time complexity of a Turing Machine. When we say that an event occurs with high probability, we typically mean that it occurs with probability at least $2/3$. An algorithm is said to have bounded error if the probability of error is at most $1/3$. By standard error-reduction techniques, we could choose this number to be any constant in $(0, 1/2)$.

A deterministic (resp. bounded-error randomized) (S, T) algorithm refers to a deterministic (resp. randomized) Turing Machine such that for all $x \in \{0, 1\}^*, |x| = n$, the machine with x on its input tape, uses at most $S(n)$ bits of space on its work tape and runs in at most $T(n)$ time. We say that an algorithm computes a family of functions $\{f_n : \{0, 1\}^n \to \{0, 1\}^*\}_{n\in\mathbb{N}}$ if for all $n \in \mathbb{N}, x \in \{0, 1\}^n$, the output of the algorithm on input x is $f_n(x)$ (with high probability if the algorithm is bounded-error randomized). These functions may be partial, i.e., defined on a strict subset of $\{0, 1\}^n$. The Turing machine is said to use R bits of randomness if on inputs of size $n \in \mathbb{N}$, the machine never reads more than $R(n)$ bits on the randomness tape.

Logspace Computation: A logspace algorithm refers to an $(O(\log(n)), \text{poly}(n))$ algorithm. The (promise) class L refers to all families of single-bit-output functions computable by deterministic logspace algorithms. The (promise) class BPL refers to all families of single-bit-output functions computable by randomized logspace algorithms with high probability. All these classes are inherently promise classes, so for the rest of the paper, we omit this prefix. We use the notation family of functions, languages and problems interchangeably. It is possible to define quantum analogues of the aforementioned complexity classes. In particular, we will be interested in BQL, the set of all families of single-bit-output functions computable by quantum logspace algorithms. (See [FR21] for the formal definition.)

2.2 Unitary Matrix Powering

We consider the following problem.

Definition 2.1 (Unitary Matrix Powering)**.** *The inputs are an $n \times n$ unitary matrix M and a parameter $T \leq \mathrm{poly}(n)$ and an $n \times n$ projection matrix Π onto a subset of standard basis states. The promise on the input is that $\|\Pi M^T(e_1)\|_2^2 \geq 4/5$ or $\|\Pi M^T(e_1)\|_2^2 \leq 1/5$. The goal is to output 1 in the former case and 0 in the latter case.*

Proposition 2.2. *The Unitary Matrix Powering Problem is logspace-complete for* BQL*.*

The proof of this is deferred to the appendix.

3 Classical Logspace Verifiers for Quantum Logspace Computations

In this section, we prove Proposition 1.2 which states that a family of functions is in BQL if and only if it has a logspace streaming proof between a quantum prover and a classical verifier that reads $O(\log n)$ random bits. First, it is clear that any streaming proof between a quantum logspace prover and a classical logspace verifier can be implemented by a BQL algorithm with success probability at least $(3/4)^2 > 1/2$ which can be amplified. It suffices to argue that the Unitary Matrix Powering Problem can be solved by a streaming proof between a quantum logspace prover and a classical logspace verifier, where the verifier uses $O(\log n)$ random bits. Towards this, we define a notion of a δ-good sequence of vectors for a matrix M.

Definition 3.1. *Let M be any $n \times n$ matrix and $T \leq \mathrm{poly}(n)$ be a natural number. Let $v_i = M^i(e_1)$ for all $i \leq T$. Let $\delta \in [0, 1]$. A sequence of vectors $v'_0, v'_1, \ldots, v'_T \in \mathbb{R}^n$ is said to be δ-good for M if for all $i \in [T]$, we have $\|v'_i - v_i\|_2 \leq \delta$ and $v_0 = e_1$.*

We make use of the following claims.

Claim 3.1. *There is a quantum logspace prover which given an $n \times n$ unitary matrix M and parameters $T \leq \mathrm{poly}(n), \delta \geq \frac{1}{\mathrm{poly}(n)}$ as input, outputs a δ-good sequence of vectors for M with probability at least $\frac{3}{4}$.*

Claim 3.2. *Let $\frac{1}{\mathrm{poly}(n)} < \delta \leq \frac{1}{10^4 T^2}$. There is a randomized logspace verifier which given any $n \times n$ unitary matrix M and parameters $T \leq \mathrm{poly}(n), \delta$ as input and read-once access to a stream of vectors $v'_0, \ldots, v'_T \in \mathbb{R}^n$ (where each vector is specified up to $\Theta(\log(n))$ bits of precision), does the following.*

- *If the sequence is δ-good for M, then the probability that the algorithm aborts is at most $1/4$.*
- *If $\|v'_T - v_T\|_2 \geq \frac{1}{5}$, then the algorithm aborts with probability at least $3/4$.*

Furthermore, this algorithm only uses $O(\log(n))$ bits of randomness.

Let us see how to complete the proof using Claim 3.1 and Claim 3.2. Given an $n \times n$ unitary matrix M as input and a parameter $T \leq \text{poly}(n)$, set $\delta = \frac{1}{10^4 T^2}$. Run the prover's algorithm from Claim 3.1 using this value of δ to produce a stream $v'_0, \ldots, v'_T$. Run the verifier's algorithm from Claim 3.2 on this stream to verify. If it doesn't abort, we have the verifier return 1 if $\|\Pi(v'_T)\|_2^2 \geq 0.6$, return 0 if $\|\Pi(v'_T)\|_2^2 \leq 0.46$ and return $\perp$ otherwise. (This computation can be easily done in classical logspace in a streaming fashion.)

Completeness: Claim 3.1 implies that an honest prover outputs a δ-good sequence with probability at least $\frac{3}{4}$. Claim 3.2 implies that an honest proof is aborted with probability at most $\frac{1}{4}$. Since $\|v'_T - v_T\|_2 \leq \delta \ll 1/100$ by assumption and Π is a projection, $\|\Pi(v'_T) - \Pi(v_T)\|_2 \leq 1/100$. Hence, if $\|\Pi(v_T)\|_2^2 \geq 4/5$, then $\|\Pi(v'_T)\|_2^2 \geq \sqrt{4/5} - 1/100 \geq 0.6$ and if $\|\Pi(v_T)\|_2^2 \leq 1/5$ then $\|\Pi(v'_T)\|_2^2 \leq \sqrt{1/5} + 1/100 \leq 0.46$. Thus, the verifier will return the correct answer whenever the sub-routine doesn't abort.

Soundness: Consider the behavior of this verifier on an arbitrary proof. If the verifier makes a mistake and returns the incorrect answer, it must be the case that either $\|\Pi(v_T)\|_2^2 \geq 4/5$ and $\|\Pi(v'_T)\|_2^2 \leq 0.46$ or $\|\Pi(v_T)\|_2^2 \leq 1/5$ and $\|\Pi(v'_T)\|_2^2 \geq 0.6$. In either case, we must have $\|v'_T - v_T\|_2 \geq \min\left(\sqrt{4/5} - \sqrt{0.46}, \sqrt{0.6} - \sqrt{1/5}\right) \geq \frac{1}{5}$. Claim 3.2 implies that such a proof is aborted with probability at least $\frac{3}{4}$. This completes the proof of Proposition 1.2.

We now proceed to prove Claim 3.1.

Proof of Claim 3.1. The prover starts by outputting $v_0 = e_1$. To output the intermediate v_i, we make use of the following result from [GRZ21]. It appears as Corollary 15 and we paraphrase it as follows.

Lemma 3.2. *Given an $n \times n$ matrix M with $\|M\|_2 \leq 1$, a positive integer $i \leq \text{poly}(n)$, two unit vectors $v, w \in \mathbb{R}^n$ and an error parameter $\delta > 0$, there is a quantum algorithm with time $\text{poly}(n/\delta)$ and space $O(\log(n/\delta))$ such that with probability $1 - 2^{-\text{poly}(n/\delta)}$, it outputs $w^\dagger M^i v$ with additive error δ.*

Note that $v_i(j) = e_j^\dagger M^i e_1$. Thus, by repeating the subroutine from Lemma 3.2 $\text{poly}\left(\frac{nT}{\delta}\right)$ times with parameters $w = e_j, v = e_1, i$ and δ/n, a quantum logspace prover can with probability at least $\frac{3}{4}$, estimate each $v_i(j)$ to δ/n additive accuracy for all $i \in [T]$ and $j \in [n]$. In this case, we have, $\|v'_i - v_i\|_2 \leq \|v'_i - v_i\|_\infty \cdot n \leq \delta$. This completes the proof of Claim 3.1. □

We now complete the proof of Claim 3.2

Proof of Claim 3.2. The verifier's algorithm is formally described in **Algorithm 2**. The informal description is as follows. The verifier will try to check that $\widetilde{M}(v'_{i-1})$ is approximately equal to v'_i for all $i \in [T]$. However, to do this in a streaming fashion, the verifier will instead test that a random linear combination of these approximate equations holds. To reduce the randomness from T to $O(\log n)$, instead of using a truly random combination of the equations the verifier uses a pseudorandom combination drawn using a 4-wise independent collection of $\{-1, 1\}$-random variables. This is similar to the ℓ_2-frequency estimation algorithm in [AMS99].

Algorithm 1 Algorithm for Verifier in Claim 3.2

Input : An $n \times n$ unitary matrix M, parameters $T \leq \text{poly}(n)$, $\frac{1}{10^4 T^2} \geq \delta \geq \frac{1}{\text{poly}(n)}$ and read-once access to a stream of vectors $v'_0, \ldots, v'_T \in \mathbb{R}^n$.

Output: If the sequence is δ-good for M, then return $\perp$ with probability at most $\frac{1}{4}$. If $\|v'_T - v_T\|_2 \geq \frac{1}{5}$, return $\perp$ with probability at least $\frac{3}{4}$.

begin

Round down each entry of the input matrix M to $\frac{\delta}{6n^2T}$ additive error to produce a matrix $\widetilde{M}$ so that $\left\|M - \widetilde{M}\right\|_2 \leq \frac{\delta}{6T}$.

Return $\perp$ if $v'_0 \neq e_1$.

for $t = 1$ **to** 11 **do**

Sample $\alpha_{i,j} \in \{-1, 1\}$ for $i \in [T], j \in [n]$ from a collection of 4-wise independent $\{-1, 1\}$-random variables with mean 0.

Compute $\Delta := \sum_{i \in [T], j \in [n]} \alpha_{i,j} \cdot w_{i,j}$ where for $i \in [T], j \in [n]$, we have $w_{i,j} := (\widetilde{M}(v'_{i-1}))(j) - v'_i(j)$.

Return $\perp$ if $|\Delta| > 30T\delta$.

end

end

Time & Space Complexity of this Algorithm: One can sample from a collection of 4-wise independent $\{-1, 1\}$-random variables of size $O(nT)$ in logspace using only $O(\log(nT))$ bits of randomness [AMS99]. Note that the quantity $\Delta \triangleq \sum_{\substack{i \in [T] \\ j \in [n]}} \alpha_{i,j} \cdot \Big((\widetilde{M}(v'_{i-1}))(j) - v'_i(j) \Big)$ can be expressed $\sum_{\substack{i \in \{0,\ldots,T\} \\ j \in [n]}} \beta_{i,j} v'_i(j)$ where $\beta_{i,j}$ are coefficients that depend only on the entries of $\widetilde{M}$ and α, and can be computed in logspace. Thus, a logspace algorithm can read the stream of $v'_i(j)$ for $i = 0, \ldots, T$ and $j \in [n]$ once from left to right and compute $\Delta \triangleq \sum_{i,j} \beta_{i,j} v'_i(j)$ in a streaming fashion. As the entries of the matrices and the vectors are $O(\log(n))$ bits long, the arithmetic can be done in logspace. The time complexity of this process is hence $\text{poly}(n)$ and the space complexity is $O(\log(n))$.

We now move on to the completeness and soundness. First, we make some observations. Let $w \in \mathbb{R}^{nT}$ be defined at $i \in [T], j \in [n]$ by $w_{i,j} \triangleq (\widetilde{M}(v'_{i-1}))(j) - v'_i(j)$. Let $\widetilde{v}_0, \ldots, \widetilde{v}_T$ be defined by $\widetilde{v}_i = \widetilde{M}^i(e_1)$ for all $i \in [T] \cup \{0\}$. Since $\left\|\widetilde{M} - M\right\|_2 \leq \frac{1}{6\delta T}$ and $\|M\|_2 \leq 1$,

$$\text{for all } i \in [T], \left\|\widetilde{M}^i - M^i\right\|_2 \leq \left(1 + \frac{\delta}{6T}\right)^i - 1 \leq \frac{\delta}{2}. \tag{1}$$

(In particular, $\left\|\widetilde{M}^i\right\|_2 \leq 1 + \delta/2$.) Thus,

$$\text{for all } i \in [T], \|\widetilde{v}_i - v_i\|_2 \triangleq \left\|\widetilde{M}^i(e_1) - M^i(e_1)\right\|_2 \leq \left\|\widetilde{M}^i - M^i\right\|_2 \leq \frac{\delta}{2}. \tag{2}$$

Completeness of the Algorithm: Suppose $v'_0, \ldots, v'_T$ is a δ-good sequence, then $\|v'_i - v_i\|_2 \leq \delta$ for all $i \in [T]$ and $v'_0 = e_1$. Since M is a contraction map with respect

to $\|\cdot\|_2$, this along with Equation (1) implies that for all $i \in [T]$,

$$\begin{aligned}\left\|\widetilde{M}(v'_{i-1}) - v'_i\right\|_2 &\leq \left\|\widetilde{M}(v'_{i-1}) - M(v'_{i-1})\right\|_2 + \left\|M(v'_{i-1}) - M(v_{i-1})\right\|_2 \\ &\quad + \|M(v_{i-1}) - v_i\|_2 + \|v_i - v'_i\|_2 \\ &\leq \left\|\widetilde{M} - M\right\|_2 \cdot \|v'_{i-1}\|_2 + \|v_{i-1} - v'_{i-1}\|_2 + \|v_i - v'_i\|_2 \\ &\leq \tfrac{\delta}{6T} \cdot (1+\delta) + \delta + \delta \leq 3\delta.\end{aligned}$$

Thus, $\|w\|_2 \leq 3T\delta$. Consider the quantity $\langle \alpha, w\rangle = \sum_{i,j} \alpha_{i,j} w_{i,j}$ that the algorithm estimates. Note that $\mathbb{E}[\langle \alpha, w\rangle] = 0$ and that $\mathbb{E}[\langle \alpha, w\rangle^2] = \sum_{i,j} w_{i,j}^2$. Chebyshev's Inequality implies that with probability at least 0.99, we have $|\langle \alpha, w\rangle| \leq 30T\delta$. This implies that with probability at least $(0.99)^{11} \geq 0.8$, every iteration of the inner loop in **Algorithm 2** does not reject.

Soundness of the Algorithm: Suppose a dishonest prover produces a stream $v'_0, \ldots, v'_T$ such that $\|v'_T - v_T\|_2 \geq \frac{1}{5}$. The verifier always returns $\perp$ if $v'_0 \neq e_1$, so we may assume that $v'_0 = e_1$. Let $\varepsilon = \frac{1}{20T}$. We argue that for some $i \in [T]$, we must have $\|w_i\|_2 \geq \varepsilon$. Assume by contradiction that $\left\|\widetilde{M}(v'_{i-1}) - v'_i\right\|_2 \leq \varepsilon$ for all $i \in [T]$. Hence, by Triangle Inequality and Equation (1), (and since $\widetilde{v}_0 = e_1$) we have

$$\begin{aligned}\|\widetilde{v}_T - v'_T\|_2 = \left\|\widetilde{M}^T(v'_0) - v'_T\right\|_2 &\leq \sum_i \left\|\widetilde{M}^{T-(i-1)}(v'_{i-1}) - \widetilde{M}^{T-i}(v'_i)\right\|_2 \\ &\leq \sum_i \left\|\widetilde{M}^{T-i}\right\|_2 \cdot \left\|\widetilde{M}(v'_{i-1}) - v'_i\right\|_2 \\ &\leq \sum_i \left(1 + \tfrac{\delta}{2}\right) \cdot \varepsilon \\ &\leq 2T\varepsilon.\end{aligned}$$

Equation (2) implies that $\|\widetilde{v}_T - v_T\|_2 \leq \frac{\delta}{2}$. This implies that $\|v'_T - v_T\|_2 \leq \frac{\delta}{2} + 2T\varepsilon$. We assumed that $\|v_T - v'_T\|_2 \geq \frac{1}{5}$. Hence, it follows that

$$\tfrac{1}{5} \leq \tfrac{\delta}{2} + 2T\varepsilon.$$

Since we chose $\varepsilon = \frac{1}{20T}$ and $\delta \leq 1/10$, this is a contradiction. Thus, we must have $\|w\|_2 \geq \varepsilon$. Note that $\mathbb{E}[\langle \alpha, w\rangle] = 0$ and $\mathbb{E}[\langle \alpha, w\rangle^2] = \|w\|_2^2$. Furthermore,

$$\mathbb{E}\left[\langle \alpha, w\rangle^4\right] = \mathbb{E}\left[\sum_{i,j,k,l} w_i w_j w_k w_l \alpha_i \alpha_j \alpha_k \alpha_l\right] \leq 6 \sum_{i,j} w_i^2 w_j^2 \leq 6\|w\|_2^4$$

Here, we used the fact that the random variables are 4-wise independent. The Paley-Zygmund Inequality implies that

$$\Pr\left[\langle \alpha, w\rangle^2 \geq \frac{1}{10} \cdot \|w\|_2^2\right] \geq \left(1 - \frac{1}{10}\right)^2 \cdot \frac{\left(\mathbb{E}[\langle \alpha, w\rangle^2]\right)^2}{\mathbb{E}[\langle \alpha, w\rangle^4]} \geq \frac{1}{8}.$$

This, along with the fact that $\|w\|_2 \geq \varepsilon$ implies that $\Pr\left[|\langle \alpha, w\rangle| \geq \frac{\varepsilon}{10}\right] \geq \frac{1}{8}$. By repeating this experiment 11 times, we can ensure that with probability at least $1 - (1 - 1/8)^{11} \geq 3/4$, we find at least one instance so that $|\langle \alpha, w\rangle| \geq \frac{\varepsilon}{10}$. Since $\delta \leq \frac{1}{10^4 T^2}$ and $\varepsilon = \frac{1}{20T}$, we have

$$\tfrac{\varepsilon}{10} > 30T\delta$$

Thus, with probability at least 3/4, we have $|\langle \alpha, w \rangle| > 30T\delta$. This implies that the algorithm returns $\perp$ with probability at least 3/4.

□

References

[ABEM17] Dorit Aharonov, Michael Ben-Or, Elad Eban, Urmila Mahadev: Interactive Proofs For Quantum Computations. Arxiv preprint 1704.04487, 2017. 1

[ACGH20] Gorjan Alagic, Andrew Childs, Alex Grilo, Shih-Han Hung: Non-interactive Classical Verification of Quantum Computation. TCC 2020. 1

[AMS99] Noga Alon, Yossi Matias, Mario Szegedy: The Space Complexity of Approximating the Frequency Moments. J. Comput. Syst. Sci. 58(1): 137-147 (1999) 5, 6

[BFK09] Anne Broadbent, Joseph Fitzsimons, Elham Kashefi: Universal Blind Quantum Computation. FOCS 2009. 1

[BKLMMVVY22] James Bartusek, Yael Tauman Kalai, Alex Lombardi, Fermi Ma, Giulio Malavolta, Vinod Vaikuntanathan, Thomas Vidick, Lisa Yang: Succinct Classical Verification of Quantum Computation. CRYPTO 2022. 1

[CBGJV19] Andrea Coladangelo, Alex Bredariol Grilo, Stacey Jeffery, Thomas Vidick: Verifier-on-a-Leash: New Schemes for Verifiable Delegated Quantum Computation, with Quasilinear Resources. EUROCRYPT 2019. 1

[CCY20] Nai-Hui Chia, Kai-Min Chung, Takashi Yamakawa: Classical Verification of Quantum Computations with Efficient Verifier. TCC 2020. 1

[FK17] Joseph Fitzsimons, Elham Kashefi: Unconditionally Verifiable Blind Quantum Computation. Phys. Rev. A, vol. 96 (2017). 1

[FR21] Bill Fefferman, Zachary Remscrim: Eliminating Intermediate Measurements in Space-bounded Quantum Computation. STOC 2021. 3, 9

[GRZ21] Uma Girish, Ran Raz, Wei Zhan: Quantum Logspace Algorithm for Powering Matrices with Bounded Norm. ICALP 2021. 5

[GRZ23] Uma Girish, Ran Raz, Wei Zhan: Is Untrusted Randomness Helpful?. ITCS 2023. 1

[Mah22] Urmila Mahadev: Classical Verification of Quantum Computations. SIAM J. Comput. 51(4): 1172-1229 (2022) 1

[RUV12] Ben Reichardt, Falk Unger, Umesh V. Vazirani: A Classical Leash for a Quantum System. Arxiv preprint arXiv:1209.0448, 2012. 1

4 Appendix

4.1 A BQL-complete Problem

We prove Proposition 2.2 which states that the Unitary Matrix Powering Problem is complete for BQL. As before, it suffices to reduce all BQL problems to this problem.

Consider any $\mathcal{F} = \{f_n : \{0,1\}^n \to \{0,1\}\}_{n\in\mathbb{N}}$ in BQL. As per the definition[1] in [FR21], this means that there exists a logspace-uniform family of quantum circuits $\{Q_n(x)\}_{n\in\mathbb{N}}$, consisting of only unitary operators where $Q_n(x)$ acts on $m = O(\log n)$ qubits with the following property. If the initial state is $|0^m\rangle$ and the first qubit of the final state is measured, then $f_n(x) = 1$ if the outcome is 0 with probability at least $4/5$ and $f_n(x) = 0$ if the outcome is 0 with probability at most $1/5$. Let $T_n(x)$ be the number of operators of the quantum circuit $Q_n(x)$ and m be the number of qubits. Define a unitary matrix $U_n(x)$ in $(T_n(x)+1)\times 2^m$ dimensions as follows. We first partition the rows and columns of $U_n(x)$ into $T_n(x)+1$ parts based on the value of the first $\log(\lceil T_n(x)+1\rceil)$ coordinates. For all $i \in [T_n(x)]$, define the $(i+1,i)$-th block of $U_n(x)$ to be the i-th operator in the circuit $Q_n(x)$. Define the $(1, T_n(x)+1)$-th block of $U_n(x)$ to be the identity matrix. All other blocks of $U_n(x)$ are defined to be zero. Since $T_n(x) \le \text{poly}(n)$ and $m \le O(\log n)$, this is a unitary operator in $\text{poly}(n)$ dimensions. Let $\Pi_n(x)$ be a projection matrix in $(T_n(x)+1)\times 2^m$ dimensions that projects onto the basis states $\{|i,j\rangle \mid i = T_n(x)+1, j \in [2^m], j_1 = 0\}$.

Firstly, each entry of the unitary matrix $U_n(x)$ and the projection matrix $\Pi_n(x)$ can be computed by a deterministic logspace algorithm. Observe that the vector $U_n^i(x)(e_1)$ is supported only on coordinates in $\{i+1\}\times[2^m]$, furthermore, when restricted to these coordinates, this vector precisely captures the state of the qubits in $Q_n(x)$ after applying the first i operators. It follows that the probability that the circuit $Q_n(x)$ outputs 1 is precisely $\|\Pi U^{T_n(x)}(e_1)\|^2$. Thus, given any $x \in \text{supp}(f_n)$, we can produce in deterministic logspace, a unitary matrix $U_n(x)$ and a projection matrix $\Pi_n(x)$ in $\text{poly}(n)$ dimensions and a parameter $T \le \text{poly}(n)$ such that $f_n(x) = 1$ if $\|\Pi U^{T_n(x)}(e_1)\|^2 \ge 4/5$ and $f_n(x) = 0$ if $\|\Pi U^{T_n(x)}(e_1)\|^2 \le 1/5$. This shows that the Unitary Matrix Powering Problem is complete for BQL.

[1] Strictly speaking, this definition is for a unitary variant of BQL, however, in [FR21] it is shown that all problems in BQL are solvable by this unitary variant.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il