A Framework of Quantum Strong Exponential-Time Hypotheses

Harry Buhrman
QuSoft, CWI, Amsterdam, The Netherlands
University of Amsterdam, The Netherlands

Subhasree Patro
QuSoft, CWI, Amsterdam, The Netherlands
University of Amsterdam, The Netherlands

Florian Speelman
QuSoft, CWI, Amsterdam, The Netherlands
University of Amsterdam, The Netherlands

Abstract

The strong exponential-time hypothesis (SETH) is a commonly used conjecture in the field of complexity theory. It essentially states that determining whether a CNF formula is satisfiable can not be done faster than exhaustive search over all possible assignments. This hypothesis and its variants gave rise to a fruitful field of research, fine-grained complexity, obtaining (mostly tight) lower bounds for many problems in P whose unconditional lower bounds are very likely beyond current techniques. In this work, we introduce an extensive framework of Quantum Strong Exponential-Time Hypotheses, as quantum analogues to what SETH is for classical computation.

Using the QSETH framework, we are able to translate quantum query lower bounds on black-box problems to conditional quantum time lower bounds for many problems in P. As an example, we provide a conditional quantum time lower bound of $\Omega(n^{1.5})$ for the Longest Common Subsequence and Edit Distance problems. We also show that the n^2 SETH-based lower bound for a recent scheme for Proofs of Useful Work carries over to the quantum setting using our framework, maintaining a quadratic gap between verifier and prover.

Lastly, we show that the assumptions in our framework can not be simplified further with relativizing proof techniques, as they are false in relativized worlds.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Quantum complexity theory

Keywords and phrases complexity theory, fine-grained complexity, longest common subsequence, edit distance, quantum query complexity, strong exponential-time hypothesis

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.19

Related Version *Full Version*: `https://arxiv.org/abs/1911.05686`

Funding SP is supported by the Robert Bosch Stiftung. HB, SP, and FS are additionally supported by NWO Gravitation grants NETWORKS and QSC, and EU grant QuantAlgo.

Acknowledgements We would like to thank Andris Ambainis, Gilles Brassard, Frédéric Magniez, Miklos Santha, Mario Szegedy, and Ronald de Wolf for helpful discussions.

1 Introduction

There is a rich diversity of computational problems that are solvable in polynomial time; some that have surprisingly fast algorithms, such as the computation of Fourier transforms or solving linear programs, and some for which the worst-case run time has not improved much for many decades. The problem is that we have no techniques for proving *superlinear* lower bounds. Of the latter category Edit Distance is a good example: this is a problem


© Harry Buhrman, Subhasree Patro, and Florian Speelman;
licensed under Creative Commons License CC-BY 4.0

with high practical relevance, and an $O(n^2)$ algorithm using dynamic programming has been known for many decades. Even after considerable effort, no algorithm has been found that can solve this problem essentially faster than n^2. The best known algorithms runs in $O(n^2/\log^2 n)$ time [34], still a nearly quadratic run time.

Traditionally, the field of (structural) complexity theory has studied the time complexity of problems in a relatively coarse manner – the class P, of problems solvable in polynomial time, is one of the central objects of study in complexity theory.

Consider CNF-SAT, the problem of whether a formula, input in conjunctive normal form, has a satisfying assignment. What can complexity theory tell us about how hard it is to solve this problem? For CNF-SAT, the notion of NP-completeness gives a convincing reason why it is hard to find a polynomial-time algorithm for this problem: if such an algorithm is found, all problems in the complexity class NP are also solvable in polynomial time, showing P = NP.

Not only is no polynomial-time algorithm known, but (if the clause-length is arbitrarily large) no significant speed-up over the brute-force method of trying all 2^n assignments is known. Impagliazzo, Paturi, and, Zane [31, 32] studied two ways in which this can be conjectured to be optimal. The first of which is called the *Exponential-Time Hypothesis* (ETH).

▶ **Conjecture 1** (Exponential-Time Hypothesis). *There exists a constant $\alpha > 0$ such that CNF-SAT on n variables and m clauses can not be solved in time $O(m2^{\alpha n})$ by a (classical) Turing machine.*

This conjecture can be directly used to give lower bounds for many natural NP-complete problems, showing that if ETH holds then these problems also require exponential time to solve. The second conjecture, most importantly for the current work, is the *Strong Exponential-Time Hypothesis* (SETH).

▶ **Conjecture 2** (Strong Exponential-Time Hypothesis). *There does not exist $\delta > 0$ such that CNF-SAT on n variables and m clauses can be solved in $O(m2^{n(1-\delta)})$ time by a (classical) Turing machine.*

The strong exponential-time hypothesis also directly implies many interesting exponential lower bounds within NP, giving structure to problems within the complexity class. A wide range of problems (even outside of just NP-complete problems) can be shown to require strong exponential time assuming SETH: for instance, recent work shows that, conditioned on SETH, classical computers require exponential time for *strong simulation* of several models of quantum computation [29, 35].

Surprisingly, SETH is not only a very productive tool for studying the hardness of problems that likely require exponential time, but can also be used to study the difficulty of solving problems within P, forming a foundation for the field of *fine-grained complexity*. The first of such a SETH-based lower bound was given in [40], via a reduction from CNF-SAT to the Orthogonal Vectors problem, showing that a truly subquadratic algorithm that can find a pair of orthogonal vectors among two lists would render SETH false.

The Orthogonal Vectors problem became one of the central starting points for proving SETH-based lower bounds, and conditional lower bounds for problems such as computing the Frechet distance between two curves [20], sequence comparison problems such as the string alignment problem [6] and Dynamic Time Warping [4], can all obtained via a reduction from Orthogonal Vectors. Both the Longest Common Subsequence (LCS) and the Edit Distance problems [11] can also be shown to require quadratic time

conditional on SETH, implying that any super-logarithmic improvements over the classic simple dynamic programming algorithm would also imply better algorithms for satisfiability – a barrier which helps explain why it has been hard to find any new algorithms for these problems.

Besides CNF-SAT, the conjectured hardness of other key problems like 3SUM and APSP is also commonly used to prove conditional lower bounds for problems in P. See the recent surveys [38, 39] for an overview of the many time lower bounds that can be obtained when assuming only the hardness of these key problems.

All these results give evidence for the hardness of problems relative to classical computation, but interestingly SETH does not hold relative to *quantum* computation. Using Grover's algorithm [28, 17], quantum computers are able to solve CNF-SAT (and more general circuit satisfiability problems) in time $2^{n/2}$, a quadratic speedup relative to the limit that SETH conjectures for classical computation.

Even though this is in violation of the SETH bound, it is not in contradiction to the concept behind the strong exponential-time hypothesis: the input formula is still being treated as a black box, and the quantum speedup comes "merely" from the general quadratic improvement in unstructured search[1].

It could therefore be natural to formulate the quantum exponential time hypothesis as identical to its classical equivalent, but with an included quadratic speedup, as a "basic QSETH". For some problems, such as Orthogonal Vectors, this conjecture would already give tight results, since these problems are themselves amenable to a speedup using Grover's algorithm. See for instance the Master's thesis [37] for an overview of some of the SETH-based lower bounds that are violated in the quantum setting.

On the other hand, since the conditional lower bound for all problems are a quadratic factor lower than before, such a "basic QSETH" lower bound for LCS or Edit Distance would be merely linear. The best currently-known quantum algorithm that computes edit distance takes quadratic time, so we would lose some of the explanatory usefulness of SETH in this translation to the quantum case.

In this work, we present a way around this limit. Realize that while finding a single marked element is quadratically faster for a quantum algorithm, there is no quantum speedup for many other similar problems. For instance, computing whether the number of marked elements is odd or even can not be done faster when allowing quantum queries to the input, relative to allowing only classical queries [15, 27].

Taking the LCS problem again as an illustrative example, after careful inspection of the reductions from CNF-SAT to LCS [3], we show that the result of such a reduction encodes more than merely the existence of an a satisfying assignment. Instead, the result of these reductions also encodes whether *many* satisfying assignments exist (in a certain pattern), a problem that could be harder for quantum computers than unstructured search. The "basic QSETH" is not able to account for this distinction, and therefore does not directly help with explaining why a linear-time quantum algorithm for LCS has not been found.

We present a framework of conjectures, that together form an analogue of the strong exponential-time hypothesis: QSETH. In this framework, we account for the complexity of computing various properties on the set of satisfying assignments, giving conjectured quantum time lower bounds for variants of the satisfiability problem that range from $2^{n/2}$ up to 2^n.

[1] For unstructured search this bound is tight [16, 19]. Bennett, Bernstein, Brassard, and Vazirani additionally show that with probability 1 relative to a random oracle all of NP cannot be solved by a bounded-error quantum algorithm in time $o(2^{n/2})$.)

Summary of results

- We define the QSETH framework, connecting quantum query complexity to the proving of fine-grained (conditional) lower bounds of quantum algorithms. The framework encompasses both different properties of the set of satisfying assignments, and is also able to handle different input circuit classes – giving a hierarchy of assumptions that encode satisfiability on CNF formulas, general formulas, branching programs, and so on.
  - To be able to handle more-complicated properties of the satisfying assignments, we require such a property to be *compression oblivious* – a notion we define to capture the cases where query complexity is a lower bound for the time complexity, even for inputs that are "compressible" as a truth table of a small formula.[2] We give various results to initiate the study of the set of compression-oblivious languages.
- Some SETH-based $\Omega(T)$ lower bounds carry over to $\Omega(\sqrt{T})$ QSETH lower bounds, from which we immediately gain structural insight to the complexity class BQP.
- We show that, assuming QSETH, the *Proofs of Useful Work* of Ball, Rosen, Sabin and Vasudevan [13] require time $\widetilde{O}(n^2)$ to solve on a quantum computer, matching the classical complexity of these proofs of work.
- We prove that the Longest Common Subsequence (and the Edit Distance) problem requires $\Omega(n^{1.5})$ time to solve on a quantum computer, conditioned on QSETH. We do this by showing that LCS (similarly, edit distance) can be used to compute a harder property of the set of satisfying assignments than merely deciding whether one satisfying assignment exists.

  Following [5], we are able to show this for a version of QSETH where the input formulas are *branching programs* instead, giving a stronger result than assuming the hardness for only CNF inputs.
- As a corollary to the proof of the conditional LCS lower bound, we can show that the query complexity of the restricted Dyck language is linear for any $k = \omega(\log n)$, partially answering an open question posed by Aaronson, Grier, and Schaeffer [2].[3]

Related work

Independently from this work, Aaronson, Chia, Lin, Wang, and Zhang [1] recently also defined a basic quantum version of the strong exponential-time hypothesis, which assumes that a quadratic speed-up over the classical SETH is optimal. They present conditional quantum lower bounds for OV, the closest pair problem, and the bichromatic closest pair problem, by giving fine-grained quantum reductions to CNF-SAT. All such lower bounds have a quadratic gap with the corresponding classical SETH lower bound.

Despite the overlap in topic, these results turn out to be complementary to the current work: In the current work we focus on defining a more extensive framework for QSETH that generalizes in various ways the basic version. Our more general framework can exhibit a quantum-classical gap that is less than quadratic, which allows us to give conditional lower bounds for LCS and edit distance ($\Omega(n^{1.5})$) and useful proofs of work (a quadratic gap between prover and verifier). For our presented applications, the requirements of the fine-grained reductions are lower, e.g., when presenting a lower bound of $n^{1.5}$ for LCS or

[2] This notion is conceptually related to the Black-Box Hypothesis introduced by [14] and studied by [30].
[3] Lower bounds for the restricted Dyck language were recently independently proven by Ambainis, Balodis, Iraids, Khadiev, Klevickis, Prūsis, Shen, Smotrovs and Vihrovs [9].

edit distance it is no problem if the reduction itself takes time $\widetilde{O}(n)$.[4] Conversely, we do not give the reductions that are given by [1]; those results are distinct new consequences of QSETH (both of the QSETH that is presented in that work, and of our more extensive QSETH framework).

Structure of the paper

In Section 2 we motivate and state the QSETH framework. Following that, in Section 3 we present the direct consequences of QSETH, including the maintaining of some current bounds (with a quadratic loss), and the Useful Proof of Work lower bound. In Section 4 we present the conditional lower bounds for LCS and the Edit Distance problem, of which the proofs can be found in the full version of the paper [22]. Additionally, the proof lower bounding the query complexity of the restricted Dyck language can be found in the full version. Finally, we conclude and present several open questions in Section 5.

2 Defining the Quantum Strong Exponential-Time Hypothesis

Almost all known lower bounds for quantum algorithms are defined in terms of *query* complexity, which measures the number of times any quantum algorithm must access the input to solve an instance of a given problem. For example the *polynomial method* [15] and the *adversary method* [8] are two of the main techniques that can be applied in many situations.

Despite the success of quantum query complexity and the fact that we know tight query lower bounds for many problems, the query model does not take into account the computational efforts required after querying the input. In particular, it is not possible to use query complexity to prove any lower bound greater than linear, since any problem is solvable in the query-complexity model after all bits are queried. In general we expect the time needed to solve most problems to be much larger than the number of queries required for the computation, but it still seems rather difficult to formalize methods to provide unconditional quantum time lower bounds for explicit problems. We overcome these difficulties by providing a framework of conjectures that can assist in obtaining *conditional* quantum time lower bounds for many problems in BQP. We refer to this framework as the QSETH framework.

Variants of the classical SETH

The Strong Exponential-Time Hypothesis (SETH) was first studied in [31, 32], who showed that the lack of a $O(2^{n(1-\delta)})$ for a $\delta > 0$ algorithm to solve CNF-SAT is deeply connected to other open problems in complexity theory. Despite it being one the most extensively studied problems in the field of (classical) complexity theory, the best known classical algorithms for solving k-SAT run in $2^{n-n/O(k)}m^{O(1)}$ time [36], while the best algorithm for the more-general CNF-SAT is $2^{n-n/O(\log \Delta)}m^{O(1)}$ [23], where m denotes the number of clauses and $\Delta = m/n$ denotes the clause to variable ratio.

Even though no refutation of SETH has been found yet, it is plausible that the CNF structure of the input formulas does allow for a speed-up. Therefore, if possible, it is preferable to base lower bounds on the hardness of more general kinds of (satisfiability) problems, where

[4] We use $\widetilde{O}$ to denote asymptotic behavior up to polylogarithmic factors.

the input consists of wider classes of circuits. For example, lower bounds based on NC-SETH, satisfiability with NC-circuits as input,[5] have been proven for LCS, Edit Distance and other problems [5], in particular all the problems that fit the framework presented in [21].

Additionally, a different direction in which the exponential-time hypothesis can be weakened, and thereby made more plausible, is requiring the computation of different properties of a formula than whether at least one satisfying assignment exists. For example, hardness of *counting* the number of satisfying assignments is captured by #ETH [26]. Computing existence is equivalent to computing the OR of the set of satisfying assignments, but it could also conceivably be harder to output, e.g., whether the number of satisfying assignments is odd or even, or whether the number of satisfying assignments is larger than some threshold. In the quantum case, generalizing the properties to be computed is not only a way to make the hypothesis more plausible: for many of such tasks it is likely that the quadratic quantum speedup, as given by Grover's algorithm, no longer exist.

2.1 The basic QSETH

To build towards our framework, first consider what would be a natural generalization of the classical SETH.

▶ **Conjecture** (Basic QSETH). *There is no bounded error quantum algorithm that solves CNF-SAT on n variables, m clauses in $O(2^{\frac{n}{2}(1-\delta)}m^{O(1)})$ time, for any $\delta > 0$.*

This conjecture is already a possible useful tool in proving conditional quantum lower bounds, as we present an example of this in Section 3.1.[6]

We first extend this conjecture with the option to consider wider classes of circuits. Let γ denote a class of representations of computational models. Such a representation can for example be polynomial-size CNF formulas, polylog-depth circuits NC, polynomial-size branching programs BP, or the set of all polynomial-size circuits. The complexity of the latter problem is also often studied in the classical case, capturing the hardness of CircuitSAT.

▶ **Conjecture** (Basic γ-QSETH). *A quantum algorithm cannot, given an input C from the set γ, decide in time $O(2^{\frac{n}{2}(1-\delta)})$ whether there exists an input $x \in \{0,1\}^n$ such that $C(x) = 1$ for any $\delta > 0$.*

We also define AC^0_2 to be the set of all depth-2 circuits consisting of unbounded fan-in, consisting only of AND and OR gates. This definition is later convenient when considering wider classes of properties, and it can be easily seen that "basic AC^0_2-QSETH" is precisely the "basic QSETH" as defined above.

Since both these basic QSETH variants already contain a quadratic speedup relative to the classical SETH, conditional quantum lower bounds obtained via these assumptions will usually also be quadratically worse than any corresponding classical lower bounds for the same problems. For some problems, lower bounds obtained using the basic QSETH, or using γ-QSETH for a wider class of computation, will be tight. However, for other problems no quadratic quantum speedup is known.

[5] NC circuits are of polynomial size and polylogarithmic depth consisting of fan-in 2 gates.
[6] Additional examples of implications from such a version of QSETH can be found in the recent independent work of [1].

2.2 Extending QSETH to general properties

We now extend the "basic γ-QSETH" as defined in the previous section, to also include computing different properties of the set of satisfying assignments. By extending QSETH in this way, we can potentially circumvent the quadratic gap between quantum and classical lower bounds for some problems.

Consider a problem in which one is given some circuit representation of a boolean function $f : \{0,1\}^n \to \{0,1\}$ and asked whether a property $\mathrm{P} : \{0,1\}^{2^n} \to \{0,1\}$ on the truth table of this function evaluates to 1, that is, given a circuit C the problem is to decide if $\mathrm{P}(\mathrm{tt}(\mathrm{C})) = 1$, where $\mathrm{tt}(\mathrm{C})$ denotes the truth table of the boolean function computed by the circuit C. If one can only access C as a black box then it is clear that the amount of time taken to compute $\mathrm{P}(\mathrm{tt}(\mathrm{C}))$ is lower bounded by the number of queries made to the string $\mathrm{tt}(\mathrm{C})$. However, if provided with the description of C, which we denote by $\mathrm{desc}(\mathrm{C})$, then one can analyze C to compute $\mathrm{P}(\mathrm{tt}(\mathrm{C}))$ possibly much faster.

For example, take the representation to be polynomial-sized CNF formulas and the property to be OR. Then for polynomial-sized CNF formulas this is precisely the CNF-SAT problem. Conjecturing quantum hardness of this property would make us retrieve the "basic QSETH" of the previous section. Do note that we cannot simply conjecture that any property is hard to compute on CNF formulas: Even though the query complexity of AND on a string of length 2^n is $\Omega(2^n)$ classically and $\Omega(2^{n/2})$ in the quantum case, this property can be easily computed in polynomial time both classically and quantumly when provided with the description of the $n^{O(1)}$ sized CNF formula.

To get around this problem, we can increase the complexity of the input representation: If we consider inputs from AC^0_2, the set of all depth-2 circuits consisting of unbounded fan-in AND and OR gates, we now have a class that is closed under complementation. For this class, it is a reasonable conjecture that both AND, the question whether the input is a tautology and all assignments are satisfying, and OR, the normal SAT problem, are hard to compute.

After this step we can look at further properties than AND and OR. For instance, consider the problem of computing whether there exists an even or an odd number of satisfying assignments. This task is equivalent to computing the PARITY of the truth table of the input formula. How much time do we expect a quantum algorithm to need for such a task?

The quadratic speedup for computing satisfiability, i.e., the OR of the truth table of the input formula, is already captured by the model where the quantum computation only tries possible assignments and then performs Grover's algorithm in a black box manner. If PARITY is also computed in such a way, then we know from query complexity [15] that there is no speedup possible, and the algorithm will have to use $\Omega(2^n)$ steps. Our QSETH framework will be able to consider more-complicated properties, like PARITY.

Finally, observe that such a correspondence, i.e., between the query complexity of a property and the time complexity of computing this property on the set of satisfying assignments, cannot hold for *all* properties, even when we consider more complicated input classes besides CNF formulas. For instance, consider a property which is 0 on exactly the strings that are truth tables of polynomial-sized circuits, and is PARITY of its input on the other strings. Such a property has high quantum query complexity, but is trivial to compute when given a polynomial-sized circuit as input. We introduce the notion of *compression oblivious* below to handle this problem.

White box and black box computation of a property

We formalize the above intuitions in the following way. Let the variable γ denote a class of representation at least as complex as the set AC^0_2, where AC^0_2 denotes the set of poly sized depth-2 circuits consisting of only OR, AND gates of unbounded fan-in and NOT gates. For every n, let $\mathrm{P} : \{0,1\}^{2^n} \to \{0,1\}$ be some function family which defines a property. We define a meta-language L_{P} such that $L_{\mathrm{P}} = \{\mathrm{desc(C)} \mid \mathrm{C}$ is an element from the set γ and $\mathrm{P(tt(C))} = 1\}$. We now define the following terms:

▶ **Definition 3** (White-box algorithms). *An algorithm* A *decides the property* P *in* ***white-box*** *if* A *decides the corresponding meta-language* L_{P}*. That is, given an input string* desc(C)*,* A *accepts if and only if* $\mathrm{P(tt(C))} = 1$*. We use* $\mathrm{qTimeWB}_\epsilon(\mathrm{P})$ *to denote the time taken by a quantum computer to decide the language* L_{P} *with error probability* ϵ*.*

▶ **Definition 4** (Black-box algorithms). *An algorithm* A *decides the property* P *in* ***black-box*** *if the algorithm* $\mathrm{A}^f(1^n, 1^m)$ *accepts if and only if* $\mathrm{P(tt}(f)) = 1$*. Here,* f *is the boolean function computed by the circuit* C *and* m *is the upper bound on* $|\mathrm{desc(C)}|$ *which is the size of the representation*[7] *that describes* f*, and* A^f *denotes that the algorithm* A *has oracle access to the boolean function* f*. We use* $\mathrm{qTimeBB}_\epsilon(\mathrm{P})$ *to denote the time taken by a quantum computer to compute the property* P *in the black-box setting with error probability* ϵ*.*

Compression oblivious properties

We define the set of *compression oblivious* properties corresponding to γ as the set of properties where the time taken to compute this property in the black-box setting is lower bounded by the quantum query complexity of this property on all strings. Formally,

$$\mathcal{CO}(\gamma) = \{\text{properties P such that } \mathrm{qTimeBB}_\epsilon(\mathrm{P}\,|_{\mathrm{S}_\gamma}) \geq \Omega(\mathrm{Q}_\epsilon(\mathrm{P}))\},$$

where $\mathrm{Q}_\epsilon(\mathrm{P})$ denotes the quantum query complexity of the property P in a ϵ-bounded error query model and $\mathrm{S}_\gamma = \{\mathrm{tt(C)} \mid \mathrm{C}$ is an element of the set $\gamma\}$.

Defining QSETH

For each class of representation γ we now define the corresponding γ-QSETH*, which states that computing any compression-oblivious property P in the *white-box* setting is at least as hard as computing P in the *black-box* setting. More formally, for every class of representation γ, such as the class of depth-2 circuits AC^0_2 or poly-sized circuits of a more complex class, we hypothesize the following:

▶ **Conjecture 5** (γ-QSETH*). *For all properties* $\mathrm{P} \in \mathcal{CO}(\gamma)$*, we have* $\mathrm{qTimeWB}_\epsilon(\mathrm{P}\,|_\gamma) \geq \Omega(\mathrm{Q}_\epsilon(\mathrm{P}))$.

2.3 Observations on the set of compression oblivious properties

As the class γ gets more complex, the corresponding γ-QSETH* becomes more credible. The set of compression oblivious properties is an interesting object of study by itself. First consider some representative examples of whether various natural properties are compression oblivious. Note here that the example property that is not compression oblivious has to be

[7] For instance a CNF/DNF formula, an NC circuit, or a general circuit.

carefully constructed for this to be the case – it is natural to conjecture that for most natural properties the knowledge that the input can be written as the truth table of a small circuit does not help in speeding up the computation.[8]

▶ **Example 6.** The properties AND and OR are in $\mathcal{CO}(\mathsf{AC}^0_2)$: The adversarial set that gives the tight query bound for the property AND (OR) are truth tables of functions that can be represented by $n^{O(1)}$ sized DNF (CNF) formulas. Namely, these are given by the formulas that reject (accept) a single possible input, which can be constructed by using n clauses that each contain a single variable or its negation. Because $\mathrm{Q}_\epsilon(\mathrm{AND}|_{\mathrm{S}_{\mathsf{AC}^0_2}}) = \mathrm{Q}_\epsilon(\mathrm{AND})$ and $\mathrm{qTimeBB}_\epsilon(\mathrm{AND}|_{\mathrm{S}_{\mathsf{AC}^0_2}}) \geq \mathrm{Q}_\epsilon(\mathrm{AND}|_{\mathrm{S}_{\mathsf{AC}^0_2}})$, we have $\mathrm{AND} \in \mathcal{CO}(\mathsf{AC}^0_2)$. The same holds for the property OR as well.

▶ **Example 7.** Consider the following property, defined on some string $z \in \{0,1\}^{2^n}$, which we view as the truth table of a formula or circuit:

$$\mathrm{P}_{\text{large-c}}(z) = \mathrm{PARITY}_{2^n}(z) \ \wedge \ [\text{there exists no circuit } C \text{ of size less than } 2^{n/100} \text{ s.t. } z = \mathrm{tt}(C).]$$

Because most strings are not a truth table of a small circuit, the query complexity of this property is close to the query complexity of PARITY, i.e., $\mathrm{Q}_\epsilon(\mathrm{P}_{\text{large-c}}) = \Omega(N)$. Nevertheless, the property is always 0 when restricted to truth tables of small circuits, and therefore trivial to compute. Therefore $\mathrm{P}_{\text{large-c}}$ is not compression oblivious for polynomial-sized circuits (or any smaller class of representations).

▶ **Example 8.** Whether PARITY is compression oblivious is unknown: the quantum query complexity of PARITY is $\Omega(N)$. Restricted to inputs which are truth tables of small formulas/circuits, the query complexity is $O(\sqrt{N})$, this is the maximum query complexity for any property when restricted to truth tables of a small circuit class [10, 33]. Conjecturing that PARITY is compression oblivious is natural, and incomparable to (but not necessarily less likely than) the main QSETH statement.

Given an explicit property P and a class of input representations γ, it would be desirable to unconditionally prove that the property P is γ-*compression oblivious*[9]. This is possible for some simple properties that have query complexity $\Theta(\sqrt{N})$ like OR, corresponding to ordinary satisfiability, and AND. Unfortunately, for more complicated properties, like computing the parity of the number of satisfying assignments, it turns out to be hard to find an unconditional proof that such a property is compression oblivious. The following theorem shows a barrier to finding such an unconditional proof: proving that such a property is compression oblivious implies separating P from PSPACE.

▶ **Theorem 9.** *If there exists a property* P *such that* $\mathrm{Q}_\epsilon(\mathrm{P}) = \widetilde{\omega}(\sqrt{N})$ *and* P *is* γ*-compression oblivious, and* $\mathrm{P} \in \mathsf{polyL}(N)$*, then* $\mathsf{P} \neq \mathsf{PSPACE}$*. Here* $N = 2^n$ *and* γ *represents the set of poly-sized circuits on* n *input variables.*

Here $\mathsf{polyL}(N)$ is same as $\mathsf{SPACE}(\mathsf{poly}\log N)$, i.e., class of properties computable in $\mathsf{poly}\log N$ amount of space. Note that SETH is already a much stronger assumption than $\mathsf{P} \neq \mathsf{PSPACE}$, therefore this observation leaves open the interesting possibility of proving

[8] In classical complexity theory, a closely related notion is the Black-Box Hypothesis introduced by [14] and studied by [30].

[9] We call a property P a γ-compression oblivious property if $\mathrm{P} \in \mathcal{CO}(\gamma)$.

that properties are compression oblivious assuming that the (Q)SETH holds for simpler properties. (For instance, these simpler properties could include OR and AND, for which it is possible to unconditionally prove that they are compression oblivious.)

Unfortunately, merely making such an assumption alone will likely not be enough to enable an easy proof that simple properties with high query complexity are compression oblivious: We show that there exists an oracle such that, if all computations and input models[10] have access to this oracle, QSETH is true but PARITY (for example) is not compression oblivious. This does give a relativization barrier to this question, showing that a non-relativizing proof will be necessary to prove that properties are compression oblivious.

▶ **Theorem 10.** *There exists an oracle relative to which the basic QSETH holds, but any property $P \in \mathsf{polyL}(N)$ for which $\mathrm{Q}_\epsilon(\mathrm{P}) = \widetilde{\omega}(\sqrt{N})$ is not γ-compression oblivious. Here γ consists of all polynomial-sized circuits (with oracle access).*

See Appendix A for the proofs of Theorems 9 and 10.

3 QSETH lower bounds for Orthogonal Vectors and Proofs of Useful Work

Recall that AC^0_2 denotes the set of polynomial-sized depth-2 circuits consisting of only OR and AND gates of unbounded fan-in. Because of the simple input structure, the AC^0_2-QSETH* conjecture is therefore closest to the classical SETH, and implies the "basic QSETH" as introduced in Section 2.1:

▶ **Corollary 11.** *If AC^0_2-QSETH* is true then there is no bounded error quantum algorithm that solves CNF-SAT on n variables, m clauses in $O(2^{(1-\delta)n/2}m^{O(1)})$ time, for any $\delta > 0$.*

Proof. Consider the property OR: $\{0,1\}^{2^n} \to \{0,1\}$. Using the fact that $\mathrm{OR} \in \mathcal{CO}(\mathsf{AC}^0_2)$, as shown in the previous section, we get $\mathrm{qTimeWB}_\epsilon(\mathrm{OR}|_{\mathsf{AC}^0_2}) \geq \Omega(\mathrm{Q}_\epsilon(\mathrm{OR})) = \Omega(2^{n/2})$. Due to the structure of the DNF formulas one can compute the property OR on DNF formulas on n variables, m clauses in $n^{O(1)}m^{O(1)}$ time. This implies that the hard cases in the set AC^0_2 for the OR property are the CNF formulas. Therefore, $\mathrm{qTimeWB}_\epsilon(\mathrm{OR}|_{\mathrm{CNF}}) \geq \Omega(2^{n/2})$ where the set CNF denotes all the polynomial sized CNF formulas. ◀

In this section we present several immediate consequences of the AC^0_2-QSETH* conjecture:

1. For some problems, classical SETH-based $\Omega(T)$ time lower bounds carry over to the quantum case, with AC^0_2-QSETH*-based $\Omega(\sqrt{T})$ quantum time lower bounds using (almost) the same reduction.
2. The *Proofs of Useful Work* of Ball, Rosen, Sabin and Vasudevan [13] require time $\widetilde{O}(n^2)$ to solve on a quantum computer, equal to their classical complexity, under AC^0_2-QSETH*.

3.1 Quantum time lower bounds based on AC^0_2-QSETH*

The statement of AC^0_2-QSETH* along with Corollary 11 can give quantum time lower bounds for some problems for which we know classical lower bounds under SETH (Conjecture 2).

[10] For example, consider circuit SAT for circuits that have access to an oracle.

▶ **Corollary 12.** *Let* P *be a problem with an* $\Omega(T)$ *time lower bound modulo* SETH. *Then,* P *has an* $\widetilde{\Omega}(\sqrt{T})$ *quantum time lower bound conditioned under* AC^0_2-QSETH* *if there exists a classical reduction from* CNF-SAT *to the problem* P *taking* $O(2^{\frac{n}{2}(1-\alpha)})$ *(for* $\alpha > 0$*) time or if there exists an efficient reduction that can access a single bit of the reduction output.*[11]

In Appendix B we explain how we can preserve the following two classical SETH lower bounds, with a quadratic gap:

▶ **Example 13.** The OV problem is defined as follows. Given two sets U and V of N vectors, each over $\{0,1\}^d$ where $d = \omega(\log N)$, determine whether there exists a $u \in U$ and a $v \in V$ such that $\Sigma_{l\in[d]} u_l v_l = 0$. The reduction of Williams [40], shows a classical lower bound of $\Omega(N^2)$ for this problem, and it can be modified to efficiently return single bits of the reduction. Therefore, assuming AC^0_2-QSETH*, any quantum algorithm requires time $\tilde{\Theta}(N)$ to solve OV for instances of size N.

▶ **Example 14.** The LCS problem is defined as follows. Given two strings a and b over an alphabet set Σ, the $\mathrm{LCS}(a,b)$ is the length of the longest subsequence common to both strings a and b. Modifying the reduction of [3], it can be shown that LCS requires time $\widetilde{\Omega}(N)$, assuming AC^0_2-QSETH*. This same bound can also be shown unconditionally, using query complexity and the observation that the majority function can be embedded in LCS.

See the recent results by Aaronson, Chia, Lin, Wang, and Zhang [1] for more examples of reductions from (a variant of) QSETH, that also hold for the basic QSETH of our framework. Additionally, there the authors define the notion of *Quantum Fine-grained Reductions* more generally, and present a study of OV that also includes the case of constant dimension.

We witness that with the AC^0_2-QSETH* conjecture, the SETH-based fine-grained lower bounds at best transfer to a square root lower complexity in the quantum case. This is definitely interesting on its own, but we are aiming for larger quantum lower bounds, in situations where the gap between the classical and quantum complexities is less than quadratic, which is why we focus on our more general framework.

3.2 Quantum Proofs of Useful Work

Other applications of AC^0_2-QSETH* include providing problems for which *Proofs of Useful Work (uPoW)* can be presented in the quantum setting. Ball et al. [13] propose uPoW protocols that are based on delegating the evaluation of low-degree polynomials to the prover. They present a classical uPoW protocol for the Orthogonal Vectors problem (OV) whose security proof is based on the assumption that OV needs $\Omega(n^{2-o(1)})$ classical time in the worst case setting, implying that the evaluation of a polynomial that encodes the instance of OV has average-case hardness. At the end of this protocol, the verifier is able to compute the number of orthogonal vectors in a given instance.

Therefore, the same protocol also works to verify the solutions to $\oplus$OV, where $\oplus$OV denotes the parity version of OV, i.e., given two sets U, V of n vectors from $\{0,1\}^d$ each, output the parity of number of pairs (u,v) such that $u \in U$, $v \in V$ and $\Sigma_{l\in[d]} u_l v_l = 0$,

[11] Note that we use a version of QSETH that relates to CNF-SAT as opposed to bounded clause-size k-SAT problems. One could also define a quantum hardness conjecture for k-CNF or k-DNF, for an arbitrary constant k, in the same way as the original SETH. This variant is required for reductions that use the fact that k is constant, which can occur through usage of the sparsification lemma [31]. For examples where this is necessary within fine-grained complexity, see the *Matching Triangles* problem mentioned in [7] or reductions like in [25].

where d is taken to be $\omega(\log n)$. Assuming AC^0_2-QSETH* and assuming PARITY $\in \mathcal{CO}(\mathsf{AC}^0_2)$ we get that $\oplus$CNF-SAT takes $\Omega(2^n)$ quantum time. Due to the classical reduction[12] given by [40], this protocol then implies a conditional quantum time lower bound of $\Omega(n^2)$ for the $\oplus$OV problem. Therefore, the uPoW protocol by [13] also requires quantum provers to take time $\widetilde{\Omega}(n^2)$.

4 Lower bounds for string problems using NC-QSETH*

In this section we discuss two consequences of the NC-QSETH* conjecture: Quantum time lower bounds for the LCS and Edit Distance problems. For length n input strings, the well-known Wagner–Fischer algorithm (based on dynamic programming) classically computes the edit distance in $O(n^2)$ time. A similar algorithm computes LCS in $O(n^2)$ time. Unfortunately, all the best known classical (and quantum) algorithms to compute these problems are also nearly quadratic. As mentioned above, results by [3, 11] prove that these near-quadratic time bounds might be tight: a sub-quadratic classical algorithm for computing LCS or edit distance would imply that SETH (Conjecture 2) is false.

SETH also implies quadratic lower bounds for many other string comparison problems, like Dynamic Time Warping and Frechet Distance, that also have (close to) optimal algorithms that are based on dynamic programming [21]. Bouroujeni et al. [18] give a sub-quadratic quantum algorithm for approximating edit distance within a constant factor which was followed by a better classical algorithm by Chakraborty et al. [24] However, no quantum improvements over the classical algorithms in the exact case are known to the best of our knowledge. Investigating why this is the case is an interesting open problem: is it possible to prove better (conditional) lower bounds, or can a better algorithm be found? We formulate the following questions for the example of LCS and the Edit Distance problem.

1. Is there a bounded-error quantum algorithm for LCS or Edit Distance that runs in a sub-quadratic amount of time?
2. Is it possible to obtain a superlinear lower bound for LCS or Edit Distance using the "basic QSETH"?
3. Can we use a different reduction to raise the linear lower bound for LCS or Edit Distance that we achieve under "basic-QSETH"?

We don't attempt to find a better algorithm for these string problems in this work, and it remains possible that no sub-quadratic quantum algorithm for these problems exists. Considering the second question: Using the basic QSETH loses a quadratic factor relative to the classical reduction, so it is clear that it will not be possible to directly translate a classical reduction to the quantum setting – since the quadratic classical SETH bound is tight. Therefore, to prove a "basic QSETH" lower bound for a problem where the gap between the best quantum and classical algorithms is less than quadratic, a fundamentally different (inherently quantum) reduction strategy would have to be found.

While the first two questions still remain open, we address the last question in this section. Using (a promise version of) the NC-QSETH* conjecture we prove conditional quantum time lower bounds of $\Omega(n^{1.5})$ for the LCS and Edit Distance problems[13].

[12] Note that here one can use the classical reduction from CNF-SAT to Orthogonal Vectors that runs in time $\widetilde{O}(2^{n/2})$.

[13] Note that, independently from our results, Ambainis et al. [9] recently presented a quantum query lower bound of $\Omega(n^{1.5-o(1)})$ for the Edit Distance problem, for algorithms that use the natural dynamic-

As global strategy, we will analyze earlier reductions [5] from branching program satisfiability to string problems, and show that solving the string problems (such as LCS) on the result of these (slightly modified) reductions can be used to compute a more complicated property of the branching program, which we call PP_{lcs}. The first step then is to give a reduction from BP-PP_{lcs}, which can be viewed as showing whether or not PP_{lcs} on a branching program is satisfied or not, to LCS. This step is formalized as the following theorem.

▶ **Theorem** (Informal statement of reduction). *There is a reduction from BP-PP_{lcs} on non-deterministic branching programs of size $2^{\text{poly}\log n}$ (length Z, width W) to an instance of the LCS problem on two sequences of length $M = 2^{n/2}(cW)^{O(\log Z)}$ for some constant c, and the reduction runs in $O(M)$ time.*

Our next step is to prove a quantum query complexity lower bound for this property, which, together with the assumption that the property is compression oblivious[14], implies a time lower bound for the LCS problem of $\widetilde{\Omega}(n^{1.5})$. The lower bound strategy for the Edit Distance problem is very similar to that of the LCS problem: the "gadgets" involved have to be constructed in a different way, but these gadgets can then be combined using a very similar method. Therefore, the reduction can be utilized to compute a property of the set of satisfying assignments that is closely related to BP-PP_{lcs}.

The full proofs of these reductions (together with the definition of PP_{lcs}) are presented in the full version of the paper [22].

5 Conclusion and Future Directions

We presented a quantum version of the strong exponential-time hypothesis, as QSETH, and demonstrated several consequences from QSETH. These included the transfer of previous Orthogonal-Vector based lower bounds to the quantum case, with a quadratically lower time bound than the equivalent classical lower bounds. We also showed two situations where the new QSETH does not lose this quadratic factor: a lower bound showing that computing edit distance or LCS takes time $n^{1.5}$ for a quantum algorithm, and an n^2 quantum lower bound for Proofs of Useful Work [13], both conditioned on QSETH.

Possible future applications for the QSETH framework are numerous. Most importantly, the QSETH can potentially be a powerful tool to prove conditional lower bounds for additional problems in BQP. The most natural candidates are other string problems, such as Dynamic Time Warping for example, but there are many other problems for which the "basic QSETH" does not immediately give tight bounds.

Additionally, the notion of *compression oblivious* properties are potentially interesting as an independent object of study. We expect most natural properties to be compression oblivious, but leave as an open question what complexity-theoretic assumptions are needed to show that, e.g., the parity function is compression oblivious.

Future directions also include a careful study of quantum time complexity of the other core problems in fine-grained complexity, such as 3SUM and APSP. Just like with satisfiability, the basic versions of these problems are amenable to a Grover-based quadratic speedup. It

programming approach of first reducing Edit Distance to connectivity on a 2D grid. However, that doesn't rule out the possibility of other $\widetilde{O}(n^{1.5-\alpha})$ quantum algorithms for the Edit Distance problem, for $\alpha > 0$.

[14] As discussed in Section 2.3, such an assumption is natural, implicit when considering more-complicated QSETH variants, and hard to prove unconditionally.

is possible that extensions of those key problems can be used to prove stronger conditional lower bounds, in a similar way to the reduction that was used for LCS or Edit Distance in the current work.

References

1 Scott Aaronson, Nai-Hui Chia, Han-Hsuan Lin, Chunhao Wang, and Ruizhe Zhang. On the quantum complexity of closest pair and related problems. *arXiv preprint*, 2019. arXiv:1911.01973.

2 Scott Aaronson, Daniel Grier, and Luke Schaeffer. A Quantum Query Complexity Trichotomy for Regular Languages. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:61, 2019. URL: https://eccc.weizmann.ac.il/report/2019/061.

3 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Quadratic-time hardness of LCS and other sequence similarity measures. *CoRR*, abs/1501.07053, 2015. arXiv:1501.07053.

4 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, pages 59–78, Washington, DC, USA, 2015. IEEE Computer Society. doi:10.1109/FOCS.2015.14.

5 Amir Abboud, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Ryan Williams. Simulating Branching Programs with Edit Distance and Friends Or: a Polylog Shaved is a Lower Bound Made. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 375–388, New York, NY, USA, 2016. ACM. doi:10.1145/2897518.2897653.

6 Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. Consequences of faster alignment of sequences. In *ICALP*, 2014.

7 Amir Abboud, Virginia Vassilevska Williams, and Huacheng Yu. Matching triangles and basing hardness on an extremely popular conjecture. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 41–50, New York, NY, USA, 2015. ACM. doi:10.1145/2746539.2746594.

8 Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 636–643, New York, NY, USA, 2000. ACM. doi:10.1145/335305.335394.

9 Andris Ambainis, Kaspars Balodis, Janis Iraids, Kamil Khadiev, Vladislavs Klevickis, Krisjanis Prusis, Yixin Shen, Juris Smotrovs, and Jevgenijs Vihrovs. Quantum lower and upper bounds for 2d-grid and dyck language. In Javier Esparza and Daniel Král', editors, *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, pages 8:1–8:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.MFCS.2020.8.

10 Andris Ambainis, Kazuo Iwama, Akinori Kawachi, Hiroyuki Masuda, Raymond H. Putra, and Shigeru Yamashita. Quantum identification of boolean oracles. In Volker Diekert and Michel Habib, editors, *STACS 2004*, pages 105–116, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

11 Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). *STOC*, 2015.

12 Theodore Baker, John Gill, and Robert Solovay. Relativizations of the P=?NP question. *SIAM Journal on computing*, 4(4):431–442, 1975.

13 Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Proofs of useful work. Cryptology ePrint Archive, Report 2017/203, 2017. URL: https://eprint.iacr.org/2017/203.

14 Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. *Journal of the ACM (JACM)*, 59(2):1–48, 2012.

15 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, July 2001. doi:10.1145/502090.502097.

16 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, October 1997. doi:10.1137/S0097539796300933.

17 E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi:10.1137/S0097539796300921.

18 Mahdi Boroujeni, Soheil Ehsani, Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, and Saeed Seddighin. Approximating edit distance in truly subquadratic time: Quantum and mapreduce. *CoRR*, abs/1804.04178, 2018. arXiv:1804.04178.

19 Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505, 1998.

20 Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless seth fails. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS '14, pages 661–670, Washington, DC, USA, 2014. IEEE Computer Society. doi:10.1109/FOCS.2014.76.

21 Karl Bringmann and Marvin Kunnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, pages 79–97, Washington, DC, USA, 2015. IEEE Computer Society. doi:10.1109/FOCS.2015.15.

22 Harry Buhrman, Subhasree Patro, and Florian Speelman. A framework of quantum strong exponential-time hypotheses, 2019. arXiv:1911.05686.

23 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. A duality between clause width and clause density for SAT. In *Proceedings of the 21st Annual IEEE Conference on Computational Complexity*, CCC '06, pages 252–260, Washington, DC, USA, 2006. IEEE Computer Society. doi:10.1109/CCC.2006.6.

24 Diptarka Chakraborty, Debarati Das, Elazar Goldenberg, Michal Koucký, and Michael E. Saks. Approximating edit distance within constant factor in truly sub-quadratic time. *CoRR*, abs/1810.03664, 2018. arXiv:1810.03664.

25 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. *ACM Trans. Algorithms*, 12(3):41:1–41:24, May 2016. doi:10.1145/2925416.

26 Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén. Exponential time complexity of the permanent and the tutte polynomial. *ACM Transactions on Algorithms (TALG)*, 10(4):21, 2014.

27 Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Limit on the speed of quantum computation in determining parity. *Phys. Rev. Lett.*, 81:5442–5444, December 1998. doi:10.1103/PhysRevLett.81.5442.

28 Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM. doi:10.1145/237814.237866.

29 Cupjin Huang, Michael Newman, and Mario Szegedy. Explicit lower bounds on strong quantum simulation. *arXiv preprint*, 2018. arXiv:1804.10368.

30 Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, Pierre McKenzie, and Shadab Romani. Does looking inside a circuit help? In *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

31 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

32 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

33 Robin Kothari. An optimal quantum algorithm for the oracle identification problem. In Ernst W. Mayr and Natacha Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, volume 25 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 482–493, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.STACS.2014.482.

34 William J. Masek and Michael S. Paterson. A faster algorithm computing string edit distances. *Journal of Computer and System Sciences*, 20(1):18–31, 1980. doi:10.1016/0022-0000(80)90002-1.

35 Tomoyuki Morimae and Suguru Tamaki. Fine-grained quantum computational supremacy. *Quantum Information & Computation*, 19(13&14):1089–1115, 2019. URL: http://www.rintonpress.com/xxqic19/qic-19-1314/1089-1115.pdf.

36 Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *J. ACM*, 52(3):337–364, May 2005. doi:10.1145/1066100.1066101.

37 Jorg Van Renterghem. The implications of breaking the strong exponential time hypothesis on a quantum computer. Master's thesis, Ghent University, 2019. URL: https://lib.ugent.be/fulltxt/RUG01/002/787/416/RUG01-002787416_2019_0001_AC.pdf.

38 Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis. *IPEC*, 2015.

39 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the ICM*, 2018. To appear.

40 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2):357–365, December 2005. doi:10.1016/j.tcs.2005.09.023.

A Observations on Compression Oblivious properties

Here we present an extra observation of the set of compression oblivious properties, and missing proofs of the statements in Section 2.3.

First, we show the following fact about how sets of compression-oblivious properties relate, relative to different computational models.

▶ **Fact 15.** *Given two classes of representations ζ and λ, if $\zeta \subseteq \lambda$ then for every property P, we have $\mathrm{P} \in \mathcal{CO}(\lambda)$ whenever $\mathrm{P} \in \mathcal{CO}(\zeta)$.*

Proof. If $\zeta \subseteq \lambda$ then also for the corresponding sets of truth tables it holds that $\mathrm{S}_\zeta \subseteq \mathrm{S}_\lambda$. If a property $\mathrm{P} \in \mathcal{CO}(\zeta)$, then $\mathrm{qTimeBB}_\epsilon(\mathrm{P}\,|_{S_\zeta}) \geq \Omega(\mathrm{Q}_\epsilon)(\mathrm{P})$ also implies $\mathrm{qTimeBB}_\epsilon(\mathrm{P}\,|_{S_\lambda}) \geq \mathrm{qTimeBB}_\epsilon(\mathrm{P}\,|_{S_\zeta})$ as S_λ is a superset of S_ζ. Therefore, $\mathrm{P} \in \mathcal{CO}(\lambda)$. ◀

▶ **Theorem 9.** *If there exists a property P such that $\mathrm{Q}_\epsilon(\mathrm{P}) = \widetilde{\omega}(\sqrt{N})$ and P is γ-compression oblivious, and $\mathrm{P} \in \mathsf{polyL}(N)$, then $\mathsf{P} \neq \mathsf{PSPACE}$. Here $N = 2^n$ and γ represents the set of poly-sized circuits on n input variables.*

Proof. By way of contradiction, assume $\mathsf{P} = \mathsf{PSPACE}$. We are given a promise that the circuit C to which we have black-box access[15] to is in the set γ, where γ is the set of poly-sized circuits on n input variables. Note that if we would have direct access to the input, instead of black-box access, we can easily solve the problem in polynomial time using the assumption $\mathsf{P} = \mathsf{PSPACE}$.

[15] By black-box access we mean that for any input $x \in \{0,1\}^n$ we can compute $C(x)$.

Using a simplified version of the algorithm for the oracle identification problem [10, 33] we can extract a compressed form of the entire input, effectively going from black-box access back to white-box access, from the set γ using only $\widetilde{O}(\sqrt{N})$ queries. The initial query-efficient algorithm is as follows:

1. Define an $N = 2^n$ bit majority string $m = m_1 m_2 ... m_N$ where $m_i = 1$ if the majority of circuits in γ have 1 in their i^{th} bit of their truth table, else $m_i = 0$.
2. Check whether there exists an index j such that the truth table of circuit C disagrees with m at j. Using Grover's algorithm on the implied string $\mathrm{tt}(C) \oplus m$ this can be achieved using $O(\sqrt{N})$ quantum queries to $\mathrm{tt}(C)$.
   If there is no disagreement, then the string m is the truth table of circuit C and without having to further query C, one can go through all the circuits in γ and compute their respective truth tables to identify C. Using the $\mathsf{P} = \mathsf{PSPACE}$ assumption, this can be done in $\mathsf{poly}(n)$ (classical) time.
3. In the case of a disagreement, remove from γ all the circuits that disagreed with $\mathrm{tt}(C)$ at index j, which, by definition of m, means at least half of the elements from γ are removed.

Repeat these steps until there is no disagreement or until $|\gamma| = 1$. Given that γ initially contained all the poly-sized circuits on n input variables. This whole algorithm requires $O(\sqrt{N} \log |\gamma|) = \widetilde{O}(\sqrt{N})$ quantum queries. Using the $\mathsf{P} = \mathsf{PSPACE}$ assumption, we can implement the same algorithm in $\widetilde{O}(\sqrt{N})$ quantum time as follows.

At any point of the algorithm we have to be able to query the index $i \in [N]$ of $\mathrm{tt}(C)$ and the i^{th} bit of the majority string m at that stage, where the majority string keeps changing every time we update the set γ. Querying any index of $\mathrm{tt}(C)$ is straight forward. On the other hand, the string m is too long to efficiently write down, but will have to be defined implicitly. To enable query access to m, the algorithm will maintain a list of tuples recording previous found positions where the truth table of C differed from the most common values: $\{(i, a_i) \mid i \in [N]$ is the index where there was a disagreement and a_i is the value of the i^{th} bit of $\mathrm{tt}(C)\}$. Now, given such a list, it takes $\mathsf{poly}(n)$ space to compute the current value m_i of the majority string at point i: simply iterate over all elements in the original circuit class up to $\mathsf{poly}(n)$ size, check whether the current circuit D is consistent with the list of previous queries, and then keep tally of $D(i)$. Now we can use the $\mathsf{P} = \mathsf{PSPACE}$ assumption to translate this to a hypothetical algorithm which takes $\mathsf{poly}(n)$ time.

Since $O(\sqrt{N})$ queries suffice to find a single disagreement between $\mathrm{tt}(C)$ and the majority string m at any stage, that means a disagreement (if any) can be found in $\widetilde{O}(\sqrt{N})$ quantum time. Given that there are only $\mathsf{poly}(n)$ such stages, that means we have found the compressed form of circuit C from the set of poly-sized circuits in $\widetilde{O}(\sqrt{N})$ time.

We now have the access to the compressed input of length $n^{O(1)}$. As the property $\mathrm{P} \in \mathsf{polyL}(N)$, we can directly compute P in $O((\log N)^{O(1)}) = O(n^{O(1)})$ amount of space, which again translates to $O((\log N)^{O(1)})$ time under the $\mathsf{P} = \mathsf{PSPACE}$ assumption. Therefore, the total number of (quantum) steps taken is $\widetilde{O}(\sqrt{N}) + O((\log N)^{O(1)})$, which is in contradiction to the assumption that P is γ-compression oblivious. ◀

▶ **Theorem 10.** *There exists an oracle relative to which the basic QSETH holds, but any property $P \in \mathsf{polyL}(N)$ for which $\mathrm{Q}_\epsilon(\mathrm{P}) = \widetilde{\omega}(\sqrt{N})$ is not γ-compression oblivious. Here γ consists of all polynomial-sized circuits (with oracle access).*

Proof. We construct the oracle in two steps. We first start with the Quantified Boolean Formula (QBF) problem as oracle, call this oracle A. Since QBF is complete for PSPACE, and since a call to A can itself be simulated in polynomial space, note that $\mathsf{P}^A = \mathsf{BQP}^A = \mathsf{PSPACE}^A$.

Recall the classic oracle from Baker, Gill, and Solovay [12], relative to which $\mathsf{P} \neq \mathsf{NP}$. This construction occasionally hides a single string of a certain length in the oracle, for a very sparse set of lengths, and shows that it is hard for a Turing machine to find the string in time less than 2^n.

This same construction also works or quantum computation: We will construct the oracle B in steps. Take the i-th oracle quantum Turing machine, with access to oracle A, and consider that it makes at most $o(2^{n_i/2})$ queries when given input 1^{n_i}, where $n_i = 2^{n_{i-1}}$. We aim to construct B such that the language

$$L_B = \{1^n \mid \text{The oracle } B \text{ contains a string of length } n\}$$

can not be decided by such a machine. Via lower bounds for unstructured search [16, 19, 8, 15], there has to exist a single-string setting of the oracle at B that makes the i-th machine fail. I.e., either B has a single string of length n_i, or the oracle is empty at n_i. Via the query lower bound of unstructured search, this language requires $2^{n/2}$ quantum time.

The final oracle C is just the direct sum of the oracle A and B:

$$C = \{(i, x) \mid (i = 0 \wedge x \in A) \vee (i = 1 \wedge x \in B)\}.$$

Relative to C, both SETH, as in Conjecture 2, and the basic QSETH are true (where we consider a relativized "basic QSETH" that takes as input circuits which can make oracle queries to C). In particular, satisfiability of the circuit which queries its input to C and outputs the result takes time $2^{n/2}$ to compute for a quantum Turing machine which has oracle access to C (since any hypothetical machine which solves this language faster, would be able to decide the hard language L_B).

Now consider the hardness of computing some property P of a string, for which we only get black box access to this string, and such that it's known that the string is a truth table of a polynomial-sized circuit which has access to oracle C. A quantum computer can first search the part of C that corresponds with B for the hidden string, using Grover's algorithm for unstructured search, taking time $2^{n/2}$. Now, after finding the hidden string, part B of the oracle is no longer relevant since any call to it can be efficiently simulated by a short computation, and therefore the oracle is effectively only a QBF oracle, meaning that after finding the string we effectively have $\mathsf{P} = \mathsf{PSPACE}$ relative to the oracle. The quantum algorithm can next use the A part, using the construction in Theorem 9, to compute the property P in total time $O^*(2^{n/2}) = \widetilde{O}(\sqrt{N})$. Since we assumed that P has query complexity at least $\widetilde{\omega}(\sqrt{N})$, it follows that P is not compression oblivious relative to the oracle. ◀

B Example lower bounds following from the basic QSETH assumption

As examples we will considered the Orthogonal Vectors (OV) and the LCS problem. The OV problem is defined as follows. Given two sets U and V of N vectors, each over $\{0,1\}^d$ where $d = \omega(\log N)$, determine whether there exists a $u \in U$ and a $v \in V$ such that $\Sigma_{l \in [d]} u_l v_l = 0$. In [40], Williams showed that SETH implies the non-existence of a sub-quadratic classical algorithm for the OV problem. In the quantum case the best-known query lower bound is $\Omega(n^{2/3})$, which can be achieved by reducing the 2-to-1 Collision problem to the Orthogonal Vectors problem; however, the known quantum time upper bound is $\widetilde{O}(n)$ [37]. First note that we cannot use Williams' classical reduction directly, since a hypothetical quantum algorithm for OV expects quantum access to the input, and writing down the entire reduction already takes time $2^{n/2}$. Instead, observe that the reduction produces a separate vector for each partial assignment: let $t(n)$ be the time needed to

compute a single element of the output of the reduction, then $t(n) = \mathsf{poly}(n)$, which is logarithmic in the size of the total reduction. Let $N = O^*(2^{n/2})$ be the size of the output of the reduction of [40], for some CNF formula with n variables. Any quantum algorithm that solves OV in time N^α, can solve CNF-SAT in time $t(n)O^*(2^{\alpha n/2}) = O^*(2^{\alpha n/2})$.[16] Assuming AC^0_2-QSETH*, this implies that a quantum algorithm requires time $\tilde{\Theta}(N)$ to solve OV for instances of size N.

The next example we consider is the LCS problem. The LCS problem is defined as follows. Given two strings a and b over an alphabet set Σ, the LCS(a, b) is the length of the longest subsequence common to both strings a and b. A reduction by [3] shows that if LCS of two strings of length $O(n)$ can be computed in time $O(n^{2-\delta})$ for some constant $\delta > 0$, then satisfiability on CNF formulas with n variables and m clauses can be computed in $O(m^{O(1)} \cdot 2^{(1-\frac{\delta}{2})n})$ which would imply that SETH (Conjecture 2) is false. Just like in the Orthogonal Vectors case, we observe that the classical reduction from CNF-SAT to LCS is local, in the sense that accessing a single bit of the exponentially-long reduction output can be done in polynomial time: Every segment of the strings that are an output of the reduction, depend only on a single partial satisfying assignment, out of the $2^{n/2}$ possible partial assignments.

This observation directly lets us use the reduction of [3] to give a quantum time lower bound of $\widetilde{\Omega}(N)$ for the LCS problem, where N here is the length of the inputs to LCS, conditioned on AC^0_2-QSETH*. However, an unconditional quantum query lower bound of $\Omega(N)$ can also be easily achieved by embedding of a problem with high query complexity, such as the majority problem, in an LCS instance.

[16] We use O^* to denote asymptotic complexity ignoring polynomial factors.