Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 116 (2021)

Quantum Meets the Minimum Circuit Size Problem

Nai-Hui Chia*[1], Chi-Ning Chou†[2], Jiayu Zhang‡[3], and Ruizhe Zhang§[4]

[1]Luddy School of Informatics, Computing, and Engineering, Indiana University Bloomington
[2]School of Engineering and Applied Sciences, Harvard University
[3]Department of Computer Science, Boston University
[4]Department of Computer Science, The University of Texas at Austin

Abstract

In this work, we initiate the study of the Minimum Circuit Size Problem (MCSP) in the quantum setting. MCSP is a problem to compute the circuit complexity of Boolean functions. It is a fascinating problem in complexity theory — its hardness is mysterious, and a better understanding of its hardness can have surprising implications to many fields in computer science.

We first define and investigate the basic complexity-theoretic properties of minimum quantum circuit size problems for three natural objects: Boolean functions, unitaries, and quantum states. We show that these problems are not trivially in NP but in QCMA (or have QCMA protocols). Next, we explore the relations between the three quantum MCSPs and their variants. We discover that some reductions that are not known for classical MCSP exist for quantum MCSPs for unitaries and states, e.g., search-to-decision reductions and self-reductions. Finally, we systematically generalize results known for classical MCSP to the quantum setting (including quantum cryptography, quantum learning theory, quantum circuit lower bounds, and quantum fine-grained complexity) and also find new connections to tomography and quantum gravity. Due to the fundamental differences between classical and quantum circuits, most of our results require extra care and reveal properties and phenomena unique to the quantum setting. Our findings could be of interest for future studies, and we post several open problems for further exploration along this direction.

*naichia@iu.edu.
†chiningchou@g.harvard.edu.
‡jyz16@bu.edu.
§ruizhe@utexas.edu.

ISSN 1433-8092

Contents

1 Introduction 3
1.1 The classical MCSP and its connections to other problems 3
1.2 Main results and technical overview 3
1.2.1 Challenges and difficulties when moving to the quantum setting 4
1.2.2 The Hardness of MQCSP and cryptography 5
1.2.3 MQCSP and learning theory 6
1.2.4 MQCSP and quantum circuit lower bounds 7
1.2.5 MQCSP and quantum fine-grained complexity 8
1.2.6 Quantum circuit complexity for states and unitaries 9
1.3 Discussion and open questions 11
1.3.1 Open problems: the complexity of quantum circuits 11
1.3.2 Open problems: potential connections to other areas 13

2 Preliminaries 15
2.1 Quantum states, unitary transformations, and quantum circuits 15
2.2 Quantum complexity classes 16
2.3 Nonuniform quantum circuit complexity classes 17

3 Minimum Quantum Circuit Size Problems 18
3.1 Problem definitions 18
3.2 Upper bounds for MQCSP 21
3.3 Hardness of quantum MCSP 21

4 Connections Between MQCSP and Other Problems 25
4.1 Cryptography and MQCSP 25
4.1.1 Quantum cryptographic primitives 25
4.1.2 Implications for quantum-secure one-way functions (qOWF) 27
4.1.3 Implication for quantum-secure $i\mathcal{O}$ 28
4.2 Learning theory 29
4.3 Circuit lower bounds 30
4.4 Fine-grained complexity 34

5 MCSP for Quantum Objects 37
5.1 Reductions for UMCSP and SMCSP 44
5.1.1 Search-to-decision reductions 45
5.1.2 Self-reduction for SMCSP 47
5.1.3 Reducing MQCSP to UMCSP 49
5.2 Applications of SMCSP and UMCSP 51
5.2.1 Applications of UMCSP 51
5.2.2 Pseudorandom states 52
5.2.3 Estimating the wormhole volume 52
5.2.4 Succinct state tomography 54

6 Acknowledgment 54

References 54

A Proof for the hardness of MQCSP 59

B Learning Theory 61
B.1 PAC learning 61
B.2 Quantum learning 62

C Proofs in Section 4.3 65
C.1 Proof for Theorem 4.19 65
C.2 Proof of Quantum Antichecker Lemma 66
C.3 Quantum Impagliazzo-Wigderson generator 68

D Quantum fine-grained hardness based on QETH 70

E Proofs for Corollary 5.10 72

F Quantum Circuit Class 73

1 Introduction

The Minimum Circuit Size Problem (MCSP) is one of the central computational problems in complexity theory. Given the truth table of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and a size parameter s (in unary) as inputs, MCSP asks whether there exists a circuit of size at most s for f. While MCSP has been studied as early as the 1950s in the Russian cybernetics program [Tra84], its complexity remains mysterious: we do not know whether it is in P or NP-hard. Meanwhile, besides being a natural computational problem, in recent years, researchers have discovered many surprising connections of MCSP to other areas such as cryptography [RR97], learning theory [CIKK16], circuit complexity [KC00], average-case complexity [Hir18], and others.

Quantum computing is of growing interest, with applications to cryptography [Sho94], machine learning [BWP$^+$17], and complexity theory [JNV$^+$20], etc. Inspired by the great success of MCSP in classical computation and the flourishing of quantum computers, we propose a new research program of studying quantum computation through the lens of MCSP. We envision MCSP as a central problem that connects different quantum computation applications and provides deeper insights into the complexity-theoretic foundation of quantum circuits.

1.1 The classical MCSP and its connections to other problems

It is immediate that MCSP $\in$ NP because the input size is 2^n so one can verify if a circuit (given as the certificate/proof) computes the input truth table in time $2^{O(n)}$. However, there is no consensus on the complexity status of this problem – MCSP could be in P, NP-complete, or NP-intermediate. Several works [MW17, KC00] showed negative evidence for proving the NP-hardness of MCSP using standard reduction techniques. We also do not know whether there is an algorithm better than brute force search (see Perebor conjecture for MCSP [Tra84]) or whether there is a search-to-decision reduction or a self-reduction[1] for MCSP[2]. On the other hand, several variants of MCSP are NP-hard under either deterministic reductions [Mas79, HOS18] or randomized reductions [Ila19, ILO20].

Researchers have discovered many surprising connections of MCSP to other fields in Theoretical Computer Science including cryptography, learning theory, and circuit lower bounds. To name a few, Razborov and Rudich [RR97] related natural properties against P/poly with circuit lower bounds and pseudorandomness. Kabanets and Cai [KC00] showed that MCSP $\in$ P implies new circuit lower bounds, and that MCSP $\in$ BPP implies that any one-way function can be inverted. Allender and Das [AD14] related the complexity class SZK (Statistical Zero Knowledge) to MCSP. Carmosino et al. [CIKK16] showed that MCSP $\in$ BPP gives efficient PAC-learning algorithms. Impagliazzo et al. [IKV18] showed that the existence of indistinguishable obfuscation implies that SAT reduces to MCSP under a randomized reduction. Hirahara [Hir18] showed that if an approximation version of MCSP is NP-hard, then the average-case and worst-case hardness of NP are equivalent. Arunachalam et al. [AGG$^+$20] proved that MCSP $\in$ BQP implies new circuit lower bounds. All these results indicate that the MCSP serves as a "hub" that connects many fundamental problems in different fields. Therefore, a deeper understanding of this problem could lead to significant progress in Theoretical Computer Science.

1.2 Main results and technical overview

In this work, we consider three different natural objects that a quantum circuit can compute: Boolean functions, unitaries, and quantum states. We start with giving the informal definitions

[1] Roughly, a problem is self-reducible if one can solve the problem with size n by algorithms for smaller size.

[2] It is worth noting that every NP-complete problem has search-to-decision reductions and self-reductions.

of the minimum circuit size problem for each of them. See Section 3 and Section 5 for the formal definitions.

Definition 1.1 (MQCSP, informal)**.** *Given the truth table of a Boolean function f and a size parameter s in unary, decide if there exists a quantum circuit C which has size at most s and uses at most s ancilla qubits such that C computes f with high probability.*

Definition 1.2 (UMCSP, informal)**.** *Given the full description of a 2^n-dimensional unitary matrix U and a size parameter s in unary, decide if there exists a quantum circuit C which has size at most s and uses at most s ancilla qubits such that C and U are close*[3]*.*

Definition 1.3 (SMCSP, informal)**.** *Let $|\psi\rangle$ be an n-qubit state. Given size parameters s and n in unary and access to arbitrarily many copies of $|\psi\rangle$ (or the classical description of $|\psi\rangle$), decide if there exists a quantum circuit C which has size at most s using at most s ancilla qubits such that $C|0^n\rangle$ and $|\psi\rangle$ are close in terms of fidelity.*

In the rest of this subsection, we first discuss several challenges and difficulties we encountered in the study of MCSP when moving from the classical setting to the quantum setting. Next, we give an overview of all the results and techniques. In particular, we focus on both interpreting the new connections we establish as well as the technical subtleties when quantizing the previous works in the classical setting. For a quick summary of the results, please take a look at Table 1.

1.2.1 Challenges and difficulties when moving to the quantum setting

In the following, we summarize several fundamental properties of quantum circuits, unitaries, and quantum states that induce problems and difficulties that would not appear in the classical setting.

Quantum computation is generally random and erroneous. It is natural to consider quantum circuits that approximate (rather than exactly computing) the desired unitary. One immediate consequence is that we have to define the quantum MCSPs as promise problems (with respect to the error)[4], which is more challenging to deal with. Moreover, since unitaries and quantum states are specified by complex numbers, we also need to properly tackle the precision issue. These quantum properties make generalizing classical results to the quantum setting non-trivial. For instance, some classical analyses (see [AGG+20] for an example) rely on the fact that the classical circuits are deterministic after the random string is made public, while any intermediate computation of a quantum circuit is inherently not deterministic.

Quantum circuits are reversible. This follows from the fact that every quantum gate is reversible. While this seems to be a restriction for quantum circuits, we observe that this enables search-to-decision reductions for UMCSP and SMCSP. Note that the existence of such reduction is a longstanding open question for classical MCSP. This suggests that quantum MCSPs can provide a new angle to leverage the reversibility of quantum circuits.

The introduction of ancilla qubits. As quantum circuits are reversible, every intermediate computation has to happen on the input qubits. Thus, it is very common to introduce *ancilla qubits* which are extra qubits initialized to all zero and can be regarded as additional registers for intermediate computation. Ancilla qubits introduce complications in quantum MCSPs. First, the quantum circuit complexity of an object could be very different when the allowed number of ancilla

[3] We say C and U are close if $|(\langle\psi|\otimes I)U^\dagger C(|\psi\rangle|0\rangle)|$ is large for all $|\psi\rangle$.

[4] The definitions above are not promise problems for simplicity. Check Section 3 and 5 for formal definitions.

qubits is different. Second, the classical simulation time of a quantum circuit scales exponentially in the number of input qubits plus the number of ancilla qubits. Namely, when the number of ancilla qubits is super-linear, classical simulations would require super-polynomial time[5]. An immediate consequence is that, unlike classical MCSP, MQCSP is not trivially in NP when allowing a super-linear number of ancilla qubits. In addition, the output of quantum circuits on ancilla qubits can be arbitrary quantum states in general. This property makes certain reductions for quantum MCSPs fail when considering many ancilla qubits.

Various universal quantum gate sets. The choice of the gate set affects the circuit complexity of the given Boolean functions (and unitaries and states). There are various universal quantum gate sets, and transforming from one to the other results in additional polylogarithmic overhead to the circuit complexity by the Solovay-Kitaev Theorem. We note that when considering certain hardness results, the choice of the gate set might matter. Take the approximate self-reduction for SMCSP (in Theorem 1.12) as an example, we start from constructing such reductions for a particular gate set. We then generalize the result to an arbitrary gate set via the Solovay-Kitaev Theorem; however, it introduces additional overhead to the approximation ratio. Another example is proving NP-hardness for multi-output MQCSP, where we show that the problem is NP-hard when considering particular gate sets, and it is still open whether the problem is NP-hard for all universal gate sets.

1.2.2 The Hardness of MQCSP and cryptography

We start with stating the hardness results of MQCSP and its implications in cryptography.

Theorem 1.4 (Informal)**.**

1. MQCSP *is in* QCMA $\subseteq$ QMA.

2. *If* MQCSP *can be solved in quantum polynomial time, then quantum-secure one-way function (*qOWF*) does not exist.*

3. *If one can solve* MQCSP *efficiently, then all problems in* SZK *have efficient algorithms.*

4. *Suppose that quantum-secure indistinguishability obfuscator ($i\mathcal{O}$) for polynomial-size circuits exists. Then,* MQCSP $\in$ BQP *implies* NP $\subseteq$ coRQP[6].

5. *Multiple-output* MQCSP *(under a gate set with some natural properties) is* NP*-hard under randomized reductions.*

We have discussed why MQCSP is not trivially in NP earlier. So, it is natural to wonder what can be a tighter upper bound for MQCSP. Instead of considering classical verifier, we allow the verifier to check the given witness circuit quantumly and thus are able to prove that MQCSP is in QCMA (which is a quantum analogue of MA allowing efficient quantum verifiers but classical witness).

For item 2 – 5, we study whether some hard problems reduce to MQCSP. Classically, many results use the fact that an MCSP oracle can break certain *pseudorandom generators* to show reductions from hard problems to MCSP. A distinguisher can break a pseudorandom generator by viewing that the string is a truth table of some Boolean function and using the MCSP oracle

[5] The running time is measured with respect to the size of the truth table or the size of the unitary/quantum state.

[6] coRQP is a complexity class of quantumly solvable problems with perfect soundness and bounded-error completeness.

to decide if the function has small circuit complexity[7]. We generalize this idea to the quantum setting by observing that if the Boolean function has small classical circuit complexity, then its quantum circuit complexity is also small. It is worth noting that the second result implies efficient algorithms for some lattice problems if MQCSP is in BQP.

For item 5, we generalize the recent breakthrough of Ilango et al. [ILO20] on the NP-hardness of MCSP. We note that the formal theorem statement depends on the gate set choices of MQCSP. To prove this theorem, we follow the proof ideas in [ILO20] and overcome some additional obstacles that appear in the quantum world. The new obstacle comes from (i) the quantum gate set is different from the one in the classical case; (ii) in the quantum world, we need to deal with error terms. We carefully handle these issues and extend the proof to the quantum setting.

1.2.3 MQCSP and learning theory

A central learning theory setting is (approximately) reconstructing a circuit for an unknown function given a limited number of samples. Learning Boolean functions in the classical setting was extensively studied (see, for example, a survey by Hellerstein and Servedio [HS07]); however, relatively few explorations have been made under the quantum setting. There are two natural quantum extensions: (i) learning a quantum circuit and (ii) adding quantumness in the learning algorithm. We study both scenarios and provide generic connections between MQCSP and the two settings

PAC learning for quantum circuits. Probabilistic approximately correct (PAC) learning [Val84] is a standard theoretical framework in learning theory. There are several variants, but for simplicity, we focus on the query model where a classical learning algorithm can query an unknown n-bit Boolean function f on inputs $x_1, \dots, x_m \in \{0,1\}^n$ and aim to output a circuit approximating f with high probability. To have efficient PAC learning algorithms for polynomial-size quantum circuits, we show that it is necessary and sufficient to have efficient algorithms for MQCSP or its variants.

Theorem 1.5 (Informal)**.** *The existence of an efficient PAC learning algorithm for* $\mathsf{BQP/poly}$ *is equivalent to the existence of an efficient randomized algorithm for* MQCSP*.*

Quantum learning. In the past two decades, there has been increased interest in quantum learning (see a survey by Arunachalam and de Wolf [AdW17]) due to the success of machine learning and quantum computing. While there have been interesting quantum speed-ups for specific learning problems such as Principal Component Analysis [LMR14] and quantum recommendation system [KP17], it is unclear whether the quantumness can provide a generic speed-up in learning theory. A recent result of Arunachalam et al. [AGG+20] suggested that this might be difficult by showing that the existence of efficient quantum learning algorithms for a circuit class would imply a breakthrough circuit lower bound. We further generalize their result by showing the equivalence of efficient quantum PAC learning and the non-trivial upper bound for MQCSP.

Theorem 1.6 (Informal)**.** *The existence of efficient quantum learning algorithms for PAC learning a circuit class* C *is equivalent to the existence of efficient quantum algorithms for* C-MQCSP[8]*.*

The proof idea is to quantize the "learning from a natural property" paradigm of [CIKK16]. Briefly speaking, the converse direction "algorithms for MQCSP imply learning algorithms" follows from the idea that one can use the Boolean function (the object to be learned) to construct a PRG

[7] If the truth table is truly random, it corresponds to a random function and must have large circuit complexity with high probability.

[8] C-MQCSP is MQCSP with respect to circuit class C.

with the property that breaking the PRG implies a reconstructing algorithm for f. Then, since an algorithm for MQCSP can break PRG, we obtain an algorithm for f. Another direction follows from the observation that we can still apply the learning algorithm given the truth table of the function. Specifically, for Theorem 1.5, it turns out that the converse direction is straightforward because $\mathsf{P/poly} \subset \mathsf{BQP/poly}$ while the forward direction requires the number of ancilla bits to be $O(n)$ due to the overhead from a classical simulation for quantum circuits. For Theorem 1.6, the difficulty lies in the fact that a quantum circuit is *inherently random* and one cannot arbitrarily compose quantum circuits as their wishes. To circumvent these issues, we invoke the techniques in [AGG+20] which built up composable tools for *reconstructing* a circuit from a quantum distinguisher. See Theorem 4.14, Theorem 4.12, and Section 4.2 for more details.

1.2.4 MQCSP and quantum circuit lower bounds

The classical MCSP is tightly connected to circuit lower bounds. We generalize the results of Oliveira and Santhanam [OS16], Arunachalam et al. [AGG+20], and Kabanets and Cai [KC00] to MQCSP.

Theorem 1.7 (Informal). *Suppose that* $\mathsf{MQCSP} \in \mathsf{BQP}$*. Then*

1. $\mathsf{BQE} \not\subset \mathsf{BQC}[n^k]$ *for any constant* $k \in \mathbb{N}$[9]*; and*

2. $\mathsf{BQP}^{\mathsf{QCMA}} \not\subset \mathsf{BQC}[n^k]$ *for any constant* $k \in \mathbb{N}$*.*

For item 1, we use MQCSP to construct a BQP-natural property against quantum circuit classes. Then, with a quantum-secure pseudorandom generator, we can use a "win-win argument" to show that $\mathsf{BQE} \not\subset \mathsf{BQC}[n^k]$ for any $k > 0$. The proof mainly follows from [AGG+20, OS16]. However, we extend their proofs to the quantum natural properties against *quantum* circuit classes. One technical contribution is a diagonalization lemma for quantum circuits.

For item 2, we follow the idea in [KC00] to show that the maximum quantum circuit complexity problem[10] can be solved in exponential time with a QCMA oracle. The main difference from the classical case is that we require a QCMA oracle instead of an NP one, which follows from the fact that we assume MQCSP is in BQP[11]. Then, the statement follows from the standard padding argument.

Another aspect of quantum circuit complexity is *hardness amplification.* Kabanets and Cai [KC00] showed that MCSP can be used as an amplifier to generate many hard Boolean functions. In this part, we show that with an MQCSP oracle, given one quantum extremely hard Boolean function, there is an efficient quantum algorithm that outputs many quantum-hard functions.

Theorem 1.8 (Hardness amplification by MQCSP, informal). *Assume* $\mathsf{MQCSP} \in \mathsf{BQP}$*. There exists a* BQP *algorithm that, given the truth table of a Boolean function with quantum circuit complexity* $2^{\Omega(n)}$*, outputs* $2^{\Omega(n)}$ *Boolean functions with* $m = \Omega(n)$ *variables such that each function has quantum circuit complexity greater than* $2^m/(c+1)m$ *for* c *some constant.*

The proof of Theorem 1.8 closely follows the proof in [KC00]. The key ingredient is a quantum Impagliazzo-Wigderson generator, which "quantizes" the construction in [IW97]. The quantum

[9] $\mathsf{BQC}[n^k]$ is the complexity class for problems that can be solved by $O(n^k)$-size quantum circuits with bounded fan-in, and BQE in the set of problems that can be solved in $2^{O(n)}$ time by quantum computers. Previously, Aaronson [Aar06] showed that $\mathsf{P}^{\mathsf{PP}} \not\subset \mathsf{BQC}[n^k]$ unconditionally. However, the relations between P^{PP}, BQE, and $\mathsf{BQP}^{\mathsf{QCMA}}$ are still unclear.

[10] The problem is, given 1^n, ask for a Boolean function $f : \{0,1\}^n \to \{0,1\}$ that has the maximum complexity.

[11] Along this line, the result still holds if we consider $\mathsf{MCSP} \in \mathsf{BQP}$ and maximum classical circuit complexity.

Impagliazzo-Wigderson generator can transform the given quantum extremely hard function to a quantum pseudorandom generator that fools quantum circuits of size $2^{O(n)}$. Since we assume $\mathsf{MQCSP} \in \mathsf{BQP}$, it means that we can construct a small quantum distinguishing circuit to accept the truth tables of hard functions. And we can show that our quantum Impagliazzo-Wigderson generator can fool the distinguishing circuit. Hence, most of the outputs of the quantum pseudorandom generator will have high quantum circuit complexity.

To quantize the Impagliazzo-Wigderson generator, we construct a quantum-secure direct-product generator, and also use the quantum Goldreich-Levin Theorem and quantum-secure Nisan-Wigderson generator developed in [AGG+20].

Hardness magnification is an interesting phenomenon in classical circuit complexity defined by [OS18]. It shows that a weak worst-case lower bound can be "magnified" into a strong worst-case lower bound for another problem. (See a recent talk by Oliveira [Oli19].) In this part, we show that MQCSP also has a quantum hardness magnification.

Theorem 1.9 (Hardness magnification for MQCSP, informal)**.** *If a gap version of* MQCSP *does not have nearly-linear size quantum circuit, then* QCMA *cannot be computed by polynomial size quantum circuits.*

We note that this is a nontrivial theorem because even if we assume $\mathsf{QCMA} \subseteq \mathsf{BQC}[\mathsf{poly}(n)]$, we can only show $\mathsf{MQCSP} \in \mathsf{BQC}[\mathsf{poly}(2^n)]$, i.e., MQCSP has a polynomial-size quantum circuit by the fact that $\mathsf{MQCSP} \in \mathsf{QCMA}$. But the theorem implies that some gap-version of MQCSP has nearly-linear size circuit!

We prove the above theorem via a quantum antichecker lemma, whose classical version was given by [OPS19, CHO+20]. And we observe that the two key ingredients: a delicate design of a Boolean circuit and a counting argument can be quantized.

1.2.5 MQCSP and quantum fine-grained complexity

Fine-grained complexity theory aims to study the *exact* lower/upper bounds of some problems. For example, most theorists believe 3-SAT is not in P, but we do not know if it can be solved in $2^{o(n)}$ time. Exponential Time Hypothesis (ETH) is a commonly used conjecture in this area which rules out this possibility (see a survey by Williams [Wil18]). Very recently, [Ila20b] showed the fine-grained hardness of MCSP for partial function based on ETH. In the quantum setting, [ACL+20, BPS21] proposed quantum fine-grained reductions and quantum strong exponential time hypothesis (QSETH) to study the quantum hardness of problems in BQP. In this part, we follow the works of [Ila20b, ACL+20] and prove the quantum hardness of MQCSP for partial functions based on the quantum ETH conjecture,which conjectures that there does not exist a $2^{o(n)}$-time quantum algorithm for solving 3-SAT[12].

Theorem 1.10 (Fine-grained hardness of $\mathsf{MQCSP}^{\star}$, informal)**.** *Quantum ETH implies $N^{o(\log\log N)}$-quantum hardness of* MQCSP *for partial functions.*

To prove the above theorem, we basically follow the reduction path in [Ila20b], which gave a reduction from a fine-grained problem studied by [LMS11] to MQCSP for partial functions. But we need to bypass two subtleties:

- The proof of [Ila20b] relies on the structure of the classical read-once formula, but there is no direct correspondence with quantum;

[12] Existing quantum SAT solvers are not much faster than Grover's search; they need $2^{\Omega(n)}$-time even for 3-SAT.

- [LMS11] only proved the classical hardness of the bipartite permutation independent set problem, but we need quantum hardness result.

For the first issue, we prove an unconditional quantum circuit lower bound for that function in the reduction. More specifically, we first show that if a small quantum circuit can compute the partial function γ in the reduction, then that circuit is a quantum read-once formula (defined by [Yao93]); and vice versa. And then, we apply a "dequantization" result by [CKP13] to show that the quantum read-once formula can be converted to a classical read-once formula with the same size. Then, by the structure of the "dequantized" read-once formula, we finally conclude that deciding MQCSP for γ is equivalent to solving the bipartite permutation independent set problem.

For the second issue, we use the quantum fine-grained reduction framework and give a reduction from 3-SAT to the bipartite permutation independent set problem. Therefore, the quantum hardness of MQCSP for partial function follows from the quantum hardness of deciding 3-SAT conjectured by the quantum ETH.

1.2.6 Quantum circuit complexity for states and unitaries

In this section, we study UMCSP and SMCSP. For SMCSP in Definition 1.3, we consider two types of inputs: quantum states and the classical description of the state. We consider the inputs as quantum states since we generally cannot have the classical description of the quantum state in the real world, and many related problems (such as shadow tomography [Aar18], quantum gravity [BFV20], and quantum pseudorandom state [JLS18]) have multiple copies of states as inputs. Although this input format makes SMCSP harder, we are able to show that SMCSP has a QCMA protocol[13]. Furthermore, the search-to-decision reduction and the self-reduction in Theorem 1.12 hold for both versions of SMCSP. We first show hardness upper bounds for UMCSP and SMCSP.

Theorem 1.11 (Informal)**.** (1) UMCSP $\in$ QCMA*.* (2) SMCSP *can be verified by* QCMA *protocols.*

To prove Theorem 1.11, we use the *swap test* to test whether the witness circuit C outputs the correct states. This suffices to show that SMCSP has a QCMA protocol. To show that UMCSP is in QCMA, checking if the circuit C and U agree on all inputs by using swap test is infeasible since there are infinitely many quantum states in the 2^n-dimensional Hilbert space. If one only checked all the computational basis states (i.e., $\{|x\rangle : x \in \{0,1\}^n\}$), it is possible that the circuit C and the given unitary U are not close on inputs in the form of superposition states. This can come from the following two sources. (a) C can introduce different phases on different computational basis states; (b) using ancilla qubits to implement U results in entanglement between the output qubits and ancilla qubits, which may fail the swap test.

To deal with these difficulties, we introduce an additional step in the test called "coherency test". This step tests the circuit output on all the initial states in the form of $|a\rangle + |b\rangle$, where $|a\rangle, |b\rangle$ are different computational basis states. We can prove that it forces the behavior of C to be coherent on all the computational basis states, and forces the phases to be roughly the same.

Reductions for UMCSP and SMCSP that are unknown to the classical MCSP. In addition to the upper bounds, we also show interesting reductions for UMCSP and SMCSP.

Theorem 1.12 (Informal)**.**

- ***Search-to-decision reductions:*** *There exist search-to-decision reductions for* UMCSP *and* SMCSP *when no ancilla qubits are allowed.*

[13] Note that since SMCSP has quantum inputs, the problem is not in QCMA under the standard definition.

- ***Self-reduction:*** SMCSP *is approximately self-reducible.*
- *A gap version of* MQCSP *reduces to* UMCSP*.*

Classically, it is unknown whether MCSP is self-reducible or has search-to-decision reductions. Ilango [Ila20a] proved that some variants of MCSP have search-to-decision reductions. Recently, Ren and Santhanam [RS21] showed that a relativization barrier applies to the deterministic search-to-decision reduction and self-reduction of MCSP. We prove the existence of search-to-decision reductions by using the property that "*quantum circuits are reversible*". In particular, we guess the i-th gate, uncompute the gate from the state or the unitary, and use the decision oracles to check whether the complexity of the new state or the new unitary reduces. By repeating this process for all gates, we can find the desired circuits. This approach suffices for the case where the quantum circuits use no ancilla qubits. On the other hand, when the quantum circuits use ancilla qubits and are not forced to turn ancilla qubits back to the all-zero state, this approach does not work. Consider UMCSP. The quantum circuit may implement a unitary $U \otimes V$. To find the circuit, the approach above needs to start from $U \otimes V$ and do the uncomputation iteratively. However, V is unknown. SMCSP has the similar issues.

For the self-reducibility of SMCSP, we show that one can approximate the circuit complexity of an n-qubit state by computing the circuit complexities of $(n-1)$-qubit states. Roughly, we find a "win-win decomposition" of an n-qubit state such that its circuit complexity is either close to the circuit complexity of an $(n-1)$-qubit state or can be approximated by two $(n-1)$-qubit states.

Finally, we show a reduction related to MQCSP and UMCSP. The proof is by encoding a Boolean function into a particular unitary and showing that the circuit complexity of that unitary gives both upper and lower bounds for the circuit complexity of the Boolean function.

Implications of Hardness of SMCSP and UMCSP For UMCSP, one application is related to a question Aaronson asked in [Aar16]: does there exist an efficient quantum process that generates a family of unitaries that are indistinguishable from random unitaries given the full description of the unitary? If there is an efficient algorithm for UMCSP, then there is no efficient quantum process that generates unitaries indistinguishable from random unitaries given the full unitary.

Moreover, several implications of MCSP carry to UMCSP by Theorem 1.12. This follows from the fact that the gap version of MQCSP suffices to break certain pseudorandom generators.

For SMCSP, we focus on the version where the inputs are copies of quantum states and present its relationships to quantum cryptography, tomography, and quantum gravity.

Theorem 1.13 (Informal)**.**

1. *If* SMCSP *has quantum polynomial-time algorithms, then there are no pseudorandom states, and thus no quantum-secure one-way functions.*
2. *Assuming additional conjectures from physics and complexity theory, the existence of an efficient algorithm for* SMCSP *implies the existence of an efficient algorithm for estimating the wormhole's volume*
3. *If* SMCSP *can be solved efficiently, then one can solve the succinct state tomography problem*[14] *in quantum polynomial time.*

[14] The succinct state tomography problem is that given many copies of a state with the promise that its circuit complexity is at most certain s, the problem is to find a circuit that computes the state.

The first result in Theorem 1.13 follows from the observation that we can use SMCSP algorithms to distinguish whether the given states have large circuit complexities. This results in algorithms for breaking pseudorandom states, and thus algorithms for inverting quantum-secure one-way functions by [JLS18]. It is worth noting that a recent work by Kretschmer [Kre21] showed some relativized results for the problem of breaking pseudorandom states. Since that problem reduces to SMCSP, his results would provide another angle for understanding the hardness of SMCSP. We show the second result under the model and assumptions considered in [BFV20]. Roughly speaking, the volumes of wormholes correspond to circuit complexities of particular quantum states. Thus efficient algorithms for one implies solving the other one efficiently if the correspondence can be computed efficiently. The third result mainly uses the *search-to-decision reduction* in Theorem 1.12 to find the circuit that computes the state.

1.3 Discussion and open questions

We lay out the following three-aspect road map for the quantum MCSP program. For each aspect, we present several results and also propose many open directions to explore. We have also summarized all results in this work in Table 1.

First, we define the Minimum Quantum Circuit Size Problem (MQCSP) and study upper bounds and lower bounds for its complexity. Furthermore, we explore the connections between MQCSP and other areas of quantum computing such as quantum cryptography, quantum learning, quantum circuit lower bounds, and quantum fine-grained complexity.

Then, we further extend MQCSP to study the quantum circuit complexities for quantum objects, including unitaries and states.[15] We want to investigate their hardness and connections to other areas in TCS. In this work, we show upper bounds and lower bounds for their complexities, search-to-decision reductions (for UMCSP and SMCSP), a self-reduction (for SMCSP), and reductions from MQCSP to UMCSP. In addition to connections generalized from classical analogues (such as cryptography, learning, and circuit lower bounds), we also find connections that might be unique in the quantum setting, such as tomography and quantum gravity.

For the last part, we want to turn around and ask what could happen when considering quantum algorithms or quantum reductions for MCSP (and also for MQCSP, UMCSP, and SMCSP)? In the previous two parts, we have already observed that efficient quantum algorithms for these problems result in surprising implications to other fields. One can further consider other influences of quantum algorithms to study quantum and classical MCSPs. For example, can SAT reduce to MCSP under quantum reductions?

Following the three-aspect road map for the quantum MCSP program, there are many open directions to explore. In particular, we are interested to understand the hardness of these problems, the relationships between them, and their connections to other fields in computer science.

1.3.1 Open problems: the complexity of quantum circuits

We start with open problems related to the hardness and relationships between quantum MCSPs. The most basic questions are to understand the complexity of different quantum MCSPs. As we have already seen, it is unclear if quantum MCSPs are in NP. Besides, we do not know if NP- or QCMA-hard problems reduce to them.

Open Problem 1. *Are* UMCSP*,* MQCSP*, and* SMCSP *in* NP*? Are these problems* NP*-hard,* QCMA*-hard, or* C*-hard for some complexity class* C *that is between* QCMA *and* SZK*?*

[15] Aaronson has raised questions about quantum circuit complexity for unitaries or states in [Aar16].

We note that the case that makes these problems not known to be in NP is when there are more than linearly many ancilla qubits. Therefore, if one can show that adding superpolynomially many ancilla qubits does not lead to significant improvement on quantum circuit complexity, then we are likely to put these problems in NP directly. Along this line, we pose the following open question:

Open Problem 2. *For every $n, s, t \in \mathbb{N}$ with $t \leq s \leq 2^{O(n)}$, is $\mathsf{BQC}(s,t) \subset \mathsf{BQC}(\mathsf{poly}(s,t), O(n))$?*

For the hardness of UMCSP and SMCSP, One potential approach for proving NP-hardness of UMCSP is as follows: Prove the NP-hardness of the gap version of certain variants of MQCSP (such as sparse MQCSP or $\mathsf{multiMQCSP}$), and then reduce it to UMCSP via the last reduction in Theorem 1.12. The hardness of SMCSP seems to be slightly more mysterious than UMCSP. One reason for this is that we do not know any relationship between SMCSP and other quantum MCSPs, and thus the approach of reducing particular variants of quantum MCSP to SMCSP does not directly work. This leads to another important open question:

Open Problem 3. *What are the relationships between* UMCSP*,* MQCSP*, and* SMCSP*?*

To answer whether quantum MCSPs are NP-complete, we can also study these problems from another angle, that is, check if quantum MCSPs have particular reductions that all NP-complete problems have. In the previous section, we observed that quantum circuits have some properties leading to search-to-decision reductions for UMCSP and SMCSP without ancilla qubits and an approximate self-reduction for SMCSP. Therefore, we ask whether we can have search-to-decision reductions and self-reductions for these quantum MCSPs.

Open Problem 4. *Are there search-to-decision reductions and self-reductions for quantum* MCSP*s?*

It is worth noting that our search-to-decision reductions fail when ancilla qubits are allowed. This mainly follows from the fact that the circuit of the solution can be an non-identity operator on the ancilla qubits in general. This could possibly be addressed by iterating all possible unitaries or states on an ϵ-net when the number of ancilla qubits are not large (e.g., at most $\log \log n$). However, we need new ideas when considering more ancilla qubits.

Moreover, it would be interesting to investigate the applications of these reductions. For instance, we have seen that the search-to-decision reductions give algorithms with UMCSP or SMCSP oracle additional power to obtain the circuits. This power may lead to interesting applications.

Open Problem 5. *Is there any application of search-to-decision reductions or self-reductions for quantum* MCSP*s?*

The hardness of average-case quantum MCSPs (which inputs are given randomly) is another interesting topic to explore. Hirahara [Hir18] showed that there is a worst-case to average-case reduction for the (gap version of) classical MCSP. We wonder if we can prove that quantum MCSPs have worst-case to average-case reductions.

Open Problem 6. *Are there worst-case to average-case reductions for quantum* MCSP*s?*

Note that there is negative evidence [BT06] showing that such classical reductions might not exist for NP-complete problems[16]. The existence of such reduction could result in important applications in cryptography, which we will discuss later.

Finally, we can also try to prove the hardness of quantum MCSPs under stronger assumptions or more powerful reductions.

[16] However, there is no evidence for the existence of quantum worst-case to average-case reductions for NP-complete since the analysis in [BT06] fails in the quantum setting. See [CHS20] for related discussion.

Open Problem 7. *Assuming* QETH *or* QSETH*, is* MQCSP*,* UMCSP*, or* SMCSP *quantumly hard?*

Open Problem 8. *Does quantum reduction provide more power to show the hardness of* MCSP*? Specifically, is* $\mathsf{NP} \subseteq \mathsf{BQP}^{\mathsf{MCSP}}$ *or* $\mathsf{NP} \subseteq \mathsf{BQP}^{\mathsf{MQCSP}}$*?*

1.3.2 Open problems: potential connections to other areas

In this work, in addition to generalizing several known connections for MCSP to quantum MCSPs, we have also discovered several connections which could be unique for quantum MCSPs. There are still many classically existing or unknown connections that we can explore. One fascinating question is whether we can base the security of one-way functions on any of these problems.

Open Problem 9. *Can we base the security of cryptographic primitives on* MQCSP*,* UMCSP*,* SMCSP*, or some variants of these problems?*

Note that since quantum MCSPs considered in this work are all worst-case problems, to answer Problem 9, we probably need worst-case to average-case reductions discussed in Problem 6. Moreover, Liu and Pass [LP20] recently showed that the existence of classical one-way function is equivalent to the average-case hardness of a type of Kolmogorov complexity on uniform distribution. However, the average-case hardness of MCSP on uniform distribution is not known to imply one-wayness even classically, and the quantum version faces a similar obstacle. Very recently, Ilango, Ren, and Santhanam [IRS21] showed that the average-case hardness of Gap-MCSP on a locally samplable distribution is equivalent to the existence of one-way function. Liu and Pass [LP21] further generalized this result to show equivalence between the existence of one-way functions and the existence of sparse languages that are hard-on-average (including Kolmogorov complexity, k-SAT, and t-Clique). It is natural to ask whether their results can be generalized to quantum MCSPs. In addition to one-way functions, We are interested in connections between quantum MCSPs and "quantum-only" primitives, e.g., quantum $i\mathcal{O}$, copy protection, quantum process learning, etc.

Along this line, as many quantum problems have quantum inputs, it is natural to consider quantum MCSPs with quantum inputs. We have shown how SMCSP connects to problems in quantum cryptography, quantum gravity, and tomography given quantum states as inputs. This fact gives the possibility that MQCSP, UMCSP, and SMCSP with "succinct" quantum or classical inputs may have surprising connections to other problems in quantum computing. For instance, one can consider inputs which are quantum circuits that encode some objects (e.g., unitaries). Then, the problem is to find another significantly smaller circuit. In [CCCW21], Chakrabarti et al. have studied this problem and show applications to quantum supremacy.

	Results	Informal Theorem Index (Formal Theorem Index)
MQCSP (Def. 3.2)	$\mathsf{MQCSP} \in \mathsf{QCMA}$	Theorem 1.4 (Theorem 3.9)
	$\mathsf{MQCSP} \in \mathsf{BQP} \Rightarrow$ No qOWF	Theorem 1.4 (Theorem 4.8)
	$\mathsf{SZK} \leq \mathsf{MQCSP}$	Theorem 1.4 (Theorem 3.13)
	$\mathsf{multiMQCSP}$ is NP-hard under a natural gate set	Theorem 1.4 (Theorem 3.14)
	$i\mathcal{O} + \mathsf{MQCSP} \in \mathsf{BQP} \Rightarrow \mathsf{NP} \subseteq \mathsf{coRQP}$	Theorem 1.4 (Theorem 4.10)
	PAC learning for $\mathsf{BQP/poly} \Leftrightarrow \mathsf{MQCSP} \in \mathsf{BPP}$	Theorem 1.5 (Theorem 4.12)
	BQP learning $\Leftrightarrow \mathsf{MQCSP} \in \mathsf{BQP}$	Theorem 1.6 (Theorem 4.14)
	$\mathsf{MQCSP} \in \mathsf{BQP} \Rightarrow \mathsf{BQE} \not\subset \mathsf{BQC}[n^k],\ \forall k \in \mathbb{N}_+$	Theorem 1.7 (Theorem 4.19)
	$\mathsf{MQCSP} \in \mathsf{BQP} \Rightarrow \mathsf{BQP}^{\mathsf{QCMA}} \not\subset \mathsf{BQC}[n^k],\ \forall k \in \mathbb{N}_+$	Theorem 1.7 (Theorem 4.22)
	$\mathsf{MQCSP} \in \mathsf{BQP} \Rightarrow$ Hardness amplification	Theorem 1.8 (Theorem 4.20)
	Hardness magnification for MQCSP	Theorem 1.9 (Theorem 4.22)
	$\mathsf{QETH} \Rightarrow$ quantum hardness of $\mathsf{MQCSP}^\star$	Theorem 1.10 (Theorem 4.27)
UMCSP (Def. 5.1)	$\mathsf{UMCSP} \in \mathsf{QCMA}$	Theorem 1.11 (Theorem 5.5)
	Search-to-decision reduction for UMCSP	Theorem 1.12 (Theorem 5.16)
	$\mathsf{gap\text{-}MQCSP} \leq \mathsf{UMCSP}$	Theorem 1.12 (Theorem 5.23)
	$\mathsf{UMCSP} \in \mathsf{BQP}$ $\Rightarrow$ No pseudorandom unitaries and no qOWF	(Theorem 5.24, Corollary 5.25)
	$i\mathcal{O} + \mathsf{UMCSP} \in \mathsf{BQP} \Rightarrow \mathsf{NP} \subseteq \mathsf{coRQP}$	(Corollary 5.26)
	$\mathsf{UMCSP} \in \mathsf{BQP} \Rightarrow$ Hardness amplification for BQP	(Corollary 5.27)
	$\mathsf{UMCSP} \in \mathsf{BQP} \Rightarrow \mathsf{BQE} \not\subset \mathsf{BQP}[n^k],\ \forall k \in \mathbb{N}$	(Corollary 5.28)
SMCSP (Def. 5.2)	SMCSP can be verified via QCMA	Theorem 1.11 (Theorem 5.9)
	Search-to-decision reduction for SMCSP	Theorem 1.12 (Theorem 5.18)
	Self-reduction for SMCSP	Theorem 1.12 (Theorem 5.20)
	$\mathsf{SMCSP} \in \mathsf{BQP}$ $\Rightarrow$ No pseudorandom states and no qOWF	Theorem 1.13 (Theorem 5.30)
	Assume conjectures from physics $\mathsf{SMCSP} \Rightarrow$ Estimating wormhole's volume	Theorem 1.13 (Theorem 5.31)
	Succinct state tomography $\leq \mathsf{SMCSP}$	Theorem 1.13 (Theorem 5.33)

Table 1: Summary of our results. A result with color Blue is a direct extension from its classical analog. A result with color Yellow requires additional techniques. A result with color Red is unique in the quantum setting.

2 Preliminaries

We start with a brief overview of quantum computation and complexity theory. We recommend the standard textbook [NC11] for a more comprehensive treatment.

2.1 Quantum states, unitary transformations, and quantum circuits

To give a brief introduction to the quantum computing, we divide the computation into three parts: *input*, *process*, and *output*.

Input. In quantum computing, we represent information in quantum states using qubits.

Definition 2.1 (Pure quantum state)**.** *A pure quantum state $|\psi\rangle$ on n qubits is represented as a unit vector in $\mathbb{C}^{2^n}$, $|\psi\rangle = (c_0, c_1, \ldots, c_{2^n-1})^T$, where $c_i \in \mathbb{C}$ for $i \in \{0, \ldots, 2^n - 1\}$ and $\sum_i |c_i|^2 = 1$.*

For example, $|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle$ represent n-bit classical messages, $0, 1, \ldots, 2^n - 1$. For convenience, we sometimes denote $N = 2^n$. Mathematically, one can think of $|i\rangle$ as the column vector with the $(i+1)$-th entry being 1 and 0 elsewhere. The input to quantum computers can be any quantum state. For classical problems, we can encode the classical input $x \in \{0,1\}^n$ as the quantum state $|x\rangle \in \mathbb{C}^{2^n}$. In general, any pure quantum state $|\psi\rangle$ can be represented as $\sum_i c_i|i\rangle$ for some $c_0, \ldots, c_{2^n-1} \in \mathbb{C}$ with $\sum_i |c_i|^2 = 1$. The complex conjugate of $|\psi\rangle$ is denoted as a row vector $\langle\psi| = (c_0^*, \cdots, c_{2^n-1}^*)$.[17]

Quantum process. Quantum process for quantum states is defined as a unitary transformation.

Definition 2.2 (Unitary transformation)**.** *A unitary transformation U for an n-qubit quantum state is an isomorphism in the 2^n-dimensional Hilbert space. For convenience, we view U as a $2^n \times 2^n$ matrix satisfying that $U \in \mathbb{C}^{N\times N}$ with $UU^\dagger = U^\dagger U = I$ where $U^\dagger$ is the Hermitian adjoint of U.*

We can represent an n-qubit quantum process acting on an n-qubit state $|\psi\rangle$ as $|\psi\rangle \mapsto U|\psi\rangle$ as a unitary matrix U in $\mathbb{C}^{2^n\times 2^n}$. Note that a unitary matrix must preserve the norm of the input state. Thus any unitary transformation is reversible. To implement a unitary transformation, we pick a set of *local unitary operations* that can generate any unitary transformation with arbitrary precision.

Definition 2.3 (Universal quantum gate set)**.** *A quantum gate set $\mathcal{G}$ is a set of unitaries such that for any unitary transformation U, U can be approximated by a finite sequence of gates in $\mathcal{G}$.*

For example, $\{\mathsf{Toffoli}, \mathsf{H}\}$ is a universal gate set [Shi02]. In this work, we consider gate sets which only contain unitaries with constant dimensions.

Note that choosing different universal gate sets may cause the circuit complexity of the same object to be different. However, the Solovay-Kitaev theorem shows that one universal gate set can approximate another one at a modest cost.

Theorem 2.4 (Solovay-Kitaev Theorem)**.** *Let $\mathcal{G}$ and $\mathcal{G}'$ be two universal gate sets. Then, any s-gate circuit $\mathcal{C}$ using gates from $\mathcal{G}$ can be approximated to precision ϵ by a $s \,\mathsf{poly}\log\frac{s}{\epsilon}$-gates circuit $\mathcal{C}'$ using gates from $\mathcal{G}'$. We say $\mathcal{C}$ approximates to $\mathcal{C}'$ with precision ϵ if*

$$\|\mathcal{C} - \mathcal{C}'\| \leq \epsilon,$$

[17] In general, people consider mixed state for quantum information. However, pure states suffice for our purpose.

where $\|\cdot\|$ is L_2 norm.

We will formally state Solovay-Kitaev Theorem when defining the problems of quantum circuit complexity.

We can represent quantum algorithms as quantum circuits by using a sequence of quantum gates from a universal quantum gate set.

Definition 2.5 (Quantum circuit $\mathsf{QC}(s,t,\mathcal{G})$)**.** *Let $s,t:\mathbb{N}\to\mathbb{N}$ and $\mathcal{G}$ be a universal quantum gate set. A quantum circuit family $\{\mathcal{C}_n : n>0\}$ is in $\mathsf{QC}(s,t,\mathcal{G})$ if the following holds: For all $n>0$,*

- *the input to $\mathcal{C}_n$ is an n-qubit quantum state $|\psi\rangle$;*
- *$\mathcal{C}_n$ extends the input layer with $t(n)$ ancilla qubits, where these ancilla qubits are initiated to $|0^{t(n)}\rangle$;*
- *$\mathcal{C}_n$ applies $s(n)$ gates from $\mathcal{G}$ on the initial state $|\psi\rangle|0^{t(n)}\rangle$.*

Here, in addition to the qubits for the input, the circuit can also have ancilla qubits as its working space. We say that a quantum algorithm is efficient if its corresponding circuit has circuit size at most polynomial in the input size. In the rest of the paper, we may write $\mathsf{QC}(s,t,\mathcal{G})$ as $\mathsf{QC}(s)$ if the number of ancilla qubits is at most $O(s)$.

Output. The outputs of quantum circuits defined in Definition 2.5 are quantum states. To extract useful information from a quantum state $|\psi\rangle$, one can *measure* the state. Mathematically, a measurement is simply a sampling process. For example, if we measure $|\psi\rangle$ in the *computational basis*, i.e., $\{|0\rangle\langle 0|,\ldots,|2^n-1\rangle\langle 2^n-1|\}$, we get the output being index i with probability $|c_i|^2$. In general, we can measure a state $|\psi\rangle$ on any orthogonal basis $\mathcal{B}$ for $\mathbb{C}^{2^n}$. Mathematically, this is equivalent to a change of basis via a *unitary transformation*.

In summary, a quantum algorithm for some Boolean function is as follows: Given $|x\rangle$, apply a quantum circuit $\mathcal{C}$ on state $|x,0^{t(n)}\rangle$, and then measure the state $\mathcal{C}|x,0^{t(n)}\rangle$ in the computational basis. If $\mathcal{C}$ computes f, then the measurement outcome will be $f(x)$ with probability good enough (e.g., $\geq 2/3$). Note that a quantum process can have measurements in the middle of the computation in general. In this case, the process is not reversible any more. However, we can always defer these measurements until all the unitaries have been applied by adding ancilla qubits. Therefore, for simplicity, we will only consider processes represented as unitaries followed by a computational-basis measurement.

Remark 1 (Deferring measurements)**.** *Let $\mathcal{M}_i$ be the computational-basis measurement on the i-th qubit. Let $|\psi\rangle$ be any n-qubit state and U,V be any n-qubit unitaries. Then, the process $U\circ\mathcal{M}_i\circ V$ operating on $|\psi\rangle$ is equivalent to $\mathcal{M}_{n+1}\circ U\circ \mathsf{CNOT}_{i,n+1}\circ V$, where $\mathsf{CNOT}_{i,n+1}$ has the i-th qubit as the control qubit and the $n+1$-th qubit as the target qubit.*

2.2 Quantum complexity classes

We introduce quantum complexity classes that are related to our study on the quantum MCSP. The classes we define in below are actually $\mathsf{PromiseBQP}$ and $\mathsf{PromiseQCMA}$. To avoid abuse of notation, we just denote them as BQP and QCMA.

We first give the definition of the quantum analogue of BPP and P.

Definition 2.6 (BQP)**.** *A promise problem $P=(P_Y,P_N)$ is in BQP if there exists a polynomial-time classical Turing Machine that on input 1^n for any $n\in\mathbb{N}$ outputs the description of a quantum circuit $\mathcal{C}_n$ with $\mathsf{poly}(n)$ gates and $\mathsf{poly}(n)$ ancilla qubits such that for $x\in\{0,1\}^n$ the following holds:*

1. *if* $x \in P_Y$, $\Pr[M_1 \circ \mathcal{C}_n|x, 0^t\rangle = 1] \geq 2/3$*;*

2. *if* $x \in P_N$, $\Pr[M_1 \circ \mathcal{C}_n|x, 0^t\rangle = 1] \leq 1/3$*,*

where M_1 *is the computational-basis measurement on the first qubit of the given state.*

We also consider the quantum analogue of NP and MA in this work.

Definition 2.7 (QCMA)**.** *A promise problem* $P = (P_Y, P_N)$ *is in* QCMA *if there exists a quantum polynomial-time (QPT) algorithm* V *such that*

1. *for* $x \in P_Y$*, there exists* $w \in \{0,1\}^{\mathsf{poly}(n)}$ *such that* $\Pr[V(x,w) = 1] \geq 2/3$*;*

2. *for* $x \in P_N$*, for all* $w \in \{0,1\}^{\mathsf{poly}(n)}$*,* $\Pr[V(x,w) = 1] \leq 1/3$*.*

Another quantum analogue of MA and NP is called QMA. The difference between QMA and QCMA is that QMA allows the certificates to be quantum states. This difference makes QCMA ⊆ QMA[18].

We also consider the class RQP, which is the one-sided error version of BQP:

Definition 2.8 (RQP)**.** *A promise problem* $P = (P_Y, P_N)$ *is in* RQP *if there exists a QPT algorithm* $\mathcal{A}$ *such that*

1. *for* $x \in P_Y$*, then* $\Pr[\mathcal{A}(x) = 1] \geq \frac{1}{2}$*;*

2. *for* $x \in P_N$*, then* $\Pr[\mathcal{A}(x) = 1] = 0$*.*

2.3 Nonuniform quantum circuit complexity classes

With the mathematical background of quantum computing, we can define nonuniform quantum circuit complexity classes. We define the quantum analogues of MCSP as promise problems. (We will justify the reason later in Section 3.) Therefore, we also define complexity classes for promise problems. A promise problem is defined as $P = \{P^n\}$, where $P^n = (P^n_Y, P^n_N)$ satisfying $P^n_Y \cap P^n_N = \emptyset$ and $P^n_Y \cup P^n_N \subseteq \{0,1\}^n$. We say a promise problem P is in some class C if there exists a language $L \in \mathsf{C}$ such that $P_Y \subseteq L$ and $P_N \subseteq \{0,1\}^* \setminus L$. In other words, for $x \in \{0,1\}^* \setminus P$, the answer could be arbitrary. Note that promise problems are naturally considered in quantum computing; for example, the local Hamiltonian problem [KSV02] (which is QMA-complete) and Identity check on basis states [WJB03] (which is QCMA-complete.)

Definition 2.9 ($\mathsf{BQC}(s,t,\mathcal{G})$)**.** *Let* $s, t : \mathbb{N} \to \mathbb{N}$ *and* $\mathcal{G}$ *be a quantum gate set.* $\mathsf{BQC}(s,t,\mathcal{G})$ *is the set of promise problems* $P = \{P^n : n > 0\}$ *for which there exists a circuit family* $\{\mathcal{C}_n : n > 0\} \in \mathsf{QC}(s,t,\mathcal{G})$ *such that for* $n > 0$*, for any* x *where* $|x| = n$*,*

- *if* $x \in P^n_Y$*, then* $\Pr[M_1 \circ \mathcal{C}_n|x, 0^t\rangle = 1] \geq 2/3$*;*
- *if* $x \in P^n_N$*,* $\Pr[M_1 \circ \mathcal{C}_n|x, 0^t\rangle = 1] \leq 1/3$*.*

Here, M_1 *is the computational-basis measurement on the first qubit.*

In the rest of the paper, we will write $\mathsf{BQC}(s,t,\mathcal{G})$ as $\mathsf{BQC}(s)$ for simplicity if the number of ancilla qubits is at most $O(s)$.

In addition to BQC, we will also consider quantum complexity classes such as QMCA and BQP. For the same reason, the classes we consider are actually PromiseBQP and PromiseQCMA. To avoid abuse of notation, we just denote them as BQP and QCMA. Also, when NP is mentioned, we are actually considering PromiseNP. The formal definitions of these classes are given in Appendix 2.2.

[18] One may expect that the quantum certificate gives the malicious prover more power to cheat in the soundness case. However, it can be shown that the existence of such a cheating prover in QMA would also imply a cheating prover in QCMA by the convexity of quantum states.

3 Minimum Quantum Circuit Size Problems

We start off the quantum MCSP program by giving the definitions of various quantum analogs of the classical MCSP in Section 3.1 and investigating some basic complexity-theoretic results in Section 3.2 and Section 3.3.

3.1 Problem definitions

While classical computation works on Boolean strings, quantum computation works on unit complex vectors. Thus, there are multiple natural notions of MCSP that can be defined and studied in the quantum realm. But first let us formally define the classical MCSP as follows.

Definition 3.1 (Classical MCSP). *Let $n, s \in \mathbb{N}$[19]. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. The problem is, given the truth table $\mathsf{tt}(f)$ of f and the size parameter s in unary, decide if there exists a classical Boolean circuit C of size at most s such that $C(x) = f(x)$ for all $x \in \{0,1\}^n$.*

Note that MCSP $\in$ NP because given a truth table $\mathsf{tt}(f)$ a circuit C, we can verify whether $C(x) = f(x)$ for all $x \in \{0,1\}^n$ in $\mathsf{poly}(|\mathsf{tt}(f)|, 1^s)$ time. On the other hand, when $s = \Omega(n)$, the number of circuits of size at most s is $2^{\Theta(s \log s)}$, which is $2^{\omega(n)}$ by the counting argument. Besides, for every Boolean function, there exists a circuit with size at most $O(2^n/n)$ [Lup58]; therefore, we can suppose the $s = O(2^n/n)$, which implies that brute-force search takes $2^{O(2^n)}$ time to solve MCSP in the worst case and it is the best known algorithm for MCSP.

As quantum computation is generally believed to be more powerful than classical computation, it is likely that the quantum circuit complexities for some Boolean functions are much different from their classical circuit complexities. Specifically, quantum circuits can create quantum entanglement between qubits that cannot be simulated classically. Therefore, we define the following problem for studying the quantum circuit complexity of the given Boolean function.

Definition 3.2 ($\mathsf{MQCSP}_{\alpha,\beta}$). *Fix a universal gate set $\mathcal{G}$. Let $n, s, t \in \mathbb{N}$ and $t \leq s$. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let $\alpha, \beta \in (1/2, 1)$ such that $\alpha - \beta \geq \frac{1}{\mathsf{poly}(2^n)}$. MQCSP is a promise problem defined as follows.*

- ***Inputs:*** *the truth table $\mathsf{tt}(f)$ of f, the size parameter s in unary representation, and the ancilla parameter t.*
- ***Yes instance:*** *there exists a quantum circuit $\mathcal{C}$ using at most s gates and operating on at most $n + t$ qubits such that for all $x \in \{0,1\}^n$, $\|(\langle f(x)| \otimes I_{n+t-1})\mathcal{C}|x, 0^t\rangle\|^2 \geq \alpha$.*
- ***No instance:*** *for every quantum circuit $\mathcal{C}$ using at most s gates and operating on at most $n + t$ qubits, there exists $x \in \{0,1\}^n$ such that $\|(\langle f(x)| \otimes I_{n+t-1})\mathcal{C}|x, 0^t\rangle\|^2 \leq \beta$.*

With the promise that the input must be either a yes instance or a no instance, the problem is to decide whether the input is a yes instance or not.

Remark 2. *Here, we set the thresholds for the yes and no instances to be α, β such that $1/2 < \beta < \alpha < 1$ and $\alpha - \beta > \frac{1}{\mathsf{poly}(2^n)}$. We require α and β to be greater than $1/2$ because a quantum circuit that outputs a uniformly random bit (e.g., measure $|+\rangle$ in the computational basis) can compute $f(x)$ with $1/2$ probability for all x. For simplicity, in the rest of the work, we will ignore the subscription α, β and will specify them when it is necessary.*

[19] For every Boolean function, there is a circuit with size at most $O(2^n/n)$. Therefore, one can suppose s is at most $O(2^n/n)$. Besides, one can also consider s is given in unary, such that the problem is still well-defined in the sense that it is trivially in NP.

For MQCSP, which gate set $\mathcal{G}$ is used is another important parameter to be considered. One may ask if circuit complexity can significantly change when considering different $\mathcal{G}$. Fortunately, according to the Solovay-Kitaev Theorem in Theorem 2.4, we can conclude that any s-gate circuit using gates from $\mathcal{G}$ can be ϵ-approximated by an $(s \cdot \mathsf{polylog}\,\frac{s}{\epsilon})$-gate circuit from another universal gate set. Hence, the circuit complexity only modestly changes when considering different gate sets.

Claim 3.3. *Fix two universal gate sets $\mathcal{G}$ and $\mathcal{G}'$. Suppose that there exists a s-gate circuit $\mathcal{C}$ that uses gates from $\mathcal{G}$ such that for all x, $\|(\langle f(x)| \otimes I_{n+t-1})\mathcal{C}|x, 0^t\rangle\| \geq 1-\delta$. Then, there exists another circuit $\mathcal{C}'$ that uses $s \cdot \mathsf{polylog}\,\frac{s}{\epsilon}$ gates in $\mathcal{G}'$ such that $\|(\langle f(x)| \otimes I_{n+t-1})\mathcal{C}|x, 0^t\rangle\| \geq 1 - \delta - \epsilon^2/2$.*

Proof. The proof simply follows from the Solovay-Kitaev Theorem in Theorem 2.4. The only subtlety is that the distance measure in Theorem 2.4 is L_2 norm distance. However, for any two states $|\psi\rangle$ and $|\phi\rangle$, we have $|\langle\psi|\phi\rangle| \geq 1 - \frac{1}{2}\||\psi\rangle - |\phi\rangle\|^2$. Thus, we can obtain the lower bound for $\|(\langle f(x)| \otimes I_{n+t-1})\mathcal{C}|x, 0^t\rangle\|$ by using the L_2 norm between $\mathcal{C}$ and $\mathcal{C}'$. □

In this work, we mainly focus on arbitrary gate sets containing one- and two-gates and $|\mathcal{G}| = O(1)$. However, for some applications, we may require a particular gate set such as $\{\mathsf{Toffoli}, \mathsf{H}\}$. We will specify $\mathcal{G}$ when it is necessary. We assume $t \leq s$ without loss of generality since we mainly consider the gate set $\mathcal{G}$ to have one- and two-qubit gates. Specifically, if there are more than s ancilla qubits, there must be ancilla qubits that are not used by any gate.

We define the problem as a promise problem for two reasons: first, applying measurements on quantum states generally gives probabilistic outputs. Similar to many probabilistic algorithms, we say a quantum algorithm solves a problem if it outputs the answer with high probability in general. Check the definition of BQP for an example. Along this line, we expect a quantum circuit $\mathcal{C}$ to implement the given Boolean function f with high probability, i.e., for each input x, the circuit outputs $f(x)$ with high probability. The second reason is about verifying the circuit. Consider the case where $\mathcal{C}$ only fails on one x with success probability $2/3 - \epsilon$, where ϵ is some extremely small number. In this case, it is hard to verify the circuit efficiently. Therefore, we require a gap for efficient verification and say that $\mathcal{C}$ does not implement f if it can only output $f(x)$ with probability with small probability for some x.

Other variants. In many applications, the *gap-version* of MCSP is much easier and more flexible to work with. Below we define the gap-version of MQCSP and the multi-output MQCSP.

Definition 3.4 ($\mathsf{MQCSP}_{a,b}[s, s', t]$)**.** *Let $n, s, s', t \in \mathbb{N}$ such that $t \leq s < s' \leq 2^{O(n)}$. Let $a - b \geq 1/\mathsf{poly}(2^n, 1^{|s|})$. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. $\mathsf{MQCSP}[s, s']$ is a promise problem defined as follows.*

- *Input: the truth table $\mathsf{tt}(f)$ of f, the size parameter s in unary, and the ancilla parameter t.*
- *Yes instance: there exists a quantum circuit $\mathcal{C}$ using at most s gates and operating on at most $n + t$ qubits such that for all $x \in \{0,1\}^n$,*
$$\|(\langle f(x)| \otimes I_{n+t-1})\mathcal{C}|x, 0^t\rangle\|^2 \geq \frac{2}{3}.$$
- *No instance: for every quantum circuit $\mathcal{C}$ using at most s' gates and operating on at most $n + t$ qubits, there exists $x \in \{0,1\}^n$ such that*
$$\|(\langle f(x)| \otimes I_{n+t-1})\mathcal{C}|x, 0^t\rangle\|^2 \leq \frac{1}{2}.$$

With the promise that the input must be either a yes instance or a no instance, the problem is to decide whether the input is a yes instance or not.

When it is clear from the context, we may use $\mathsf{MQCSP}^\star$ to denote $\mathsf{MQCSP}_{a,b}[s, s', t]$.

Definition 3.5 ($\mathcal{G}$-$\mathsf{multiMQCSP}_{\alpha,\beta}(s,t)$)**.** *Let m, s, t be functions of n such that $t \leq s \leq 2^{o(n)}$ and $m \leq n+t$. Let $\alpha, \beta \in [2^{-m}, 1]$ such that $\alpha - \beta > \frac{1}{\mathsf{poly}(2^n)}$. Let $f : \{0,1\}^n \to \{0,1\}^m$ be a multioutput function. $\mathcal{G} - \mathsf{multiMQCSP}_{\alpha,\beta}(s,t)$ is a promise problem that*

1. *Input: the truth table $\mathsf{tt}(f)$ of f.*
2. *Yes instance: there exists a quantum circuit $\mathcal{C}$ using at most s gates from $\mathcal{G}$ and operating on at most $n+t$ qubits such that for all $x \in \{0,1\}^n$,*

$$\|(\langle f(x)| \otimes I_{n+t-m})\mathcal{C}|x, 0^t\rangle\|^2 \geq \alpha,$$

3. *No instance: for any quantum circuit $\mathcal{C}$ using at most s gates from $\mathcal{G}$ and operating on at most $n+t$ qubits, there exists $x \in \{0,1\}^n$ such that*

$$\|(\langle f(x)| \otimes I_{n+t-m})\mathcal{C}|x, 0^t\rangle\|^2 \leq \beta.$$

With the promise that the input must be either a yes instance or a no instance, the problem is to decide whether the input is a yes instance or not.

Natural property. It is worth noting that we can view an efficient quantum algorithm for MQCSP as *quantum natural property against quantum circuit classes.* Natural properties against circuit classes were first defined by Razborov and Rudich [RR97], and recently, Arunachalam et al. [AGG$^+$20] further considered quantum natural properties against circuit classes.

Definition 3.6 (Natural Property [RR97])**.** *Let C be a uniform complexity class and C' be a circuit class. We say that a property $\Gamma = \{\Gamma_n : n \in \mathbb{N}\}$ is C-natural against C' if the following holds.*

1. ***Constructivity:*** *for all $L \in \Gamma$, $L \in C$.*
2. ***Largeness:*** *There exists $n_0 \in \mathbb{N}$, for $n \geq n_0$, $|\Gamma_n|/|\mathcal{F}_n| \geq \frac{1}{2}$, where $\mathcal{F}_n$ is the set of all Boolean functions with input length n.*
3. ***Usefulness:*** *There exists $n_0 \in \mathbb{N}$, for $n \geq n_0$, $\Gamma_n \cap C'_n = \emptyset$, where C'_n is the set of circuits in C' on n (qu)bits.*

Note that an MQCSP oracle can be used to construct natural properties against quantum circuit classes $\mathsf{BQC}[s]$ for any s. Therefore, if we suppose that MQCSP is in BQP, then we can have properties that are BQP-natural against quantum circuit classes. For simplicity, we call properties that are BQP-natural as *quantum natural properties.* Arunachalam et al. [AGG$^+$20] first considered quantum natural properties against circuit classes, and proved circuit lower bounds for quantum complexity classes. Our work can also be viewed as a study of quantum natural properties against quantum circuit classes. The formal definition of BQP-natural property is in below:

Definition 3.7 (BQP-Natural Property [AGG$^+$20])**.** *We say that a combinatorial property Γ is C-natural against polynomial-size quantum circuits* ($\mathsf{BQC}[\mathsf{poly}]$) *if the following holds.*

1. ***Constructivity:*** *for any string $L \in \Gamma$, L can be accepted by a BQP algorithm.*
2. ***Largeness:*** *There exists $n_0 \in \mathbb{N}$, for $n \geq n_0$, $\frac{|\Gamma_n|}{|\mathcal{F}_n|} \geq \frac{1}{2}$.*

3. ***Usefulness:*** *There exists $n_0 \in \mathbb{N}$, for $n \geq n_0$, any string accepted by* $\mathsf{BQC}[\mathsf{poly}]$ *is not in* Γ_n.

Then, our observation on the connection between MQCSP and quantum natural property is formally stated as follows:

Observation 1. *If* MQCSP $\in$ BQP*, then there exists a* BQP*-natural property against quantum circuits* $\mathsf{QC}[n^k]$ *for any $k \in \mathbb{N}_+$.*

3.2 Upper bounds for MQCSP

It turns out that, unlike the classical MCSP, MQCSP is not trivially in NP. The best upper bound we are able to get for MQCSP is QCMA, the quantum analogue of NP (or MA). Before showing that MQCSP is in QCMA, we first discuss why it is not trivially in NP like the classical MCSP. One obvious reason is that MQCSP is a promise problem. Therefore, we consider PromiseNP, which definition is the same as NP except that PromiseNP relax the definition of NP to contain promise problems that have NP certificates. For the ease of presentation, we will use NP for both NP and PromiseNP. Then, when the number of ancilla qubits is linear, one can verify the given circuit by simply writing down the corresponding unitary.

Theorem 3.8. MQCSP *is in* NP *when only a linear number of ancilla qubits are allowed.*

However, when the number of ancilla qubits is superlinear, e.g., n^2, the quantum circuit $\mathcal{C}$ operates on $2^{O(n^2)}$ qubits, and thus the corresponding unitary $U_{\mathcal{C}}$ has dimension $2^{O(n^2)}$ which is superpolynomial in 2^n. In this case, the verifier cannot compute $U_{\mathcal{C}}$ classically in time $\mathsf{poly}(2^n)$. Therefore, the trivial approach does not work.

Note that although the trivial approach fails to show that MQCSP is in NP, it does not rule out the possibility that MQCSP can be efficiently verified via other approaches. In the following theorem, we show that a quantum verifier can efficiently verify the given quantum circuit, and thus MQCSP is in QCMA.

Theorem 3.9. MQCSP $\in$ QCMA*.*

We leave the proof to Appendix A for completeness.

3.3 Hardness of quantum MCSP

It is a major open problem in complexity theory to understand the hardness of classical MCSP. Here, we show that the state-of-the-art hardness results on MCSP (and its variants) can be extended to MQCSP. We remark that this is actually not straightforward to see because the classical MCSP is incomparable with MQCSP.

First, we show that the SZK-hardness result of MCSP by Allender and Das [AD14] can be extended to MQCSP. Here, SZK stands for the complexity class *Statistical Zero Knowledge* that lies between P and NP. We first define SZK and the statistical distance as follows.

Definition 3.10 (Statistical Distance $SD(X,Y)$)**.** *Let X and Y be two probability distributions, the statistical distance between X and Y can be defined as follows:*

$$\max_{S \subseteq \{0,1\}^{m'}} |\Pr[X \in S] - \Pr[Y \in S]|$$

Definition 3.11 (SZK)**.** *A promise problem $P = (P_Y, P_N)$ is in* SZK *if there exists a* PPT *verifier V and an interactive proof system (P,V) satisfying the following properties:*

1. ***Completeness:*** *For $x \in P_Y$, there exists P such that* $\Pr[\langle P, V\rangle(x) = 1] \geq \frac{2}{3}$.
2. ***Soundness:*** *For $x \in P_N$, for all P,* $\Pr[\langle P, V\rangle(x) = 1] \leq \frac{1}{3}$.
3. ***Statistical zero-knowledge:*** *There exists a* PPT *simulator S, for all* PPT *verifier V^*, for all $x \in P_Y$,*
$$\mathsf{SD}(S(V^*)(x), \langle P, V^*\rangle(x)) \leq \mathsf{negl}(n).$$

We introduce an SZK-complete problem by Ben-Or and Gutfreund [BOG08].

Definition 3.12 (Polarized Image Intersection Density (PIID), [BOG08])**.** *Given two circuits $C_0, C_1 : \{0,1\}^m \to \{0,1\}^{m'}$ of size n^k with the promise that either*

1. $\max_{S\subseteq\{0,1\}^{m'}} |\Pr_x[C_0(x) \in S] - \Pr_x[C_1(x) \in S]| \leq \frac{1}{2^n}$, *or*
2. $\Pr_{x\in\{0,1\}^{m'}}[\exists y \in \mathsf{Im}(C_0)$ *such that* $C_1(x) = y] \leq \frac{1}{2^n}$,

where $n = \mathsf{poly}(m)$ and $\mathsf{Im}(C) := \{C(x) : x \in \{0,1\}^m\}$. The problem is to decide which case is true.

Theorem 3.13. $\mathsf{SZK} \subseteq \mathsf{BPP}^{\mathsf{MQCSP}}$

To prove Theorem 3.13, we first observe that the existence of small classical circuit implies the existence small quantum circuits and an MQCSP oracle can invert one-way functions (which we will prove in Section 4.1.1). Then, we can show that PIID is in $\mathsf{BPP}^{\mathsf{MQCSP}}$ following the framework of [AD14]. We leave the proof to Appendix A for completeness.

Next, we quantize the recent breakthrough of Ilango et al. [ILO20] on the NP-hardness of classical MCSP. There are two main differences between the classical and quantum settings: (i) the circuit model is different and hence makes the combinatorics different, and (ii) the quantum setting allows the output to have some errors. We partially overcome these two difficulties and prove the following theorem.

Theorem 3.14. *Suppose* $\mathsf{CNOT} \circ (\mathsf{I} \otimes \mathsf{X}), \mathsf{Toffoli} \in \mathcal{G}$*. Every multi-bit gate in $\mathcal{G}$ behaves classically on classical inputs and has at most 1 target wire and at most 2 control wire. (That is, except 1 wire, the outputs of the other wires, at most 2, are the same as their corresponding classical inputs. For example,* CNOT *gate.) Then $\mathcal{G}$-*$\mathsf{multiMQCSP}$ *is* NP*-hard under randomized reduction.*

$\mathsf{CNOT} \circ (\mathsf{I} \otimes \mathsf{X})$ is the following operation on two input wires, denoted as control wire and target wire: first do a X on the target wire, and do a CNOT from the control wire to the target wire. We consider it as a single gate, as the analog of the classical NOT gate.

Here the choice of gate set matter: we need the quantum gate set to contain the analog of the usual classical gate set. $\mathsf{CNOT} \circ (\mathsf{I} \otimes \mathsf{X})$ is the analog of classical single-bit NOT operation, and $\mathsf{Toffoli}$ is the analog of classical AND operation. Here the correspondence has two properties: (1) if the target wire is in the zero state and the control wire is classical, the output of the target wire will be the corresponding classical logical computation result; (2) if the input of the control wire is classical, the output of the control wire will remain the same. Since in the quantum world data copy is not for free, the second property is important for deriving our result.

The proof follows the outline of the proof in [ILO20]. We note there are two differences during the proof in the quantum case compared to the classical case:

- The circuit model is different. In the classical world the gates are single-output and we assume free-copy. And the basic gate set contains AND, OR, NOT gates. In quantum world, data copy is not for free and we need to use the $\mathsf{Toffoli}$ gate to implement the AND/OR gates.

  Remark 3. *One idea might be to use the Solovay-Kitaev theorem to switch the gate set and make the theorem general. But this does not work here in an immediate way. Our proof does not imply the problem is also* NP*-hard to approximate multiplicatively. On the other hand, the classical result [ILO20] is not known to be general on different gate set either.*

- In the definition of multi-output minimum quantum circuit size problem, we allow the output to have some errors, which is not considered in the classical world.

Proof of Theorem 3.14. We consider the same construction as [ILO20]. Let's restate it here for completeness.

1. Choose a large enough constant r so that 20-approximating r-bounded set cover problem is NP-hard. Consider an instance $(1^n, \mathcal{S})$ of this problem.

2. m is the least power of 2 that is greater than n^3. Sample the truth table T representing a function on $\{0,1\}^{\log m} \to \{0,1\}$ uniformly at random. Construct $g := \bullet_{S\in\mathcal{S}}\text{Eval-DNF}_{T_{\langle S^m\rangle}}$ where:

   - To define DNF_f that encode the truth table f, we first repeat the construction in [ILO20] for completeness:

   $$\text{DNF}_f := ((x_1 = y_1^1) \wedge \cdots \wedge (x_n = y_n^1)) \vee \cdots \vee ((x_1 = y_1^t) \wedge \cdots \wedge (x_n = y_n^t))$$

     where $y^1, \cdots y^t$ are YES inputs of f in lexicographical order, $x_1, \cdots x_n$ index the bits of the input string x, $y_1^j, \cdots y_n^j$ index the bits of y^j, and $(x_i = y_i^t)$ denotes $(x_i \oplus (1 \oplus y_i^j))$. We use the same construction with one difference: here $\vee$ is further decomposed to $\neg$ and $\wedge$.
   - $T_{\langle S\rangle}$ is the truth table that is equal to T for input in S and 0 everywhere else.
   - $S^m := \cup_{i\in S} P_i^{m,n}$ where $P_i^{m,n} := \{j \in [m] : j \equiv i \mod n\}$. This step closes the gap between $[m]$ (the MCSP size) and $[n]$ (the set cover size).
   - "$\bullet$" is used on two functions that have the same input domain, and it concatenates the outputs of these functions to get a new function.
   - To define Eval-C, we first consider $x_1 \bullet x_2 \bullet \cdots x_n \bullet g_1 \bullet g_2 \bullet \cdots g_s$ where $g_1, \cdots g_s$ are the output of each gate in circuit C. Then we remove the gate output that are the same on all the inputs.

3. As in [ILO20], define k as the number of distinct components of g that are not directly a function identical to an input. Note that this can be efficiently computed.

   Take $\alpha = 1, \beta = 0.99, t = 10s$ (s is the output number of our construction).

   Define $CC_{\alpha,\beta}(t, \mathsf{tt}(f))$ as the subroutine that uses binary search to find the minimum s such that $\mathcal{G} - \mathsf{multiMQCSP}_{\alpha,\beta}(s,t)(\mathsf{tt}(f)) = \text{true}$.[20] Use the $\mathsf{multiMQCSP}$ oracle and compute

   $$\Delta := CC_{\alpha,\beta}(t, \mathsf{tt}(T \bullet g)) - k$$

   as the approximation of the set cover instance $(1^n, \mathcal{S})$.

[20] Since multiMQCSP is a promise problem this routine does not necesarrily find the minimum s but should return a value that there exists a circuit of this size that approximate the function everywhere with correct probability β. This is sufficient for later proof.

To analyze this reduction, we need to prove the followings steps:

1. $CC_{\alpha,\beta}(t, \mathsf{tt}(g)) = k$
2. $\Delta \leq 3 \cdot \mathrm{cover}([n], \mathcal{S}) + 1$ where $\mathrm{cover}([n], \mathcal{S})$ is the size of the minimum set cover solution for $\mathcal{S}$.
3. $\Delta \geq \mathrm{cover}([n], \mathcal{S})/6 - 6$ with probability $1 - 2^{-\Omega(m)}$.

Then we get an approximation to the set cover problem.

Let us prove the three statements step-by-step.

Step 1: The $\leq$ part is proved by the function construction itself. We implement $\neg$ with the $\mathsf{CNOT} \circ (\mathsf{I} \otimes \mathsf{X})$ gate (and write the output on an empty ancilla system) and implement $\wedge$ with the $\mathsf{Toffoli}$ gate.

The $\geq$ part is slightly different since in quantum case the gate model is different. In classical world all the gates are single-output, while in quantum world there are multi-output gates. However, for the multi-output gates like CNOT and $\mathsf{Toffoli}$, there is only one target wire, and the other wires are control wire. Thus for each output component, we can always find the nearest gate that does not use it as a control wire (if there is such a gate along the way, ignore it). In this way each different output component corresponds to a different gate in the circuit, which completes the proof.

Step 2: As [ILO20], when $\mathrm{cover}([n], \mathcal{S}) = \ell$, without loss of generality assume $S_1, \cdots S_\ell$ are a set cover. Then $T = T_{\langle S_1^m \rangle} \vee \cdots \vee T_{\langle S_\ell^m \rangle}$. This can be computed using $3\ell + 1$ extra gates on the minimum circuit of Eval-g. (Note that in the quantum world we need slightly more gates than the classical world. And we need to evaluate the OR gate by NOT-AND-NOT gates to get T.)

Step 3: Denote $\ell = \lfloor \mathrm{cover}([n], \mathcal{S})/6 \rfloor$. The goal is to show that the probability that $\Delta \leq \ell$ is small by showing that T satisfying $\Delta \leq \ell$ must have a short description. Suppose T is a truth table such that the condition $\Delta > \ell$ does not hold. We need to find a circuit of gate number $\leq 2\ell$ where:

- The inputs are: the bits of x; and the output of g.
- It encodes the output of T.

We use the similar idea to [ILO20] but we need to address the two problems discussed before this proof.

As what we did in Step 1, we can associate each output component ($g_i(x)$, for example) to a unique gate in the circuit. As [ILO20], we remove these gates from the circuit. There might be some gates between this gate and the output $g_i(x)$ that use the wire as control wires. For these gates, simply use $g_i(x)$ as the control value.

As in [ILO20] we have $CC_{\alpha,\beta}(t, \mathsf{tt}(T \bullet g)) \leq \ell + k$. And since for each g_i at least one gate is removed, the remaining circuit is a circuit D that takes $\log(m) + k$ inputs and has at most ℓ gates such that

$$D(x, g_1(x), \cdots g_k(x)) \text{ encodes } T(x)$$

Then since each gate has fan-in at most 3 the circuit uses at most 3ℓ components of g. Then after a possible relabling of $g_1 \cdots g_k$ we can assume D takes $\log(m) + 3\ell$ inputs such that

$$D(x, g_1(x), \cdots g_{3\ell}(x)) \text{ encodes } T(x)$$

The new circuit does not necessarily behave the same as the original circuit, but they do behave the same (up to a global phase) on the subspace that all the outputs are computed correctly. By the definition of multiMQCSP and the choices of parameters this is true with norm ≥ 0.99. Thus we can view the shrinked circuit as an encoding of T by focusing on the most-possible outputs of this circuit. Then by the same argument as [ILO20] such a shrinked circuit has a description of $(1-\Omega(1))m$ bits, which implies such T has at most $2^{(1-\Omega(1))m}$ choices thus a random T falls into this case with exponentially small probability. □

However, we don't know whether this problem is NP-complete, since it's not known to be in NP. With a proof similar to that of Theorem 3.9, we only know multiMQCSP $\in$ QCMA. Namely, there remains a gap between our understandings of the upper bound and hardness of multiMQCSP. We pose it as an open problem to settle the complexity of multiMQCSP.

4 Connections Between MQCSP and Other Problems

4.1 Cryptography and MQCSP

Classically, we have already known connections between MCSP and one-way functions [KC00, RR97] and indistinguishable obfuscation [IKV18]. In this section, we show the quantum analogies of these results.

4.1.1 Quantum cryptographic primitives

We first introduce relevant primitives in cryptography.

Definition 4.1 (Pseudorandom Generator (PRG))**.** *Let $G : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. Let $\ell : \mathbb{N} \to \mathbb{N}$ be a polynomial-time computable function such that $\ell(n) > n$ for all n. G is a pseudorandom generator of stretch $\ell(n)$ if it satisfies:*

1. *$|G(x)| = \ell(|x|)$ for all $x \in \{0,1\}^*$, and*
2. *for all Probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$, there exists a negligible function $\epsilon : \mathbb{N} \to [0,1]$ such that for all $n \in \mathbb{N}$*

$$\left| \Pr_{x \sim \{0,1\}^n}[\mathcal{A}(G(x)) = 1] - \Pr_{y \sim \{0,1\}^{\ell(n)}}[\mathcal{A}(y) = 1] \right| \leq \epsilon(n).$$

We say that a PRG is *local* if every output bit of the PRG can be computed in time poly(n). In the following, we define PRG secure against any quantum polynomial-time adversary.

Definition 4.2 (Quantum-Secure Pseudorandom Generator (qPRG))**.** *Let $G : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function[21]. Let $\ell : \mathbb{N} \to \mathbb{N}$ be a polynomial-time computable function such that $\ell(n) > n$ for all n. G is a pseudorandom generator secure against quantum adversaries of stretch $\ell(n)$ if it satisfies:*

1. *$|G(x)| = \ell(|x|)$ for all $x \in \{0,1\}^*$, and*

[21] It is worth noting that G can be any function that is efficiently computable in either quantum or classical polynomial time.

2. *for all quantum polynomial-time (QPT) algorithm $\mathcal{A}$, there exists a negligible function $\epsilon : \mathbb{N} \to [0,1]$ such that for all $n \in \mathbb{N}$*

$$\left| \Pr_{x \sim \{0,1\}^n}[\mathcal{A}(G(x)) = 1] - \Pr_{y \sim \{0,1\}^{\ell(n)}}[\mathcal{A}(y) = 1] \right| \leq \epsilon(n).$$

In this work, we consider two ways of constructing quantum-secure PRGs based on different cryptographic primitives. One is based on the quantum-secure one-way functions and the other one is based on the hard function.

Definition 4.3 (Quantum-Secure One-Way function (qOWF))**.** *A function $f : \{0,1\}^* \to \{0,1\}^*$ is a quantum-secure one-way function, if the following conditions hold: For every $n \in \mathbb{N}$, for any $x \in \{0,1\}^n$ picked uniformly at random,*

1. *There exists a* $\mathsf{poly}(n)$*-time deterministic algorithm for computing f.*
2. *For any* $\mathsf{poly}(n)$*-time quantum algorithm $\mathcal{A}'$, $\Pr_x[\mathcal{A}'(f(x)) \in f^{-1}(f(x))] = \mathsf{negl}(n)$.*

Definition 4.4 (GGM Construction [GGM86])**.** *Let $G : \{0,1\}^n \to \{0,1\}^{2n}$ be a* (q)PRG*. For every $z \in \{0,1\}^m$, the GGM construction of a pseudorandom function family $\{h_z : \{0,1\}^n \to \{0,1\}^n\}_{z \in \{0,1\}^m}$ is defined as follows:*

$$f_z(x) = G_{z_m} \circ G_{z_{m-1}} \circ \cdots \circ G_{z_1}(x),$$

where we denote by $G_0(x)$ the first n bits of G, and by $G_1(x)$ the last n qubits.

Lemma 4.5 ([HILL99])**.** *If* OWF*s exist, then for every $c \in \mathbb{N}$, there exists a secure* PRG *with stretch $\ell(n) = n^c$.*

Since the security proof of Lemma 4.5 is black-box, the analysis carries over to the quantum setting directly if the one-way function is secure against quantum adversaries. Therefore, we can obtain Lemma 4.6.

Lemma 4.6 (Folklore)**.** *If* qOWF*s exist, then for every $c \in \mathbb{N}$, there exist* qPRG*s with stretch $\ell(n) = n^c$.*

Lemma 4.7. *Suppose that there exists a* qPRG *$G : \{0,1\}^n \to \{0,1\}^{2n}$. Then, for $m = O(\log n)$, there exists a local* qPRG *$\hat{G} : \{0,1\}^n \to \{0,1\}^{2^m}$.*

Proof. We first give the construction of $\hat{G}$. Follow the GGM construction in Definition 4.4, we let

$$h'_x(z) = G_{z_m} \circ G_{z_{m-1}} \circ \cdots \circ G_{z_1}(x)$$

where $z \in \{0,1\}^m$, $x \in \{0,1\}^n$. We let $h_x(z)$ be the first output bit of $h'_x(z)$ and define the qPRG as

$$\hat{G}(x) = h_x(0) \mid h_x(1) \mid \cdots \mid h_x(2^m - 1).$$

It is obvious that each bit of $\hat{G}(x)$ can be computed in time m times the runtime of G.

We then prove that $\hat{G}(x)$ is indistinguishable from a truly random string by the standard hybrid approach. For $i \in [m]$, we define

$$H^i(z) = (G_{z_m} \circ G_{z_{m-1}} \circ \cdots \circ G_{z_i}(y_{z,i}))_1,$$

where $y_{z,i}$ is drawn independently and uniformly randomly from $\{0,1\}^n$. Note that $H^1(z) = h_z(x)$ and $H^m(z)$ is a random bit. Let

$$\hat{G}^i = H^i(0) \mid H^i(1) \mid \cdots \mid H^i(2^m - 1) \quad \forall i \in [m].$$

Suppose that there exists a QPT algorithm $\mathcal{A}$ such that

$$\left| \Pr_{x \sim \{0,1\}^n} [\mathcal{A}(\hat{G}(x)) = 1] - \Pr_{u \sim \{0,1\}^{2^m}} (\mathcal{A}(u)) \right| \geq 1/\mathsf{poly}(n).$$

Then, by the triangular inequality,

$$\sum_{i=1}^{m-1} \left| \Pr[\mathcal{A}(\hat{G}^i) = 1] - \Pr[\mathcal{A}(\hat{G}^{i+1}) = 1] \right| \geq 1/\mathsf{poly}(n)$$

which implies that there exists i^* such that $|\Pr[\mathcal{A}(\hat{G}^{i^*}) = 1] - \Pr[\mathcal{A}(\hat{G}^{i^*+1}) = 1]| \geq 1/\mathsf{poly}(n)$. Since distinguishing $\hat{G}^{i^*}$ and $\hat{G}^{i^*+1}$ implies that one can distinguish $G(x)$ from a random string, G is not a qPRG. This completes the proof.

□

4.1.2 Implications for quantum-secure one-way functions (qOWF)

Here, we show a quantum analogous result for [KC00, RR97] by considering the implication of the existence of efficient quantum algorithms for either classical or quantum MCSP.

Theorem 4.8. *If* $\mathsf{MQCSP} \in \mathsf{BQP}$*, then there is no quantum-secure one-way function* (qOWF)*.*

Proof. Let $f : \{0,1\}^* \to \{0,1\}^*$ be any function. By Lemma 4.6, we construct $G_f : \{0,1\}^n \to \{0,1\}^{n^a}$ that is a qPRG if f is a qOWF. We denote the runtime for G_f as $O(n^b)$ for some constant b.

Given G_f, we construct a qPRG $\hat{G} : \{0,1\}^n \to \{0,1\}^{2^m}$ where $m = O(\log n)$ by Lemma 4.7. Then, we view the outputs of $\hat{G}(x)$ as a truth table of some Boolean function $g_x : \{0,1\}^m \to \{0,1\}$. Note that according to the construction in Lemma 4.7, the time for evaluating g_x on $z \in \{0,1\}^m$ is $O(m \cdot n^b) = \tilde{O}(n^b)$. On the other hand, for a random Boolean function from $\{0,1\}^m$ to $\{0,1\}$, we know from Claim F.1 that its circuit complexity is greater than $\frac{2^m}{(c+1)m}$ with high probability. Therefore, by setting $m = d \log n$ for some constant $d \gg b$, the circuit complexity of the random function is $\tilde{O}(n^d) \gg \tilde{O}(n^b)$ with high probability.

Algorithm 1 A quantum algorithm for breaking qPRG

```
Input: Given tt(h) for $h : \{0,1\}^m \to \{0,1\}$ constructed from $\hat{G}$ in Lemma 4.7.
1: Runs the quantum algorithm for MQCSP with $s = \frac{2^m}{(c+1)m}$
2: return "Yes" if the algorithm in previous step outputs yes.
3: return "No", otherwise
```

Since we assume $\mathsf{MQCSP} \in \mathsf{BQP}$, we obtain a quantum polynomial-time algorithm $\mathcal{A}$ for distinguishing $\{g_x\}_{x \in \{0,1\}^n}$ and the random function family $\mathcal{F}_m$ as in Algorithm 1. The circuit complexity for g_x is at most $\tilde{O}(n^b)$ and the for a random function h is greater than $\frac{2^m}{(c+1)m} = \tilde{O}(n^d)$ for $d \gg b$. thus, we obtain

$$\left| \Pr_{x \sim \{0,1\}^n} [\mathcal{A}(\mathsf{tt}(g_x)) = 1] - \Pr_{h \sim \mathcal{F}_m} [\mathcal{A}(\mathsf{tt}(h)) = 1] \right| \geq 1/\mathsf{poly}(n).$$

This implies that we can use $\mathcal{A}$ to break G in quantum polynomial time by Lemma 4.7. Finally, by Lemma 4.6, we obtain a quantum polynomial-time algorithm $\mathcal{A}_{inv}$ for inverting any f. □

4.1.3 Implication for quantum-secure $i\mathcal{O}$

In this section, we use Theorem 4.8 and quantum-secure $i\mathcal{O}$ to show that if MQCSP can be solved by a BQP algorithm, then $\mathsf{NP} \subset \mathsf{coRQP}$, which is the class of one-sided error quantum polynomial-time algorithms such that a "Yes" instance will always be accepted while a "NO" instance will be rejected with high probability.

We define the quantum-secure $i\mathcal{O}$ as follows:

Definition 4.9 (Quantum-secure indistinguishability obfuscation, $i\mathcal{O}$)**.** *A probabilistic polynomial-time machine iO is an indistinguishability obfuscator for a circuit class $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ if the following conditions are satisfied for all $\lambda \in \mathbb{N}$:*

- ***Functionality:*** *For any $C \in \mathcal{C}_\lambda$, for all inputs x, $i\mathcal{O}(C)(x) = C(x)$.*
- ***Indistinguishability:*** *For any $C_1, C_2 \in \mathcal{C}_\lambda$ such that $|C_1| = |C_2|$ and $C_1(x) = C_2(x)$ for all inputs x, any quantum polynomial-time distinguisher $\mathcal{A}$ cannot distinguish the distributions $i\mathcal{O}(C_1)$ and $i\mathcal{O}(C_2)$ with noticeable probability, i.e., $\big|\Pr[\mathcal{A}(i\mathcal{O}(C_1)) = 1] - \Pr[\mathcal{A}(i\mathcal{O}(C_2)) = 1]\big| \le \mathsf{negl}(\lambda)$.*

Remark 4. *We note that there are some (candidate) constructions of post-quantum $i\mathcal{O}$, based on different assumptions. For example, [BDGM20] constructed $i\mathcal{O}$ based on the circular security of LWE-based encryption schemes, which is conjectured to be quantum-secure. [WW20] showed a construction of $i\mathcal{O}$ based on the indistinguishability of two distributions which is also arguably quantum-secure.*

Theorem 4.8 implies the following result for quantum-secure $i\mathcal{O}$:

Theorem 4.10. *Suppose that quantum-secure $i\mathcal{O}$ for polynomial-size circuits exists. Then,* $\mathsf{MQCSP} \in \mathsf{BQP}$ *implies* $\mathsf{NP} \subseteq \mathsf{coRQP}$.

Proof. Let $f_C(r) := i\mathcal{O}(C, r)$, where r is the random string. Then, by Theorem 4.8, we know that there exists a quantum polynomial-time algorithm $\mathcal{A}_{inv}$ with access to an MQCSP oracle and a non-negligible function p such that for any circuit C,

$$\Pr_r\left[f_C(\mathcal{A}_{inv}^{\mathsf{MQCSP}}(C, i\mathcal{O}(C, r))) = f_C(r)\right] \ge p(|r|). \tag{1}$$

Then, we can use $\mathcal{A}_{inv}$ to solve the $\mathsf{Circuit\text{-}SAT}$ problem. The algorithm is as follows:

Algorithm 2 A quantum algorithm for $\mathsf{Circuit\text{-}SAT}$

```
Input: The description of a circuit $C : \{0,1\}^n \to \{0,1\}$.
$s \leftarrow |C|$.
Compute $\bot_s$.                                  ▷ A canonical unsatisfiable circuit
$\hat{C} \leftarrow i\mathcal{O}(C, r)$.
$r' \leftarrow \mathcal{A}_{inv}^{\mathsf{MQCSP}}(\bot_s, \hat{C})$.
return "No" if $\hat{C} = i\mathcal{O}(\bot_s, r')$.
```

We assume that for any $s \ge 0$, we can compute a canonical unsatisfiable circuit of size s in $\mathsf{poly}(s)$ time.

If $C \in \mathsf{UNSAT}$, then $C \equiv \bot_s$. If $C = \bot_s$, by Eq. (1), $\mathcal{A}_{inv}^{\mathsf{MQCSP}}$ finds r with probability at least $p(|r|)$. Otherwise, by the indistinguishability of $i\mathcal{O}$ and $\mathsf{MQCSP} \in \mathsf{BQP}$, $\mathcal{A}_{inv}^{\mathsf{MQCSP}}$ is a quantum polynomial-time algorithm and hence cannot distinguish $C \in \mathsf{UNSAT} \setminus \{\perp_s\}$ and $\perp_s$ with more than $\mathsf{negl}(|r|)$ probability. Therefore, Algorithm 2 will reject C with probability $O(p(|r|))$.

If $C \in \mathsf{SAT}$, then $C \not\equiv \perp_s$. By the functionality of $i\mathcal{O}$, for any r, r', $i\mathcal{O}(C, r) \neq i\mathcal{O}(\perp_s, r')$. Hence, Algorithm 2 will always accept C.

Hence, by repeatedly running Algorithm 2 many times, we get that $\mathsf{NP} \subset \mathsf{coRQP}$, the one-sided error analog of BQP □

Remark 5. *It is worth noting that in the classical setting, the existence of $i\mathcal{O}$ implies that* NP *and* MCSP *are equivalent under randomized reductions; the other direction directly follows from the fact that* $\mathsf{MCSP} \in \mathsf{NP}$*. However, since it is unclear if* $\mathsf{MQCSP} \in \mathsf{NP}$*, we can only conclude that* $\mathsf{NP} \subseteq \mathsf{RQP}^{\mathsf{MQCSP}}$ *assuming the existence of quantum-secure $i\mathcal{O}$.*

4.2 Learning theory

In this section, we discuss connections between MQCSP and learning theory. We consider two standard settings: probably approximately correct (PAC) learning and quantum learning. We postpone the details to Appendix B.

PAC learning. Let C be a circuit class. We are interested in how to efficiently learn a function in C. PAC learning is a theoretical framework to evaluate how well a learning algorithm is. Here we focus on a special setting of PAC learning where the algorithm is able to query any input to the unknown function. In the following, we denote C-MCSP as the classical MCSP problem with respect to the circuit class C.

Definition 4.11 (PAC learning over the uniform distribution with membership queries)**.** *Let* C *be a circuit class and let $\epsilon, \delta > 0$. We say an algorithm (ϵ, δ)-PAC-learns* C *over the uniform distribution with membership queries if the following hold. For every $n \in \mathbb{N}$ and n-variate $f \in$* C*, given membership query access to f, the algorithm outputs a circuits C such that with probability at least $1 - \delta$ over its internal randomness, we have $\Pr_{x \in \{0,1\}^n}[f(x) \neq C(x)] < \epsilon$. The running time of the learning algorithm is measured as a function of $n, 1/\epsilon, 1/\delta$ and, $\mathsf{size}(f)$.*

The seminal paper of Carmosino et al. [CIKK16] showed that efficient PAC learning for a (classical) circuit class C is *equivalent* to the corresponding MCSP being easy. Here, we quantize this connection and show in the following theorem that efficient PAC-learning for $\mathsf{BQP/poly}$ is equivalent to efficient algorithm for MQCSP. Here, $\mathsf{BQP/poly}$ is defined as $\bigcup_{s \leq \mathsf{poly}(n)} \mathsf{BQC}(s)$.

For technical reason, we need to work on a gap version of MQCSP in one direction of the equivalence. Let $\tau : \mathbb{N} \to (0, 1/2)$, $\mathsf{MQCSP}[s, s', t, \tau]$ is defined as the gap problem where the No instances in Definition 3.4 becomes "for every quantum circuit $\mathcal{C}$ using at most s' gates and operating on at most $n + t$ qubits, there are at least τ fraction of $x \in \{0,1\}^n$ such that $\|(\langle f(x)| \otimes I_{n+t-1})\mathcal{C}|x, 0^t\rangle\|^2 \leq \frac{1}{2}$".

Theorem 4.12 (Equivalence of efficient PAC learning for $\mathsf{BQP/poly}$ and efficient randomized algorithm for MQCSP)**.**

- *If* $\mathsf{MQCSP} \in \mathsf{BPP}$*, then there is a randomized algorithm that $(1/\mathsf{poly}(n), \delta)$-PAC learns $f \in$* $\mathsf{BQP/poly}$ *under the uniform distribution with membership queries for every $\delta > 0$. Specifically, the algorithm runs in quasi-polynomial time.*
- *If there is a randomized algorithm that $(1/\mathsf{poly}(n), \delta)$-PAC learns $f \in$* $\mathsf{BQP/poly}$ *under the uniform distribution with membership queries for some $\delta > 0$ in $2^{O(n)}$ time, then we have* $\mathsf{MQCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), \mathsf{poly}(n), \tau] \in \mathsf{BQP}$ *and* $\mathsf{MQCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), O(n), \tau] \in$ BPP *for every $\tau > 0$.*

Similarly, the positive resolution of Open Problem 2 would strengthen the conclusion of the second item in Theorem 4.12 to $\mathsf{MQCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), \mathsf{poly}(n), \tau] \in \mathsf{BPP}$.

Quantum learning. As it could be the case that MQCSP might have non-trivial quantum algorithm, it is also of interest to study the connection to quantum learning [AGG+20].

Definition 4.13 (Quantum learning)**.** *Let* C *be a circuit class of boolean functions and let* $\epsilon, \delta > 0$*. We say a quantum algorithm* (ϵ, δ)*-learns* C *if the following hold. For every* $n \in \mathbb{N}$ *and* n*-variate* $f \in \mathsf{C}$*, given quantum oracle access to* f*, the algorithm outputs a polynomial-size quantum circuit* U *such that with probability at least* $1 - \delta$*, we have* $\mathbb{E}_{x \in \{0,1\}^n}[|(\langle f(x)| \otimes I)U|x, 0^m\rangle|^2] > 1 - \epsilon$*. The running time of the learning algorithm is measured as a function of* $n, 1/\epsilon, 1/\delta$ *and,* $\mathsf{size}(f)$*.*

It turns out that efficient quantum learning for a circuit class C (could be either a classical circuit class or a quantum circuit class) is equivalent to efficient quantum algorithm for C-MCSP. Similarly, C-$\mathsf{MCSP}[s, s', \tau]$ is defined as the gap problem with the No instances being the truth tables where every circuit $\mathcal{C}$ of size s' errs on τ fraction of the inputs.

Theorem 4.14 (Equivalence of efficient quantum learning and efficient quantum algorithm for C-MCSP)**.** *Let* C *be a circuit class.*

- *If* C-$\mathsf{MCSP} \in \mathsf{BQP}$*, then there exists a quantum algorithm that* $(1/\mathsf{poly}(n), \delta)$*-learns* C *for every* $\delta > 0$*. Specifically, the algorithm runs in polynomial time.*
- *If there exists a quantum algorithm that* (ϵ, δ)*-learns* C *in time* $2^{O(n)}$ *for some constants* $\epsilon, \delta \in (0, 1/2)$*, then we have* C-$\mathsf{MCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), \tau] \in \mathsf{BQP}$ *for every* $\tau > 0$*.*

4.3 Circuit lower bounds

The classical MCSP is tightly connected to circuit lower bounds. Many results show that a fast algorithm for MCSP will lead to breakthrough in circuit lower bounds, which on the other hand indicates that MCSP might be very difficult to solve. In this section, we "quantize" four results relating MQCSP and quantum circuit lower bounds.

Quantum circuit lower bound via quantum natural proof By Observation 1, we know that MQCSP gives a BQP-quantum natural property. Then, we follow a recent work by Arunachalam et al. [AGG+20] and prove the following theorem:

Theorem 4.15. *If* $\mathsf{MQCSP} \in \mathsf{BQP}$*, then* $\mathsf{BQE} \not\subset \mathsf{BQC}[n^k]$ *for any constant* $k \in \mathbb{N}_+$*, where* $\mathsf{BQE} = \mathsf{BQTIME}[2^{O(n)}]$*.*

Remark 6. *A key difference between Theorem 4.15 and [AGG+20] is that their circuit lower bound for* BQE *is against* classical *circuits, while ours is against* quantum *circuits by proving a diagonalization lemma for quantum circuits.*

An ingredient of our proof is a conditional pesudorandom generator against uniform quantum computation. We first recall the definition of PRG against uniform quantum circuits given by [AGG+20].

Definition 4.16 (Pesudorandom generator against uniform quantum circuit, [AGG+20])**.** *A family of functions* $\{G_n\}_{n \geq 1}$ *is an infinitely often* (ℓ, m, s, ϵ)*-generator against uniform quantum circuits if the following properties holds:*

1. *Stretch:* $G_n : \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$.
2. *Uniformity and efficiency:* *There is a deterministic algorithm A that when given 1^n and $x \in \{0,1\}^{\ell(n)}$ runs in time $O(2^{\ell(n)})$ and outputs $G_n(x)$.*
3. *Pseudorandomness:* *For every deterministic algorithm A such that when given $1^{m(n)}$ runs in time $s(m)$ and outputs a quantum circuit C_m of size at most $s(m)$ computing a m-input Boolean function, for infinitely many $n \geq 1$,*

$$\left| \Pr_{x \sim \{0,1\}^{\ell(n)}, C_m} [C_m(G_n(x)) = 1] - \Pr_{y \sim \{0,1\}^{m(n)}, C_m} [C_m(y) = 1] \right| \leq \epsilon(m).$$

[AGG+20] constructed the following infinitely often PRG based on the assumption $\mathsf{PSPACE} \not\subseteq \mathsf{BQSUBEXP}$.

Theorem 4.17 (Conditional PRG against uniform quantum computations, [AGG+20])**.** *Suppose that $\mathsf{PSPACE} \not\subseteq \mathsf{BQSUBEXP}$. Then, for some choice of constants $\alpha \geq 1$ and $\lambda \in (0, 1/5)$, there is an infinitely often $(\ell, m, s, \varepsilon)$-generator $G = \{G_n\}_{n \geq 1}$, where $\ell(n) \leq n^\alpha$, $m(n) = \lfloor 2^{n^\lambda} \rfloor$, $s(m) = 2^{n^{2\lambda}} \geq \mathsf{poly}(m)$ (for any polynomial), and $\varepsilon(m) = 1/m$.*

Now, we are ready to prove the lower bound for BQE based on the conditional PRG and a diagonalization theorem for quantum circuits.

Proof of Theorem 4.15. We use a win-win argument to prove the circuit lower bound.

Case 1: Suppose $\mathsf{PSPACE} \subseteq \mathsf{BQSUBEXP}$, i.e., for every $\gamma \in (0,1]$, $\mathsf{PSPACE} \subseteq \mathsf{BQTIME}[2^{n^\gamma}]$. Then, for a fixed $k \in \mathbb{N}$, by a diagonalization lemma for quantum circuits (Claim F.3), we know that there exists a language $L \in \mathsf{PSPACE}$ such that $L \notin \mathsf{BQC}[n^k]$. However, by the assumption, $L \in \mathsf{BQE}$, which implies that $\mathsf{BQE} \not\subset \mathsf{BQC}[n^k]$.

Case 2: $\mathsf{PSPACE} \not\subseteq \mathsf{BQSUBEXP}$, that is, there exists a language $L \in \mathsf{PSPACE}$ and $\gamma > 0$ such that $L \notin \mathsf{BQTIME}[2^{n^\gamma}]$. By Theorem 4.17, for some $\alpha \geq 1, \lambda \in (0, 1/5)$, there exists an infinitely often (ℓ, m, s, ϵ)-PRG $\mathcal{G} = \{G_n\}_{n \geq 1}$, where $\ell(n) = n^\alpha$, $m(n) = \lfloor 2^{n^\lambda} \rfloor$, $s(m) = \lfloor 2^{n^{2\lambda}} \rfloor$, $\epsilon(m) = 1/m$.

For each $w \in \{0,1\}^{n^\alpha}$, we consider $G_n(w)$ as the truth table of Boolean function $\mathsf{fnc}(G_n(w)) : \{0,1\}^d \to \{0,1\}$, where $d := \log(m(n))$ is the input length of the function. We will show that $\mathsf{fnc}(G_n(w))$ is a hard function for $\mathsf{BQC}[d^{O(1)}]$ for most $w \in \{0,1\}^{\ell(n)}$.

Suppose that this is not true, i.e., there exists a $k > 0$ such that for almost every $n > 0$, $\mathsf{fnc}(G_n(w)) \in \mathsf{BQC}[d(n)^k]$ for a constant fraction of seeds $w \in \{0,1\}^{\ell(n)}$. Then, consider a quantum circuit C_m^{MQCSP} which takes a m-bit string s and accepts it if and only if $\mathsf{MQCSP}(s, 1^{d^k}) = 1$, where s is the truth table and d^k is the size parameter. Since we assume $\mathsf{MQCSP} \in \mathsf{BQP}$, the quantum circuit C_m^{MQCSP} can be generated by a deterministic algorithm in time $\mathsf{poly}(m) \leq s(m)$[22]. This implies that

$$\Pr_{w \sim \{0,1\}^{\ell(n)}, C_m^{\mathsf{MQCSP}}} \left[C_m^{\mathsf{MQCSP}}(G_n(w)) = 1 \right] \geq \delta$$

[22] For all problems in BQP, there exists a classical Turing machine that can efficiently uniformly generate the quantum circuits.

for some constant $\delta \in (0,1)$. On the other hand, by the pseudorandomness property of G_n (part 3 in Definition 4.16), for infinitely many n, we have

$$\left| \Pr_{w\sim\{0,1\}^{\ell(n)}, C_m^{\mathsf{MQCSP}}} \left[C_m^{\mathsf{MQCSP}}(G_n(w)) = 1 \right] - \Pr_{y\sim\{0,1\}^{m(n)}, C_m^{\mathsf{MQCSP}}} \left[C_m^{\mathsf{MQCSP}}(y) = 1 \right] \right| \leq \frac{1}{m}. \tag{2}$$

However, only $o(1)$-fraction of random functions have polynomial-size quantum circuits, i.e.,

$$\Pr_{y\sim\{0,1\}^{m(n)}, C_m^{\mathsf{MQCSP}}} \left[C_m^{\mathsf{MQCSP}}(y) = 1 \right] \leq o(1),$$

which means Eq. (2) cannot hold. Therefore, for infinitely many n, and almost all w, the function $\mathsf{fnc}(G_n(w)) \notin \mathsf{BQC}[n^k]$ for every $k \in \mathbb{N}_+$.

Therefore, we can construct a hard language $L^{\mathcal{G}}$ as follows:

- For any $n > 0$ and every $x \in \{0,1\}^n$, check if x can be written as (w, y), where $|w| = \ell(t)$ and $|y| = \lceil \log m(t) \rceil$ for some $t \in \mathbb{N}$.
- If not, then $L^{\mathcal{G}}(x) := 0$.
- Otherwise, $L^{\mathcal{G}}(x) := \mathsf{fnc}(G_t(w))(y)$.

We first show that $L^{\mathcal{G}} \in \mathsf{BQE}$. By the running time property of G_n (part 2 in Definition 4.16), $G_n(w)$ can be computed in deterministic time $O(2^{\ell(t)}) \leq O(2^n)$. Hence, $L^{\mathcal{G}} \in \mathsf{E} \subset \mathsf{BQE}$.

Then, we show that $L^{\mathcal{G}} \notin \mathsf{BQC}[n^k]$ for every $k \in \mathbb{N}_+$. Fix $k > 0$. Suppose there exists a quantum circuit family $\{C_n\}_{n\geq 1}$ that computes $L^{\mathcal{G}}$ and C_n has size n^k for every $n \geq 1$. However, we already proved that there exists an infinite-size subset $\{\mathcal{S} \subset \mathbb{N}\}$ such that for $n \in \mathcal{S}$, there exists many "hard seed" w_n such that

$$\mathsf{fnc}(G_t(w_n)) \notin \mathsf{BQC}[t^{2\alpha k}]. \tag{3}$$

Then, for any $n \in S$ and any w_n that makes Eq. (3) hold, define a new quantum circuit family $\{C \lfloor_{w_n}\}_{n\geq 1}$ such that $C \lfloor_{w_n} (y) := C(w_n, y)$, i.e., $C \lfloor_{w_n}$ computes the hard function $\mathsf{fnc}(G_t(w_n))$. Hence, $C \lfloor_{w_n}$ must have size larger than $t^{2\alpha k}$. Since $n = \ell(t) + \log m(t) = t^\alpha + t^\lambda \leq t^{2\alpha}$, and the size of C_n should be least the size of its restriction $C \lfloor_{w_n}$, we conclude that C_n has size larger than n^k for these infinitely many $n \in \mathcal{S}$. Therefore, the BQE language $L^{\mathcal{G}} \notin \mathsf{BQC}[n^k]$, which implies $\mathsf{BQE} \not\subset \mathsf{BQC}[n^k]$.

Combining Case 1 and 2 completes the proof of the theorem. □

Circuit lower bound for $\mathsf{BQP}^{\mathsf{QCMA}}$ Our second result shows that if $\mathsf{MQCSP} \in \mathsf{BQP}$, then $\mathsf{BQP}^{\mathsf{QCMA}}$ cannot be computed by polynomial-size quantum circuits. Our result follows the seminal work of Kabanets and Cai [KC00], which showed a circuit lower bound for P^{NP} based on MCSP is easy. More specifically, we consider the following "hard problem":

Definition 4.18 (Maximum quantum circuit complexity problem)**.** *The input of this problem is 1^n for $n \in \mathbb{N}_+$. The output is the truth table of a function $f : \{0,1\}^n \to \{0,1\}$ such that for any $f' : \{0,1\}^n \to \{0,1\}$, the quantum circuit complexity $\mathrm{qCC}(f) \geq \mathrm{qCC}(f')$.*

We first prove that $\mathsf{BPE}^{\mathsf{QCMA}}$ can solve the maximum quantum circuit complexity problem, which implies that $\mathsf{BPE}^{\mathsf{QCMA}}$ contains the hardest Boolean function. Then, by the standard padding argument, we can show quantum circuit lower bound for $\mathsf{BQP}^{\mathsf{QCMA}}$.

Theorem 4.19. *If* $\mathsf{MQCSP} \in \mathsf{BQP}$*, then* $\mathsf{BPE}^{\mathsf{QCMA}}$ *contains a function with maximum quantum circuit complexity. Furthermore,* $\mathsf{BQP}^{\mathsf{QCMA}} \not\subset \mathsf{BQC}[n^k]$ *for any constant* $k > 0$*.*

We note that there are two subtle differences between Theorem 4.19 and [KC00]'s result:

- We need a QCMA oracle while [KC00] used an NP oracle. This is because we assume that $\mathsf{MQCSP} \in \mathsf{BQP}$. In order to decide the maximum quantum circuit complexity, we can non-deterministically guess a truth table and use the BQP algorithm to verify its quantum circuit complexity. This process can be achieved by an QCMA oracle.
- Another difference is that we consider the BPE class while [KC00] considered the E class. This is because our QCMA oracle can only output correct answers with high probability. Thus, the whole algorithm will be a randomized algorithm.

The formal proof is deferred to Section C.1.

Hardness amplification using MQCSP [KC00] showed that the classical MCSP can be used for hardness amplification, i.e., given one very hard Boolean function, there exists an efficient algorithm to find many hard functions via an MCSP oracle. We show that it also holds for quantum circuits:

Theorem 4.20. *Assume* $\mathsf{MQCSP} \in \mathsf{BQP}$*. Then, there exists a* BQP *algorithm that, given the truth table of an* n*-variable Boolean function of quantum circuit complexity* $2^{\Omega(n)}$*, outputs* $2^{\Omega(n)}$ *Boolean functions on* $m = \Omega(n)$ *variables each, such that all of the output functions have quantum circuit complexity greater than* $\frac{2^m}{(c+1)m}$ *for any* $c > 0$*.*

In order to prove Theorem 4.20, we first construct a "quantum version" of the Impagliazzo-Wigderson generator [IW97]. We note that the construction in the following lemma is stronger than the Definition 4.16, based on the truth table of a very hard function.

Lemma 4.21 (Quantum Impagliazzo-Wigderson generator)**.** *For every* $\epsilon > 0$*, there exist* $c, d \in \mathbb{N}$ *such that the truth table of a Boolean function* $f : \{0,1\}^{cn} \to \{0,1\}$ *of quantum circuit complexity* $2^{\epsilon cn}$ *can be transformed in time* $O(2^n)$ *into a pseudorandom generator* $G : \{0,1\}^{dn} \to \{0,1\}^{2^n}$ *running in time* $O(2^n)$ *that can fool quantum circuits of size* $2^{O(n)}$*, i.e., for any* $p > 0$*, any quantum circuit* $\mathcal{C}$ *of size at most* 2^{pn}*,*

$$\left| \Pr_{x \sim \{0,1\}^{dn}, \mathcal{C}}[\mathcal{C}(G(x)) = 1] - \Pr_{y \sim \{0,1\}^{2^n}, \mathcal{C}}[\mathcal{C}(y) = 1] \right| \leq 2^{-n}.$$

Proof of Theorem 4.20. Let $c > 0$ and $s(n) = \frac{2^n}{(c+1)n}$. Assuming that $\mathsf{MQCSP} \in \mathsf{BQP}$, we get a polynomial-size quantum circuit family $\{\mathcal{D}_n\}$ that only accept n-variable Boolean functions of quantum circuit complexity greater than $s(n)$. By Claim F.1, the acceptance probability is close to one.

However, the size of $\mathcal{D}_n$ is bounded by a fixed polynomial in the input size, by Lemma 4.21, the quantum Impagliazzo-Wigderson generator G will fool $\mathcal{D}_n$. That is, almost all 2^n-bit strings output by G will have quantum circuit complexity greater than $s(n)$. We can then use the MQCSP circuit to decide the quantum circuit complexity of these strings and only output hard functions. □

The proof of Lemma 4.21 relies on a quantum-secure direct product generator and several hardness amplification steps. It is deferred to Section C.3.

Hardness magnification for MQCSP. Hardness magnification refers to a transformation of a weak circuit lower bound (e.g., linear size lower bound) to a stronger circuit lower bound (e.g., polynomial size lower bound). Note that a magnification theorem for a circuit class is highly dependent on the structure of the circuits. Specifically, it is not immediately clear that every circuit class is *magnifiable*. Here, we show that there exists hardness magnification for quantum circuits when it comes to MQCSP.

Theorem 4.22. *If* $\mathsf{MQCSP}\left[2^{n^{1/2}}/2n, 2^{n^{1/2}}\right]$ *is hard for* $\mathsf{BQC}\left[2^{n+O(n^{1/2})}\right]$*, then* $\mathsf{QCMA} \not\subseteq \mathsf{BQC}[\mathsf{poly}(n)]$.

The proof of Theorem 4.22 is via antichecker lemma, which was first given by [OPS19, CHO+20] for proving hardness magnification for MCSP.

Lemma 4.23 (Antichecker lemma for quantum circuits)**.** *Assume* $\mathsf{QCMA} \subseteq \mathsf{BQC}[\mathsf{poly}]$*. Then for any* $\lambda \in (0,1)$ *there are circuits* $\{C_{2^n}\}_{n=1}^{\infty}$ *of size* $2^{n+O(n^\lambda)}$ *which given the truth table* $\mathsf{tt}(f) \in \{0,1\}^{2^n}$ *, outputs* $2^{O(n^\lambda)}$ *n-bit strings* $y_1, \ldots, y_{2^{O(n^\lambda)}}$ *together with bits* $f(y_1), \ldots, f(y_{2^{O(n^\lambda)}})$ *forming a set of anticheckers for* f*, i.e. if* f *is hard for quantum circuits of size* 2^{n^λ} *then every quantum circuit of size* $2^{n^\lambda}/2n$ *fails to compute* f *on one of the inputs* $y_1, \ldots, y_{2^{O(n^\lambda)}}$*.*

With Lemma 4.23, we can prove Theorem 4.22 by using a small quantum circuit to verify the given circuits only on the anticheckers.

Proof of Theorem 4.22. Suppose $\mathsf{QCMA} \subseteq \mathsf{BQC}[\mathsf{poly}]$. Let $\mathsf{tt}(f)$ be the input of $\mathsf{MQCSP}[2^{n^{1/2}}/2n, 2^{n^{1/2}}]$. By Lemma 4.23, we can find a set of anticheckers $y_1, \ldots, y_{2^{O(n^{1/2})}}$ by a quantum circuit of size $2^{n+O(n^{1/2})}$. Then, we use a QCMA algorithm to decide if there exists a quantum circuit of size $2^{n^\lambda}/2n$ that computes f correctly on $\{(y_1, f(y_1)), \ldots, (y_{2^{O(n^\lambda)}}, f(y_{2^{O(n^\lambda)}}))\}$. By the assumption, it can be done by a $2^{O(n^\lambda)}$ size quantum circuit. Then, there are two cases:

- If the QCMA algorithm returns "Yes", it means that $y_1, \ldots, y_{2^{O(n^{1/2})}}$ are not antiecheckers. By Lemma 4.23, f is *not* hard for $2^{n^{1/2}}$ size quantum circuit.
- If the QCMA algorithm returns "No", then no $2^{n^{1/2}}/2n$ size quantum circuit can compute f on $y_1, \ldots, y_{2^{O(n^{1/2})}}$. So, f is hard for $2^{n^{1/2}}/2n$ size quantum circuit.

Hence, $\mathsf{MQCSP}[2^{n^{1/2}}/2n, 2^{n^{1/2}}] \in \mathsf{BQC}[2^{n+O(n^{1/2})}]$. □

The proof of Lemma 4.23 is deferred to Section C.2.

4.4 Fine-grained complexity

It is a long-standing open problem to show the hardness of MCSP based on some fine-grained complexity hypotheses, like the Exponential-Time Hypothesis (ETH), which was conjectured by Impagliazzo, Paturi, and Zane [IPZ01] and becomes a widely used assumption in fine-grained complexity area.

Definition 4.24 (Exponential Time Hypothesis (ETH))**.** *There exists* $\delta > 0$ *such that 3-SAT with* n *variables cannot be solved in time* $2^{\delta n}$*.*

Very recently, a breakthrough result by Ilango [Ila20b] proved the ETH-hardness of MCSP for partial Boolean functions. On the other hand, Quantum fine-grained complexity was studied very recently by [ACL+20, BPS21, AL20, GS20]. Motivated by the fact that currently there is no quantum algorithm for 3-SAT that is significantly faster than Grover's search, we conjecture that 3-SAT with n variables cannot be solved in $2^{o(n)}$ quantum time (QETH). And based on QETH, we want show that MQCSP for partial Boolean function is also hard.

We first formally define QETH and MQCSP for partial functions (MQCSP*).

Definition 4.25 (Quantum Exponential Time Hypothesis (QETH))**.** *There exists $\delta' > 0$ such that 3-SAT with n variables cannot be solved in time $2^{\delta' n}$ in quantum.*

Definition 4.26 (MQCSP for partial functions (MQCSP*))**.** *The input is the truth table $\{0,1,\star\}^{2^n}$ of a partial function $f : \{0,1\}^n \to \{0,1,\star\}$ and an integer parameter s. The goal is to decide whether there exists a quantum circuit C of size at most s (using single-qubit and 2-qubit gates) that computes f. That is, for all $x \in \{0,1\}^n$ such that $f(x) \neq \star$, we have*

$$\Pr[C(x) = f(x)] \geq \frac{2}{3}.$$

Our main result of this section is as follows:

Theorem 4.27 (QETH-hardness of MQCSP*)**.** MQCSP* *cannot be solved in $N^{o(\log \log N)}$-time quantumly on truth tables of length N assuming QETH.*

Our reduction reveals the connections between MQCSP*, quantum read-once formula and classical read-once formula. The proof is given in Section D.

Classical reduction for MCSP*. We first give a brief overview of the classical reduction for MCSP* in [Ila20b]. They reduced MCSP* to a fine-grained problem: $2n \times 2n$ *Bipartite Permutation Independent Set problem*, which is defined as follows:

Definition 4.28 (Bipartite Permutation Independent Set problem)**.** *A $2n \times 2n$ bipartite permutation independent set problem is defined on a directed graph G with vertex set $[n] \times [n]$ and edge set E. The goal is to decide whether there exists a permutation $\pi \in \mathcal{S}_{2n}$ such that*

- $\pi([n]) = [n]$,
- $\pi(\{n+i : i \in [n]\}) = \{n+i : i \in [n]\}$,
- *if $((j,k),(j',k')) \in E$, then either $\pi(j) \neq k$ or $\pi(n+j') \neq \pi(n+k')$.*

Lokshtanov, Marx, and Saurabh [LMS11] proved that this problem is $2^{o(n \log n)}$-hard under ETH, which implies the ETH-hardness of MCSP*.

The reduction from $2n \times 2n$ bipartite permutation independent set problem to MCSP* is via the following partial function γ. Consider an instance $G = ([n] \times [n], E)$ of $2n \times 2n$ bipartite permutation independent set problem. The reduction outputs the truth table of a partial Boolean function $\gamma : \{0,1\}^{2n} \times \{0,1\}^{2n} \times \{0,1\}^{2n} \to \{0,1,\star\}$ such that

$$\gamma(x,y,z) := \begin{cases} \bigvee_{i \in [2n]} (y_i \wedge z_i) & \text{if } x = 0^{2n}, \\ \bigvee_{i \in [2n]} z_i & \text{if } x = 1^{2n}, \\ \bigvee_{i \in [2n]} (x_i \vee y_i) & \text{if } z = 1^{2n}, \\ 0 & \text{if } z = 0^{2n}, \\ \bigvee_{i \in [n]} x_i & \text{if } z = 1^n 0^n \text{ and } y = 0^{2n}, \\ \bigvee_{i \in \{n+1,\cdots,2n\}} x_i & \text{if } z = 0^n 1^n \text{ and } y = 0^{2n}, \\ 1 & \text{if } \exists ((j,k),(j',k')) \in E \text{ s.t. } (x,y,z) = (\overline{e_k e_{k'}}, 0^{2n}, e_j e_{j'}), \\ \star & \text{otherwise.} \end{cases} \tag{4}$$

In particular, the small circuit size of γ implies that G is a "Yes" instance of the bipartite permutation independent set problem:

Lemma 4.29 ([Ila20b]). *Each of the following are equivalent:*

1. $\mathsf{MCSP}^\star(\gamma, 6n-1) = 1$*;*
2. γ *can be computed by a read-once formula;*
3. *there exists a* $\pi \in \mathcal{S}_{2n}$ *such that* $\bigvee_{i\in[2n]}((x_{\pi(i)} \vee y_i) \wedge z_i)$ *computes* γ*;*
4. *there exists a* $\pi \in \mathcal{S}_{2n}$ *that satisfies the instance of bipartite permutation independent set problem given by* G*.*

Quantum reduction for $\mathsf{MQCSP}^\star$ We follow the proof in [Ila20b] but adapt it to quantum circuits. More specifically, we want to show that for the partial function γ defined by Eq. (4), $\mathsf{MQCSP}^\star(\gamma, 6n-1) = 1$ is equivalent to the case that γ can be computed by a read-once formula.

The reverse direction is easy:

Claim 4.30. *If* γ *can be computed by a read-once formula, then* $\mathsf{MQCSP}^\star(\gamma, 6n-1) = 1$*.*

Proof. It is easy to see that a read-once formula on $6n$ input variables has at most $6n-1$ Boolean gates. Hence, it implies that $\mathsf{MCSP}^\star(\gamma, 6n-1) = 1$. Then, we have $\mathsf{MQCSP}^\star(\gamma, 6n-1) = 1$ because we can use a quantum circuit with all 2-qubit gates to simulate a Boolean circuit without increasing the circuit size. □

For the forward direction, we consider an intermediate model: *read-once quantum formula*. The quantum formula was defined by Yao [Yao93] as follows:

Definition 4.31. *A quantum formula is a single-output quantum circuit such that every gate has at most one output that is used as an input to a subsequent one.*

If a quantum formula only uses every input qubit at most once, then we say it is a read-once quantum formula.

We first prove the forward direction for the quantum read-once formula:

Claim 4.32. *If* $\mathsf{MQCSP}^\star(\gamma, 6n-1) = 1$*, then* γ *can be computed by a read-once quantum formula. Here, we assume that the quantum circuits only use single-qubit and 2-qubit gates.*

Proof. It is easy to verify that γ depends on all of the $6n$ input variables. Hence, by a light-cone argument, the topology of the quantum circuit that computes γ using $6n-1$ 2-qubit gates must be a full binary tree with $6n$ leaves. Hence, that circuit is a read-once quantum formula. □

Cosentino, Kothari, and Paetznick [CKP13] proved that any read-once quantum formula can be "dequantized" to the classical read-once quantum formula:

Theorem 4.33 ([CKP13]). *If a language is accepted by a bounded-error read-once quantum formula over single-qubit and 2-qubit gates, then it is also accepted by an exact read-once classical formula with the same size, using* NOT *and all 2-bit Boolean gates.*

Hence, we can apply Theorem 4.33 to dequantize Claim 4.32:

Claim 4.34. *If* $\mathsf{MQCSP}^\star(\gamma, 6n-1) = 1$*, then* γ *can be computed by a classical read-once formula with* $6n-1$ *2-bit gates. In particular, all the* NOT *gates can be pushed to the leaf level and the high level gates are* $\{\mathsf{AND}, \mathsf{OR}, \mathsf{XOR}\}$*.*

Proof. By Theorem 4.33, there is a read-once classical formula that computes γ using $6n-1$ 2-bit logical gates. We can enumerate all of the 2-bit Boolean function and check that they can be expressed by one of $\mathsf{AND}, \mathsf{OR}, \mathsf{XOR}$ gate with some NOT gates on the input wire. Then, by De Morgan's laws, we can push the NOT gate to the bottom level. Note that these transformations will preserve the read-once property. □

The next claim shows that NOT and XOR gates do not help computing γ:

Claim 4.35. *The classical read-once formula computing* γ *only uses* AND *and* OR *gates.*

Proof. The proof is similar to the proof of Claim 13 in [Ila20b].

We first note that the XOR gate is not monotone. Then, by setting $x = 0^{2n}$, we have $\gamma(0^{2n}, y, z) = \bigvee_{i \in [2n]}(y_i \wedge z_i)$, which is a monotone function in y and z. Hence, the XOR gates in the formula cannot depend on the all the y and z variables. Similarly, by setting $z = 1^{2n}$, we have $\gamma(x, y, 1^{2n}) = \bigvee_{i \in [2n]}(x_i \vee y_i)$, which is monotone in x and y. It implies that the XOR gates cannot depend on all the x variables. Hence, the formula will not use the XOR gate.

For the NOT gate, since the function is monotone in the positive input variables after some restrictions, and the formula is read-once, the NOT gate will also not be used. □

By Claim 4.30, 4.34 and 4.35, we get that $\mathsf{MQCSP}^\star(\gamma, 6n-1) = 1$ is equivalent to the case that γ can be computed by a read-once formula using AND and OR gates. This statement corresponds to showing that (1) $\Leftrightarrow$ (2) in Lemma 4.29 for $\mathsf{MCSP}^\star$. Then, by (2) $\Leftrightarrow$ (4) in Lemma 4.29, we prove the following reduction for $\mathsf{MQCSP}^\star$:

Lemma 4.36. $\mathsf{MQCSP}^\star(\gamma, 6n-1) = 1$ *is equivalent to the existence of* $\pi \in \mathcal{S}_{2n}$ *that satisfies the instance of bipartite permutation independent set problem given by* G*.*

The remaining thing is to prove the quantum hardness of the $2n \times 2n$ $\mathsf{Bipartite\ Permutation\ Independent\ Set}$ problem. We follow the quantum fine-grained reduction framework by [ACL+20] and show the following QETH-hardness result. The proof is given in Section D.

Lemma 4.37. *Assuming* QETH*, there is no* $2^{o(n \log n)}$*-time quantum algorithm that solves* $2n \times 2n$ $\mathsf{Bipartite\ Permutation\ Independent\ Set}$ *problem.*

Now, we can prove the QETH-hardness of $\mathsf{MQCSP}^\star$:

Proof of Theorem 4.27. By Lemma 4.36, $\mathsf{MQCSP}^\star$ can be reduced to $2n \times 2n$ $\mathsf{Bipartite\ Permutation\ Independent\ Set}$ problem and the hardness follows from Lemma 4.37. □

5 MCSP for Quantum Objects

In this section, we generalize the problem to considering circuit complexities of quantum objects, including unitaries and quantum states. In particular, we study their hardness, related reductions, and their implications to other subjects in quantum computer science. We start by defining the two problems.

Definition 5.1 ($\mathsf{UMCSP}_{\alpha,\beta}$)**.** *Let* $n, s, t \in \mathbb{N}$ *and* $t \leq s$*. Let* $\alpha, \beta \in (0, 1]$*. Let* $U \in \mathbb{C}^{2^n \times 2^n}$ *be a unitary.* UMCSP *is a promise problem defined as follows.*

- ***Inputs:*** *the unitary matrix U, the size parameter s in unary representation, and the ancilla parameter t.*
- ***Yes instance:*** *there exists a quantum circuit $\mathcal{C}$ using at most s gates and operating on at most $n+t$ qubits such that for all $|\psi\rangle \in \mathbb{C}^{2^n}$,*

$$\|(\langle\psi| \otimes I_t)(U^\dagger \otimes I_t)\mathcal{C}|\psi, 0^t\rangle\|^2 \geq \alpha, \tag{5}$$

- ***No instance:*** *for every quantum circuit $\mathcal{C}$ using at most s gates and operating on at most $n+t$ qubits, there exists some $|\psi\rangle \in \mathbb{C}^{2^n}$ such that*

$$\|(\langle\psi| \otimes I_t)(U^\dagger \otimes I_t)\mathcal{C}|\psi, 0^t\rangle\|^2 \leq \beta. \tag{6}$$

With the promise that the input must be either a yes instance or a no instance, the problem is to decide whether the input is a yes instance or not.

Remark 7. *Since the input to* UMCSP *is a unitary matrix U and each entry is a complex number, we cannot fully describe U and hence need to specify a precision parameter. Moreover, the precision issue is subtle in the search-to-decision reduction. For a gate set $\mathcal{G}$, we denote $\ell_{\mathcal{G}} \in \mathbb{N}$ as the maximum number of bits used to encode an entry of a gate. Note that if a circuit uses s gates from $\mathcal{G}$, then each entry in the resulting unitary can be written down with at most $s \cdot \ell_{\mathcal{G}}$ bits. Thus, by the triangle inequality for the distance between unitaries, it suffices to use $s \cdot \ell_{\mathcal{G}}$ bits to encode each entry of the input unitary. Also, note that when $\alpha - \beta < 2^{-s \cdot \ell_G}$,* $\mathsf{UMCSP}_{\alpha,\beta}$ *becomes a non-promise problem since effectively the gap between Yes and No instances does not matter. In the definition of* UMCSP*, we hide the introduction of precision parameter for simplicity. Note that from the above reasoning and the fact that the input unitary is $2^n \times 2^n$, it would not affect the complexity of the problem even one chooses the bit complexity to be $2^{O(n)}$, which is more than enough for most interesting situations.*

Definition 5.2 ($\mathsf{SMCSP}_{\alpha,\beta}$)**.** *Let $n, s, t \in \mathbb{N}$, where $t \leq s$. Let $\alpha, \beta \in (0, 1]$. Let $|\phi\rangle \in \mathbb{C}^{2^n}$ be a quantum state.* SMCSP *is a promise problem defined as follows.*

- ***Inputs:*** *size parameters s and n in unary, access to arbitrary many copies of $|\psi\rangle$, and the ancilla parameter t.*
- ***Yes instance:*** *there exists a quantum circuit $\mathcal{C}$ using at most s gates and operating on at most $n+t$ qubits such that*

$$\|(\langle\phi| \otimes I_{n+t-1})\mathcal{C}|0^{n+t}\rangle\|^2 \geq \alpha,$$

- ***No instance:*** *for every quantum circuit $\mathcal{C}$ using at most s gates and operating on at most $n+t$ qubits,*

$$\|(\langle\phi| \otimes I_{n+t-1})\mathcal{C}|0^{n+t}\rangle\|^2 \leq \beta.$$

With the promise that the input must be either a yes instance or a no instance, the problem is to decide whether the input is a yes instance or not.

Remark 8. *Similarly, the precision of the input parameters α, β of* SMCSP *has to depend on the bit complexity of the gate set. See Remark 7 for more discussion.*

Remark 9. *For the thresholds α, β, it is worth noting that a quantum circuit that outputs a mixed state can always have nonzero inner product with an arbitrary state. Therefore, we cannot set β to be arbitrarily small; otherwise, there will not be any U or $|\phi\rangle$ satisfying the no instance.*

For SMCSP, we focus on the version where the inputs are multiple quantum states. The input format is quite different from UMCSP and MQCSP; instead of having the full classical description, SMCSP is given access to many copies of the quantum state. Hence, we say an algorithm for SMCSP is efficient if it runs in time $\mathsf{poly}(n,t,s)$, i.e., an efficient algorithm can use at most $\mathsf{poly}(n,t,s)$ copies of $|\psi\rangle$. We choose this input format because that in the quantum setting, we generally cannot have the classical description of the quantum state. For instance, in shadow tomography[Aar18], quantum gravity[BFV20], and quantum pseudorandom states[JLS18], the problem is given many copies of a quantum state, identify some properties of the state. Furthermore, although this problem seems to be much harder than having the full description or a succinct description (e.g, a circuit that generates the state) of the state, we will see that this problem has a QCMA protocol. [23]

Remark 10. *On the other hand, the hardness results including the problem is in* QCMA *(Theorem 5.9), the search-to-decision reduction (Theorem 5.18), and the approximate self-reduction (Theorem 5.20) all hold for the version where the input is a classical description for the state.*

Before proving the main theorems in this section, we introduce some notations and the swap test. Swap test [BCWdW01] is a quantum subroutine for testing whether two pure quantum states are close to each other.

Notation 1. *We write $a \approx_\epsilon b$ for $a, b \in \mathbb{R}$ to mean $\|a-b\| \leq \epsilon$.*

Notation 2. *We write $|\varphi\rangle \approx_\epsilon |\phi\rangle$ to mean $\||\varphi\rangle - |\phi\rangle\| \leq \epsilon$.*

Lemma 5.3 (Correctness of Swap Test)**.** *For any two states $|\phi\rangle, |\psi\rangle$, consider the following state*

$$(\mathsf{H} \otimes \mathsf{I})(c\text{-}\mathsf{SWAP})(\mathsf{H} \otimes \mathsf{I})|0\rangle|\phi\rangle|\psi\rangle$$

Measuring the first qubit gives outcome 1 with probability $\frac{1}{2} - \frac{1}{2}|\langle\phi|\psi\rangle|^2$.

Claim 5.4. *Let $|\phi\rangle, |\psi\rangle \in \mathbb{C}^{2^n}$ be two quantum states such that $|\phi\rangle \approx_\epsilon |\phi\rangle$. Then, for any $|\psi'\rangle$ which is a state on at most n qubits,*

$$\|(\langle\psi'| \otimes I)|\phi\rangle\| - \epsilon \leq \|(\langle\psi'| \otimes I)|\psi\rangle\| \leq \|(\langle\psi'| \otimes I)|\phi\rangle\| + \epsilon.$$

Proof. Without loss of generality, we can write $|\psi\rangle = |\phi\rangle + |\epsilon\rangle$, where $\||\epsilon\rangle\| \leq \epsilon$. Then, $\|(\langle\psi'| \otimes I)|\psi\rangle\| = \|(\langle\psi'| \otimes I)|\phi\rangle + (\langle\psi'| \otimes I)|\epsilon\rangle\|$. By using triangular inequality, we obtain the following two inequalities:

$$\begin{aligned} &\|(\langle\psi'| \otimes I)|\psi\rangle\| \leq \|(\langle\psi'| \otimes I)|\phi\rangle\| + \|(\langle\psi'| \otimes I)|\epsilon\rangle\|, \text{ and} \\ &\|(\langle\psi'| \otimes I)|\psi\rangle\| \geq \|(\langle\psi'| \otimes I)|\phi\rangle\| - \|(\langle\psi'| \otimes I)|\epsilon\rangle\|. \end{aligned}$$

Since $\||\epsilon\rangle\| \leq \epsilon$, $\|(\langle\psi'| \otimes I)|\epsilon\rangle\| \leq \epsilon$. This completes the proof. □

Theorem 5.5. $\mathsf{UMCSP}_{\alpha,\beta}$ *where $\beta \leq 1 - \mathsf{poly}(1/2^n)$ and $\alpha > 1 - 2^{-2n-20}(1-\beta)^4$ (for example, $\alpha = 1 - \mathsf{exp}(-2^n), \beta = 1 - \mathsf{poly}(1/2^n)$) is in* QCMA.

To design the verifier (that verifies a quantum circuit $\mathcal{C}$ really implements U as we want), what we will do is the following checking:

[23] Since SMCSP takes quantum inputs, the problem is not in QCMA under the standard definition. However, problems with quantum inputs in quantum computing is natural, so, it is also reasonable to study the complexity classes that allow quantum inputs.

1. Standard basis check: check whether Eq. (5) is satisfied on standard basis states.
2. Coherency check: Check Eq. (5) on superposition states in the form of $|a\rangle + |b\rangle$. This step has two goals: (1) checking whether the operation does behave similar to a unitary (instead of, for example, a collapsing measurement). (2) the unitary does not introduce different phases on different basis states.

Proof. Our checking algorithm follows the two steps above. The certificate is the circuit that implements the unitary such that Eq. (5) is satisfied. The following algorithm verifies it (assuming the promise):

1. (Standard basis check) For each $i \in [2^n]$, evaluate $(U^\dagger \otimes I_t)\mathcal{C}(|i\rangle \otimes |0^t\rangle)$ for $\mathsf{poly}_1(2^n)$ times. Store the output state (which requires only polynomial memory); denote the j-th sample on input i as $|\varphi_i^j\rangle$. Measure each of the states and check whether the output for $|\varphi_i^j\rangle$ is i. If not, mark it as a negative sample.

   If for any i, the ratio of negative samples is $\geq 2^{-2n-18}(1-\beta)^4$, reject.

2. (Coherency check) Do the following for each $i, j \in [2^n], i \neq j$ for $\mathsf{poly}_2(2^n)$ times:

   Apply $(U^\dagger \otimes I_t)\mathcal{C}$ on $\frac{1}{\sqrt{2}}(|i\rangle + |j\rangle) \otimes |0^t\rangle$. Project the output system on $\frac{1}{\sqrt{2}}(|i\rangle + |j\rangle)$. If the projection does not succeed, consider it as a negative sample.

   If for any of i, the ratio of negative samples is $\geq 2^{-2n-18}(1-\beta)^4$, reject.

We will show, when $\mathsf{poly}_1, \mathsf{poly}_2$ are all chosen to be some sufficiently big polynomials, this test can be used as the QCMA-verifier we need.

First, if a circuit satisfies Eq. (5), we can prove the verifier succeeds with probability $1 - 2^{-O(\mathsf{poly}(2^n))}$.

1. First, in the standard basis check, by Eq. (5), the expected ratio of negative sample is at most $1 - \alpha \leq \frac{1}{4} \cdot \text{threshold}$ (threshold $:= 2^{-2n-18}(1-\beta)^4$). By the Chernoff bound we have, $\forall a \in [2^n]$,

$$
\begin{aligned}
&\Pr[\text{negative ratio} \geq \text{threshold}] \\
=\ &\Pr[\text{negative samples} \geq \text{threshold} \cdot \mathsf{poly}_1(2^n)] \\
\leq\ &2^{-O(\mathbb{E}[\text{negative samples}])} && \text{(Chernoff bound)} \\
\leq\ &2^{-O(\mathsf{poly}_1(2^n) \cdot 2^{-2n-20}(1-\beta)^4)} && (\text{threshold} \cdot \mathsf{poly}_1(2^n) \geq 4 \cdot \mathbb{E}[\text{negative samples}])
\end{aligned}
$$

   which is $2^{-O(\mathsf{poly}(2^n))}$ when poly_1 is taken to be big enough. (Since $1 - \beta = \mathsf{poly}(1/2^n)$)

   Summing this failure probability for all $a \in [2^n]$ altogether we know with probability is at most

$$2^n \cdot 2^{-O(\mathsf{poly}(2^n))} = 2^{-O(\mathsf{poly}(2^n))},$$

   which means it could not pass the first step.

2. For the coherency check we can apply Eq. (5) directly again and know for each a, b, the expected error ratio is $\leq 1 - \alpha \leq \frac{1}{4} \cdot \text{threshold}$. (Similarly threshold $:= 2^{-2n-18}(1-\beta)^4$). Thus by the Chernoff bound and similar arguments

$$\forall a, b \in [2^n], \quad \Pr[\text{error ratio} \geq \text{threshold}] \leq 2^{-O(\mathsf{poly}_2(2^n) \cdot 2^{-2n-20}(1-\beta)^4)}$$

which is $2^{-O(\mathsf{poly}(2^n))}$ when poly_2 is taken to be big enough. (Since $1-\beta = \mathsf{poly}(1/2^n)$)

Thus summing this failure probability for all $a, b \in [2^n]$ we know this step fails with probability at most

$$2^n \cdot 2^n \cdot 2^{-O(\mathsf{poly}(2^n))} = 2^{-O(\mathsf{poly}(2^n))}.$$

Thus we get the completeness.

Then we prove a circuit that satisfies Eq. (6) will be rejected with probability $1 - 2^{-O(\mathsf{poly}(2^n))}$. To prove that, we need to understand how the coherency check help us control the form of the states. We will prove the following lemmas step by step.

First, we show the success of coherency check implies the ancilla states have to be close to each other:

Lemma 5.6. *Suppose for some $a, b \in [2^n], a \neq b$, the following equations hold:*

$$\begin{aligned}
&\|(\langle a| \otimes I_t)(U^\dagger \otimes I_t)\mathcal{C}(|a\rangle \otimes |0^t\rangle)\|^2 \geq 1-\delta,\\
&\|(\langle b| \otimes I_t)(U^\dagger \otimes I_t)\mathcal{C}(|b\rangle \otimes |0^t\rangle)\|^2 \geq 1-\delta,\\
&\left\|\left(\frac{\langle a| + \langle b|}{\sqrt{2}} \otimes I_t\right)(U^\dagger \otimes I)\mathcal{C}\left(\frac{|a\rangle + |b\rangle}{\sqrt{2}} \otimes |0^t\rangle\right)\right\|^2 \geq 1-\delta. \qquad (7)
\end{aligned}$$

Define the ancilla states $|\chi_a\rangle$, $|\chi_b\rangle$ via

$$(U^\dagger \otimes I_t)\mathcal{C}(|a\rangle \otimes |0^t\rangle) \approx_{\sqrt{\delta}} |a\rangle \otimes |\chi_a\rangle \qquad (8)$$

$$(U^\dagger \otimes I_t)\mathcal{C}(|b\rangle \otimes |0^t\rangle) \approx_{\sqrt{\delta}} |b\rangle \otimes |\chi_b\rangle \qquad (9)$$

where the right hand sides are the states from projecting $(U^\dagger \otimes I_t)\mathcal{C}(|a\rangle \otimes |0^t\rangle)$, and projecting $(U^\dagger \otimes I_t)\mathcal{C}(|b\rangle \otimes |0^t\rangle)$ on to $|a\rangle, |b\rangle$ respectively.

Then we have

$$|\chi_a\rangle \approx_{4\delta^{1/4}} |\chi_b\rangle \qquad (10)$$

Proof. We can evaluate the left hand side of Eq. (7) and get

$$\begin{aligned}
&\left\|\frac{1}{\sqrt{2}}((\langle a| + \langle b|) \otimes I_t)((U^\dagger \otimes I)\mathcal{C})\frac{1}{\sqrt{2}}((|a\rangle + |b\rangle) \otimes |0^t\rangle)\right\|\\
\approx_{\sqrt{2\delta}}\ &\frac{1}{2}\|((\langle a| + \langle b|) \otimes I_t)(|a\rangle \otimes |\chi_a\rangle + |b\rangle \otimes |\chi_b\rangle)\| \qquad \text{(By Eqs. (8),(9))}\\
=\ &\frac{1}{2}\||\chi_a\rangle + |\chi_b\rangle\|\\
=\ &\sqrt{1 - \frac{1}{4}\||\chi_a\rangle - |\chi_b\rangle\|^2}
\end{aligned}$$

Substitute Eq. (7), we know

$$\sqrt{1 - \frac{1}{4}\||\chi_a\rangle - |\chi_b\rangle\|^2} \geq \sqrt{1-\delta} - \sqrt{2\delta},$$

$$\||\chi_a\rangle - |\chi_b\rangle\| \leq 2\sqrt{2\sqrt{2\delta(1-\delta)} - \delta} \leq 4\delta^{1/4}.$$

The lemma is then proved. □

Furthermore, we can show, when Eq. (10) holds for all pairs (a, b), the operation $(U^\dagger \otimes I)\mathcal{C}$ is indeed close to identity:

Lemma 5.7. *Suppose for all $a, b \in [2^n], a \neq b$, Eqs. (8),(9),(10) holds. Then for all $|\psi\rangle \in \mathbb{C}^{2^n}$,*

$$\|(\langle\psi| \otimes I_t)(U^\dagger \otimes I_t)\mathcal{C}|\psi, 0^t\rangle\|^2 \geq 1 - 10 \cdot 2^{n/2}\delta^{1/4}$$

Proof. Decompose $|\psi\rangle = \sum_{i\in[2^n]} c_i|e_i\rangle$. Take $|\mathrm{aux}\rangle = |\chi_0\rangle$. Then

$$\begin{aligned}
(U^\dagger \otimes I_t)\mathcal{C}(|\psi\rangle \otimes |0^t\rangle) &= \sum_{i\in[2^n]} c_i(U^\dagger \otimes I_t)\mathcal{C}(|e_i\rangle \otimes |0^t\rangle) \\
&\approx_{\sum_i c_i\sqrt{\delta}} \sum_{i\in[2^n]} c_i|e_i\rangle \otimes |\chi_i\rangle && \text{(By Eqs. (8),(9))} \\
&\approx_{\sum_i 4c_i\delta^{1/4}} \sum_{i\in[2^n]} c_i|e_i\rangle \otimes |\mathrm{aux}\rangle && \text{(By Eq. (10))} \\
&= |\psi\rangle \otimes |\mathrm{aux}\rangle,
\end{aligned}$$

which implies

$$\begin{aligned}
\|(\langle\psi| \otimes I_t)(U^\dagger \otimes I_t)\mathcal{C}|\psi, 0^t\rangle\|^2 &\geq (1 - 5\delta^{1/4}\sum_i c_i)^2 \\
&\geq (1 - 5 \cdot 2^{n/2}\delta^{1/4})^2 \\
&\geq 1 - 10 \cdot 2^{n/2}\delta^{1/4}.
\end{aligned}$$

And the proof is completed. □

Then we prove a circuit that satisfies Eq. (6) will be rejected with probability $1 - 2^{-O(\mathsf{poly}(2^n))}$.

1. After the standard basis check, $\mathcal{C}$ has to satisfy the following property, otherwise the verifier will reject with probability $1 - 2^{-O(\mathsf{poly}(2^n))}$:

$$\forall a \in [2^n], \quad \|(\langle a| \otimes I_t)(U^\dagger \otimes I_t)\mathcal{C}(|a\rangle \otimes |0^t\rangle)\|^2 \geq 1 - 2^{-2n}(\frac{1}{11}(1-\beta))^4 \tag{11}$$

That's because otherwise the standard basis test for some $a \in [2^n]$ will have an expected negative ratio $\geq 2^{-2n}(\frac{1}{11}(1-\beta))^4 \geq 4 \cdot \text{threshold}$ (recall threshold $:= 2^{-2n-18}(1-\beta)^4$).

A more detailed calculation is as follows.

$$\begin{aligned}
\Pr[\text{negative ratio} < \text{threshold}] = &\ \Pr[\text{negative samples} < \text{threshold} \cdot \mathsf{poly}_1(2^n)] \\
\leq &\ \exp(-O(\mathbb{E}[\text{negative samples}])) \\
\leq &\ \exp\left(-O(\mathsf{poly}_1(2^n) \cdot 2^{-2n}((1-\beta)/11)^4)\right),
\end{aligned}$$

where the second step follows from the Chernoff bound, and the last step follows from

$$\text{threshold} \cdot \mathsf{poly}_1(2^n) < \frac{1}{4}\mathbb{E}[\text{negative samples}].$$

Thus

$$\Pr[\text{negative ratio} \geq \text{threshold}] \geq 1 - 2^{-O(\mathsf{poly}_1(2^n)\cdot 2^{-2n}((1-\beta)/11)^4)}$$

2. After the coherency check, $\mathcal{C}$ has to satisfy the following property, otherwise the verifier will reject with probability $1-2^{-O(\mathsf{poly}(2^n))}$: for all $a, b \in [2^n]$, $a \neq b$,

$$\|\frac{1}{\sqrt{2}}((\langle a| + \langle b|) \otimes I_t)(U^\dagger \otimes I)\mathcal{C}(\frac{1}{\sqrt{2}}(|a\rangle + |b\rangle) \otimes |0^t\rangle)\|^2 \geq 1 - 2^{-2n}(\frac{1}{11}(1-\beta))^4 \tag{12}$$

   The calculation is similar as the first step.

3. And from Eqs. (11) and (12), Lemma 5.7 implies that for all $|\psi\rangle \in \mathbb{C}^{2^n}$,

$$\|(\langle\psi| \otimes I_t)(U^\dagger \otimes I_t)\mathcal{C}|\psi, 0^t\rangle\|^2 \geq 1 - 10 \cdot 2^{n/2}(2^{-2n}(\frac{1}{11}(1-\beta))^4)^{1/4} > \beta.$$

   However, by the promise this is not possible to be in the no instance.

This completes the proof. □

Claim 5.8. $\mathsf{UMCSP}_{\alpha,\beta}$ *is in* NP *when only linear ancilla qubits are allowed and* $1-\alpha < 2^{-2n-20}(1-\beta)^4$ *and* $1-\beta \geq \mathsf{poly}(1/2^n)$ *(for example,* $1-\alpha = \exp(-2^n), 1-\beta = \mathsf{poly}(1/2^n)$*). However,* $\mathsf{UMCSP}_{\alpha,\beta}$ *is not trivially in* NP *in general.*

Proof. The certificate is the circuit implementation $\mathcal{C}$ that achieves Eq. (5). Now since the circuit only operates on a polynomial-dimension system, the unitary transformation of the whole circuit can be computed and written down using only a polynomial-time classical computer.

The subtlety is to verify whether the unitary computed here satisfies Eq. (5). We can prove it following the same way as the proof of Theorem 5.5. Here the quantum space is always polynomially bounded and a classical polynomial time verifier can simulate the protocol in the proof of Theorem 5.5 classically. (One note is the quantum output samples there can be lazy-sampled.) This completes the proof. □

Next, we showed that SMCSP has a QCMA protocol. Note that since SMCSP is given access to quantum states, it is even not a promise problem under the standard definition. Therefore, we can only say there is a QCMA protocol for this problem.

Theorem 5.9. $\mathsf{SMCSP}_{\alpha,\beta}$ *with gap* $|\alpha-\beta| \geq \mathsf{poly}(s)$ *has a* QCMA *protocol.*

Proof. We use the swap test to check whether the given states and the state generated from the certificate circuit are close. The verifier's algorithm is as follows:

Algorithm 3 The efficient verifier for SMCSP.

```
Input: $s, t \in \mathbb{N}$, $\mathsf{poly}(s)$ copies of $|\psi\rangle$, and quantum circuit $\mathcal{C}$.
Generate $\mathsf{poly}(s)$ $|\phi\rangle = \mathcal{C}|0\rangle$.
Apply swap test to $|\psi\rangle$ and $|\phi\rangle$.
return "Yes" if there are at least $\frac{a+b}{2}$ trials outputs 0.
return "No", otherwise.
```

Given $s, t \in \mathbb{N}$ and $\mathsf{poly}(s)$ copies of $|\psi\rangle$, we first consider the case where there exists a circuit $\mathcal{C}$ such that $\|(\langle\psi| \otimes I_t)\mathcal{C}|0^{n+t}\rangle\|^2 \geq \alpha$. Let $\mathcal{C}$ be the certificate. Then, by applying the swap test to $|\psi\rangle$ and $\mathcal{C}|0\rangle$, the probability that we get 0 (which means identical) is $\frac{1}{2} + \frac{|\langle\psi|\mathcal{C}|0\rangle|^2}{2}$, which is at least $\frac{1+\alpha}{2}$ in this case. We denote the probability of outputs 0 at the i-th trial as X_i. Then, By the Chernoff inequality,

$$\Pr\left[\sum_{i=1}^{\ell} X_i \geq (\frac{1}{2} + \frac{\alpha+\beta}{4})\ell\right] \leq \exp\left(-\frac{(\alpha-\beta)^2\ell}{16}\right).$$

Since $|\alpha - \beta| \geq \frac{1}{\mathsf{poly}(s)}$, the success probability of Algorithm 3 in this case is at least 2/3 by having $\ell = \mathsf{poly}(s)$ trials. Similarly, we can prove the case when there exists no circuit $\mathcal{C}$ such that $\|(\langle\psi| \otimes I_t)\mathcal{C}|0^{n+t}\rangle\|^2 > \beta$. This completes the proof.

□

Given Theorem 5.9, we can also obtain the following result when given classical descriptions of quantum states.

Corollary 5.10. SMCSP *with classical descriptions of quantum states as inputs is in* QCMA*.*

The subtlety is that the verifier needs to construct the state $|\psi\rangle$ given the classical description of $|\psi\rangle$. If the verifier can do this efficiently (in time $\mathsf{poly}(2^n)$), then the rest of the analysis follows the proof for Theorem 5.9. We leave the proof to Appendix E.

For the ease of notation, we will simply denote $\mathsf{UMCSP}_{\alpha,\beta}$ and $\mathsf{SMCSP}_{\alpha,\beta}$ as UMCSP and SMCSP and will specify α and β when it is necessary in the rest of the section.

5.1 Reductions for UMCSP and SMCSP

In this section, we will show search-to-decision reductions for UMCSP and SMCSP. To prove the above results, it is easier for us to consider UMCSP and SMCSP as problems for computing the circuit complexity of given unitaries and states.

We first give formal definitions of approximating functions, unitaries, and states and the corresponding quantum circuit complexities.

Definition 5.11 (Approximating f with precision δ)**.** *We say that a quantum circuit $\mathcal{C}$ that approximates a function $f : \mathbb{Z}^n \to \mathbb{Z}^m$ with precision δ if for all $x \in \mathbb{Z}^n$, there exists $\epsilon' \leq \epsilon$ such that*

$$\mathcal{C}_{f,\delta}|x\rangle|0^t\rangle = \sqrt{1-\epsilon'}|f(x)\rangle|\psi_{f(x)}\rangle + \sqrt{\epsilon'}|\phi_x\rangle. \tag{13}$$

Definition 5.12 (Approximating U with precision δ)**.** *Let U be as a $2^n \times 2^n$ unitary. We define $\mathcal{C}_{U,\epsilon}$ as the circuit that approximates U with precision δ such that for all $|\psi\rangle \in \mathbb{C}^{2^n}$ there exists $\delta' \leq \delta$*

$$\mathcal{C}_{U,\delta}|\psi\rangle|0^t\rangle = \sqrt{1-\delta'}(U|\psi\rangle) \otimes |\psi'\rangle + \sqrt{\delta'}|\phi'\rangle.$$

Here, the additional t qubits for $\mathcal{C}_{U,\delta}$ are ancilla qubits.

Definition 5.13 (Approximating $|\psi\rangle$ with precision δ)**.** *Let $|\psi\rangle \in \mathbb{C}^{2^n}$ be a quantum state. We define $\mathcal{C}_{|\psi\rangle,\epsilon}$ as the circuit that approximates $|\psi\rangle$ with precision δ*

$$\mathcal{C}_{|\psi\rangle,\delta}|0^{n+t}\rangle = \sqrt{1-\delta'}|\psi\rangle|\psi'\rangle + \sqrt{\delta'}|\phi'\rangle$$

Here, $\delta' \leq \delta$ and the additional t qubits are ancilla qubits.

We use $CC(\cdot,\epsilon)$ to denote the quantum circuit complexity of the minimum quantum circuit that approximates the given Boolean functions, states, or unitaries with precision ϵ.

Remark 11 (Upper bounds on $CC(\cdot,\epsilon)$)**.** *For any universal gate set, any unitary U in $\mathbb{C}^{2^n\times 2^n}$ can be ϵ-approximated by a circuit with size at most $\widetilde{O}(n^2 2^{2n}\log\frac{1}{\epsilon})$ [NC11]. The same upper bound also holds for states. The existence of $2^{O(n)}$ upper bounds implies that $CC(\cdot,\epsilon)$ can be computed efficiently given efficient algorithms for* SMCSP *and* UMCSP*.*

5.1.1 Search-to-decision reductions

In the following, we prove search-to-decision reductions for UMCSP and SMCSP. The main intuition for these reductions is that quantum circuits are reversible, which gives us the ability to do some "rewinding tricks". We define the search versions of UMCSP and SMCSP as follows:

Definition 5.14 ($\mathsf{SearchUMCSP}_\epsilon$)**.** *Let $n,t\in\mathbb{N}$. Let $U\in\mathbb{C}^{2^n\times 2^n}$ be a unitary matrix and $\epsilon\in(0,1)$. Let s be the smallest integer such that there exists a quantum circuit $\mathcal{C}$ of size s that uses at most t ancilla bits and for all $|\psi\rangle\in\mathbb{C}^{2^n}$,*

$$\|(\langle\psi|\otimes I_t)(U^\dagger\otimes I_t)\mathcal{C}|\psi,0^t\rangle\|^2\geq 1-\epsilon.$$

Given U, t, and ϵ, the problem is to output a circuit $\mathcal{C}'$ of size at most s that uses at most t ancilla bits and for all $|\psi\rangle\in\mathbb{C}^{2^n}$, $\|(\langle\psi|\otimes I_t)(U^\dagger\otimes I_t)\mathcal{C}'|\psi,0^t\rangle\|^2\geq 1-\epsilon-2^{-cn}$ for every constant $c>0$.

Definition 5.15 ($\mathsf{SearchSMCSP}_{\epsilon,s}$)**.** *Let $n,s,t\in\mathbb{N}$ and $\epsilon\in(0,1)$. Let $|\psi\rangle\in\mathbb{C}^{2^n}$ be a quantum state with the promise that there exists a circuit $\mathcal{C}$ of size at most s and t ancilla bits such that*

$$\|(\langle\psi|\otimes I_{n+t-1})\mathcal{C}|0^{n+t}\rangle\|^2\geq 1-\epsilon.$$

Given (n,s,t) in unary, ϵ, and access to arbitrary many copies of $|\psi\rangle$, the problem is to find a circuit $\mathcal{C}'$ of size at most s and t ancilla bits such that $\|(\langle\psi|\otimes I_{n+t-1})\mathcal{C}'|0^{n+t}\rangle\|^2\geq 1-\epsilon-2^{-cn}$ for every constant $c>0$.

Remark 12. *In Definition 5.15, we have included the upper bound 1^s (the unary representation) as part of the inputs. This mainly follows from the fact that we are considering problems with copies of quantum states. One may expect that we can find s by using binary search with an efficient algorithm for* SMCSP*. However, efficient algorithms for* SMCSP *with $s=2^n$ can run in time* $\mathsf{poly}(2^n)$*, and efficient algorithms for* SearchSMCSP *without 1^s as part of the inputs need to run in time* $\mathsf{poly}(n)$*. Hence, this prevents us from finding s efficiently (in time* $\mathsf{poly}(n)$*) with an efficient algorithm for* SMCSP *(in time* $\mathsf{poly}(s)$*). On the other hand, if we consider the case where* SMCSP *and* SearchSMCSP *have the classical description of the state (instead of copies of the quantum state) as part of the inputs, then there is no need to have 1^s in the inputs of* SearchSMCSP *since we can find s via binary search with efficient algorithms for* SMCSP*.*

In the following, we show search-to-decision reductions for UMCSP and SMCSP when $t=0$ (i.e., no ancilla qubits)[24].

Theorem 5.16. *There exists a search-to-decision reduction for* UMCSP *for $t=0$ (i.e., no ancilla qubits). In particular, if there is a time $T(n)$ algorithm for $\mathsf{UMCSP}_{\alpha,\beta}$ where $\alpha>1-2^{-c_1n}$ and $\alpha-\beta\geq 2^{-c_2n}$ for every constants $c_1,c_2>0$, then there is a time $\mathsf{poly}(T(n),2^n)$ algorithm for $\mathsf{SearchUMCSP}_\epsilon$ where $\epsilon\geq 2^{-c_3n}$ for every constant $c_3>0$ and $t=0$.*

[24] In general, search-to-decision reductions for SMCSP and UMCSP mean that SearchSMCSP reduces to SMCSP and SearchUMCSP reduces to UMCSP for any $n,s,t\in\mathbb{N}$.

Remark 13. *Here we require the gap $\alpha - \beta$ in the decision* UMCSP *oracle to be at least* $\mathsf{poly}(2^{-n})$ *because our* QCMA *upper bound for* UMCSP *(see Theorem 5.5) only works in this regime.*

Proof. Let us first state the reduction in the form of an algorithm with oracle queries to UMCSP as follows.

Algorithm 4 Search-to-decision reduction for UMCSP.

```
Input: $\epsilon \in (0,1)$, $U \in \mathbb{C}^{2^n \times 2^n}$, and a constant $c_3 > 0$.
 1: Let $U_0 = U$, $\Delta = 2^{-2c_3 n}$, $\epsilon_0 = \epsilon$ , and $\epsilon_i = \epsilon_0 + i \cdot \Delta$ for all $i \in \mathbb{N}$.
 2: Use the oracle $\mathsf{UMCSP}_{1-\epsilon_0, 1-\epsilon_0-\Delta}$ to binary-search $s$, the minimum circuit size of $U$.
 3: Set $i = 1$.
 4: while $i < s$ do
 5:     for all gates $h_i$ in $\mathcal{G}$ on all $n$ qubits do
 6:         if $\mathsf{UMCSP}_{1-\epsilon_i, 1-\epsilon_i-\Delta}(U_{i-1}h_i^\dagger, s-i)$ = Yes then
 7:             Set $g_i = h_i$.
 8:             Let $U_i = g_i^\dagger U_{i-1}$.
 9:             Set $i = i + 1$.
10:             Break.
11: return $g_1, \ldots, g_s$.
```

We inductively prove the following claim.

Claim 5.17. *For every $0 < i < s$, at the i-th iteration in line 5, we know that there exists a circuit $\mathcal{C}$ of size at most $s - i + 1$ such that $\min_{|\psi\rangle} |\langle\psi|U_{i-1}^\dagger \mathcal{C}|\psi\rangle|^2 \geq 1 - \epsilon_i$.*

Proof. For the base case we consider $i = 1$ and note that after line 2 in Algorithm 4, we know that there exists a circuit $\mathcal{C}$ of size at most s such that $\min_{|\psi\rangle} |\langle\psi|U_0^\dagger \mathcal{C}|\psi\rangle|^2 \geq 1 - \epsilon - \Delta = 1 - \epsilon_1$. This proves the base case.

Now, suppose the induction statement holds for some i, we first claim that the algorithm must go into the if-loop in line 6. Note that by induction hypothesis there exists a circuit $\mathcal{C}$ of size at most $s - i + 1$ such that $\min_{|\psi\rangle} |\langle\psi|U_{i-1}^\dagger \mathcal{C}|\psi\rangle|^2 \geq 1 - \epsilon_i$. Let g_i be the last gate in $\mathcal{C}$, we know that $\min_{|\psi\rangle} |\langle\psi|(U_{i-1}^\dagger g_i)(g_i^\dagger \mathcal{C})|\psi\rangle\|^2 \geq 1 - \epsilon_i$ and $g_i^\dagger \mathcal{C}$ is a circuit of size at most $s - i + 1 - 1 = s - i$. This shows that the algorithm will go into the if-loop in line 6 in the i-th iteration. Next, after the algorithm goes into line 6 in the i-th iteration, by the correctness of $\mathsf{UMCSP}_{1-\epsilon_i, 1-\epsilon_i-\Delta}$, we know that there is a circuit $\mathcal{C}'$ $(= \mathcal{C} g_i^\dagger)$ of size at most $s - i$ such that $\min_{|\psi\rangle} |\langle\psi|U_i^\dagger \mathcal{C}'|\psi\rangle|^2 \geq 1 - \epsilon_i - \Delta = 1 - \epsilon_{i+1}$. This completes the induction step and hence proves Claim 5.17. □

Finally, with the same argument in the proof of Claim 5.17, we know that

$$\min_{|\psi\rangle} |\langle\psi|U^\dagger g_1 \cdots g_s|\psi\rangle|^2 \geq 1 - \epsilon_s = 1 - \epsilon - s \cdot 2^{-2c_3 n} \geq 1 - \epsilon - 2^{-c_3 n}$$

as desired. Also, notice that the algorithm only queries the UMCSP oracle at most 2^n times and hence the running time is $\mathsf{poly}(T(n), 2^n)$ where $T(n)$ is the running time of the UMCSP oracle. □

Theorem 5.18. *There exists a search-to-decision reduction for* SMCSP *for $t = 0$. In particular, if there is a time $T(n)$ algorithm for $\mathsf{SMCSP}_{\alpha,\beta}$ where $\alpha > 1 - 2^{-c_1 n}$ and $\alpha - \beta \geq 2^{-c_2 n}$ for every constants $c_1, c_2 > 0$, then there is a time $\mathsf{poly}(T(n), s)$ quantum algorithm for $\mathsf{SearchSMCSP}_{\epsilon,s}$ where $\epsilon \geq 2^{-c_3 n}$ for every constant $c_3 > 0$ and $t = 0$.*

Proof. The proof is similar to the proof for Theorem 5.16. We describe the reduction as follows:

Algorithm 5 Search-to-decision reduction for SMCSP.

```
Input: $s \in \mathbb{N}$, $\epsilon \in (0,1)$, access to copies of $|\psi\rangle$, and a constant $c_3 > 0$.
Let $|\psi_0\rangle = |\psi\rangle$, $\Delta = 2^{-2c_3 n}$, $\epsilon_0 = \epsilon$, and $\epsilon_i = \epsilon_0 + i \cdot \Delta$ for all $i \in \mathbb{N}$.
Use the oracle $\mathsf{SMCSP}_{1-\epsilon_0,1-\epsilon-\Delta}$ to binary-search $s^* \leq s$, the minimum circuit size of $|\psi\rangle$.
Set $i = 1$
while $i < s^*$ do
    for all gates $h_i$ in $\mathcal{G}$ on all $n + t$ qubits do
        if $\mathsf{SMCSP}_{1-\epsilon_i,1-\epsilon_i-\Delta}(|\psi_i\rangle, s^* - i) =$ Yes then
            Set $g_i = h_i$.
            Let $|\psi_i\rangle = g_i^\dagger |\psi_{i-1}\rangle$.
            Set $i = i + 1$.
            Break.
return $g_1, \ldots, g_{s^*}$.
```

The analysis is similar to the proof of Theorem 5.16. Notice that given access to the quantum state $|\psi\rangle$, we can uncompute the gates using a quantum computer. Therefore, the search-to-decision reduction still holds. □

Regarding SMCSP and $\mathsf{SearchSMCSP}$ which have the classical description of $|\psi\rangle$ as part of the inputs (instead of copies $|\psi\rangle$), we can also obtain the search-to-decision reduction following the same framework. The only difference is that the algorithm uncomputes the gates from the states by matrix-vector multiplication instead of applying the inverse of the gates on the states. The runtime of the matrix-vector multiplication is $\mathsf{poly}(2^n)$. Note that, as we have mentioned in Remark 12, $\mathsf{SearchSMCSP}$ in this case does not need to have the upper bound s in the inputs.

Corollary 5.19. *There exists a search-to-decision reduction for* SMCSP*, where the search and the decision problems are given the classical descriptions of the states in inputs.*

It is worth noting that Algorithm 5 and Algorithm 4 do not directly work when considering quantum circuits that are allowed to use ancilla qubits (i.e., $t > 0$). This follows from the fact that, based on definitions of UMCSP and SMCSP, a quantum circuit $\mathcal{C}$ that implements the target unitary or state can apply an arbitrary operator on the ancilla qubits, i.e., $C^\dagger(U \otimes I) \neq I$. In this case, we do not know the unitary of $\mathcal{C}$ or the state of $\mathcal{C}|0\rangle$, and thus we cannot run Algorithm 5 and Algorithm 4.

5.1.2 Self-reduction for SMCSP

In this section, we show that SMCSP is approximately self-reducible. In other words, one can approximate the circuit complexity of an n-qubit state by computing the circuit complexity of an $(n-1)$-qubit state.

Theorem 5.20. *Let $\mathcal{A}_\delta$ be an efficient algorithm for computing $CC(|\phi\rangle, \delta)$ for any $(n-1)$-qubit state $|\phi\rangle$. Let $|\psi\rangle$ be any n-qubit state. Given (n, s) in unary, $\epsilon \in (0,1)$, and access to copies of $|\psi\rangle$, $CC(|\psi\rangle, \epsilon)$ can be approximated efficiently using $\mathcal{A}_\delta$.*

Recall that $CC(\cdot, \epsilon)$ denotes the quantum circuit complexity of the minimum quantum circuit that approximates the given states with precision ϵ.

Proof. We first fix the gate set to be $CNOT$ and all single-qubit rotations and prove the theorem under this particular gate set. Then, we generalize the theorem to all gate sets by the Solovay-Kitaev Theorem in Theorem 2.4.

Let $|\psi\rangle \in \mathbb{C}^{2^n}$ be an arbitrary n-qubit quantum state. Without loss of generality, we can represent $|\psi\rangle$ as

$$c_0|0\rangle|\psi_0\rangle + c_1|1\rangle|\psi_1\rangle,$$

where $c_0, c_1 \in \mathbb{C}$ and $|c_0|^2 + |c_1|^2 = 1$. $|1\rangle$ and $|0\rangle$ are single-qubit states, and $|\psi_0\rangle$ and $|\psi_1\rangle$ are states on $n-1$ qubits and are not orthogonal in general. Our goal is show upper and lower bounds for $CC(|\psi\rangle, \epsilon)$ from $CC(|\psi_0\rangle, \delta)$ and $CC(|\psi_1\rangle, \delta)$.

To prove the upper and the lower bounds, we first estimate $|c_0|^2$ and $|c_1|^2$ to precision $\epsilon/4$ by using quantum amplitude estimation. We denote the estimated values as $|c_0'|^2$ and $|c_1'|^2$ and consider the following two cases.

1. $|c_0'|^2$ or $|c_1'|^2 < \epsilon/2$; and
2. $|c_0'|^2, |c_1'|^2 \geq \epsilon/2$.

Upper bound In case that $|c_0'|^2$ (or $|c_1'|^2$) is less than $\frac{\epsilon}{2}$, $|c_1|^2$ (or $|c_0|^2$) must be greater than $1 - \frac{3\epsilon}{4}$, which implies that the square of the inner product of $|\psi\rangle$ and $|1\rangle|\psi_1\rangle$ (or $|0\rangle|\psi_0\rangle$) is at least $1 - \frac{3\epsilon}{4}$. Therefore,

$$CC(|\psi\rangle, \epsilon) \leq CC(|\psi_1\rangle, \epsilon/4) \text{ or } CC(|\psi_0\rangle, \epsilon/4).$$

In case that both $|c_0'|^2$ and $|c_1'|^2$ are at least $\frac{\epsilon}{2}$, Let $\mathcal{C}_0 = \mathcal{C}_{|\psi_0\rangle,\epsilon}$ and $\mathcal{C}_1 = \mathcal{C}_{|\psi_1\rangle,\epsilon}$. Then, there exists $\mathcal{C}^*$ that approximates $|\psi\rangle$ with precision ϵ as follows:

$$\begin{aligned}
|0^n\rangle &\xrightarrow{R\otimes I_{n-1}} c_0|0\rangle|0^{n-1}\rangle + c_1|1\rangle|0^{n-1}\rangle \\
&\xrightarrow{control-\mathcal{C}_1} c_0|0\rangle|0^{n-1}\rangle + c_1|1\rangle\mathcal{C}_1|0^{n-1}\rangle \\
&\xrightarrow{X\otimes I_{n-1}} c_0|1\rangle|0^{n-1}\rangle + c_1|0\rangle\mathcal{C}_1|0^{n-1}\rangle \\
&\xrightarrow{control-\mathcal{C}_0} c_0|1\rangle\mathcal{C}_0|0^{n-1}\rangle + c_1|0\rangle\mathcal{C}_1|0^{n-1}\rangle \\
&\xrightarrow{X\otimes I_{n-1}} c_0|0\rangle\mathcal{C}_0|0^{n-1}\rangle + c_1|1\rangle\mathcal{C}_1|0^{n-1}\rangle
\end{aligned}$$

Here R is a single-qubit rotation gate that rotates $|0\rangle$ to $c_0|0\rangle + c_1|1\rangle$. Since our gate set includes all single-qubit rotations, the cost of R is just 1. For $control - \mathcal{C}_0$ and $control - \mathcal{C}_1$, we can think of it as every gate in $\mathcal{C}_i$ is controlled by an additional qubit, i.e., R becomes $control - R$ and CNOT becomes Toffoli gate. By the composition methods in [NC11], we can implement these control gates with only constant multiplicative overhead. Hence, $|\mathcal{C}^*| \leq k \cdot (|\mathcal{C}_0| + |\mathcal{C}_1|) + 3$ for some constant k, and we can conclude that

$$CC(|\psi\rangle, \epsilon) \leq k \cdot (CC(|\psi_0\rangle, \epsilon) + CC(|\psi_1\rangle, \epsilon)) + 3.$$

Lower bound Let $\mathcal{C}$ be the minimum quantum circuit that approximates $|\psi\rangle$ with precision ϵ.

When $|c_0'|^2$ and $|c_1'|^2$ are both at least $\epsilon/2$, $|c_0|^2$ and $|c_1|^2$ are at least $\epsilon/4$ where $|c_0'|^2$ is the estimated value of $|c_0|^2$. Intuitively, we can obtain $|\psi_0\rangle$ or $|\psi_1\rangle$ by parallelly applying $\mathcal{C}$ on $O(\frac{1}{\epsilon})$-many $|0^n\rangle$ states and measuring the first qubits of all the outputs states in the computational basis. By deferring all these measurements toward the end of the computation, we obtain

$$CC(|\psi_i\rangle, \epsilon') \leq k^*(CC(|\psi\rangle, \epsilon) + h)$$

for $i = 0, 1$, $h = O(1)$, and $k^* = O(1/\epsilon)$. Here $\epsilon \leq \epsilon' \leq (1 - \frac{\epsilon}{4})^{k^*} + \epsilon$. The additional constant cost h is from the overhead of deferring measurements.

When $|c_0'|^2$ or $|c_0'|^2$ is at least $1 - \epsilon/2$, the circuit for $|\psi\rangle$ is already a good approximation for $|\psi_1\rangle$ following the same reason for proving the upper bound in the same case. This implies that

$$CC(|\psi_i\rangle, 4\epsilon) \leq CC(|\psi\rangle, \epsilon).$$

The reduction The algorithm is as follows:

1. Estimating $|c_0|$ and $|c_1|$ with precision $\epsilon/4$.
2. Approximate $CC(|\psi\rangle, \epsilon)$ according to $|c_0'|$ and $|c_1'|$.
   - When $|c_0'|^2$ or $|c_1'|^2 \leq \frac{\epsilon}{2}$, compute $CC(|\psi_i\rangle, \epsilon/4)$ and $CC(|\psi_i\rangle, 4\epsilon)$ for $i = 0, 1$. Then,
   $$CC(|\psi_i\rangle, 4\epsilon) \leq CC(|\psi\rangle, \epsilon) \leq CC(|\psi_i\rangle, \epsilon/4).$$
   - When $|c_0'|^2, |c_1'|^2 \geq \epsilon/2$, compute $CC(|\psi_i\rangle, \epsilon')$ and $CC(|\psi_i\rangle, \epsilon)$ for $i = 0, 1$. Then,
   $$\frac{1}{k^*} \cdot \max_{i=0,1} (CC(|\psi_i\rangle, \epsilon')) - h \leq CC(|\psi\rangle, \epsilon) \leq k \cdot (CC(|\psi_0\rangle, \epsilon) + CC(|\psi_1\rangle, \epsilon)) + 3$$

For the running time of the reduction, we can estimate $|c_0|^2$ and $|c_1|^2$ with precision $\epsilon/4$ in time $\mathsf{poly}(1/\epsilon)$ using quantum amplitude estimation. In case that $|c_0'|^2$ (or $|c_1'|^2$) is less than $\frac{\epsilon}{2}$, we only need to compute $CC(|\psi_1\rangle, \epsilon/4)$ by having many enough copies of $|\psi_1\rangle$, which can be efficiently obtained by measuring $|\psi\rangle$. In case that both $|c_0'|^2$ and $|c_1'|^2$ are at least $\frac{\epsilon}{2}$, $|c_0|$ and $|c_1|$ must be at least $\frac{\epsilon}{4}$. Then, we can still obtain sufficiently many copies of $|\psi_0\rangle$ and $|\psi_1\rangle$ in time $\mathsf{poly}(\frac{1}{\epsilon})$ to compute $CC(\psi_0, \epsilon)$ and $CC(\psi_1, \epsilon)$.

Finally, we generalize the results above to arbitrary universal gate set by applying the Solovay-Kitaev Theorem. This gives upper bounds multiplicative overhead $\mathsf{polylog}\,\frac{CC(|\psi_i\rangle,\delta)}{\epsilon}$ and lower bounds multiplicative overhead $\mathsf{polylog}^{-1}\,\frac{CC(|\psi_i\rangle,\delta)}{\epsilon}$, where the choices of i and δ depend on the cases.

□

Remark 14. *Theorem 5.20 also holds when the problem is given the classical description of the quantum state. When considering the version with classical descriptions of states, the reduction becomes even simpler since c_0 and c_1 can be easily computed from the input.*

5.1.3 Reducing MQCSP to UMCSP

In the following, we present a reduction from MQCSP to UMCSP. We first introduce a unitary that trivially encode a given Boolean function.

Definition 5.21 (Trivial unitary encoding of Boolean functions (U_f))**.** *Let $f : \mathbb{Z}^n \to \mathbb{Z}^m$. We define U_f as a $2^{n+m} \times 2^{n+m}$ unitary such that for all $x \in \mathbb{Z}^n$*

$$U_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle$$

Obviously, given the truth table of a function $f : \{0,1\}^n \to \{0,1\}^m$, one can compute U_f in time $\mathsf{poly}(2^n)$. Then, one might expect that the circuit complexity of f is equal to of U_f (in Definition 5.21). However, this is not the case in general since there are many unitaries that can compute f without the form of U_f. In the following lemma, we show that one can give both upper and lower bounds for $CC(f)$ by the quantities $CC(U_f, \epsilon)$ and $CC(U_f, 2\epsilon)$

Lemma 5.22.

$$\frac{CC(U_f, 2\epsilon)}{2} - m \leq CC(f,\epsilon) \leq CC(U_f,\epsilon)$$

Proof. It is easy to see that given $\mathtt{tt}(f)$, one can compute U_f in time $2^{O(n+m)}$ which is polynomial in $|T(f)| = 2^{n+m}$.

We first consider the case where $CC(f)$ and $CC(U_f)$ can be computed with probability 1. We can prove the first inequality as follows:

$$\begin{aligned} |x\rangle|0\rangle &\xrightarrow{\mathcal{C}_f} e^{-i\theta_x}|f(x)\rangle|\psi_x\rangle \\ &\xrightarrow{copy} e^{-i\theta_x}|f(x)\rangle|f(x)\rangle|\psi_x\rangle \\ &\xrightarrow{\mathcal{C}_f^\dagger} |f(x)\rangle|x\rangle|0\rangle, \end{aligned} \tag{14}$$

where $e^{-i\theta}$ are the global coefficient that C_f might have for each θ_x. $C_f^\dagger(copy)C_f$ perfectly computes U_f on all $x \in \{0,1\}^n$ without any global coefficient. This implies that for all $|\psi\rangle \in \mathbb{C}^{2^n}$, $C_f^\dagger(copy)C_f$ computes $U_f|\psi\rangle$ perfectly. The cost for applying this circuit is $2CC(f) + m$. Therefore, we can conclude that $CC(U_f) \leq 2CC(f)+m$. The second inequality is true since a circuit for implementing U_f is also a circuit for f by definition. Note that the global phase in Eq. (14) can be absorbed into the second register; however, we write it down here to help explain why $C_f^\dagger(copy)C_f$ implements U_f not just only on the computational basis, but on all the states.

In the following, we consider the case where we allow U_f and f to be computed with probability at least some thresholds.

$$\begin{aligned} |x\rangle|0\rangle &\xrightarrow{\mathcal{C}_{f,\epsilon}} \sqrt{1-\epsilon}|f(x)\rangle|\psi_{f(x)}\rangle + \sqrt{\epsilon}(\sum_{y\neq f(x)} c_y|y\rangle|\phi'_{x,y}\rangle) \\ &\xrightarrow{Copy} \sqrt{1-\epsilon}|f(x)\rangle|f(x)\rangle|\psi_{f(x)}\rangle + \sqrt{\epsilon}(\sum_{y\neq f(x)} c_y|y\rangle|y\rangle|\phi'_{x,y}\rangle) \\ &= |f(x)\rangle(\sqrt{1-\epsilon}|f(x)\rangle|\psi_{f(x)}\rangle + \sqrt{\epsilon}(\sum_{y\neq f(x)} c_y|y\rangle|\phi'_{x,y}\rangle)) \\ &\quad + \sqrt{\epsilon}(\sum_{y\neq f(x)} c_y|y\rangle|y\rangle|\phi'_{x,y}\rangle - \sum_{y\neq f(x)} c_y|f(x)\rangle|y\rangle|\phi'_{x,y}\rangle) \\ &\xrightarrow{\mathcal{C}^\dagger_{f,\epsilon}} |f(x)\rangle|x\rangle|0\rangle + |\psi'_x\rangle. \end{aligned} \tag{15}$$

Since $\langle f(x), x, 0|\psi'_x\rangle = -\epsilon$ and $\langle\psi'_x|\psi'_x\rangle = 2\epsilon$, we have that

$$|\psi'_x\rangle = -\epsilon|f(x),x,0\rangle + \sqrt{2\epsilon-\epsilon^2}|\psi''_x\rangle.$$

Therefore, we can rewrite Eq. (15) as

$$(1-\epsilon)|f(x),x,0\rangle + \sqrt{2\epsilon-\epsilon^2}|\psi''_x\rangle,$$

which implies that the circuit $C^\dagger_{f,\epsilon}(Copy)C_{f,\epsilon}$ can compute U_f with probability $(1-\epsilon)^2 < 1-2\epsilon$, i.e., $CC(U_f, 2\epsilon) \leq 2CC(f,\epsilon) + m$. $CC(f,\epsilon) \leq CC(U_f,\epsilon)$ is also trivial by the definition. □

We describe an algorithm to approximate $CC(f)$ given an oracle to UMCSP.

Algorithm 6 A reduction from MQCSP to UMCSP

```
Input: Given tt(f) for $f : \{0,1\}^n \to \{0,1\}^m$
Construct $U_f$.
Use UMCSP oracle to compute $s = CC(U_f)$.
return $(\frac{s}{2} - m, s)$.
```

Theorem 5.23. MQCSP$[s/2 - 1, s] \leq$ UMCSP.

Proof. By Lemma 5.22, $CC(f, \epsilon)$ is between $\frac{CC(U_f, 2\epsilon)}{2} - 1$ and $CC(U_f, \epsilon)$ when f is a Boolean function. To compute $CC(U_f, \epsilon)$, we can use the oracle for $\mathsf{UMCSP}_{1-\epsilon,\beta}$, where $\beta \leq 1 - \epsilon - \frac{1}{\mathsf{poly}}$. For $CC(U_f, 2\epsilon)$, we use the oracle for $\mathsf{UMCSP}_{1-2\epsilon,\beta'}$, where $\beta' \leq 1 - 2\epsilon - \frac{1}{\mathsf{poly}}$. This completes the proof. □

Remark 15. *One may expect that we can use Algorithm 6 and* NP*-hardness result about* multiMCSP *to prove* NP*-hardness of* UMCSP*. However, since the reduction for the multioutput MCSP generates functions with exponential-size output string, it make the first inequality in Lemma 5.22 fail. Therefore, whether* UMCSP *is* NP*-hard or not is still open.*

5.2 Applications of SMCSP and UMCSP

In this part, we give applications of UMCSP and SMCSP to other fields in computer science and physics. For SMCSP, we focus on the version with multiple quantum states as inputs.

5.2.1 Applications of UMCSP

A question Aaronson raised in [Aar16] is whether there exists an efficient quantum process that generates a family of unitaries that are indistinguishable from random unitaries given the full description of the unitary. Obviously, if we can solve UMCSP efficiently, we can distinguish truly random unitaries from unitaries generated from efficient quantum process.

Theorem 5.24. *If* UMCSP *has efficient (quantum) algorithms, then there is no efficient quantum process that generates a family of unitaries indistinguishable from random unitaries given the full description of the unitary.*

Besides, some results about MQCSP in Section 4 also hold for UMCSP by Theorem 5.23 and Algorithm 6. In the following, we list some results that trivially holds.

Corollary 5.25. *If* UMCSP $\in$ BQP*, then there is no* qOWF*.*

Corollary 5.26. *If there exists a quantum-secure $i\mathcal{O}$, then* UMCSP $\in$ BQP *implies* NP $\subseteq$ coRQP*.*

Corollary 5.27. *Assume* UMCSP $\in$ BQP*. Then, there exists a* BQP *algorithm that, given the truth-table of an n-variable Boolean function of quantum circuit complexity $2^{\Omega(n)}$, output $2^{\Omega(n)}$ Boolean functions on $m = \Omega(n)$ variables each, such that all of the output functions have quantum circuit complexity greater than $\frac{2^m}{(c+2)m}$ for any $c > 0$.*

Corollary 5.25, Corollary 5.26 and Corollary 5.27 hold since we use the MQCSP oracle as a distinguisher to distinguish functions whose sizes have a large gap, i.e., functions with quantum circuit complexity $\mathsf{poly}(n)$ from functions with quantum circuit complexity $2^{\Omega(n)}$. As the UMCSP oracle can solve MQCSP$[\frac{s}{2} - 1, s]$, the existence of efficient algorithms for UMCSP also implies the same results.

Corollary 5.28. *If* $\mathsf{UMCSP} \in \mathsf{BQP}$*, then* $\mathsf{BQE} \not\subset \mathsf{BQC}[n^k]$ *for all constant* $k \in \mathbb{N}$*.*

Corollary 5.28 holds because for the gap version of MQCSP with a constant gap, it gives a *promise* BQP-natural property, which is defined in [AGG+20]. Suppose we have an efficient quantum algorithm for solving $\mathsf{MQCSP}[2^{\epsilon n}/2-1, 2^{\epsilon n}]$ for small constant ϵ, then it will reject any function with quantum circuit complexity less than $2^{\epsilon n}/2$ and will accept another large subset of functions with quantum circuit complexity larger than $2^{\epsilon n}$. Then, we can use the technique in [AGG+20] to construct the hard language $\mathcal{L}$ from the quantum PRG (Theorem 4.17) and promise quantum natural property. The remaining proof of Theorem 4.15 will work after this adaptation.

5.2.2 Pseudorandom states

An efficient algorithm for SMCSP gives an efficient distinguisher for separating states with large circuit complexity from states with small circuit complexity given many copies of the state. Obviously, this gives us a way to distinguish random states from states that are generated from some efficient process.

Definition 5.29 (Pseudorandom states (PRS) ([JLS18]))**.** *Let* κ *be the security parameter. Let* K *be the key space and* $\mathcal{H}$ *be the state space both parameterized by* κ*. A family of quantum states* $\{|\psi_k\rangle\}_{k\in K} \subset \mathcal{H}$ *is pseudorandom if the following properties hold.*

1. ***Efficiency:*** *There is a quantum polynomial-time algorithm* G *that given* $k \in K$*, can generate* $|\psi_k\rangle$*.*
2. ***Indistinguishability:*** *For all quantum polynomial-time algorithm* $\mathcal{A}$ *and any* $m = \mathsf{poly}(\kappa)$

$$|\Pr_k[\mathcal{A}(|\psi_k\rangle) = 1] - \Pr_{|\psi\rangle\leftarrow\mu}[\mathcal{A}(|\psi\rangle) = 1]| \leq \mathsf{negl}(\kappa),$$

where μ *is the Haar measure on* $\mathcal{H}$*.*

Theorem 5.30. *If* $\mathsf{SMCSP} \in \mathsf{BQP}$*, then there is no* PRS *and* qOWF*.*

Proof. Let $|\psi\rangle$ be the state and $\mathcal{A}$ be the algorithm to distinguish whether $|\psi\rangle$ is a truely random state or from a particular efficient algorithm. In the definition of PRS, $\mathcal{A}$ knows the algorithm for constructing the PRS (but it does not know the key.) Therefore, $\mathcal{A}$ also knows the circuit complexity s for generating the PRS $|\psi\rangle$. Suppose $|\psi\rangle$ is an n-qubit PRS generated by a quantum circuit with size s, by solving SMCSP with size parameter s and $\mathsf{poly}(s)$ copies of $|\psi\rangle$, the adversary can distinguish $|\psi\rangle$ from a Haar random state with high probability since a Haar random state has complexity exponential in n.

Finally, by [JLS18], there exist PRS assuming the existence of qOWF. Since we can break any PRS scheme by solving SMCSP, we can also invert any qOWF by solving SMCSP. □

5.2.3 Estimating the wormhole volume

Integrating general relativity and quantum mechanics into a comprehensive theorem for quantum gravity is one of the most challenging physics problems. The AdS/CFT correspondence plays an important role in this line of research. The AdS/CFT correspondence conjectures the duality between the Anti-de Sitter space (i.e., the bulk) and a conformal field theory (i.e., the boundary). In particular, it conjectures the dictionary maps from wormholes and operators in the bulk to quantum states and operators on the boundary. One fascinating puzzle in Ads/CFT correspondence is about the volume of the wormhole. The volume of the wormhole grows steadily with time; what is the quantity of the corresponding quantum state on the boundary that has this feature? Susskind proposed the *Complexity=Volume Conjecture* [Sus16]. It states that the wormhole volume equals the quantum circuit complexity of the corresponding quantum state times some constant c. In the following, we give a brief description of the Complexity=Volume Conjecture and related backgrounds. One can see [Sus16, BFV20] for detailed discussions.

AdS/CFT Correspondence AdS/CFT correspondence conjectures a dual map Φ between wormholes (AdS side) and quantum systems (CFT side). The setting we consider here is wormholes with two-sided blackholes. Under this setting, the CFT side is divided into left and right systems denoted by Hamiltonians H_L and H_R, where the left and right CFT systems are on n qubits (compatible with the entropy of the wormhole 2^n). We denote the whole system (with both left and right systems) as $H = H_L + H_R$. An early model of AdS/CFT goes under the ER=EPR slogan: the wormhole (Einstein-Rosen Bridge) is dual to maximally entangled (EPR) pair. The corresponding state is usually called the thermal field double (TFD) state $|TFD\rangle$ [MS13]

$$|TFD\rangle = \frac{1}{\sqrt{2^n}} \sum_i e^{-E_i/\beta} |i\rangle_L |i\rangle_R, \tag{16}$$

where $|i\rangle_L$ and $|i\rangle_R$ are energy eigenstates of H_L and H_R.

The quantum state after time-t evolution is

$$|TFD(t)\rangle = e^{-iHt}|TFD\rangle.$$

Recall the dual map Φ between a wormhole (AdS side) and a quantum system (CFT side), one can represent the wormhole after time t as $\Phi(e^{-iHt}|TFD\rangle)$ (and view $\Phi(|TFD\rangle)$ as the wormhole at time 0).

The statement of Complexity=Volume Conjecture can be stated as follows:

Conjecture 1 (Complexity=Volume Conjecture [Sus16])**.** *Consider a wormhole and its corresponding CFT system H, for some suitable ϵ, c, and $0 \leq t \leq O(2^n)$,*

$$CC_\epsilon(|TFD\rangle, |TFD(t)\rangle) = c \cdot Volume(\Phi(e^{-iHt}|TFD\rangle)),$$

where $CC_\epsilon(|TFD\rangle, |TFD(t)\rangle)$ is the circuit complexity for constructing $|TFD(t)\rangle$ from $|TFD\rangle$ with at most ϵ error.

The SMCSP oracle gives a way to identify the quantum circuit complexity of the given state. This implies that if the dictionary map between the wormhole and the quantum state is efficient, one can estimate the wormhole volume in two ways. 1) Apply the dictionary map to transfer the wormhole to the corresponding state and then apply the SMCSP oracle for the circuit complexity, which gives the wormhole volume. 2) As it is hard to imagine mapping wormholes to states, one can view the SMCSP oracle as a POVM and then uses the dictionary map to transfer the POVM to the corresponding operators in the bulk to measure the volume. This gives the following lemma.

Theorem 5.31. *Assuming the Volume=Complexity Conjecture, if the dictionary map can be computed in quantum polynomial time and* SMCSP $\in$ BQP*, then one can estimate the wormhole volume in quantum polynomial time when the volume is at most polynomially large.*

Here, we require the volume is at most polynomially large. This follows from the fact that we need a upper bound polynomial in n for doing binary search to find the circuit complexity with an efficient SMCSP algorithm. If the upper bound is $2^{O(n)}$, the running time of the SMCSP algorithm can be $\mathsf{poly}(2^n)$. Therefore a quantum polynomial-time algorithm for SMCSP in this case would not imply a quantum polynomial-time algorithm for estimating the wormhole's volume.

Besides, recall that the wormhole is initially described by $|TFD\rangle$. So, we also need to modify the definition of SMCSP to allow such an initial state.

Bouland et al. in [BFV20] used this correspondence in a reverse way. In particular, they showed that if the dictionary map and simulating the state in the bulk are efficient (i.e., the quantum Extended Church-Turing thesis holds for quantum gravity), then one can efficiently distinguish certain PRS from Haar random state by mapping the state to the wormhole in the bulk and do the simulation in the bulk to estimate the volume. Following this idea, we can also conclude that if there is a quantum polynomial time algorithm for estimating the wormhole's volume, then one can compute the circuit complexity of the corresponding quantum state efficiently assuming the the Volume=Complexity Conjecture and that the dictionary map is efficient[25].

5.2.4 Succinct state tomography

In the following, we show that solving SMCSP can help to have a succinct answer to state tomography for states which are generated from a polynomial-size circuit without any measurement.

Definition 5.32 (Succinct state tomography)**.** *Let $|\psi\rangle$ be an n-qubit quantum state that is generated from a quantum circuit $\mathcal{C}$ of size s without using measurement and ancilla qubits. Given $\mathsf{poly}(n)$ copies of $|\psi\rangle$ and an upper bound s' where $s \leq s' \leq \mathsf{poly}(n)$, the problem is to output a succinct description (e.g., $\mathcal{C}$) of $|\psi\rangle$.*

Theorem 5.33. *Succinct state tomography in Def. 5.32 reduces to SMCSP.*

Proof. Obviously, succinct state tomography reduces to the search version of SMCSP. By the search-to-decision reduction in Theorem 5.18, we can solve succinct state tomography by solving SMCSP. □

6 Acknowledgment

We are grateful to Scott Aaronson and Boaz Barak for helpful discussions and valuable comments on our manuscript. We would like to thank Lijie Chen, Kai-Min Chung, Matthew Coudron, Yanyi Liu, and Fang Song for useful discussions.

NHC's research is support by the U.S. Department of Defense and NIST through the Hartree Postdoctoral Fellowship at QuICS and by NSF through IUCRC Planning Grant Indiana University: Center for Quantum Technologies (CQT) under award number 2052730.

RZ's research is supported by NSF Grant CCF-1648712 and Scott Aaronson's Vannevar Bush Faculty Fellowship from the US Department of Defense.

CNC's research is supported by Boaz Barak's NSF awards CCF 1565264 and CNS 1618026.

JZ's research is supported by Adam Smith's NSF awards 1763786.

[25] Note that this does not give an efficient algorithm for solving SMCSP in general since it can only solve SMCSP for CFT states.

References

[Aar06] Scott Aaronson. Oracles are subtle but not malicious. In *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, pages 15–pp. IEEE, 2006.

[Aar16] Scott Aaronson. The complexity of quantum states and transformations: from quantum money to black holes. *arXiv preprint arXiv:1607.05256*, 2016.

[Aar18] Scott Aaronson. Shadow tomography of quantum states. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 325–338, New York, NY, USA, 2018. Association for Computing Machinery.

[ACL$^+$20] Scott Aaronson, Nai-Hui Chia, Han-Hsuan Lin, Chunhao Wang, and Ruizhe Zhang. On the quantum complexity of closest pair and related problems. In *35th Computational Complexity Conference (CCC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[AD14] Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014*, pages 25–32, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[AdW17] Srinivasan Arunachalam and Ronald de Wolf. Guest column: A survey of quantum learning theory. *ACM SIGACT News*, 48(2):41–67, 2017.

[AGG$^+$20] Srinivasan Arunachalam, Alex B Grilo, Tom Gur, Igor C Oliveira, and Aarthi Sundaram. Quantum learning algorithms imply circuit lower bounds. *arXiv preprint arXiv:2012.01920*, 2020.

[AL20] Andris Ambainis and Nikita Larka. Quantum algorithms for computational geometry problems. In *15th Conference on the Theory of Quantum Computation, Communication and Cryptography*, 2020.

[BCWdW01] Harry Buhrman, Richard Cleve, John Watrous, and Ronald de Wolf. Quantum fingerprinting. *Phys. Rev. Lett.*, 87:167902, Sep 2001.

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for io: Circular-secure lwe suffices. *IACR Cryptol. ePrint Arch*, 2020:1024, 2020.

[BFV20] Adam Bouland, Bill Fefferman, and Umesh Vazirani. Computational Pseudorandomness, the Wormhole Growth Paradox, and Constraints on the AdS/CFT Duality (Abstract). In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 63:1–63:2, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[BOG08] Michael Ben-Or and Dan Gutfreund. Trading help for interaction in statistical zero-knowledge proofs. *Journal of Cryptology*, 16:95–116, 03 2008.

[BPS21] Harry Buhrman, Subhasree Patro, and Florian Speelman. A Framework of Quantum Strong Exponential-Time Hypotheses. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer*

Science (STACS 2021), volume 187 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[BT06] Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for np problems. *SIAM Journal on Computing*, 36(4):1119–1159, 2006.

[BWP$^+$17] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[CCCW21] Shouvanik Chakrabarti, Chi-Ning Chou, Kai-Min Chung Chung, and Xiaodi Wu. Scalable verification of quantum supremacy based on circuit obfuscation. *Manuscript*, 2021.

[CHO$^+$20] Lijie Chen, Shuichi Hirahara, Igor C Oliveira, Ján Pich, Ninad Rajgopal, and Rahul Santhanam. Beyond natural proofs: Hardness magnification and locality. *Leibniz International Proceedings in Informatics*, 151, 2020.

[CHS20] Nai-Hui Chia, Sean Hallgren, and Fang Song. On Basing One-way Permutations on NP-hard Problems under Quantum Reductions. *Quantum*, 4:312, August 2020.

[CIKK16] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In *Proceedings of the 31st Conference on Computational Complexity*, CCC '16, Dagstuhl, DEU, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[CKP13] Alessandro Cosentino, Robin Kothari, and Adam Paetznick. Dequantizing read-once quantum formulas. In *8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, August 1986.

[GS20] François Le Gall and Saeed Seddighin. Quantum meets fine-grained complexity: Sublinear time quantum algorithms for string problems. *arXiv preprint arXiv:2010.12122*, 2020.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[Hir18] Shuichi Hirahara. Non-black-box worst-case to average-case reductions within np. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 247–258. IEEE, 2018.

[HOS18] Shuichi Hirahara, Igor C Oliveira, and Rahul Santhanam. Np-hardness of minimum circuit size problem for or-and-mod circuits. In *33rd Computational Complexity Conference (CCC 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[HS07] Lisa Hellerstein and Rocco A Servedio. On PAC learning algorithms for rich Boolean function classes. *Theoretical Computer Science*, 384(1):66–76, 2007.

[IKV18] Russell Impagliazzo, Valentine Kabanets, and Ilya Volkovich. The power of natural properties as oracles. In *33rd Computational Complexity Conference (CCC 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[Ila19] R. Ilango. AC0[p] lower bounds and np-hardness for variants of mcsp. *Electron. Colloquium Comput. Complex.*, 26:21, 2019.

[Ila20a] Rahul Ilango. Connecting Perebor Conjectures: Towards a Search to Decision Reduction for Minimizing Formulas. In Shubhangi Saraf, editor, *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:35. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[Ila20b] Rahul Ilango. Constant depth formula and partial function versions of mcsp are hard. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 424–433. IEEE, 2020.

[ILO20] Rahul Ilango, Bruno Loff, and Igor C. Oliveira. Np-hardness of circuit minimization for multi-output functions. In *35th Computational Complexity Conference (CCC 2020)*, CCC '20, Dagstuhl, DEU, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[IRS21] Rahul Ilango, Hanlin Ren, and Rahul Santhanam. Hardness on any samplable distribution suffices: New characterizations of one-way functions by meta-complexity. *Electron. Colloquium Comput. Complex.*, 28:82, 2021.

[IW97] Russell Impagliazzo and Avi Wigderson. P= BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 220–229, 1997.

[Jeř09] Emil Jeřábek. Approximate counting by hashing in bounded arithmetic. *Journal of Symbolic Logic*, 74(3):829–860, 2009.

[JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 126–152, Cham, 2018. Springer International Publishing.

[JNV$^+$20] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP*=RE. *arXiv preprint arXiv:2001.04383*, 2020.

[KC00] Valentine Kabanets and Jin-Yi Cai. Circuit minimization problem. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 73–79, 2000.

[KP17] Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, 2017.

[Kre21] William Kretschmer. Quantum Pseudorandomness and Classical Complexity. In *16th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2021)*, volume 197 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:20, 2021.

[KSV02] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation.* American Mathematical Society, USA, 2002.

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.

[LMS11] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 760–776. SIAM, 2011.

[LP20] Yanyi Liu and R. Pass. On one-way functions and kolmogorov complexity. *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1243–1254, 2020.

[LP21] Yanyi Liu and Rafael Pass. A note on one-way functions and sparse languages. *Electron. Colloquium Comput. Complex.*, 28:92, 2021.

[Lup58] Oleg B. Lupanov. On the synthesis of switching circuits. *Doklady Akademii Nauk SSSR*, 119(1):23–26, 1958.

[Mas79] William J Masek. Some np-complete set covering problems. *Unpublished Manuscript*, 1979.

[MS13] J. Maldacena and L. Susskind. Cool horizons for entangled black holes. *Fortschritte der Physik*, 61(9):781–811, Aug 2013.

[MW17] Cody D Murray and R Ryan Williams. On the (non) np-hardness of computing circuit complexity. *Theory of Computing*, 13(1):1–22, 2017.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, USA, 10th edition, 2011.

[Oli19] Igor Carboni Oliveira. Advances in hardness magnification. https://www.dcs.warwick.ac.uk/~igorcarb/documents/papers/magnification-note.pdf, 2019.

[OPS19] Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. Hardness magnification near state-of-the-art lower bounds. In *34th Computational Complexity Conference (CCC 2019).* Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[OS16] Igor C Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds and pseudorandomness. *arXiv preprint arXiv:1611.01190*, 2016.

[OS18] Igor Carboni Oliveira and Rahul Santhanam. Hardness magnification for natural problems. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 65–76. IEEE, 2018.

[RR97] Alexander A Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 1(55):24–35, 1997.

[RS21] Hanlin Ren and Rahul Santhanam. A relativization perspective on meta-complexity. *Electron. Colloquium Comput. Complex.*, 28:89, 2021.

[Shi02] Yaoyun Shi. Both toffoli and controlled-not need little help to do universal quantum computation. *arXiv preprint quant-ph/0205115*, 2002.

[Sho94] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[Sus16] Leonard Susskind. Computational complexity and black hole horizons. *Fortschritte der Physik*, 64(1):24–43, 2016.

[Tra84] Boris A Trakhtenbrot. A survey of russian approaches to perebor (brute-force searches) algorithms. *Annals of the History of Computing*, 6(4):384–400, 1984.

[Val84] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Wil18] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the ICM*, volume 3, pages 3431–3472. World Scientific, 2018.

[WJB03] Pawel Wocjan, Dominik Janzing, and Thomas Beth. Two QCMA-complete problems. *Quantum Info. Comput.*, 3(6):635–643, November 2003.

[WW20] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious lwe sampling. *IACR Cryptol. ePrint Arch*, 2020:1042, 2020.

[Yao93] A Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 352–361. IEEE, 1993.

A Proof for the hardness of MQCSP

Theorem 3.9. $\mathsf{MQCSP} \in \mathsf{QCMA}$.

Proof. The certificate is still the classical description of a quantum circuit $\mathcal{C}$ that has size at most s and operates on at most $n+t$ qubits. The verifier first implements $\mathcal{C}$. Then, the verifier repeats evaluating $\mathcal{C}|x, 0^t\rangle$ and measuring the first qubit $\ell = \mathsf{poly}(2^n)$ times. We denote the measurement outcomes of the ℓ trials as binary random variables $X_1, \dots, X_\ell$ which are all independent. Finally, the verifier checks if for all $x \in \{0,1\}^n$, there are at least $\frac{\alpha+\beta}{2}$ of the outcomes are consistent with $f(x)$.

For the yes instance, we have the promise that $\|(\langle f(x)| \otimes I_{n+t-1})\mathcal{C}|x, 0^t\rangle\|^2 \geq \alpha$ for all $x \in \{0,1\}^n$. Let $X = \sum_{i=1}^n X_i$. By using the second statement of Chernoff inequality, we have that $\Pr[X \leq \frac{(\alpha+\beta)\ell}{2}] \leq \exp\left(-\frac{(\alpha+\beta)^2\ell}{8\alpha}\right)$. By setting $\ell = \mathsf{poly}(2^n)$, we obtain $\Pr[X \leq \frac{(\alpha+\beta)\ell}{2}] \leq e^{-\mathsf{poly}(2^n)}$. This implies that $\Pr[X \geq \frac{(\alpha+\beta)\ell}{2}$ for all $x \in \{0,1\}^n] \geq 1 - e^{-\mathsf{poly}(2^n)}$. For the no instance, we can do the similar analysis using Chernoff bound and show that there exists $x \in \{0,1\}^n$ such that $\Pr[X \geq \frac{(\alpha+\beta)\ell}{2}]$ is negligible. □

Theorem 3.13. $\mathsf{SZK} \subseteq \mathsf{BPP}^{\mathsf{MQCSP}}$

Proof of Theorem 3.13. Let (n, C_0, C_1) be a PIID instance, where $C_0, C_1 : \{0,1\}^m \to \{0,1\}^{m'}$ of size n^k. For $b = 0, 1$ and $x \in \{0,1\}^m$, we let $f_b(x) = C_b(x)$. Then, similar to the proof for Theorem 4.8, the idea is using f_b to construct a pseudorandom generator $\hat{G}$ and break $\hat{G}$ by applying the MQCSP oracle. Specifically, the algorithm is as follows:

Algorithm 7 A PPT algorithm $\mathcal{A}$ for PIID with MQCSP oracle

```
Input: $C_0, C_1$ of size $n^k$ and $m$-qubit input.
1: Pick $x$ uniformly randomly from $\{0,1\}^m$.
2: Compute $f_0(x)$.
3: Use $f_0(x)$ to generate a pseudorandom string $G_{f_0(x)}(r)$ as in Lemma 4.6.
4: Use $G_{f_0(x)}(r)$ to generate the truth table $\mathsf{tt}(g) = \hat{G}(r)$ as in Lemma 4.7.
5: Apply the inverting algorithm $\mathcal{A}_{inv}^{\mathsf{MQCSP}}$ with access to function $f_1$ in Theorem 4.8 to invert $f_1$
   for $x'$. Note that the function used in the inverting algorithm is $f_1$ instead of $f_0$.
6: return "Yes" if $C_0(x) = C_1(x')$; "No" if $C_0(x) \neq C_1(x')$.
```

In Algorithm 7, we do not explicitly describe the inverting algorithms $\mathcal{A}_{inv}$. However, based on Theorem 4.8, such algorithms must exist.

Then, when (C_0, C_1) is a no instance, i.e., $\Pr_{x \in \{0,1\}^m}[\exists y \in \mathsf{Im}(C_0) \text{ such that } C_1(x) = y] \leq \frac{1}{2^n}$, the probability that there exists x' such that $C_1(x') = C_0(x)$ over x is at most $1/2^n$. In this case, Algorithm 7 outputs "Yes" with probability at most $1/2^n$.

When (C_0, C_1) is a yes instance, C_0 and C_1 has statistical distance $1/2^n$ over $x \in \{0,1\}^m$. Then, the success probability of the algorithm $\mathcal{A}$ in Algorithm 7 is

$$\begin{aligned}\Pr[\mathcal{A}(C_0, C_1) = \text{“Yes”}] &= \Pr_x[f_1(\mathcal{A}_{inv}^{\mathsf{MQCSP}}(f_1, f_0(x))) = f_0(x)] \\ &= \sum_{y \in \{0,1\}^{m'}} \Pr_x[f_0(x) = y] \Pr_x[f_1(\mathcal{A}_{inv}^{\mathsf{MQCSP}}(f_1, y)) = y | y]\end{aligned}$$

Note that if we compute $f_1(x)$ (instead of $f_0(x)$) at step 2 in Algorithm 7, then the success probability of $\mathcal{A}$ is

$$\begin{aligned}\Pr_x[f_1(\mathcal{A}_{inv}^{\mathsf{MQCSP}}(f_1, f_1(x))) = f_1(x)] &= \sum_{y \in \{0,1\}^{m'}} \Pr_x[f_1(x) = y] \Pr_x[f_1(\mathcal{A}_{inv}^{\mathsf{MQCSP}}(f_1, y)) = y | f_1(x) = y] \\ &\geq 1/\mathsf{poly}(n).\end{aligned}$$

The last inequality follows from Theorem 4.8. The MQCSP oracle can break $\hat{G}$ due to the fact that the construction of $\hat{G}$ is a small classical circuit and thus also a small quantum circuit. Therefore, we can use the MQCSP oracle to distinguish it from a truely random string.

The difference between these two probabilities above is

$$\begin{aligned}&\Pr_x[f_1(\mathcal{A}_{inv}^{\mathsf{MQCSP}}(f_1, f_0(x))) = f_0(x)] - \Pr_x[f_1(\mathcal{A}_{inv}^{\mathsf{MQCSP}}(f_1, f_1(x))) = f_1(x)] \\ &= \sum_y \Pr_x[f_1(\mathcal{A}_{inv}^{\mathsf{MQCSP}}(f_1, y)) = y | f_1(x) = y](\Pr_x[f_0(x) = y] - \Pr_x[f_1(x) = y]) \\ &\leq \sum_y (\Pr_x[f_0(x) = y] - \Pr_x[f_1(x) = y]) \leq \frac{1}{2^n}.\end{aligned}$$

The last inequality follows from the definition of statistical distance. Therefore, Algorithm 7 succeeds with probability at least $1/\mathsf{poly}(n) - 2^{-n}$ for a "Yes" instance. Finally, we can amplify the success probability for the yes instance to 2/3 by repetition. Thus, $\mathsf{PIID} \in \mathsf{BPP}^{\mathsf{MQCSP}}$.

□

B Learning Theory

In this section, we provide the details of Section 4.2 on the connection between learning theory and MQCSP.

B.1 PAC learning

Let us recall the definition of PAC learning.

Definition 4.11 (PAC learning over the uniform distribution with membership queries)**.** *Let* C *be a circuit class and let* $\epsilon, \delta > 0$. *We say an algorithm* (ϵ, δ)*-PAC-learns* C *over the uniform distribution with membership queries if the following hold. For every* $n \in \mathbb{N}$ *and* n*-variate* $f \in \mathsf{C}$*, given membership query access to* f*, the algorithm outputs a circuits* C *such that with probability at least* $1-\delta$ *over its internal randomness, we have* $\Pr_{x\in\{0,1\}^n}[f(x) \neq C(x)] < \epsilon$. *The running time of the learning algorithm is measured as a function of* $n, 1/\epsilon, 1/\delta$ *and,* $\mathsf{size}(f)$.

The following theorem shows that efficient PAC-learning for $\mathsf{BQP/poly}$ is equivalent to efficient algorithms for MQCSP. Here, $\mathsf{BQP/poly}$ is defined as $\bigcup_{s\leq \mathsf{poly}(n)} \mathsf{BQC}(s)$

Theorem 4.12 (Equivalence of efficient PAC learning for $\mathsf{BQP/poly}$ and efficient randomized algorithm for MQCSP)**.**

- *If* $\mathsf{MQCSP} \in \mathsf{BPP}$*, then there is a randomized algorithm that* $(1/\mathsf{poly}(n), \delta)$*-PAC learns* $f \in \mathsf{BQP/poly}$ *under the uniform distribution with membership queries for every* $\delta > 0$. *Specifically, the algorithm runs in quasi-polynomial time.*
- *If there is a randomized algorithm that* $(1/\mathsf{poly}(n), \delta)$*-PAC learns* $f \in \mathsf{BQP/poly}$ *under the uniform distribution with membership queries for some* $\delta > 0$ *in* $2^{O(n)}$ *time, then we have* $\mathsf{MQCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), \mathsf{poly}(n), \tau] \in \mathsf{BQP}$ *and* $\mathsf{MQCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), O(n), \tau] \in \mathsf{BPP}$ *for every* $\tau > 0$.

Proof.

- The key ingredient to show $\mathsf{MQCSP} \in \mathsf{BPP}$ implies efficient PAC learning for $\mathsf{BQP/poly}$ is the "learning from a natural property" framework by [CIKK16]. First, note that $\mathsf{BQP/poly}$ is a circuit class that contains $\mathsf{P/poly}$ and hence can implement both the *Nisan Wigderson generator* and the *Direct Product + Goldreich-Levin amplification*. Second, $\mathsf{MQCSP} \in \mathsf{BPP}$ implies there is a BPP-natural property against $\mathsf{BQP/poly}$. Finally, by Theorem 5.1 of [CIKK16], there is a randomized algorithm that $(1/\mathsf{poly}(n), \delta)$-PAC learns $f \in \mathsf{BQP/poly}$ under the uniform distribution with membership queries for every $\delta > 0$ in quasipolynomial time.

- Let ALG be a randomized algorithm that $(1/\mathsf{poly}(n), \delta)$-PAC learns $f \in \mathsf{BQP/poly}$ under the uniform distribution with membership queries for some $\delta > 0$. We design the following randomized algorithm for $\mathsf{MQCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), t(n), \tau]$ where $t(n)$ is the number of ancilla bits that will be determined later. For every $\tau > 0$, let $\epsilon = \tau/2$.

Algorithm 8 A quantum algorithm for $\mathsf{MQCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), t(n), \tau]$

```
Input: The truth table $T$ of a $n$-variate Boolean function $f$.
1: for $i = 1, \dots, 10\lceil \log 1/\delta \rceil$ do
2:     Run $ALG$ and supply the membership query with the truth table $T$. Let $C_i$ be the output of $ALG$.
3:     Uniformly and independently sample $x_1, \dots, x_\ell \in \{0,1\}^n$ where $\ell = \lceil 100 \log(1/\delta)/\epsilon^2 \rceil$.
4:     if $|\{j \in [\ell] : C_i(x_j) \neq f(x_j)\}| < \frac{\epsilon}{10} \cdot \ell$ then
5:         Break and output "Yes".
6: Output "No".
```

Let us analyze the correctness of the above algorithm. First, if f is an Yes instance, i.e., there exists a polynomial size quantum circuit C that computes f, then due to the correctness of ALG, $\Pr_{C_i}[|\{x \in \{0,1\}^n : C_i(x) \neq f(x)\}| < 2^n/\mathsf{poly}(n)] > \delta$ for each i. Namely, with probability at least 9/10, there exists an $i \in [10\lceil \log 1/\delta \rceil]$ such that $|\{x \in \{0,1\}^n : C_i(x) \neq f(x)\}| < 2^n/\mathsf{poly}(n)$. For this specific i, by Chernoff bound, with probability at least 9/10 the algorithm will go to line 5 and output "Yes". That is, the above algorithm accepts an Yes instance with probability at least 2/3 as desired.

Next, if f is a No instance, i.e., for every polynomial size quantum circuit C, we have $|\{x \in \{0,1\}^n : C(x) \neq f(x)\}| \geq \tau \cdot 2^n > \epsilon \cdot 2^n$. For each $i \in [10\lceil \log 1/\delta \rceil]$, C_i is a polynomial size circuit and hence by Chernoff bound, the algorithm goes to line 5 with probability at most $2^{-\Omega(\epsilon^2 \ell)}$. Due to the choice of ℓ, we know that the algorithm will output "No" with probability at least 2/3. That is, the above algorithm rejects an No instance with probability at least 2/3 as desired.

Finally, the running time of the algorithm is $\mathsf{poly}(\text{Time}(ALG), 1/\delta, 1/\epsilon, n, m)$ where the dependency on $\mathsf{poly}(n, m)$ is for calculating $C_i(x_j)$ using the quantumness. Note that this running time is $\mathsf{poly}(2^n)$ and hence we conclude that $\mathsf{MQCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), t(n)] \in \mathsf{BQP}$.

When the number of ancilla bits is $O(n)$, note that we can calculate $C_i(x_j)$ in $\mathsf{poly}(2^n)$ time and hence $\mathsf{MQCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), t(n)] \in \mathsf{BPP}$

□

B.2 Quantum learning

As it could be the case that MQCSP might have non-trivial quantum algorithm, it is also of interest to study the connection to quantum learning.

Definition 4.13 (Quantum learning)**.** *Let C be a circuit class of boolean functions and let $\epsilon, \delta > 0$. We say a quantum algorithm (ϵ, δ)-learns C if the following hold. For every $n \in \mathbb{N}$ and n-variate $f \in \mathsf{C}$, given quantum oracle access to f, the algorithm outputs a polynomial-size quantum circuit U such that with probability at least $1 - \delta$, we have $\mathbb{E}_{x \in \{0,1\}^n}[|(\langle f(x)| \otimes I)U|x, 0^m\rangle|^2] > 1 - \epsilon$. The running time of the learning algorithm is measured as a function of $n, 1/\epsilon, 1/\delta$ and, $\mathsf{size}(f)$.*

It turns out that efficient quantum learning for a circuit class C is equivalent to efficient quantum algorithm for its corresponding MCSP, i.e., C-MCSP.

Theorem 4.14 (Equivalence of efficient quantum learning and efficient quantum algorithm for C-MCSP)**.** *Let C be a circuit class.*

- *If* $\mathsf{C}\text{-}\mathsf{MCSP} \in \mathsf{BQP}$*, then there exists a quantum algorithm that* $(1/\mathsf{poly}(n), \delta)$*-learns* C *for every* $\delta > 0$*. Specifically, the algorithm runs in polynomial time.*
- *If there exists a quantum algorithm that* (ϵ, δ)*-learns* C *in time* $2^{O(n)}$ *for some constants* $\epsilon, \delta \in (0, 1/2)$*, then we have* $\mathsf{C}\text{-}\mathsf{MCSP}[\mathsf{poly}(n), \omega(\mathsf{poly}(n)), \tau] \in \mathsf{BQP}$ *for every* $\tau > 0$*.*

Proof.

- The key idea is to quantize the "learning from a natural property" framework [CIKK16]. Let us start with three important lemmas from [AGG+20].

  Lemma B.1 (Corollary of Lemma 4.3 and Lemma 4.4 in [AGG+20])**.** *Let* $L, s_D : \mathbb{N} \to \mathbb{N}$ *be constructive functions and* $\gamma \in (0, 1)$ *with* $1 \leq L(n) \leq 2^n$ *for every* $n \in \mathbb{N}$*. There exists an algorithm* A_{NW} *on input* 1^n *and* 1^L *outputs* $\mathsf{code}(C_{NW})$ *for a quantum circuit* C_{NW} *in time* $S(n) = \mathsf{poly}(n, L(n), s_D(n))$ *with the following properties. In the following, we abbreviate* $L = L(n)$ *and* $s_D = s_D(n)$*.*

  There exists a constant $c > 0$ *and an oracle function* $NW^{\mathcal{O}} : \{0,1\}^m \to \{h : \{0,1\}^{\log L} \to \{0,1\}\}$ *where* $m = cn^2$ *and* $size(NW^{\mathcal{O}}(z)) = \mathsf{poly}(n, size(\mathcal{O}))$ *for all* $z \in \{0,1\}^m$*. Let* $g : \{0,1\}^n \to \{0,1\}$*. Suppose there is a quantum circuit* D *of size at most* s_D *with*

  $$\left| \Pr_{z \in \{0,1\}^m, D}[D(NW^g(z)) = 1] - \Pr_{y \in \{0,1\}^L}[D(y) = 1] \right| \geq \gamma\,.$$

  Then C_{NW} *on input* $\mathsf{code}(D)$ *and with oracle access to* g*, outputs* $\mathsf{code}(C)$ *for a quantum circuit* C *of size* $O(L^2 \cdot s_D)$*. With probability* $\Omega(\gamma/L^2)$ *over the output measurement of* C_{NW}*, we have*

  $$\Pr_{x \in \{0,1\}^n, C}[C(x) = g(x)] \geq \frac{1}{2} + \frac{\gamma}{2L}\,.$$

  Lemma B.2 (Lemma 4.5 in [AGG+20])**.** *Let* $k, s : \mathbb{N} \to \mathbb{N}$ *be constructive functions and* $\gamma > 0$*. There exists an algorithm* A_{GL} *such that on input* 1^n *and* $1^{k(n)}$ *outputs a circuit* C_{GL} *of size* $\mathsf{poly}(n, k(n), s(n))$ *in time* $\mathsf{poly}(n, k(n), s(n))$ *with the following properties. In the following, we abbreviate* $k = k(n)$ *and* $s = s(n)$*.*

  Let $f : \{0,1\}^{kn} \to \{0,1\}^k$*. Suppose there is a quantum circuit* C *of size at most* s *satisfying*

  $$\mathop{\mathbb{E}}_{x \in \{0,1\}^{kn}} \mathop{\mathbb{E}}_{r \in \{0,1\}^k} [|(\langle f(x) \cdot r| \otimes I) C |x, r, 0^m\rangle|^2] \geq \frac{1}{2} + \gamma\,.$$

  Then C_{GL} *on input* $\mathsf{code}(C)$ *outputs* $\mathsf{code}(G^{\mathcal{O}})$ *for a quantum oracle circuit* $G^{\mathcal{O}}$ *of size* $O(kn)$ *such that*

  $$\mathop{\mathbb{E}}_{x, G^C}[|(\langle f(x)| \otimes I) G^C |x, 0^{k+m+1}\rangle|^2] \geq \frac{\gamma^3}{2}\,.$$

  Lemma B.3 (Theorem in 4.28 [AGG+20])**.** *Let* $k, s : \mathbb{N} \to \mathbb{N}$ *be constructive functions and* $\epsilon, \delta \in (0, 1)$*. There exists a constant* $c \geq 1$ *and an algorithm* A_{IJKW} *such that on input* 1^n *and* $1^{k(n)}$ *outputs a circuit* C_{IJKW} *of size* $\mathsf{poly}(n, k(n), s(n), \log 1/\delta, 1/\epsilon)$ *in time* $\mathsf{poly}(n, k(n), s(n), \log 1/\delta, 1/\epsilon)$ *with the following properties. In the following, we abbreviate* $k = k(n)$ *and* $s = s(n)$*.*

  Let $g : \{0,1\}^n \to \{0,1\}$*. Suppose* k *is an even integer with*

  $$k \geq c \cdot \frac{1}{\delta} \left[\log \frac{1}{\delta} + \log \frac{1}{\epsilon} \right]\,,$$

and suppose G is a quantum circuit of size at most s defined over $S_{n,k} := \{S \subset \{0,1\}^n : |S| = k\}$ with k output bits with

$$\mathop{\mathbb{E}}_{B\sim S_{n,k},G}[G(B) = g^k(B)] \geq \epsilon\,.$$

Then C_{IJKW} on input $\mathsf{code}(G)$ outputs $\mathsf{code}(C)$ for a quantum circuit C of size $\mathsf{poly}(n,k,s,\log(1/\delta),1/\epsilon)$ such that

$$\mathop{\mathbb{E}}_{x\sim\{0,1\}^n,C}[C(x) = g(x)] \geq 1-\delta\,.$$

Now, we are ready to describe our quantum learning algorithm for C.

Algorithm 9 A quantum learning algorithm for C

```
Input: $1^n$, quantum oracle access to $n$-variate $f \in \mathsf{C}$, and parameters $\delta \in (0,1)$.
Let $L = \mathsf{poly}(n)$, $\epsilon = 1/\mathsf{poly}(n)$, and $k = \lceil c \cdot \frac{1}{\delta}(\log\frac{1}{\delta} + \log\frac{1}{\epsilon})\rceil$.
$C_{NW} \leftarrow A_{NW}(1^{kn+k})$; $C_{GL} \leftarrow A_{GL}(1^n, 1^k)$; $C_{IJKW} \leftarrow A_{IJKW}(1^n, 1^k)$.
Let $\mathsf{code}(D)$ be the description of a quantum circuit solving $\mathsf{C}$-$\mathsf{MCSP}$ with truth table size $L$.
Use the oracle access to $f$ to build an oracle access to $NW^g$ where $g : \{0,1\}^{kn} \times \{0,1\}^k \to \{0,1\}$
  with $g(x_1,\dots,x_k,r_1,\dots,r_k) = \oplus_{i=1}^k (r_i \cdot f(x_i))$ for every $x_1,\dots,x_k \in \{0,1\}^n$ and $r_1,\dots,r_k \in \{0,1\}$.
$\mathsf{code}(\tilde{C}) \leftarrow C_{NW}^g(\mathsf{code}(D))$
$\mathsf{code}(G^{\mathcal{O}}) \leftarrow C_{GL}(\mathsf{code}(\tilde{C}))$.
$C \leftarrow C_{IJKW}(\mathsf{code}(G^{\tilde{C}}))$.
Output $C$.
```

Let us analyze the correctness and running time of Algorithm 9 simultaneously. Let $f : \{0,1\}^n \to \{0,1\} \in \mathsf{C}$ be the function we want to learn. Let $g : \{0,1\}^{kn} \times \{0,1\}^k \to \{0,1\}$ be $g(x_1,\dots,x_k,r_1,\dots,r_k) = \oplus_{i=1}^k (r_i \cdot f(x_i))$ for every $x_1,\dots,x_k \in \{0,1\}^n$ and $r_1,\dots,r_k \in \{0,1\}$. Observe that if $\text{size}(f) = \mathsf{poly}(n)$, then $\text{size}(NW^g) = \mathsf{poly}(n) = \mathsf{poly}(\log L)$.

Next, if C-$\mathsf{MCSP} \in \mathsf{BQP}$, then there exists a quantum algorithm D running in time $\mathsf{poly}(L)$ with

$$\left|\Pr_{z\in\{0,1\}^m,D}[D(NW^g(z)) = 1] - \Pr_{y\in\{0,1\}^L}[D(y) = 1]\right| \geq \frac{1}{3}\,.$$

By Lemma B.1, $C_{NW}^g(\mathsf{code}(D))$ outputs the description of a quantum circuit C of size $O(L^2 \cdot \text{size}(D)) = \mathsf{poly}(n)$ in time $\mathsf{poly}(L, \mathsf{size}(D))$ such that with probability $\Omega(1/L^2)$,

$$\Pr_{\substack{x_1,\dots,x_r\in\{0,1\}^n\\ r_1,\dots,r_k\in\{0,1\},C}}[C(x_1,\dots,x_k,r_1,\dots,r_k) = g(x_1,\dots,x_k,r_1,\dots,r_k)] \geq \frac{1}{2} + \frac{1}{6L}\,.$$

Next, by Lemma B.2, $C_{GL}(\mathsf{code}(C))$ outputs the description of an oracle quantum circuit $G^{\mathcal{O}}$ of size $O(kn \cdot \text{size}(C)) = \mathsf{poly}(n)$ in time $\mathsf{poly}(n,k)$ such that

$$\mathop{\mathbb{E}}_{x_1,\dots,x_k,G^C}\left[|(\langle f^k(x_1,\dots,x_k)| \otimes I)G^C|x, 0^{k+m+1}\rangle|^2\right] \geq \Omega\left(\frac{1}{L^3}\right) = \frac{1}{\mathsf{poly}(n)}\,.$$

Finally, by Lemma B.3, $C_{IKJW}(\mathsf{code}(G))$ outputs the description of a quantum circuit C of size $\mathsf{poly}(n,k,\text{size}(G),\log(1/\delta),1/\epsilon) = \mathsf{poly}(n,1/\delta,1/\epsilon) = \mathsf{poly}(n)$ in time $\mathsf{poly}(n)$ such that

$$\mathop{\mathbb{E}}_{x\sim\{0,1\}^n,C}[C(x) = g(x)] \geq 1-\delta\,.$$

We conclude that there is a polynomial time $(1/3,\delta)$-quantum learning algorithm for C.

- Let ALG be a (ϵ,δ)-quantum learning algorithm for C for some $\epsilon,\delta\in(0,1/2)$. We design the following quantum algorithm for $\mathsf{C}\text{-}\mathsf{MCSP}[\mathsf{poly}(n),\omega(\mathsf{poly}(n)),\tau]$. For every $\tau>0$, let $\epsilon=\tau/4$ and $\epsilon'=\tau/2$.

Algorithm 10 A quantum algorithm for $\mathsf{C}\text{-}\mathsf{MCSP}[\mathsf{poly}(n),\omega(\mathsf{poly}(n)),\tau]$

```
Input: The truth table $T$ of a $n$-variate Boolean function $f$.
1: for $i = 1,\dots,10\lceil\log 1/\delta\rceil$ do
2:     Run $ALG$ and supply quantum oracle access to $f$ using the truth table $T$. Let $C_i$ be the output of $ALG$.
3:     Uniformly and independently sample $x_1,\dots,x_\ell\in\{0,1\}^n$ where $\ell=\lceil 100\log(1/\delta)/\epsilon^2\rceil$.
4:     if $\sum_{j\in[\ell]}|(\langle f(x_j)|\otimes I)U|x,0^m\rangle|^2\ge(1-\frac{\epsilon+\epsilon'}{2})\cdot\ell$ then
5:         Break and output "Yes".
6: Output "No".
```

Let us analyze the correctness of the above algorithm. First, if f is an Yes instance, i.e., there exists a polynomial size quantum circuit C that computes f, then due to the correctness of ALG, $\Pr_{C_i}[\mathbb{E}_{x\in\{0,1\}^n}[|(\langle f(x)|\otimes I)U|x,0^m\rangle|^2]>1-\epsilon]>\delta$ for each i. Namely, with probability at least $9/10$, there exists an $i\in[10\lceil\log 1/\delta\rceil]$ such that $\mathbb{E}_{x\in\{0,1\}^n}[|\{x\in\{0,1\}^n : |(\langle f(x)|\otimes I)U|x,0^m\rangle|^2\}|]\ge 1-\epsilon$. For this specific i, by Chernoff bound, with probability at least $9/10$ the algorithm will go to line 5 and output "Yes". That is, the above algorithm accepts an Yes instance with probability at least $2/3$ as desired.

Next, if f is an No instance, i.e., for every polynomial size quantum circuit C, at least τ fraction of $x\in\{0,1\}^n$ has $|(\langle f(x)|\otimes I)U|x,0^m\rangle|^2\le 1/2$. Hence, by the choice of ϵ', we have $\mathbb{E}_{x\in\{0,1\}^n}[|(\langle f(x)|\otimes I)U|x,0^m\rangle|^2]<(1-\epsilon')$. For each $i\in[10\lceil\log 1/\delta\rceil]$, C_i is a polynomial size circuit and hence by Chernoff bound, the algorithm goes to line 5 with probability at most $2^{-\Omega(\epsilon^2 m)}$. Due to the choice of m, we know that the algorithm will output "No" with probability at least $2/3$. That is, the above algorithm rejects an No instance with probability at least $2/3$ as desired.

Finally, the running time of the algorithm is $\mathsf{poly}(\mathrm{Time}(ALG),1/\delta,1/\epsilon,n,m)$ where the dependency on $\mathsf{poly}(n,m)$ is for calculating $C_i(x_j)$ using the quantumness. Note that this running time is polynomial in the size of the truth table and hence we conclude that $\mathsf{C}\text{-}\mathsf{MQCSP}[\mathsf{poly}(n),\omega(\mathsf{poly}(n)),\tau]\in\mathsf{BQP}$.

□

C Proofs in Section 4.3

In this section, we provide some missing proofs in Section 4.3.

C.1 Proof for Theorem 4.19

The goal of this section is to prove Theorem 4.19.

Theorem 4.19. *If* $\mathsf{MQCSP}\in\mathsf{BQP}$, *then* $\mathsf{BPE}^{\mathsf{QCMA}}$ *contains a function with maximum quantum circuit complexity. Furthermore,* $\mathsf{BQP}^{\mathsf{QCMA}}\not\subset\mathsf{BQC}[n^k]$ *for any constant* $k>0$.

Proof. We follow the proof of a classical result in [KC00, Theorem 10].

We first determine the maximum quantum circuit complexity for all Boolean functions using an MQCSP oracle. For each $s = 2^{O(n)}, 2^{O(n)} - 1, \cdots$, decide if there exists a function f_s such that $\mathrm{qCC}(f_s) \geq s$. The first s we meet such that f_s exists is the maximum quantum circuit complexity. It can be achieved by a QCMA algorithm with input 1^s, by the assumption $\mathsf{MQCSP} \in \mathsf{BQP}$. Hence, in classical $2^{O(n)}$ time with query access to a QCMA oracle, we can find the maximum quantum circuit complexity $s_\star$ with high probability.

Then, we can construct the truth table by guessing bit-by-bit. We start from the empty truth table $T = \emptyset$. We first try to choose the first bit $T_1 = 0$ and decide if T can be extended to a truth table with quantum circuit complexity $s_\star$, which can be done by a QCMA oracle query. If the answer is "No", we set $T_1 = 1$. Then, we iterate over all bits of T. It is easy to see that in $O(2^n)$ time we can construct T with high probability.

Therefore, we get a $\mathsf{BPE}^{\mathsf{QCMA}}$ algorithm for the maximum quantum circuit complexity problem, which immediately gives a $\mathsf{BPE}^{\mathsf{QCMA}}$ algorithm for computing such hard functions. By Claim F.1, this function has quantum circuit complexity at least $\Omega(2^n/n)$. Hence, by a padding argument for quantum circuits, we obtain a polynomial lower bound for $\mathsf{BQP}^{\mathsf{QCMA}}$. □

C.2 Proof of Quantum Antichecker Lemma

The goal of this section is to prove Lemma 4.23.

Lemma 4.23 (Antichecker lemma for quantum circuits)**.** *Assume* $\mathsf{QCMA} \subseteq \mathsf{BQC}[\mathsf{poly}]$. *Then for any* $\lambda \in (0,1)$ *there are circuits* $\{C_{2^n}\}_{n=1}^{\infty}$ *of size* $2^{n+O(n^\lambda)}$ *which given the truth table* $\mathsf{tt}(f) \in \{0,1\}^{2^n}$ *, outputs* $2^{O(n^\lambda)}$ *n-bit strings* $y_1, \ldots, y_{2^{O(n^\lambda)}}$ *together with bits* $f(y_1), \ldots, f(y_{2^{O(n^\lambda)}})$ *forming a set of anticheckers for f, i.e. if f is hard for quantum circuits of size* 2^{n^λ} *then every quantum circuit of size* $2^{n^\lambda}/2n$ *fails to compute f on one of the inputs* $y_1, \ldots, y_{2^{O(n^\lambda)}}$.

Proof. The proof follows [CHO+20].

Let $\lambda \in (0,1)$ and f be a Boolean function with n input bits that is hard for 2^{n^λ}-size quantum circuits.

For $i \geq 0$ and $s \in [0,1]$, define the predicate:

$$\begin{aligned}&P_f(y_1, \ldots, y_i)[s] = 1 \iff \\ &\leq s \text{ fraction of all quantum circuits of size } 2^{n^\lambda}/2n \text{ compute } f \text{ correctly on } y_1, \ldots, y_i.\end{aligned}$$

We also define the function:

$$R_f(y_1, \ldots, y_i) := \# \left\{ \text{quantum circuits of size } 2^{n^\lambda}/2n \text{ compute } f \text{ correctly on } y_1, \ldots, y_i \right\}.$$

Then, we construct $y_1, \ldots, y_{2^{O(n^\lambda)}}$ iteratively. It is easy to see that $P_f(\cdot)[1] = 1$. Suppose we already have $y_1, \ldots, y_{i-1}$ such that $P_f(y_1, \ldots, y_{i-1})[(1 - 1/4n)^{i-1}] = 1$ holds. We want to find y_i such that $P_f(y_1, \ldots, y_i)[(1 - 1/4n)^i] = 1$. We will construct a formula F of size $2^{O(n^\lambda)}$ such that if $R_f(y_1, \ldots, y_{i-1}) \geq 2n^2$, then

$$\begin{aligned}&P_f(y_1, \ldots, y_{i-1}) \left[(1 - 1/4n)^{i-1}\right] = 1 \\ \Rightarrow\ &\exists y_i\ F(y_1, \ldots, y_i, f(y_1), \ldots, f(y_i)) = 1 \\ \Rightarrow\ &P_f(y_1, \ldots, y_i) \left[(1 - 1/4n)^i\right] = 1.\end{aligned}$$

We first show how to find y_i given this formula F. The idea is to use Valiant-Vazirani Isolation Lemma. Let r be uniformly chosen from $\{2, n+1\}$ and let $h : \{0,1\}^n \to \{0,1\}^r$ be uniformly chosen from a pairwise independent hash family $\mathcal{H}_{n,r}$. Consider the following predicate

$$\begin{aligned}&F^{r,h}(y_1, \ldots, y_{i-1}, z, f(y_1), \ldots, f(y_{i-1}), f(z)) := \\ &F(y_1, \ldots, y_{i-1}, z, f(y_1), \ldots, f(y_{i-1}), f(z)) \wedge h(z) = 0^r.\end{aligned}$$

The quantum circuit size of $F^{r,h}$ is $2^{O(n^\lambda)}$.

By the Isolation Lemma, for fixed $y_1, \ldots, y_{i-1}$, with probability at least $1/8n$, there is a unique z such that

$$F^{r,h}(y_1, \ldots, y_{i-1}, z, f(y_1), \ldots, f(y_{i-1}), f(z)) = 1.$$

If we sample $2^{O(n^\lambda)}$ many tuples of (r, h), then the probability that none of those (r, h) will lead to unique solution of $F^{r,h}$ is less than $2^{-2^{O(n^\lambda)}/8n} \leq 2^{-2^{O(n^\lambda)}}$ by choosing proper constant. On the other hand, the total number of all possible $y_1, \ldots, y_{i-1}, f(y_1), \ldots, f(y_{i-1})$ is at most $2^{2^{O(n^\lambda)}}$. It means that there exists a set $\mathcal{R}$ of $2^{O(n^\lambda)}$ tuples of (r, h) such that for any $y_1, \ldots, y_{i-1}, f(y_1), \ldots, f(y_{i-1})$, there exists an $(r, h) \in \mathcal{R}$ that makes $F^{r,h}$ have unique solution. Note that $\mathcal{R}$ can be hard-wired into the circuit C_n. Hence, the j-th bit of the antichecker y_i can be computed by the following formula of size $2^{n+O(n^\lambda)}$:

$$\bigvee_{z \in \{0,1\}^n} z_j \wedge F^{r,h}(y_1, \ldots, y_{i-1}, z, f(y_1), \ldots, f(y_{i-1}), f(z)). \tag{17}$$

Then, we need to select an (r, h) from $\mathcal{R}$ that gives the unique y_i. This task is in QCMA, and by assumption, $\mathsf{QCMA} \subseteq \mathsf{BQC}[\mathsf{poly}]$. So, we just need to apply a $2^{O(n^\lambda)}$-size quantum circuit. Once we have y_i, $f(y_i)$ can be obtained from $\mathsf{tt}(f)$ via an Address function, which can be implemented by a circuit of size $2^{n+O(\log n)}$.

By repeating this process, we can get $y_1, \ldots, y_{2^{O(n^\lambda)}}$ and $f(y_1), \ldots, f(y_{2^{O(n^\lambda)}})$ by a $2^{n+O(n^\lambda)}$ circuit. Then, we need to check $R_f(y_1, \ldots, y_{2^{O(n^\lambda)}}) \geq 2n^2$. Deciding whether $R_f(y_1, \ldots, y_i) \geq 2n^2$ is in $\mathsf{QCMA} \subseteq \mathsf{BQC}[\mathsf{poly}]$ with input $(y_1, \ldots, y_i, f(y_1), \ldots, f(y_i), 1^{2^{O(n^\lambda)}})$ since the witness is $2n^2$ quantum circuits each of size $2^{n^\lambda}/2n$, which can be represented by a $2^{O(n^\lambda)}$ binary string. The witness can be checked by simulating the quantum circuits. Therefore, there exists a $2^{O(n^\lambda)}$ quantum circuit for it. When $R_f(y_1, \ldots, y_i) \leq 2n^2$, the $2n^2$ circuits of size $2^{n^\lambda}/2n$ can be generated by an $\mathsf{QCMA}^{\mathsf{coQCMA}}$ algorithm. And since $\mathsf{QCMA} \subseteq \mathsf{BQC}[\mathsf{poly}]$, by uncomputing the garbage, we can show that $\mathsf{QCMA}^{\mathsf{coQCMA}} \subseteq \mathsf{BQC}[\mathsf{poly}]$ and this step can be done by a $2^{O(n^\lambda)}$ quantum circuit. For each circuit, by exhaustively searching, we can find an n-bit string that witness the error. The circuit size of this step is $2^{n+O(n^\lambda)}$.

In order to construct F, we use a result in [OPS19] (Lemma 23) showing that if $P_f(y_1, \ldots, y_{i-1})[(1-1/4n)^{i-1}] = 1$ and $R_f(y_1, \ldots, y_{i-1}) \geq 2n^2$, then

$$\exists y_i \; P_f(y_1, \ldots, y_i) \left[(1 - 1/4n)^{i-1}(1 - 1/2n)\right] = 1. \tag{18}$$

The proof is by a standard counting argument, and by examining the proof, we find that it also holds for quantum circuits.

By Eq. (18), we know that there exists a y_i such that $\leq (1 - 1/4n)^{i-1}(1 - 1/2n) < (1 - 1/4n)^i$ fraction of circuits of size $2^{n^\lambda}/2n$ that can compute f on $y_1, \ldots, y_i$. The remaining task is to find a witness (which is F) that can certify $P_f(y_1, \ldots, y_i) \left[(1 - 1/4n)^i\right] = 1$. We can use an approximate counting with linear hash functions to construct F. More specifically, by [Jeř09], the witness is a set of matrices $A_1, \ldots, A_{2^{O(n^\lambda)}}$ defining an injective map from the Cartesian power of the set of all circuits of size $2^{n^\lambda}/2n$ that compute f on $y_1, \ldots, y_i$ to the same Cartesian power of $(1-1/4n)^i$ fraction of the set of all circuits of size $2^{n^\lambda}/2n$. The existence of these matrices can be decided by an $\mathsf{QCMA}^{\mathsf{coQCMA}}$ algorithm, which can also be decided by a $2^{O(n^\lambda)}$ quantum circuit, by our assumption. □

C.3 Quantum Impagliazzo-Wigderson generator

The goal of this section is to prove Lemma 4.21.

Lemma 4.21 (Quantum Impagliazzo-Wigderson generator)**.** *For every $\epsilon > 0$, there exist $c, d \in \mathbb{N}$ such that the truth table of a Boolean function $f : \{0,1\}^{cn} \to \{0,1\}$ of quantum circuit complexity $2^{\epsilon cn}$ can be transformed in time $O(2^n)$ into a pseudorandom generator $G : \{0,1\}^{dn} \to \{0,1\}^{2^n}$ running in time $O(2^n)$ that can fool quantum circuits of size $2^{O(n)}$, i.e., for any $p > 0$, any quantum circuit $\mathcal{C}$ of size at most 2^{pn},*

$$\left| \Pr_{x\sim\{0,1\}^{dn},\mathcal{C}}[\mathcal{C}(G(x)) = 1] - \Pr_{y\sim\{0,1\}^{2^n},\mathcal{C}}[\mathcal{C}(y) = 1] \right| \leq 2^{-n}.$$

Before giving the proof, we first recall some necessary definitions and lemmas in the previous work.

Lemma C.1 (A variant of Lemma 4.29 in [AGG+20])**.** *Let $L : \{0,1\}^* \to \{0,1\}$ be a language that is randomly reducible to the language L'. For every n, suppose we have the description of a quantum circuit U such that*

$$\mathop{\mathbb{E}}_{x\in\{0,1\}^n} \left[\|\Pi_{L'(x)} U |x, 0^q\rangle\|^2 \right] \geq 1 - \frac{1}{n^k},$$

for some $k \geq 2b + a$.

There is a $O(|U| \cdot \mathsf{poly}(n))$-size quantum circuit $\widetilde{U}$ that satisfies

$$\|\widetilde{\Pi}_x \widetilde{U} |0, x, 0^{q^*}\rangle\|^2 \geq 1 - 2^{-2n+1} \qquad \text{for every } x \in \{0,1\}^n,$$

where $\widetilde{\Pi}_x = |L(x)\rangle\langle L(x)| \otimes |x\rangle\langle x| \otimes |0^{q^*}\rangle\langle 0^{q^*}|$ and $q^* = \mathsf{poly}(n)$.

Definition C.2 (Expander walks)**.** *Let $\mathcal{G}$ be a graph with vertex set $\{0,1\}^n$ and degree 16. Let the expander walk generator $\mathsf{EW} : \{0,1\}^n \times [16]^k \to \{0,1\}^{nk}$ such that $\mathsf{EW}(v, d) := (v_1, \ldots, v_k)$, where $v_1 = v$ and v_{i+1} is the d_i-th neighbor of v_i in $\mathcal{G}$.*

Definition C.3 (Nearly disjoint subsets)**.** *Let $\Sigma = \{S_1, \ldots, S_k\}$ be a family of subsets of $[m]$ of size n. We say Σ is γ-disjoint if $|S_i \cap S_j| \leq \gamma n$ for any $i \neq j$.*

For $r \in \{0,1\}^m$, $S \subseteq [m]$, let $r|_S$ be the restriction of r to S. Then, for a γ-disjoint Σ, $\mathsf{ND}^\Sigma : \{0,1\}^m \to \{0,1\}^{nk}$ is defined by $\mathsf{ND}^\Sigma(r) := r|_{S_1}, \ldots, r|_{S_k}$.

Definition C.4 (M-restrictable)**.** *We say $G_n : \{0,1\}^m \to \{0,1\}^{nk}$ is M-restrictible if there exists a polynomial-time computable function $h : [n] \times \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^m$ such that*

- *For any $i \in [n], x \sim \{0,1\}^n, \alpha \sim \{0,1\}^m$, $h(i, x, \alpha)$ is uniformly distributed.*
- *For any i, x, α, let $G(h(i, x, \alpha)) := x_1, \ldots, x_k$. Then, we have $x_i = x$.*

- *For any $i, j \neq i$, for any α, there exists a set $S \subseteq \{0,1\}^n$, $|S| \leq M$ such that for any x, $x_j \in S$.*

Definition C.5 ((k', q, δ)-hitting)**.** *We say $G_n : \{0,1\}^m \to \{0,1\}^{nk}$ is (k', q, δ)-hitting if for any sets $H_1, \ldots, H_k \subseteq \{0,1\}^n$, $|H_i| \geq \delta 2^n$, we have*

$$\Pr[|\{i : x_i \in H_i\}| < k'] < q.$$

Proof of Lemma 4.21. We follow the proof in [IW97]. We first assume that there exists a function $f_0 : \{0,1\}^n \to \{0,1\}$ such that the quantum circuit complexity of f_0 is $2^{\Omega(n)}$. We may assume that $f_0 \in \mathsf{BQE}$. Then, encoding the truth table of f_0 by a locally list-decodable code, we obtain a function $f_1 : \{0,1\}^{O(n)} \to \{0,1\}$ such that $f_1 \in \mathsf{BQE}$, and for any quantum circuit $\mathcal{B}_1$ of size less than $2^{\Omega(n)}$,

$$\mathop{\mathbb{E}}_{x \sim \{0,1\}^{O(n)}, \mathcal{B}_1} [B_1(x) = f_1(x)] := \mathop{\mathbb{E}}_{x \sim \{0,1\}^{O(n)}, \mathcal{B}_1} [\|\Pi_{f_1(x)} \mathcal{B}_1 |x, 0\rangle \|] \leq 1 - n^{-O(1)}.$$

The properties of f_1 can be proved by Lemma C.1.

Then, by Lemma B.3 with $k = \mathsf{poly}(n), \epsilon = O(1), \delta = \frac{1}{\mathsf{poly}(n)}$, we have a function $f_2 = f_1^{\otimes k} : \{0,1\}^{kn} \to \{0,1\}^k$ such that for any quantum circuit $\mathcal{B}_2$ of size less than $2^{\Omega(n)}$,

$$\mathop{\mathbb{E}}_{x \in \{0,1\}^{nk}, \mathcal{B}_2} [\mathcal{B}_2(x) = f_2(x)] \leq O(1).$$

We can apply the quantum Goldreich-Levin Theorem (Lemma B.2) to f_2 and get a function $f_3 : \{0,1\}^n \to \{0,1\}$ (scaling the input size) such that for any quantum circuit $\mathcal{B}_3$ of size less than $2^{\Omega(n)}$,

$$\mathop{\mathbb{E}}_{x \in \{0,1\}^n, \mathcal{B}_3} [\mathcal{B}_3(x) = f_2(x)] \leq \frac{2}{3}.$$

The remaining thing is to "quantize" the direct-product generator defined by [IW97] using f_3. More specifically, we say G is a $(s, s', \epsilon, \delta)$ *quantum direct-product generator* if $G : \{0,1\}^m \to \{0,1\}^{nk}$ such that for every Boolean function g that is δ-hard for any quantum circuit of size s, we have $g^{\otimes} \circ G$ is ϵ-hard for any quantum circuit of size s'. The main result of [IW97] is the construction of $(2^{\Omega(n)}, 2^{\Omega(n)}, 2^{-\Omega(n)}, \frac{1}{3})$ direct-product generator. We first briefly describe the construction and then show that it also works for quantum circuits.

The direct-product generator in [IW97] is constructed from the expander random walks (Definition C.2) and nearly disjoint subsets (Definition C.3). They defined the direct-product generator $\mathsf{XG}(r, r', v, d) := \mathsf{EW}(v, d) \oplus \mathsf{ND}^{\Sigma}(r')$, where $\Sigma \subseteq [m]$ is selected by r such that $|r| = O(n), |r'| = m = O(n), |v| = n, |d| = O(n)$. They proved that XG is $2^{\Omega(n)}$-restrictible and $(O(n), 2^{-\Omega(n)}, 1/3)$-hitting. It's easy to see that the restrictible and hitting properties are pure combinatorial and circuit independent, which means that they also hold for quantum circuits. Then, they proved that these combinatorial properties imply XG is also a direct product generator. This step, however, need to be reproved for quantum circuits.

Claim C.6. *Let $s > 0$, $G(r) : \{0,1\}^m \to \{0,1\}^{nk}$ be a $(\rho k, q, \delta)$-hitting, M-restrictible pseudorandom generator, where $q > 2^{-\rho k/3}, s > 2Mnk$. Then, G is a $(s, \Omega(sq^2 n^{-O(1)}), O(q), \delta)$-quantum direct product generator.*

Proof. Let $\epsilon = (4\delta/\rho + 1)q$. Suppose there is a quantum circuit $\mathcal{C}$ such that

$$\mathop{\mathbb{E}}_{x\sim\{0,1\}^m,\mathcal{C}}\left[\mathcal{C}(x) = g^{\otimes k} \circ G(x)\right] \geq \epsilon.$$

Then, we construct a quantum circuit $\mathcal{F}$ of size $O(|C|+kMn)$ such that for any $H \subseteq \{0,1\}^n, |H| \geq \delta 2^n$,

$$\mathop{\mathbb{E}}_{y\sim H,\mathcal{F}}[\mathcal{F}(y) = g(y)] \geq \frac{1}{2} + \frac{q}{2}.$$

We use the same construction as [IW97]. Let $i \sim [k], \alpha_0 \sim \{0,1\}^m$. Let $x_1, \ldots, x_k$ be the output of $G(h(i,x,\alpha_0))$. For each $j \neq i$, we non-uniformly construct a table of $g(x_j)$ for any x_j that is a possible output of $G(h(i,x,\alpha_0))$ for different x. Since G is M-restrictible, each table has at most M values. Then, on input $y \in \{0,1\}^n$, the circuit $\mathcal{F}$ simulates $\mathcal{C}$ on $h(i,y,\alpha_0)$ and let $c_1, \ldots, c_k$ be the output. Then, for $y_1, \ldots, y_k := G(h(i,y,\alpha))$, $\mathcal{F}$ counts the number of indices $j \neq i$ such that $c_i \neq g(y_i)$ using the tables. Let t be the number. Then, with probability 2^{-t}, $\mathcal{F}$ outputs c_i; otherwise, $\mathcal{F}$ outputs a random bit.

For analysis of quantum circuits, as in [AGG$^+$20], we first consider $\mathcal{C}$ being an inherently probabilistic circuit. For any $H \subseteq \{0,1\}^n$, let $y \sim H$ uniformly at random. Then, for any $y_1, \ldots, y_k \in (\{0,1\}^n)^k$,

$$\Pr_{y\sim H}[y_1, \ldots, y_k \text{ generated by } \mathcal{F}] = \frac{u}{\delta k} \cdot \Pr_{r\sim\{0,1\}^m}[y_1, \ldots, y_k \text{ generated by } G(r)]. \tag{19}$$

where u is the number of $y_i \in H$. Since $\mathbb{E}_{r,\mathcal{C}}[\mathcal{C}(r) = g^{\otimes k}(G(r))] \geq \epsilon$, for a random r, the probability that $u \geq \rho k$ and $\mathcal{C}(r) = g^{\otimes k}(G(r))$ is at least $\epsilon - q$, by the hitting property of G. Hence, the probability that $u \geq \rho k$ and $\mathcal{C}$ succeeds for $y_1, \ldots, y_k$ generated by F on a random $x \in H$ is $(\epsilon - q) \cdot \rho/\delta = 4q$, since each $(y_1, \ldots, y_k)$ has at least ρ/δ of its probability under $G(r)$ by Eq. (19). Then, we can compute the expected success probability of $\mathcal{F}$ on $y \in H$ given $u \geq \rho k$ by Theorem 3.2 in [IW97], which is

$$\mathop{\mathbb{E}}_{y\sim H}[\mathcal{F}(y) = g(y) \mid u \geq \rho k] \geq \frac{1}{2} + q.$$

Since $u \geq \rho k$ has probability at least $1-q$, the overall success probability is at least $(1+q)/2$. Finally, by Lemma 2.7 in [AGG$^+$20], we can change the inherently probabilistic circuit by a quantum circuit and the result still holds.

Hence, $\mathcal{F}$ has expected probability $(1+q)/2$ on $1-\delta$ fraction of inputs. Then, we can take $O(n/q^2)$ copies and take the majority of them, which gives a circuit of size $O((|\mathcal{C}| + kM)n/q^2) \leq s$ if $|\mathcal{C}| = \Omega(sq^2 n^{-O(1)})$, and has success probability at least $1-\delta$. The Claim is then proved. □

By Claim C.6, we know that $\mathsf{XG} : \{0,1\}^{O(n)} \to \{0,1\}^{n^2}$ is a $(2^{\Omega(n)}, 2^{\Omega(n)}, 2^{-\Omega(n)}, 1/3)$-quantum direct product generator.

Finally, feeding the output of XG to the quantum Nisan-Wigderson generator (Lemma B.1) C_{NW} gives the desired quantum pseudo-random generator, which completes the proof of the lemma. □

D Quantum fine-grained hardness based on QETH

In this section, we will show that $2n \times 2n$ bipartite permutation independent set problem is hard under QETH.

Lemma 4.37. *Assuming* QETH*, there is no $2^{o(n\log n)}$-time quantum algorithm that solves $2n \times 2n$* Bipartite Permutation Independent Set *problem.*

More specifically, We "quantize" the fine-grained reduction in [LMS11]. The reduction chain is as follows:

$$\begin{aligned}&\textsf{3-SAT} \le_{FG} \textsf{3-Coloring} \le_{FG} n \times n \textsf{ Clique} \le_{FG} n \times n \textsf{ Permutation Clique}\\ &\le_{FG} n \times n \textsf{ Permutation Independent Set}\\ &\le_{FG} 2n \times 2n \textsf{ Bipartite Permutation Independent Set}\end{aligned}$$

We first define some intermediate fine-grained problems.

Definition D.1 ($n \times n$ Clique problem)**.** *Given a graph on the vertex set $[n] \times [n]$, decide if there exists $i_1, \ldots, i_n \in [n]$ such that the subgraph on $(1, i_1), \ldots, (n, i_n)$ forms an n-clique.*

Definition D.2 ($n \times n$ Permutation Clique/Independent Set problem)**.** *Given a graph on the vertex set $[n]\times[n]$, decide if there exists a permutation $\pi \in \mathcal{S}_n$ such that the subgraph on $(1, \pi(1)), \ldots, (n, \pi(n))$ forms an n-clique/independent set.*

The following claims shows that the aforementioned reductions work for quantum lower bounds.

Claim D.3. *Under* QETH*, there is no $2^{o(n)}$-time quantum algorithm for* 3-Coloring*, where n is the number of vertices in the input graph.*

Proof. By the NP-complete proof of 3-Coloring, we know that a 3-CNF formula with n variables and m clauses can be reduced to a 3-Coloring instance in time $O(n + m)$. Hence, a $2^{o(n)}$-time quantum algorithm for 3-Coloring implies a $2^{o(n)}$-time quantum algorithm for 3-SAT, which implies that QETH fails. □

Claim D.4. *If $n \times n$* Clique *can be solved in $2^{o(n\log n)}$ time quantumly, then 3-*Coloring *can be solved in $2^{o(n)}$ time quantumly.*

Proof. We use the reduction given by [LMS11]. Let G be an instance of 3-Coloring with n vertices. The reduction can produce a graph H with vertices $[k] \times [k]$ such that $n \le k \log_3 k - k$. Then, G is 3-colorable if and only if H is a "Yes" instance of $k \times k$ Clique. The reduction takes $\mathsf{poly}(k)$-time classically.

Hence, if there exists a quantum algorithm for $k \times k$ Clique in time $2^{o(k\log k)}$, then it gives a quantum algorithm for 3-Coloring that runs in time $2^{o(n)}$. □

Claim D.5. *If $n \times n$* Permutation Clique/Independent Set *can be solved in $2^{o(n\log n)}$ time quantumly, then $n \times n$* Clique *can also be solved in $2^{o(n\log n)}$ time quantumly.*

Proof. By [LMS11], there is a reduction from $n \times n$ Clique to $n \times n$ Permutation Clique that takes $2^{O(n\log\log n)} = 2^{o(n\log n)}$ time classically. Hence, the reduction also works for quantum $2^{o(n\log n)}$-time lower bound.

Note that $n \times n$ Permutation Clique and $n \times n$ Permutation Independent Set are equivalent problem, since we can reduce them by taking the complement graph. □

Claim D.6. *If $2n \times 2n$* Bipartite Permutation Independent Set *can be solved in $2^{o(n\log n)}$ quantumly, then $n \times n$* Permutation Independent Set *can be solved in $2^{o(n\log n)}$ time quantumly.*

Proof. By [LMS11], the classical reduction takes time $O(n^2)$. Hence, it also works for quantum algorithms. □

Finally, we can prove the QETH-hardness of $2n \times 2n$ bipartite permutation independent set problem:

Proof of Lemma 4.37. It follows from Claim D.3, D.4, D.5 and D.6. □

E Proofs for Corollary 5.10

Corollary 5.10. SMCSP *with classical descriptions of quantum states as inputs is in* QCMA*.*

Lemma E.1. *Given $v = [v_0, \ldots, v_{2^n-1}]$ for $v_i \in \mathbb{C}$ for $i = 0, \ldots, 2^n - 1$, there exists a quantum circuit such that the state $|v\rangle$ can be computed in time $\mathsf{poly}(2^n)$ with $\langle i|v\rangle = v_i$.*

Proof. We show that one can use single-qubit rotations to construct $|v\rangle$.

We first prepare $|0^{n+1}\rangle$. Then, we do a single-qubit rotation on the first qubit such that

$$|0^{n+1}\rangle \to \sqrt{\frac{\sum_{i=0}^{2^{n-1}-1}|v_i|^2}{\sum_{i=0}^{2^n-1}|v_i|^2}}|0\rangle|0^n\rangle + \sqrt{\frac{\sum_{i=2^{n-1}}^{2^n-1}|v_i|^2}{\sum_{i=0}^{2^n-1}|v_i|^2}}|1\rangle|0^n\rangle.$$

Then, let the first qubit be the control qubit and apply the controlled rotation to rotate the second qubit to be

$$\sqrt{\frac{\sum_{i=0}^{2^{n-2}-1}|v_i|^2}{\sum_{i=0}^{2^{n-1}-1}|v_i|^2}}|0\rangle + \sqrt{\frac{\sum_{i=2^{n-2}}^{2^{n-1}-1}|v_i|^2}{\sum_{i=0}^{2^{n-1}-1}|v_i|^2}}|1\rangle, \text{if the first qubit is } |0\rangle,$$

$$\sqrt{\frac{\sum_{i=2^{n-1}}^{2^{n-1}+2^{n-2}-1}|v_i|^2}{\sum_{i=2^{n-1}}^{2^n-1}|v_i|^2}}|0\rangle + \sqrt{\frac{\sum_{i=2^{n-1}+2^{n-2}}^{2^n-1}|v_i|^2}{\sum_{i=2^{n-1}}^{2^n-1}|v_i|^2}}|1\rangle, \text{if the first qubit is } |1\rangle.$$

By doing these controlled rotations in sequence, we can obtain $||v|\rangle$ where $\langle v|i\rangle = |v_i|$ for all i. Let $v_j = e^{-i\theta_j}|v_j|$ without loss of generality. Then, condition on j, we do the following rotation on the $(n+1)$-th qubit:

$$|0\rangle \to e^{-i\theta_j}|0\rangle$$

for all j. This gives $|v\rangle$.

Finally, we use at most $2^{O(n)}$ (control) rotations. By Remark 11, each controlled rotation can be implemented with at most $2^{O(n)}$ overhead. Hence, the verifier can construct $|v\rangle$ in time $\mathsf{poly}(2^n)$. □

Proof of Corollary 5.10. Following Lemma E.1, we can make $\mathsf{poly}(n, s, t)$ copies of the state in polynomial time. Then, following the proof for Theorem 5.9, the problem is in QCMA. □

F Quantum Circuit Class

In this section, we will show some properties of the quantum circuit $\mathsf{QC}(s, \mathcal{G})$. Note that $\mathcal{G}$ considered in this paper are universal gate set with constant fan-in. So, the results here are also for constant fan-in universal gate sets.

Claim F.1. *For $n \in \mathbb{N}$, there exists a constant c such that a random Boolean function $f : \{0,1\}^n \to \{0,1\}$ has quantum circuit complexity greater $\frac{2^n}{(c+1)n}$ with probability at least $1 - 2^{\frac{2^n}{c+1}}$.*

Proof. For any s-gate and $(n+t)$-qubit quantum circuit (where $n + t \leq s$), there are at most

$$\binom{n+qs+t}{q}^s |\mathcal{G}|^s \leq 2^{cs \log s}$$

possible circuits for some constant c large enough, where $\mathcal{G}$ is the quantum gate set, and q is the maximum number of qubits for any gate in $\mathcal{G}$ can operate on. Let $s = \frac{2^n}{(c+1)n}$. Then the number of circuits of size s is at most $2^{cs \log s} < 2^{\frac{c}{c+1} \cdot 2^n}$.

There are 2^{2^n} Boolean functions from $\{0,1\}^n$ to $\{0,1\}$. Suppose we pick one function uniformly randomly, then for every fixed quantum circuit $\mathcal{C}$ and input $x \in \{0,1\}^n$, the probability that $\|(\langle f(x)| \otimes I_{n+t-1})\mathcal{C}|x, 0^t\rangle\| \geq \frac{1}{2}$ is $\frac{1}{2}$. Therefore, the probability that a fixed quantum circuit can compute $f(x)$ for all $x \in \{0,1\}^n$ is at most $\frac{1}{2^{2^n}}$. By using union bound, the probability that there exists $\mathcal{C}$ of size $\frac{2^n}{(c+1)n}$ that can compute f is at most $\frac{2^{\frac{c}{c+1} \cdot 2^n}}{2^{2^n}} = 2^{\frac{2^n}{c+1}}$. □

Claim F.2. *For $s = \mathsf{poly}(n)$ and $\mathcal{G}$ a gate set that contains only constant fan-in gates, $\mathsf{BQC}(s, \mathcal{G})$ is in $\mathsf{DSPACE}(O(s^2))/O(s^2)$.*

Proof. The proof follows from the idea of showing $\mathsf{BQP} \subset \mathsf{PSPACE}$. Let $L \in \mathsf{BQC}(s, \mathcal{G})$ and $\{\mathcal{C}_n\}$ be the quantum circuit family in $\mathsf{QC}(s, \mathcal{G})$ that can solve L. Then, we show that there is a $O(s^2)$-space TM T with $O(s \log s)$-bit advice that can simulates $\mathcal{C}_n$.

Let $\mathcal{C}_n$ be the advice to T. We first calculate the number of bits needed to represent s-gate circuit. For each gate, we need $O(\log s)$ to specify its wires and 2^a register to record the corresponding unitary, where a is the maximum fan-in of gates in $\mathcal{G}$. Note that a unitary U may has entries that cannot be written down in bounded bits. Therefore, we let the precision to every entry in U be $\epsilon = \frac{1}{c2^s}$ for some constant c large enough, which requires number of bits $\log \frac{1}{\epsilon} = O(s)$. The total number of bits required for each gate is $O(s)$. and thus the number bits for the circuit is $O(s^2)$.

Now, suppose $\mathcal{C}_n = U_s U_{s-1} \cdots U_1$. For any $x \in \{0,1\}^n$ the probability that $\mathcal{C}_n$ accepts is

$$\sum_{y \in A} |\langle y|U_s U_{s-1} \cdots U_1|x\rangle|^2,$$

where $A := \{y : y \text{ has the first bit as } 1\}$. Then, the TM T computes each branch one-by-one. for any $y \in A$

$$\langle y|U_s U_{s-1} \cdots U_1|x\rangle = \sum_{z_1, \ldots, z_{s-1} \in \{0,1\}} \langle y|U_s|z_{s-1}\rangle\langle z_{s-1}|U_{s-1}|z_{s-2}\rangle\langle z_{s-2}| \cdots |z_1\rangle\langle z_1|U_1|x\rangle. \tag{20}$$

Note that U_i is a constant-dimensional unitary and x and z_j's are vectors with exactly one non-zero entry. So, computing $\langle z_j|U_j|z_{j-1}\rangle$ only requires $O(s)$ (since the entries in U takes $O(s)$ space for the precision). Then, since we can also compute $\langle z_j|U_j|z_{j-1}\rangle$ one by one, the space required for each branch in Eq. (20) is just $O(s)$. Therefore, the space we need is at most $O(s^2)$ (including the space for the advice).

Note that our calculation in Eq. (20) will have error since our precision to each entry in the unitary is $\epsilon = \frac{1}{c2^s}$. Let $\tilde{U}_s\tilde{U}_{s-1}\cdots\tilde{U}_1$ be what we really compute. Then,

$$\sum_{y\in A}|\langle y|U_sU_{s-1}\cdots U_1|x\rangle|^2 - \sum_{y\in A}|\langle y|\tilde{U}_s\tilde{U}_{s-1}\cdots\tilde{U}_1|x\rangle|^2 \leq O(2^{s+n}\epsilon).$$

By setting $\epsilon = \frac{1}{c2^s}$ for some constant c large enough, T can solve L with probability at least $2/3$ by having an amplified version of $\mathcal{C}_n$ at first (e.g., parallel repetition).

□

Claim F.3 (Diagonalization for quantum circuits). *For every $k \in \mathbb{N}_+$, there exists a language $L_k \in \mathsf{PSPACE}$ but $L_k \notin \mathsf{BQC}[n^k]$ for sufficiently large n.*

Proof. By Claim F.2, we know that $\mathsf{BQC}[n^k]$ is contained in $\mathsf{DSPACE}[n^{2k}]/n^{2k}$. By a nonuniform almost everywhere hierarchy for space complexity (Lemma 11 in [OS16]), we know that $\mathsf{DSPACE}[n^{3k}] \not\subset \mathsf{DSPACE}[n^{2k}]/n^{2k}$ for sufficiently large n. Hence, we can find a language $L_k \notin \mathsf{BQC}[n^k]$. □

Claim F.4 (BQC size hierarchy). *For $n > 0$, let $s(n) = o(\frac{2^n}{n})$. Then, there exists a Boolean function f in $\mathsf{BQC}[s(n)]\backslash\mathsf{BQC}[s(n) - O(n)]$, i.e., f can be computed by an $s(n)$-size quantum circuit but not computed by any $(s(n) - O(n))$-size quantum circuit.*

Proof. The proof is very similar to the argument for classical circuits. By Claim F.1, we can find a function g that requires quantum circuit of size $2^n/cn$ for some $c > 1$. Suppose there are t inputs $x_1, \ldots, x_t$ such that $g(x_i) = 1$ for $i \in [t]$. Then, we construct a series of functions g_i for $i = 0, 1, \cdots, t$ such that $g_i(x) = 1$ if and only if $x \in \{x_1, \ldots, x_i\}$. It's easy to see that the following properties are satisfied:

- $g_0 \in \mathsf{BQC}[0]$ and $g_t \in \mathsf{BQC}[2^n/cn]$.
- For $0 \leq i < t$, the difference of the quantum circuits size of g_i and g_{i+1} is at most $O(n)$. It follows since g_i and g_{i+1} are only different at x_i.

Hence, there exists an $i > 0$ such that the quantum circuit size of g_i is at most $s(n)$ but lager than $s(n) - O(n)$, since $s(n) = o(2^n/cn)$.

□