UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

QSETH Strikes Again

Finer Quantum Lower Bounds for Lattice Problem, Strong Simulation, Hitting Set Problem, and More

Chen, Yanlin; Chen, Yilei; Kumar, Rajendra; Patro, Subhasree; Speelman, Florian

DOI
10.4230/LIPIcs.APPROX/RANDOM.2025.6

Publication date
2025

Document Version
Final published version

Published in
Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques

License
CC BY

Link to publication

Citation for published version (APA):
Chen, Y., Chen, Y., Kumar, R., Patro, S., & Speelman, F. (2025). QSETH Strikes Again: Finer Quantum Lower Bounds for Lattice Problem, Strong Simulation, Hitting Set Problem, and More. In A. Ene, & E. Chattopadhyay (Eds.), *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: APPROX/RANDOM 2025, August 11-13, 2025, Berkeley, CA, USA* Article 6 (Leibniz International Proceedings in Informatics; Vol. 353). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2025.6

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:05 Sept 2026

QSETH Strikes Again: Finer Quantum Lower Bounds for Lattice Problem, Strong Simulation, Hitting Set Problem, and More

Yanlin Chen
University of Maryland (QuICS), College Park, MD, USA

Yilei Chen
Tsinghua University, Beijing, China

Rajendra Kumar
Indian Institute of Technology Delhi, New Delhi, India

Subhasree Patro
Eindhoven University of Technology, The Netherlands

Florian Speelman
University of Amsterdam and QuSoft, The Netherlands

Abstract

Despite the wide range of problems for which quantum computers offer a computational advantage over their classical counterparts, there are also many problems for which the best known quantum algorithm provides a speedup that is only quadratic, or even subquadratic. Such a situation could also be desirable if we *don't want* quantum computers to solve certain problems fast - say problems relevant to post-quantum cryptography. When searching for algorithms and when analyzing the security of cryptographic schemes, we would like to have evidence that these problems are difficult to solve on quantum computers; *but how do we assess the exact complexity of these problems?*

For most problems, there are no known ways to directly prove time lower bounds, however it can still be possible to relate the hardness of disparate problems to show *conditional* lower bounds. This approach has been popular in the classical community, and is being actively developed for the quantum case [1, 15, 14, 7].

In this paper, by the use of the QSETH framework [15] we are able to understand the quantum complexity of a few natural variants of CNFSAT, such as parity-CNFSAT or counting-CNFSAT, and also are able to comment on the non-trivial complexity of approximate versions of counting-CNFSAT. Without considering such variants, the best quantum lower bounds will always be quadratically lower than the equivalent classical bounds, because of Grover's algorithm; however, we are able to show that quantum algorithms will likely not attain even a quadratic speedup for many problems. These results have implications for the complexity of (variations of) lattice problems, the strong simulation and hitting set problems, and more. In the process, we explore the QSETH framework in greater detail and present a useful guide on how to effectively use the QSETH framework.

2012 ACM Subject Classification Theory of computation → Quantum computation theory

Keywords and phrases Quantum conditional lower bounds, Fine-grained complexity, Lattice problems, Quantum strong simulation, Hitting set problem, QSETH

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2025.6

Category APPROX

Related Version *Full Version*: `https://arxiv.org/abs/2309.16431` [17]

Funding *Yanlin Chen*: Most of this work was completed while the author was at Centrum Wiskunde & Informatica (QuSoft).
Yilei Chen: Supported by Tsinghua University startup funding, and Shanghai Qi Zhi Institute Innovation Program SQZ202405.

© Yanlin Chen, Yilei Chen, Rajendra Kumar, Subhasree Patro, and Florian Speelman;
licensed under Creative Commons License CC-BY 4.0

Rajendra Kumar: Supported by the Chandrukha New Faculty Fellowship from IIT Delhi and the Prime Minister Early Career Research Grant from the Anusandhan National Research Foundation (ANRF).
Florian Speelman: Supported by the Dutch Ministry of Economic Affairs and Climate Policy (EZK), as part of the Quantum Delta NL program, and the project Divide and Quantum "D&Q" NWA.1389.20.241 of the program "NWA-ORC", which is partly funded by the Dutch Research Council (NWO).

1 Introduction

A popular classical hardness conjecture known as the Strong Exponential Time-Hypothesis (SETH) says that determining whether an input CNF formula is satisfiable or not cannot be done in $\mathcal{O}(2^{n(1-\delta)})$ time for any constant $\delta > 0$ [27, 28]. Several fine-grained lower bounds based on SETH have been shown since then; see [36, 37] for a summary of many such results. Some of the SETH-hard problems are building blocks for fine-grained cryptography [8, 30]. Besides finding a satisfying assignment, natural variants of the CNFSAT problem include computing the count or the parity of the count of satisfying assignments to a CNF formula – #SETH and $\oplus$SETH conjecture complexity of these problems, respectively. These conjectures are weaker (i.e., more believable) than SETH, and can still be used to show fine-grained hardness of various problems [20].

When considering quantum computation, the SETH conjecture is no longer true, as using Grover's algorithm for unstructured search [24] one can solve the CNFSAT problem in $2^{\frac{n}{2}} \cdot \text{poly}(n)$ time. Aaronson, Chia, Lin, Wang, and Zhang assume this Grover-like quadratic speedup is nearly optimal for CNFSAT and (independent of [15]) initiate the study of quantum fine-grained complexity [1]. However, conjectures such as #SETH or $\oplus$SETH are likely to still hold in the quantum setting because a Grover-like quantum speedup is not witnessed when the task is to compute the total number of satisfying assignments or the parity of this number. This situation can in some cases be exploited to give better quantum lower bounds than one would get from the conjectured quantum lower bound for the vanilla CNFSAT problem. This makes it at least as relevant (if not more) to study variants of CNFSAT and their implications in the quantum setting, as has been done classically. In fact, motivated by this exact observation, Buhrman, Patro, and Speelman [15] introduced a framework of Quantum Strong Exponential-Time Hypotheses (QSETH) as quantum analogues to SETH, with a striking feature that allows one to "technically" unify conjectures such as quantum analogues of $\oplus$SETH, #SETH, maj-SETH, etc. under one *umbrella* conjecture.

The QSETH framework

In their framework, Buhrman et al. consider the problem in which one is given a formula or a circuit representation of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and is asked whether a property $\mathrm{P} := \left(\mathrm{P}_n\right)_{n\in\mathbb{N}}$ where $\mathrm{P}_n : \mathcal{D} \subseteq \{0,1\}^{2^n} \to \{0,1\}$ on the truth table[1] of this formula evaluates to 1. They conjectured that when the circuit representation is obfuscated enough then for *most* properties P (that are compression-oblivious properties as we will see in Definition 1), the time taken to compute P_n on the truth table of poly(n)-sized circuits is lower bounded by $\mathsf{Q}(\mathrm{P}_n)$, i.e. the 1/3-bounded error quantum query complexity of P_n, on all bit strings of length 2^n.

[1] Truth table of a formula ϕ on n variables, denoted by $tt(\phi)$, is a 2^n bit string derived in the following way $tt(\phi) = \bigcirc_{a\in\{0,1\}^n} \phi(a)$; the symbol $\circ$ denotes concatenation.

It is not hard to see that such a conjecture cannot be true for *all* properties. In principle, one can construct properties for which the above statement would not hold. For instance, consider a property P that is trivial on truth tables of small formulas (i.e., poly(n) size) but complicated on formulas of longer length. These kinds of properties are likely to have very high quantum query complexity, but in reality, it will be trivial to compute such a P_n of P on formulas of poly(n) size. In order to prevent such scenarios the authors in [15] introduce the notion of compression-oblivious properties which they believe encompasses most of the naturally occurring properties. See Sections 2.2 and 2.3 of [15] for a detailed discussion on this topic and also see [18] for some new observations about the notion of compression-oblivious properties. To give a bit of intuition, first consider the set of truth tables corresponding to the set of poly(n) size formulas on n variables and consider the set of all the 2^n bit strings. Compression-oblivious properties are those properties for which one cannot save in computational time to compute them on a string from the former set in comparison to computing the same property on strings from the latter set. More formally,[2]

▶ **Definition 1** ($\mathsf{AC}^0_{2,p}$- and AC^0_2-Compression-Oblivious Properties [15, 18])**.** *Let $p \in \mathbb{N}$. We say a property P is* $\mathsf{AC}^0_{2,p}$-compression-oblivious[3]*, denoted by $P \in \mathcal{CO}(\mathsf{AC}^0_{2,p})$, if for every constant $\delta > 0$, for every quantum algorithm $\mathcal{A}$ that computes P in the black-box setting, $\forall n' \in \mathbb{N}, \exists n \geq n'$ and $\exists$ a set $L = \{L^1, L^2, \ldots\} \subseteq \mathsf{AC}^0_{2,p}$ of "hard languages", such that $\forall$ circuit families $\{C^1_{n''}\}_{n'' \in \mathbb{N}}$ corresponding to L^1, $\forall$ circuit families $\{C^2_{n''}\}_{n'' \in \mathbb{N}}$ corresponding to L^2, …, $\mathcal{A}$ uses at least $\mathsf{Q}(P_n)^{1-\delta}$ quantum time on at least one of the inputs in $\{C^i_n\}_{i \in [|L|]}$.*

In particular, we say a property P is AC^0_2-compression-oblivious *(denoted by $P \in \mathcal{CO}(\mathsf{AC}^0_2)$) if $\exists p \in \mathbb{N}$ such that $P \in \mathcal{CO}(\mathsf{AC}^0_{2,p})$.*

With that, we can conjecture the following using the QSETH framework by [15].

▶ **Conjecture 2** (AC^0_2-QSETH, consequences of [15])**.** *Let P be an* AC^0_2-compression-oblivious *property. The AC^0_2-QSETH conjecture states that $\exists p \in \mathbb{N}$, such that for every constant $\delta > 0$, for every quantum algorithm $\mathcal{A}$ that computes P in the white-box setting, $\forall n' \in \mathbb{N}, \exists n \geq n'$, $\exists$ a set $L = \{L^1, L^2, \ldots\} \subseteq \mathsf{AC}^0_{2,p}$, $\forall$ circuit families $\{C^1_{n''}\}_{n'' \in \mathbb{N}}$ corresponding to L^1, $\forall$ circuit families $\{C^2_{n''}\}_{n'' \in \mathbb{N}}$ corresponding to L^2, …, the algorithm $\mathcal{A}$ uses at least $\mathsf{Q}(P_n)^{1-\delta}$ quantum time on at least one of the inputs in $\{C^i_n\}_{i \in [|L|]}$.*

Informally, the notion of AC^0_2-compression-obliviousness captures properties whose query complexity is a lower bound for the time complexity to compute the property even for truth tables of small CNF/DNF formulas. And, the AC^0_2-QSETH conjecture states that having access to this succinct representation of the truth table, i.e., the description of the formula itself, must not help towards improving the computation time in computing these properties. The terms *black-box* and *white-box* setting are used to highlight this difference - in the former setting (i.e., the black-box setting as stated in Definition 1) even though the input is a CNF/DNF formula, the algorithm is only allowed to evaluate the formula on required inputs and has no access to the description of the inputted CNF/DNF formula, and, in the latter setting (i.e., the white-box setting as stated in Conjecture 2) the algorithm is allowed to use the description as well.

[2] The definition of compression-oblivious properties as stated in this paper is a more formal version of its original definition in [15]. Also, see [18] for a discussion on this topic.

[3] We say a language $L \in \mathsf{AC}^0_{2,p}$ iff there exists a family of Boolean circuits $\{C_n\}_{n \in \mathbb{N}}$ corresponding to L such that $\forall n$, C_n has depth at most 2 and circuit size $|C_n| \leq n^p$.

In comparison to the original QSETH paper (by Buhrman et al. [15]) where the framework was introduced and applied to a more complex class of formulas,[4] this paper instead serves as a guide to using QSETH for the lowest level of formulas, i.e., poly-sized CNF and DNF formulas, in a more elaborate fashion.

Table 1 Overview of conditional lower bounds for variants of CNFSAT and k-SAT. The variable $\hat{h}$ in the above table is an arbitrary natural number satisfying $\gamma\hat{h} \geq 1$. Our results hold for the multiplicative factor $\gamma \in \left[\frac{1}{2^n}, 0.4999\right)$ and the additive error $\Delta \in [1, 2^n)$.

Problem	Variants	Quantum lower bound	Reference
CNFSAT	Vanilla	$2^{\frac{n}{2}-o(n)}$	Corollary 6
	Parity	$2^{n-o(n)}$	Corollary 14
	Majority	$2^{n-o(n)}$	Corollary 16
	Strict Majority	$2^{n-o(n)}$	Corollary 16
	Count	$2^{n-o(n)}$	Theorem 13
	Count_q	$2^{n-o(n)}$	Corollary 15
	Δ-Additive error	$\left(\sqrt{\frac{2^n}{\Delta}} + \frac{\sqrt{\hat{h}(2^n-\hat{h})}}{\Delta}\right)^{1-o(1)}$	Theorem 21
	γ-Multiplicative factor	$\left(\frac{1}{\gamma}\sqrt{\frac{2^n-\hat{h}}{\hat{h}}}\right)^{1-o(1)}$	Corollary 26
k-SAT $k = \Theta(\log(n))$	Vanilla	$2^{\frac{n}{2}-o(n)}$	Section 5, [1]
	Parity	$2^{n-o(n)}$	Corollary 28
	Count	$2^{n-o(n)}$	Corollary 28
	Count_q	$2^{n-o(n)}$	Corollary 28
	γ-Multiplicative factor	$\left(\frac{1}{\gamma}\sqrt{\frac{2^n-\hat{h}}{\hat{h}}}\right)^{1-o(1)}$	Corollary 29

Summary and technical overview

In this paper, we use the QSETH framework (or precisely, AC^0_2-QSETH) to "generate" natural variations of QSETH such as $\oplus$QSETH, #QSETH, maj-QSETH, etc., which could (arguably) already be acceptable standalone conjectures in the quantum setting, and study some of their interesting implications. Additionally, we also use the QSETH framework to prove quantum lower bounds for *approximately* counting the number of satisfying assignments to CNF formulas, a problem whose complexity has been of interest in the classical setting [22]; we study its quantum implications. See Section 3 for details. Proof of this result follows from a more detailed exploration of the QSETH framework than what was required in the original paper. Thus, as another contribution of this paper, we present a useful guide on how to effectively use the QSETH framework. Here we carefully summarize our contributions and technical overview below:

- We zoom into Buhrman et al.'s QSETH framework at the lowest-level formula class, i.e., the class of polynomial-size CNFs and DNFs, and use it to study the quantum complexity of variations of CNFSAT problems. The QSETH framework is quite general which also makes it not entirely trivial to use it thus, we present a useful guide on how to

[4] The authors in [15] extensively used QSETH framework for branching programs or equivalently NC circuits to show non-trivial lower bounds for edit distance and longest common subsequence problems.

Table 2 Overview of lower bounds based on AC^0_2-QSETH - all the relevant details such as the definitions, reductions, etc., can be found in the full version of this paper [17]. The variable $\hat{h}$ in the above table is an arbitrary natural number satisfying $\gamma\hat{h} \geq 1$. Our results hold for the multiplicative factor $\gamma \in \left[\frac{1}{2^n}, 0.4999\right)$ and the additive error $\Delta' \in [\frac{1}{2^n}, 1)$.

Problem	Variants	Quantum lower bound	Reference
Strong Simulation	Exact (with n bits precision)	$2^{n-o(n)}$	[17, Theorem 4.2]
	Exact (with 2 bits precision)	$2^{n-o(n)}$	[17, Corollary 4.3]
	Δ-Additive error	$\left(\sqrt{\frac{1}{\Delta'}} + \frac{\sqrt{\hat{h}(2^n-\hat{h})}}{2^n\Delta'}\right)^{1-o(1)}$	[17, Corollary 4.5]
	γ-Multiplicative factor	$\left(\frac{1}{\gamma}\sqrt{\frac{2^n-\hat{h}}{\hat{h}}}\right)^{1-o(1)}$	[17, Theorem 4.7]
CVP_p for $p \notin 2\mathbb{Z}$		$2^{\frac{n}{2}-o(n)}$	[17, Section 5]
Lattice Counting (for non-even norm)	Vanilla	$2^{n-o(n)}$	[17, Corollary 5.6]
	γ-Multiplicative factor	$\left(\frac{1}{\gamma}\sqrt{\frac{2^n-\hat{h}}{\hat{h}}}\right)^{1-o(1)}$	[17, Corollary 5.7]
	q-count	$2^{n-o(n)}$	[17, Corollary 5.6]
Orthogonal Vectors	Vanilla	$n^{1-o(1)}$	[1, 15]
	Parity	$n^{2-o(1)}$	[17, Corollary 6.8]
	Count	$n^{2-o(1)}$	[17, Corollary 6.8]
	γ-Multiplicative factor	$\left(\frac{1}{\gamma}\sqrt{\frac{n^2-\hat{h}}{\hat{h}}}\right)^{1-o(1)}$	[17, Corollary 6.8]
Hitting Set	Vanilla	$2^{\frac{n}{2}-o(n)}$	[17, Corollary 6.4]
	Parity	$2^{n-o(n)}$	[17, Corollary 6.4]
	Count	$2^{n-o(n)}$	[17, Corollary 6.4]
	γ-Multiplicative factor	$\left(\frac{1}{\gamma}\sqrt{\frac{2^n-\hat{h}}{\hat{h}}}\right)^{1-o(1)}$	[17, Corollary 6.4]
$\oplus$Set Cover		$2^{n-o(n)}$	[17, Corollary 6.11]

effectively use the AC^0_2-QSETH conjecture, for e.g., what lemmas need to be proved and what assumptions are needed to be made in order to understand the quantum complexity of CNFSAT and its variants; see Figure 1.

- We can categorise the several variants of CNFSAT in two ways. First classification can be done by the width of the CNF formulas, i.e., k-CNFs versus CNFs of unbounded clause length. Second classification is made with the choice of the property of the truth table one desires to compute. See the summary of complexity all CNFSAT variants and their respective quantum time lower bounds in Table 1 and see below for the overview of the techniques used.
  - To prove the quantum time lower bounds for the property variants of CNFSAT problem we invoke AC^0_2-QSETH (Conjecture 2). But, AC^0_2-QSETH conjectures the hardness of properties on a set of CNF and DNF formulas. For properties like count, parity, majority, etc., it easily follows from De Morgan's laws that these properties are equally hard on both CNF and DNF formulas. However, such arguments no longer hold when the properties are approximate variants of count for which we give nontrivial proofs; see Sections 3.1.2 and 3.1.3.
  - Additionally, we also use AC^0_2-QSETH to understand quantum complexity of k-SAT and its property variants. Firstly we study the classical reduction from CNFSAT to k-SAT given by [16] and observe that the $2^{\frac{n}{2}}$ quantum lower bound for k-SAT, for $k = \Theta(\log n)$, follows from the quantum lower bound of CNFSAT. Moreover, we make an important observation that this reduction by [16] is count-preserving and can be used to conclude lower bounds for other counting variants of k-SAT. See Table 1.

- Having (somewhat) understood the complexities of these variants of CNFSAT, we then prove conditional quantum time lower bounds for lattice problem, strong simulation, orthogonal vectors, set cover, hitting set, and their respective variants; see Table 2.
  - The quantum $2^{\frac{n}{2}}$ time lower bound we present for CVP_p (for $p \notin 2\mathbb{Z}$) follows from a reduction from k-SAT to CVP_p by [11, 2] and from the hardness result of k-SAT we present. Though such a result would also trivially follow by using Aaronson et al.'s version of QSETH, we stress that our hardness result of k-SAT is based on basic-QSETH which is a more believable conjecture.[5]
  - Additionally, we also discuss the quantum complexity of the lattice counting problem (for non-even norm). We present a reduction, using a similar idea of [11], from $\#k$-SAT to the lattice counting problem and we show a 2^n time quantum lower bound for the latter when $k = \Theta(\log n)$. As mentioned earlier, we get a 2^n time quantum lower bound for $\#k$-SAT, when $k = \Theta(\log n)$, using AC^0_2-QSETH.
  - As another application to the bounds we get from the property variants of CNFSAT we look at the strong simulation problem. It was already established by [19, 35] that strong simulation of quantum circuit is a #P-hard problem but in this work we give exact lower bounds for the same. Additionally, using the lower bounds of approximate counts of CNFSAT we are able to shed light on how hard it is to quantumly solve the strong simulation problem with additive and multiplicative error approximation.
  - Last but not least, we are also able to use the lower bounds for the property variants of CNFSAT to give interesting lower bounds for orthogonal vectors, hitting set problem and their respective variants. See Section 6 for more details.

  Our motivation to study the worst-case complexities of counting versions of these problems stems from the fact that worst-case complexity of counting versions of problems have been used in past to understand average-case complexity of other related problems. And, computational problems that have high average-case complexities usually find their place in cryptographic primitives. For example, Goldreich and Rothblum in [23] present a worst-case to average-case reduction for counting t-cliques in graph and use the average-case hardness result towards constructing an interactive proof system. Another such example is that of the OV problem - Ball et al. in [9] use the worst-case hardness of the counting variant of OV to first prove average-case hardness of evaluating low-degree polynomials which they use towards a Proofs of Work (PoW) protocol. Furthermore, Buhrman et al. in [15] observed that this PoW protocol in combination with QSETH ensures that the quantum provers also require the same time as the classical provers.[6]

Related work

Our paper is a follow-up work to the original QSETH paper by [15]; also the list of problems for which we show lower bounds does not overlap with the problems studied in [15]. A basic version of QSETH was also introduced by Aaronson et al. [1] where they primarily used it to study the quantum complexity of closest pair and bichromatic pair problems; they also discuss the complexity of the (vanilla version of) orthogonal vector problem. Prior to this work, a quantum hitting-set conjecture was proposed and its implications were discussed in

[5] If basic-QSETH from Buhrman et al.'s framework is false then Aaronson et al.'s QSETH is also false, but the implication in the other direction is not obvious.

[6] Note that counting the number of OV pairs on *average* has a fast algorithm [21] so a worst-case to average-case reduction for counting OV is not possible under standard fine-grained complexity assumptions.

Schoneveld's bachelor thesis [33], but their definition of hitting set is different from ours. In our work, we observe that the parsimonious reduction from CNFSAT to hitting set given by [20] is easily quantizable, using which we get a QSETH-based lower bound. Recently, Huang et al. [26] showed a significant barrier to establishing fine-grained quantum reductions from k-SAT to lattice problems in the Euclidean norm. In contrast, our work focuses on lattice problems in the ℓ_p norm, where p is not an even integer.

2 Preliminaries

2.1 Quantum query complexity of Boolean properties

We start with defining Boolean properties, and then we will define the bounded-error quantum complexity of computing those properties.

▶ **Definition 3** (Property). *A Boolean property (or just property) is a sequence* $P := (P_n)_{n\in\mathbb{N}}$ *where each* P_n *is a set of Boolean functions defined on* 2^n *variables.*

The (bounded-error) quantum query complexity is defined only in a non-uniform setting, therefore, it is defined for P_n for every $n \in \mathbb{N}$ instead of directly defining for P. A quantum query algorithm $\mathcal{A}$ for $\mathrm{P}_n : \{0,1\}^{2^n} \to \{0,1\}$, on an input $x \in \{0,1\}^{2^n}$ begins in a fixed initial state $|\psi_0\rangle$, applies a sequence of unitaries $U_0, O_x, U_1, O_x, \ldots, U_T$, and performs a measurement whose outcome is denoted by z. Here, the initial state $|\psi_0\rangle$ and the unitaries $U_0, U_1, \ldots, U_T$ are independent of the input x. The unitary O_x represents the "query" operation, and maps $|i\rangle|b\rangle$ to $|i\rangle|b + x_i \bmod 2\rangle$ for all $i \in [2^n] - 1$. We say that $\mathcal{A}$ is a 1/3-bounded-error algorithm computing P_n if for all x in the domain of P_n, the success probability of outputting $z = \mathrm{P}_n(x)$ is at least 2/3. Let $\mathrm{cost}(\mathcal{A})$ denote the number of queries $\mathcal{A}$ makes to O_x throughout the algorithm. The 1/3-bounded-error quantum query complexity of P_n, denoted by $\mathsf{Q}(\mathrm{P}_n)$, is defined as $\mathsf{Q}(\mathrm{P}_n) = \min\{\mathrm{cost}(\mathcal{A}) : \mathcal{A} \text{ computes } \mathrm{P}_n \text{ with error probability } \leq 1/3\}$.

2.2 CNFSAT and k-SAT

A Boolean formula over variables $x_1, \ldots, x_n$ is in CNF form if it is an AND of OR's of variables or their negations. More generally, a CNF formula has the form

$$\bigwedge_i \left(\bigvee_j v_{ij} \right)$$

where v_{ij} is either x_k or $\neg x_k$. The terms v_{ij} are called *literals* of the formula and the disjunctions $\bigvee_j v_{ij}$ are called its clauses. A k-CNF is a CNF formula in which all clauses contain at most k literals (or the clause width is at most k). Note that when $k > n$, then clauses must contain duplicate or trivial literals (for example, $x_k \vee \neg x_k$ and $x_k \vee x_k$), therefore we can assume without loss of generality that k is at most n. A DNF is defined in the exact same way as CNF, except that it is an OR of AND's of variables or their negations, that is, a DNF formula has the form $\bigvee_i \left(\bigwedge_j v_{ij} \right)$. We also define computational problems k-SAT and CNFSAT.

▶ **Definition 4** (CNFSAT). *Given as input a CNF formula* ϕ *defined on* n *variables, the goal is to determine if* $\exists x \in \{0,1\}^n$ *such that* $\phi(x) = 1$.

▶ **Definition 5** (k-SAT). *Given as input a* k*-CNF formula* ϕ *defined on* n *variables, the goal is to determine if* $\exists x \in \{0,1\}^n$ *such that* $\phi(x) = 1$.

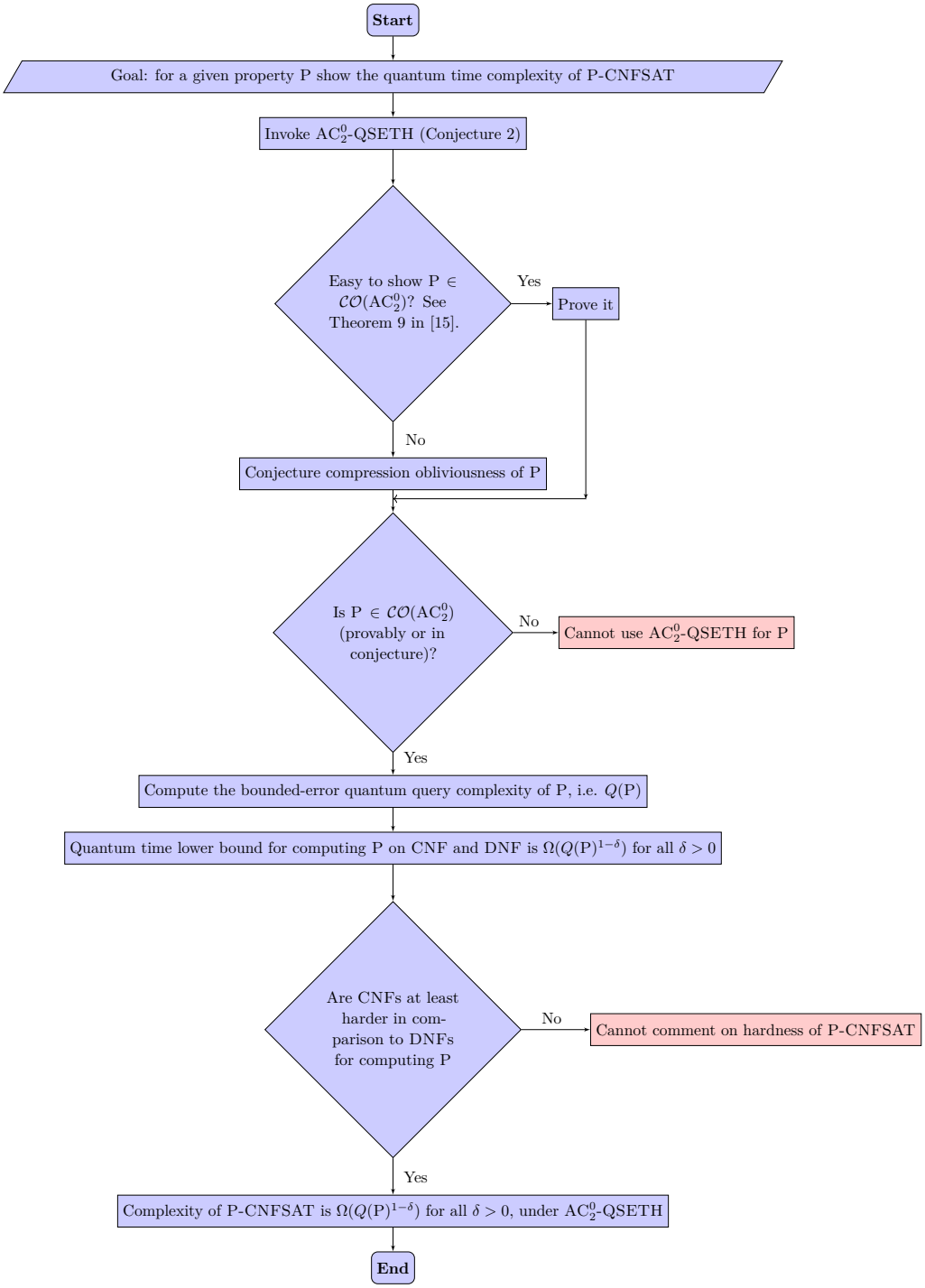

Figure 1 Step-by-step guide on how to use the QSETH framework in a plug-and-play manner to show hardness results for P-CNFSAT. Here P can be any (partial or total) Boolean property of truth tables.

3 Lower bounds for variants of CNFSAT using AC^0_2-QSETH

We will now define several variants of CNFSAT problem and using AC^0_2-QSETH understand the quantum complexities of all of them. The consequences of these hardness results, some of which follow immediately and the rest with some work, will be discussed in Sections 4–6. We begin with some common variants of CNFSAT problem (such as k-SAT) which are also very well studied classically [20]; we do this in Section 3.1.1. And, proceed with some less popular variants (Sections 3.1.2, 3.1.3, and 3.2) but with interesting consequences (presented in Sections 4–6).

3.1 Quantum complexity of CNFSAT and other related problems

We first restate the quantum hardness of CNFSAT before delving into showing hardness results for its other variants. Interestingly, for the property $\mathrm{OR} : \{0,1\}^{2^n} \to \{0,1\}$, where for $x \in \{0,1\}^{2^n}$ we define $\mathrm{OR}(x) = 1$ if $|x| \geq 1$ and $\mathrm{OR}(x) = 0$ whenever $|x| = 0$, we can explicitly prove that $\mathrm{OR} \in \mathcal{CO}(\mathsf{AC}^0_2)$ [15, 18]. Also, note that computing OR on truth tables of DNF formulas of $\mathrm{poly}(n)$ length can be computed in $\mathrm{poly}(n)$ time. Hence, using AC^0_2-QSETH we can recover the following Basic-QSETH conjecture.

▶ **Corollary 6** (Basic-QSETH [15]). *For each constant $\delta > 0$, there exists $c > 0$ such that there is no bounded-error quantum algorithm that solves CNFSAT (even restricted to formulas with $m \leq cn^2$ clauses) in $\mathcal{O}\left(2^{\frac{n(1-\delta)}{2}}\right)$ time, unless AC^0_2-QSETH (Conjecture 2) is false.*

Note that Aaronson et al. [1] directly conjecture the above statement, while in our case the above conjecture is implied by AC^0_2-QSETH (Conjecture 2), and we will show how AC^0_2-QSETH can imply other conjectured time lower bound for variants of CNFSAT problems in this subsection.

3.1.1 Quantum complexity of #CNFSAT, ⊕CNFSAT, $\#_q$CNFSAT and maj-CNFSAT

To give conditional quantum lower bounds for variants of CNFSAT, we should understand their corresponding quantum query lower bound (on the 2^n-bit truth table). Here we introduce the properties that correspond to those popular variants of CNFSAT (which will be defined later.)

▶ **Definition 7.** *Let $|x| = |\{i : x_i = 1\}|$ denote the Hamming weight of N-bit binary string x. We here list some properties defined on binary strings.*

1. *count: Let $\textsc{count} : \{0,1\}^N \to [N] \cup \{0\}$ be the non-Boolean function defined by $\textsc{count}(x) = |x|$.*
2. *parity: Let $\textsc{parity} : \{0,1\}^N \to \{0,1\}$ be the Boolean function defined by $\textsc{parity}(x) = |x| \bmod 2$.*
3. *$\textsc{count}_q$: Let q be an integer and let $\textsc{count}_q : \{0,1\}^N \to [q] - 1$ be the non-Boolean function defined by $\textsc{count}_q(x) = |x| \mod q$.*
4. *majority: Let $\textsc{majority} : \{0,1\}^N \to \{0,1\}$ be the Boolean function defined by*

$$\textsc{majority}(x) = \begin{cases} 1 & \text{if } |x| \geq \frac{N}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

And, there is also the following function almost similar to majority.

5. *st-MAJORITY: Let st-MAJORITY* $: \{0,1\}^N \to \{0,1\}$ *be the Boolean function with*

$$\textit{st-MAJORITY}(x) = \begin{cases} 1 & \text{if } |x| > \frac{N}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

Here, we define variants of CNFSAT corresponding to the above-mentioned properties.

▶ **Definition 8** (variants of CNFSAT). *Let* $|\phi| = \{y \in \{0,1\}^n : \phi(y) = 1\}$ *denote the Hamming weight of the truth table of* ϕ*. The following lists five variants of CNFSAT:*

1. *#CNFSAT: Given a CNF formula* ϕ *on* n *input variables, output* $|\phi|$*.*
2. $\oplus$*CNFSAT: Given a CNF formula* ϕ *on* n *input variables, output* $|\phi| \bmod 2$*.*
3. $\#_q$*CNFSAT: Given a CNF formula* ϕ *on* n *input variables and an integer* $q \in [2^n] \setminus \{1\}$*, output* $|\phi| \bmod q$*.*
4. *MAJ-CNFSAT: Given a CNF formula* ϕ *on* n *input variables, output* 1 *if* $|\phi| \geq 2^n/2$ *(else output* 0*).*
5. *st-MAJ-CNFSAT: Given a CNF formula* ϕ *on* n *input variables, output* 1 *if* $|\phi| > 2^n/2$ *(else output* 0*).*

Now again, we use the quantum query lower bound for P whenever we want to discuss the time complexity of P-CNFSAT as in the QSETH framework by [15]. Therefore, immediately after the definitions for variants of CNFSAT (with respect to property P), we will introduce the corresponding bounded-error quantum *query* lower bound for computing P, and then conjecture the *time* lower bound for P-CNFSAT (P variant CNFSAT) using this query lower bound. After that, we can use lower bounds for those variants of CNFSAT to understand the quantum complexity of (variants of) k-SAT. We include the quantum query lower bounds for those properties for completeness.

▶ **Lemma 9** ([10]). *The bounded-error quantum query complexity for* COUNT*,* PARITY*,* MAJORITY *and st-*MAJORITY *on inputs of* N*-bit Boolean strings is* $\Omega(N)$*.*

Proof. [10] showed that the bounded-error quantum query complexity of a (total) Boolean function $f : \{0,1\}^N \to \{0,1\}$, denoted by $\mathsf{Q}(f)$ is lower bounded by $1/2$ of the degree of a minimum-degree polynomial p that approximates f on all $X \in \{0,1\}^N$, i.e., $|p(X) - f(X)| \leq 1/3$; let us denote this approximate degree by $\widetilde{deg}(f)$. Another important result by Paturi [32] showed that if f is a non-constant, symmetric[7] and total Boolean function on $\{0,1\}^N$ then $\widetilde{deg}(f) = \Theta(\sqrt{N(N - \Gamma(f))})$ where $\Gamma(f) = \min\{|2k - N + 1| : f_k \neq f_{k+1} \text{ and } 0 \leq k \leq N-1\}$ and $f_k = f(X)$ for $|X| = k$.

Using the above two results we can show the following:

1. $\Gamma(\text{PARITY}) = 0$ for odd N and $\Gamma(\text{PARITY}) = 1$ whenever N is even. Hence $\mathsf{Q}(\text{PARITY}) = \Omega(N)$.[8]
2. Similar to the above item $\Gamma(\text{MAJORITY}) = \Gamma(\textit{st-MAJORITY}) = 0$ for odd N and $\Gamma(\text{MAJORITY}) = \Gamma(\textit{st-MAJORITY}) = 1$ otherwise. Hence, $\mathsf{Q}(\text{MAJORITY}) = \Omega(N)$ and $\mathsf{Q}(\textit{st-MAJORITY}) = \Omega(N)$.
3. Any of the above three properties can be computed from COUNT. Hence, $\mathsf{Q}(\text{COUNT}) = \Omega(N)$. ◀

[7] A symmetric Boolean function $f : \{0,1\}^N \to \{0,1\}$ implies $f(X) = f(Y)$ for all X, Y whenever $|X| = |Y|$.

[8] One can actually immediately give $\mathsf{Q}(\text{PARITY}) \geq N/2$ by an elementary degree lower bound without using Paturi's result.

▶ **Lemma 10.** *Let $q \in [3, \frac{N}{2}]$ be an integer and $\textsc{count}_q : \{0,1\}^N \to [q]-1$ be the function defined by $\textsc{count}_q(x) = \textsc{count}(x) \mod q$. Then $\mathsf{Q}(\textsc{count}_q) = \Omega(\sqrt{N(N-2q+1)})$.*

Proof. Let $\textsc{dec-count}_q$ be a decision version of the $\textsc{count}_q$ defined for all $x \in \{0,1\}^N$ as

$$\textsc{dec-count}_q(x) = \begin{cases} 1, & \text{if } \textsc{count}_q(x) = q-1, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

When the function is non-constant and symmetric then one can use Paturi's theorem to characterize the approximate degree of that function [32]. It is easy to see that $\textsc{dec-count}_q$ is a non-constant symmetric function. Combining both these results we get that $\mathsf{Q}(\textsc{dec-count}_q) = \Omega(\sqrt{N(N-\Gamma(\textsc{dec-count}_q))})$.

We now compute the value of $\Gamma(\textsc{dec-count}_q)$. For any symmetric Boolean function $f : \{0,1\}^N \to \{0,1\}$ the quantity $\Gamma(f)$ is defined as $\Gamma(f) = \min_k\{|2k-N+1|\}$ such that $f_k \neq f_{k+1}$ and $f_k = f(x)$ for $|x| = k$ with $1 \leq k \leq N-1$. It is easy to see that $\textsc{dec-count}_q(x) = 1$ only for x with Hamming weight $|x| = rq-1$ where r is an integer and $\textsc{dec-count}_q(x) = 0$ elsewhere. Let r' be the integer such that $r'q - 1 \leq \frac{N}{2} \leq (r'+1)q - 1$ then the k minimizing $\Gamma(\textsc{dec-count}_q)$ is either $r'q-1$ or $(r'+1)q-1$. This implies that $\Gamma(\textsc{dec-count}_q) \leq 2q-1$, which in turn implies that $N - \Gamma(\textsc{dec-count}_q) \geq N - 2q + 1$. Therefore, $\mathsf{Q}(\textsc{dec-count}_q) = \Omega(\sqrt{N(N-2q+1)})$.

As one can compute $\textsc{dec-count}_q$ using an algorithm that computes $\textsc{count}_q$, we therefore have $\mathsf{Q}(\textsc{count}_q) = \Omega(\sqrt{N(N-2q+1)})$. ◀

As we don't yet know how to prove compression-obliviousness of properties with high query complexities (Theorem 9 in [15]) we instead conjecture that COUNT, PARITY, MAJORITY and *st*-MAJORITY are compression oblivious for poly-sized CNF and DNF formulas. We think it is reasonable to make this conjecture since it will falsify certain commonly-used cryptography assumptions if those properties are not compression oblivious. See [18] for a discussion on this topic.

▶ **Conjecture 11.** *The following properties*

1. $\textsc{parity} : \{0,1\}^{2^n} \to \{0,1\}$,
2. $\textsc{count}_q : \{0,1\}^{2^n} \to [q-1] \cup \{0\}$ *where* $2 < q \leq 2^{n-1}$,
3. $\textsc{majority} : \{0,1\}^{2^n} \to \{0,1\}$, *and*
4. *st*-$\textsc{majority} : \{0,1\}^{2^n} \to \{0,1\}$

stated in Definition 7 are in $\mathcal{CO}(AC^0_2)$.

▶ **Corollary 12.** *Let AC^0_2 denote the class of* poly(n) *sized CNF and DNF formulas on n input variables. If any one item of Conjecture 11 is true then the property $\textsc{count} : \{0,1\}^{2^n} \to [2^n] \cup \{0\}$ is in $\mathcal{CO}(AC^0_2)$.*

We can now invoke AC^0_2-QSETH (as mentioned in Conjecture 2) to prove the quantum hardness for #CNFSAT, $\oplus$CNFSAT, MAJ-CNFSAT and st-MAJ-CNFSAT.

▶ **Theorem 13** (#QSETH). *For each constant $\delta > 0$, there exists $c > 0$ such that there is no bounded-error quantum algorithm that solves #CNFSAT (even restricted to formulas with $m \leq cn^2$ clauses) in $\mathcal{O}(2^{n(1-\delta)})$ time, unless AC^0_2-QSETH (Conjecture 2) is false or $\textsc{count} \notin \mathcal{CO}(AC^0_2)$ (i.e., each item of Conjecture 11 is false).*

Proof. By way of contradiction, let us assume that there exists a bounded-error quantum algorithm $\mathcal{A}$ that solves #CNFSAT on n variables (and on m clauses with some $m \leq cn^2$) in $\mathcal{O}(2^{n(1-\delta)})$ time for some $\delta > 0$. Then given a circuit $C \in \text{AC}^0_2$ we do one of the following:

- if C is a poly-sized CNF formula then we use the algorithm $\mathcal{A}$ to compute the number of satisfying assignments to C in $\mathcal{O}(2^{n(1-\delta)})$ time. Or,
- if C is a poly-sized DNF formula then we first construct the negation of C, let us denote by $\neg C$, in poly(n) time; using De Morgan's law we can see that the resulting formula $\neg C$ will be a poly(n) CNF formula. Using $\mathcal{A}$ we can now compute the number of satisfying assignments t to $\neg C$ in $\mathcal{O}(2^{n(1-\delta)})$ time. The number of satisfying assignments to C would be then $2^n - t$.

The existence of an algorithm such as $\mathcal{A}$ would imply that AC^0_2-QSETH is false. Hence, proved. ◀

Using similar arguments as in the proof of Theorem 13 we can conclude the following statements.

▶ **Corollary 14** ($\oplus$QSETH). *For each constant $\delta > 0$, there exists $c > 0$ such that there is no bounded-error quantum algorithm that solves $\oplus$CNFSAT (even restricted to formulas with $m \leq cn^2$ clauses) in $\mathcal{O}(2^{n(1-\delta)})$ time, unless AC^0_2-QSETH (Conjecture 2) is false or PARITY $\notin \mathcal{CO}(AC^0_2)$ (i.e., Item 1 of Conjecture 11 is false).*

▶ **Corollary 15** ($\#_q$QSETH). *Let $q \in [3, \frac{N}{2}]$ be an integer. For each constant $\delta > 0$, there exists $c > 0$ such that there is no bounded-error quantum algorithm that solves $\#_q$CNFSAT (even restricted to formulas with $m \leq cn^2$ clauses) in $\mathcal{O}\left(2^{n(1-\delta)}\right)$ time, unless AC^0_2-QSETH (Conjecture 2) is false or COUNT$_q \notin \mathcal{CO}(AC^0_2)$ (i.e., Item 2 of Conjecture 11 is false).*

▶ **Corollary 16** (Majority-QSETH). *For each constant $\delta > 0$, there exists $c > 0$ such that there is no bounded-error quantum algorithm that solves*

1. *MAJ-CNFSAT (even restricted to formulas with $m \leq cn^2$ clauses) in $\mathcal{O}(2^{n(1-\delta)})$ time, unless AC^0_2-QSETH is false or MAJORITY $\notin \mathcal{CO}(AC^0_2)$ (i.e., Item 3 of Conjecture 11 is false);*
2. *st-MAJ-CNFSAT (even restricted to formulas with $m \leq cn^2$ clauses) in $\mathcal{O}(2^{n(1-\delta)})$ time, unless AC^0_2-QSETH is false or st-MAJORITY $\notin \mathcal{CO}(AC^0_2)$ (i.e., Item 4 of Conjecture 11 is false).*

Akmal and Williams showed that one can actually compute the Majority on the truth table of k-CNF formulas for constant k in polynomial time, while computing the strict-Majority on the truth table of such formulas is NP-hard [6]. Therefore, here we define both majority and strict-majority and their variants of CNFSAT problems for clarity (and state the hardness of both problems in one conjecture). Note that for CNFSAT, each clause is allowed to contain n literals (which means k is no longer a constant), and in this case, it is not clear if one can solve MAJ-CNFSAT in polynomial time or not. Therefore, none of AC^0_2-QSETH, Items 3 and 4 of Conjecture 11 is immediately false yet. (See also the discussion at the bottom of page 5 in the arXiv version of [6] for reductions between MAJ-CNFSAT and st-MAJ-CNFSAT.)

3.1.2 Quantum complexity of Δ-add-#CNFSAT

Instead of the exact number of satisfying assignments to a formula, one might be interested in an additive-error approximation. Towards that, we define the problem Δ-ADD-#CNFSAT as follows.

▶ **Definition 17** (Δ-ADD-#CNFSAT). *Given a CNF formula ϕ on n variables. The goal of the problem is to output an integer d such that $|d - |\phi|| < \Delta$ where $\Delta \in [1, 2^n]$.*

This problem (Definition 17) can be viewed as computing the following property on the truth table of the given formula.

▶ **Definition 18** (Δ-ADD-COUNT). *Given a Boolean string $x \in \{0,1\}^N$, Δ-ADD-COUNT asks to output an integer w such that $|w - |x|| < \Delta$ where $\Delta \in [1, N)$.*

Note that Δ-ADD-COUNT is a relation instead of a function now because its value is not necessarily uniquely defined. The bounded-error quantum query complexity for computing Δ-ADD-COUNT was studied in [31]. They showed the following result.

▶ **Theorem 19** (Theorem 1.11 in [31]). *Let $\Delta \in [1, N)$. Every bounded-error quantum algorithm that computes Δ-ADD-COUNT uses $\Omega\left(\sqrt{\frac{N}{\Delta}} + \frac{\sqrt{t(N-t)}}{\Delta}\right)$ quantum queries on inputs with t ones.*

For values of $\Delta = o(\sqrt{t})$ we are unable to prove the compression-obliviousness of this property. Hence, we make the following conjecture.

▶ **Conjecture 20.** *Δ-ADD-COUNT $\in \mathcal{CO}(AC^0_2)$.*

One can now establish the time lower bound for computing the Δ-ADD-COUNT on poly(n)-sized CNF and DNF formulas. However, this doesn't automatically imply the same lower bound for the case when there are only CNF formulas to consider. Fortunately, Δ-ADD-COUNT is defined in such a way that computing this property is equally hard for both CNF and DNF formulas. More precisely, the following statement holds.

▶ **Theorem 21** (Δ-ADD-#QSETH). *Let $\Delta \in [1, 2^n)$. For each constant $\delta > 0$, there exists $c > 0$ such that there is no bounded-error quantum algorithm that solves Δ-ADD-#CNFSAT (even restricted to formulas with $m \leq cn^2$ clauses) in $\mathcal{O}\left(\left(\sqrt{\frac{N}{\Delta}} + \frac{\sqrt{\hat{h}(N-\hat{h})}}{\Delta}\right)^{1-\delta}\right)$ time where $\hat{h}$ is the number of satisfying assignments, unless AC^0_2-QSETH (Conjecture 2) is false or Δ-ADD-COUNT $\notin \mathcal{CO}(AC^0_2)$ (i.e., Conjecture 20 is false).*

Proof. By way of contradiction let's assume that there is an algorithm $\mathcal{A}$ such that given a CNF formula it can compute the Δ-ADD-COUNT on its truth table in $\mathcal{O}\left(\left(\sqrt{\frac{N}{\Delta}} + \frac{\sqrt{t(N-t)}}{\Delta}\right)^{1-\beta}\right)$ time for some constant $\beta > 0$.

Then, given a poly(n) sized DNF formula on n variables, let us denote that by ϕ, we can run Algorithm $\mathcal{A}$ on $\neg\phi$ and use its output which is a Δ additive error approximation of the number of satisfying assignments to $\neg\phi$ to compute a Δ additive error approximation of the number of satisfying assignments to ϕ.

Let us denote the number of satisfying assignments of $\neg\phi$ by d' and the output of Algorithm $\mathcal{A}$ by d. This means we have $|d - d'| < \Delta$. We claim that $2^n - d$ will be a Δ additive error approximation of $2^n - d'$, which is the number of satisfying assignments of ϕ; $|(2^n - d) - (2^n - d')| = |d' - d| < \Delta$.

Therefore, a $\mathcal{O}\left(\left(\sqrt{\frac{N}{\Delta}} + \frac{\sqrt{t(N-t)}}{\Delta}\right)^{1-\beta}\right)$ time algorithm for computing Δ-ADD-COUNT on truth table of CNF formulas also implies a $\mathcal{O}\left(\left(\sqrt{\frac{N}{\Delta}} + \frac{\sqrt{t(N-t)}}{\Delta}\right)^{1-\beta}\right)$ time algorithm for computing Δ-ADD-COUNT on truth table of DNF formulas; this violates the combination of AC^0_2-QSETH and Conjecture 20. Hence, proved. ◀

3.1.3 Quantum complexity of γ-#CNFSAT and other related problems

One other approximation of the count of satisfying assignments is the multiplicative-factor approximation, defined as follows.

▶ **Definition 22** (γ-#CNFSAT). *Let $\gamma \in (0,1)$. The γ-#CNFSAT problem is defined as follows. Given a CNF formula formula ϕ on n Boolean variables, The goal of the problem is to output an integer d such that $(1-\gamma)|\phi| < d < (1+\gamma)|\phi|$.* [9]

The expression $(1-\gamma)|\phi| < d < (1+\gamma)|\phi|$ can be categorized into the following two cases.

- Case 1 is when $\gamma|\phi| \leq 1$: in this regime, the algorithm solving γ-#CNFSAT is expected to return the value $|\phi|$, which is the *exact* count of the number of solutions to the CNFSAT problem. From Theorem 13 we postulate that for each constant $\delta > 0$, there is no $\mathcal{O}(2^{n(1-\delta)})$ time algorithm that can compute the exact number of solutions to input CNF formula; this is a tight lower bound.
- Case 2 is when $\gamma|\phi| > 1$: in this regime, the algorithm solving γ-#CNFSAT is expected to return a value d which is a γ-*approximate relative count* of the number of solutions to the CNFSAT problem.

In order to understand the hardness of γ-#CNFSAT in the second case, we will first try to understand how hard it is to compute the following property. Let $f_{\ell,\ell'} : \mathcal{D} \to \{0,1\}$ with $\mathcal{D} \subset \{0,1\}^N$ be a partial function defined as follows

$$f_{\ell,\ell'}(x) = \begin{cases} 1, & \text{if } |x| = \ell, \\ 0, & \text{if } |x| = \ell'. \end{cases}$$

Nayak and Wu in [31] analyzed the approximate degree of $f_{\ell,\ell'}$. By using the polynomial method [10] again we have a lower bound on the quantum query complexity of $f_{\ell,\ell'}$ as mentioned in the following statement.

▶ **Lemma 23** ([31, Corollary 1.2]). *Let $\ell, \ell' \in \mathbb{N}$ be such that $\ell \neq \ell'$, $f_{\ell,\ell'} : \mathcal{D} \to \{0,1\}$ where $\mathcal{D} \subset \{0,1\}^N$, and*

$$f_{\ell,\ell'}(x) = \begin{cases} 1, & \textit{if } |x| = \ell, \\ 0, & \textit{if } |x| = \ell'. \end{cases}$$

Let $\Delta_\ell = |\ell - \ell'|$ and $p \in \{\ell, \ell'\}$ be such that $|\frac{N}{2} - p|$ is maximized. Then every bounded-error quantum algorithm that computes $f_{\ell,\ell'}$ uses $\Omega\left(\sqrt{\frac{N}{\Delta_\ell}} + \frac{\sqrt{p(N-p)}}{\Delta_\ell}\right)$ queries.

Using AC^0_2-QSETH we will now show that for a choice of ℓ, ℓ' computing $f_{\ell,\ell'}$ on truth tables of CNF formulas of poly(n) size requires $\Omega\left(\sqrt{\frac{2^n}{\Delta_\ell}} + \frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)$ time where $\Delta_\ell = |\ell - \ell'|$ and $p \in \{\ell, \ell'\}$ that maximises $|2^{n-1} - p|$. The only caveat (as also witnessed several times earlier) is that we cannot prove the compression obliviousness of $f_{\ell,\ell'}$ hence we state and use the following conjecture.

[9] The same results hold if the approximation is defined with the equalities, i.e., $(1-\gamma)|\phi| \leq d \leq (1+\gamma)|\phi|$. An additional observation under this changed definition of γ-#CNFSAT is as follows. Given a CNF formula as input, the algorithm for γ-#CNFSAT outputs 0 only when there is no satisfying assignment to that formula. Hence, one can decide satisfiability of a given CNF formula using the algorithm for γ-#CNFSAT. Therefore, the same lower bound holds for this changed definition of γ-#CNFSAT.

▶ **Conjecture 24.** *For any pair of integers $\ell, \ell' \in [2^n] \cup \{0\}$ satisfying that $\ell \neq \ell'$, $f_{\ell,\ell'} \in \mathcal{CO}(AC^0_2)$.*[10]

And we can show the following.

▶ **Lemma 25.** *Let $\ell, \ell' \in [2^n] \cup \{0\}$ be such that $\ell \neq \ell'$. If both AC^0_2-QSETH(Conjecture 2) and Conjecture 24 hold, then at least one of the following is true:*

- *For each constant $\delta > 0$, there exists $c > 0$ such that there is no bounded-error quantum algorithm that computes $f_{\ell,\ell'}$ on the truth table of CNF formulas defined on n variables in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}} + \frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta}\right)$ time (even restricted to formulas with $m \leq cn^2$ clauses);*
- *For each constant $\delta > 0$, there exists $c > 0$ such that there is no bounded-error quantum algorithm that computes $f_{N-\ell,N-\ell'}$ on the truth table of CNF formulas defined on n variables in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}} + \frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta}\right)$ time (even restricted to formulas with $m \leq cn^2$ clauses);*

here $\Delta_\ell = |\ell - \ell'|$ and $p \in \{\ell, \ell'\}$ such that $|2^{n-1} - p|$ is maximized. In particular, when $\ell + \ell' = 2^n$, the above immediately implies the following:

- *For each constant $\delta > 0$, there exists $c > 0$ such that there is no bounded-error quantum algorithm that computes $f_{\ell,\ell'}$ on the truth table of CNF formulas defined on n variables in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}} + \frac{\sqrt{\ell\ell'}}{\Delta_\ell}\right)^{1-\delta}\right)$ time (even restricted to formulas with $m \leq cn^2$ clauses).*

Proof. Let N be an integer that we will fix later and let $f'_{\ell,\ell'} : \{0,1\}^N \to \{0,1\}$ be defined as follows

$$f'_{\ell,\ell'} = \begin{cases} 1, & \text{if } |x| = N - \ell, \\ 0, & \text{if } |x| = N - \ell'. \end{cases}$$

It is not hard to see $f'_{\ell,\ell'}$ is the same as function $f_{N-\ell,N-\ell'}$. Fortunately, both the functions $f_{N-\ell,N-\ell'}$ and $f_{\ell,\ell'}$ have the same value of Δ_ℓ and h where $h = p(N-p)$. Therefore the bounded error quantum query complexity of $f'_{\ell,\ell'}$ is $\Omega\left(\sqrt{\frac{N}{\Delta_\ell}} + \frac{\sqrt{p(N-p)}}{\Delta_\ell}\right)$ where $\Delta_\ell = |\ell - \ell'|$ and $p \in \{\ell, \ell'\}$ such that $|\frac{N}{2} - p|$ is maximised; same as the bounded error quantum query complexity of $f_{\ell,\ell'}$ as mentioned in Lemma 23.

Moreover, as $f'_{\ell,\ell'}$ is the same function $f_{N-\ell,N-\ell'}$ it is therefore clear from Conjecture 24 that $f'_{\ell,\ell'} \in \mathcal{CO}(\mathrm{AC}^0_2)$ which means there is no bounded error quantum algorithm that can compute $f'_{\ell,\ell'}$ or $f_{\ell,\ell'}$ on truth tables of $\mathrm{poly}(n)$ size CNF or DNF formulas defined on n input variables in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}} + \frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta}\right)$ time for any constant $\delta > 0$ unless AC^0_2-QSETH is false. We will now show that conditional on AC^0_2-QSETH this result holds even when we restrict ourselves to only $\mathrm{poly}(n)$ sized CNF formulas.

Having introduced $f'_{\ell,\ell'}$ we will now prove Lemma 25 using the following propositions.

- **Proposition A** There is no bounded error quantum algorithm that can compute $f_{\ell,\ell'}$ on truth table of CNF formulas defined on n variables in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}} + \frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta}\right)$ time for any $\delta > 0$.

[10] Note that, there are some values of ℓ, ℓ' for which $f_{\ell,\ell'}$ will be provably compression oblivious, for e.g., $\ell = 1$ and $\ell' = 0$ would capture the OR property which is compression oblivious; see Section 3.1.

- **Proposition B** There is no bounded error quantum algorithm that can compute $f_{\ell,\ell'}$ on truth table of DNF formulas defined on n variables in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}}+\frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta}\right)$ time for any $\delta>0$.
- **Proposition C** There is no bounded error quantum algorithm that can compute $f'_{\ell,\ell'}$ on truth table of CNF formulas defined on n variables in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}}+\frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta}\right)$ time for any $\delta>0$.
- **Proposition D** There is no bounded error quantum algorithm that can compute $f'_{\ell,\ell'}$ on truth table of DNF formulas defined on n variables in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}}+\frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta}\right)$ time for any $\delta>0$.

Conditional on Conjecture 24 and AC^0_2-QSETH the following statements hold.

- **Claim 1** At least one of the propositions A or B is true.
- **Claim 2** At least one of the propositions C or D is true.
- **Claim 3** At least one of the propositions A or C is true; by way of contradiction let us assume that both propositions A and C are false, this means there exist algorithms $\mathcal{A},\mathcal{A}'$ that for an $\delta>0$ and $\delta'>0$ compute $f_{\ell,\ell'}$ and $f'_{\ell,\ell'}$ on the truth table of $\mathrm{poly}(n)$ size CNF formulas defined on n input variables in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}}+\frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta}\right)$ time and in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}}+\frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta'}\right)$ time, respectively. Moreover, if propositions A and C are false then from Claims 1 and 2 we can deduce that both B and D must be true which means there is no quantum algorithm that can compute $f_{\ell,\ell'}$ or $f'_{\ell,\ell'}$ on the truth table of $\mathrm{poly}(n)$ size DNF formulas on n input variables in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}}+\frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta}\right)$ time for any $\delta>0$. However, given a DNF formula ϕ as an input to compute $f_{\ell,\ell'}$ on its truth table one can instead compute $f'_{\ell,\ell'}$ on the negation of ϕ, let us denote by $\neg\phi$, using algorithm $\mathcal{A}'$ on $\neg\phi$ in $\mathcal{O}\left(\left(\sqrt{\frac{2^n}{\Delta_\ell}}+\frac{\sqrt{p(2^n-p)}}{\Delta_\ell}\right)^{1-\delta'}\right)$ time which is a contradiction. This means at least one of the two propositions A or C must be true which is exactly the statement of Lemma 25. ◀

Inspired by the arguments used in the proof of Theorem 1.13 in [31], we will now show that Lemma 25 implies the following result. Our result holds for $\gamma\in\left[\frac{1}{2^n},0.4999\right)$; this range of γ suffices for our reductions presented in the later sections.

▶ **Corollary 26** (γ-#QSETH). *Let $\gamma\in\left[\frac{1}{2^n},0.4999\right)$. For each constant $\delta>0$, there exists $c>0$ such that there is no bounded-error quantum algorithm that solves γ-#CNFSAT (even restricted to formulas with $m\le cn^2$ clauses) in time*

1. $\mathcal{O}\left(\left(\frac{1}{\gamma}\sqrt{\frac{2^n-\hat{h}}{\hat{h}}}\right)^{1-\delta}\right)$ *if $\gamma\hat{h}>1$, where $\hat{h}$ is the number of satisfying assignments;*
2. $\mathcal{O}(2^{n(1-\delta)})$ *otherwise,*

unless AC^0_2-QSETH(Conjecture 2) is false or $\ell\neq\ell'$, $f_{\ell,\ell'}\notin\mathcal{CO}(AC^0_2)$ (i.e., Conjecture 24 is false).

We show the first part of Corollary 26 in the following way and use the result from Theorem 13 for the second part. Given a value of $\gamma \in \left[\frac{1}{2^n}, 0.4999\right)$ we will fix values of $\ell \in [2^n] \cup \{0\}$ and $\ell' \in [2^n] \cup \{0\}$ such that we are able to compute $f_{\ell,\ell'}$ on the truth table of an input CNF formulas on n variables using the algorithm that solves γ-#CNFSAT. Hence, we can show a lower bound on γ-#CNFSAT using the lower bound result from Lemma 25.

Proof. Let $N = 2^n$. Let $\ell = \frac{N}{2} + \lceil \gamma t \rceil = \left\lceil \frac{N}{2} + \gamma t \right\rceil$ and $\ell' = \frac{N}{2} - \lceil \gamma t \rceil = \left\lfloor \frac{N}{2} - \gamma t \right\rfloor$; here $t \in [N]$ is a value that we will fix later but in any case, we have $1 \leq \lceil \gamma t \rceil < \frac{N}{2}$. With that, we are ensured that $\gamma\ell > \frac{1}{2}$. We also make sure to choose values ℓ, ℓ' in such a way that $\gamma\ell' = \Omega(1)$. Clearly, $\ell + \ell' = N$ and $\Delta_\ell = |\ell - \ell'| = 2\lceil \gamma t \rceil$. Therefore by invoking the result from Lemma 25 we can say that for these values of ℓ, ℓ' there is no bounded-error quantum algorithm that can solve $f_{\ell,\ell'}$ on the truth table of CNF formulas in $\mathcal{O}\left(\left(\sqrt{\frac{N}{\lceil \gamma t \rceil}} + \frac{\sqrt{\ell(N-\ell)}}{\lceil \gamma t \rceil}\right)^{1-\delta}\right)$ time, for each $\delta > 0$; let us denote this claim by (*).

Let $\mathcal{A}$ be an algorithm that computes γ-#CNFSAT, i.e., Algorithm $\mathcal{A}$ returns a value h such that $(1-\gamma)\hat{h} < h < (1+\gamma)\hat{h}$. Given $\ell = \frac{N}{2} + \lceil \gamma t \rceil$ and $\ell' = \frac{N}{2} - \lceil \gamma t \rceil$, there are values of $t \in [N]$ such that we will be able to distinguish whether the number of satisfying assignments to a formula is ℓ or ℓ' using Algorithm $\mathcal{A}$. As $\ell > \ell'$ in our setup, we want t such that $\ell'(1+\gamma) < \ell(1-\gamma)$; it is then necessary that $\gamma N < 2\lceil \gamma t \rceil$; let us denote this as Condition 1.

Now we set the values of ℓ and ℓ'. Given a value of $\gamma \in \left[\frac{1}{N}, \frac{1}{2}\right)$, we set $\ell = \left\lceil \frac{N}{2(1-\gamma)} \right\rceil$ and $\ell' = N - \ell$. This implies $\frac{N}{2(1-\gamma)} \leq \ell < \frac{N}{2(1-\gamma)} + 1$, $\frac{N(1-2\gamma)}{2(1-\gamma)} - 1 < \ell' \leq \frac{N(1-2\gamma)}{2(1-\gamma)}$, and $\frac{\gamma N}{(1-\gamma)} \leq |\ell - \ell'| < \frac{\gamma N}{(1-\gamma)} + 2$. Therefore we obtain $2\gamma\ell - 2\gamma \leq |\ell - \ell'| < 2\gamma\ell + 2$.[11] We know from claim (*) that every quantum algorithm that (for these values of ℓ, ℓ') computes $f_{\ell,\ell'}$ on CNF formulas requires $\Omega(L^{1-\delta})$ time for each $\delta > 0$, where $L = \frac{1}{\gamma}\sqrt{\frac{N-\ell}{\ell+1}} = \Omega\left(\frac{1}{\gamma}\sqrt{\frac{N-\ell}{\ell}}\right)$. Moreover, ℓ' is $(\ell-1)(1-2\gamma) - 1 < \ell' \leq \ell(1-2\gamma)$. Therefore, we can see that $L = \Omega\left(\frac{1}{\gamma}\sqrt{\frac{N-\ell}{\ell}}\right) = \Omega\left(\frac{1-2\gamma}{\gamma}\sqrt{\frac{N-\ell'}{\ell'}}\right) = \Omega\left(\frac{1}{\gamma}\sqrt{\frac{N-\ell'}{\ell'}}\right)$.

It is also easy to see that if $\ell = \lceil \frac{N}{2(1-\gamma)} \rceil$ were to be expressed as $\frac{N}{2} + \lceil \gamma t \rceil$ (i.e. denote ℓ to be $\frac{N}{2} + \lceil \gamma t \rceil$), then for that value of t we have $\lceil \gamma t \rceil = \lceil \frac{N}{2(1-\gamma)} \rceil - \frac{N}{2} \geq \frac{N\gamma}{2(1-\gamma)} > \frac{N\gamma}{2}$, which satisfies Condition 1. Hence here we can use Algorithm $\mathcal{A}$ to distinguish whether the number of satisfying assignments to a formula is ℓ or ℓ'. Hence given a CNF formula as input, we will be able to use Algorithm $\mathcal{A}$ to distinguish whether the number of satisfying assignments is ℓ or ℓ'. Let $T = \frac{1}{\gamma}\sqrt{\frac{N-\ell}{\ell}} + \frac{1}{\gamma}\sqrt{\frac{N-\ell'}{\ell'}} = \mathcal{O}(\frac{1}{\gamma}\sqrt{\frac{N-\ell'}{\ell'}})$. If for some constant $\delta > 0$, $\mathcal{A}$ can solve γ-#CNFSAT on an input CNF formula that has $\hat{h}$ number of satisfying assignments in $\mathcal{O}((\frac{1}{\gamma}\sqrt{\frac{N-\hat{h}}{\hat{h}}})^{1-\delta})$ time, then we are essentially computing $f_{\ell,\ell'}$ in $\mathcal{O}(T^{1-\delta})$ time, which is a contradiction to claim (*). Hence the first part of the statement of Corollary 26 proved.

Proof of the second part of this theorem follows from Theorem 13 as the regime $\gamma\hat{h} \leq 1$ translates to exactly counting the number of satisfying assignments. ◀

[11] To view the calculations in a less cumbersome way one can use the fact that asymptotically $\ell = \frac{N}{2(1-\gamma)}$, $\ell' = \frac{N(1-2\gamma)}{2(1-\gamma)}$ and $|\ell - \ell'| = \frac{\gamma N}{(1-\gamma)} = 2\gamma\ell$.

3.2 Quantum complexity of #k-SAT and other related problems

In the previous subsection, we discussed the quantum complexity of variants of CNFSAT problems. However, it is not clear how to immediately derive a similar quantum complexity result for variants of k-SAT problems with constant k by using the quantum (conditional) hardness results for variants of CNFSAT problems. Of course we could make a further conjecture about variants of k-SAT problems like we did in the previous subsection, but it would introduce too many conjectures. Moreover, some variants of k-SAT (for constant k) are even shown to be solvable in polynomial time [6].

To give the (quantum) complexity of some optimization problems (for example, lattice problems [11]), on the other hand, we might want to have some (quantum) conditional lower bounds for (variants of) k-SAT problems with not too large k. This is because we might make $2^k \cdot \mathrm{poly}(n)$ calls to a solver of those problems to solve k-SAT. This is undesirable for giving the (quantum) complexity of those optimization problems when k approaches n, while it is tolerable for a relatively small k (like $k = \mathrm{poly}\log n$). Hence in this subsection, we would like to say something interesting about quantum hardness for #k-SAT and $\oplus k$-SAT when $k = \Theta(\log n)$, only using the hardness assumptions on counting-CNFSAT (that is, #QSETH). Here, variants of k-SAT are defined exactly the same way as Definition 8, Definition 17, and Definition 22, except that the input is now a k-CNF formula.

We use the classical algorithm by Schuler [34].[12] This algorithm can be viewed as a Turing reduction from SAT with bounded clause density to SAT with bounded clause width, which was analyzed in [16]. The time complexity of this algorithm is upper bounded by $\binom{m+n/k}{n/k} \cdot \mathrm{poly}(m, n)$, where m is number of clauses.

Algorithm 1 $\mathrm{ReduceWidth}_k(\psi)$.

Input: CNF formula ψ

if ψ has no clause of width $> k$ **then**
    output ψ
else
    let $C' = \{l_1, \ldots, l_{k'}\}$ be a clause of ψ of width $k' > k$
    $C = \{l_1, \ldots, l_k\}$
    $\psi_0 \leftarrow \psi - \{C'\} \cup \{C\}$
    $\psi_1 \leftarrow \psi \wedge \neg l_1 \wedge \neg l_2 \wedge \cdots \wedge \neg l_k$
    $\psi_1 \leftarrow$ Remove variables corresponding to $l_1, \ldots, l_k$ from ψ_1 by setting $l_1 = 0, \ldots, l_k = 0$
    $\mathrm{ReduceWidth}_k(\psi_0)$
    $\mathrm{ReduceWidth}_k(\psi_1)$

Algorithm 1 takes as input a CNF formula of width greater than k, and then outputs a list of k-CNF formulas ψ_i where the solutions of the input formula is the union of the solutions of the output formulas, i.e., $\mathrm{sol}(\psi) = \cup_i \mathrm{sol}(\psi_i)$, where $\mathrm{sol}(\phi)$ denotes the set of satisfying assignments to a formula ϕ. In fact, it is not hard to see that the count of the number of satisfying assignments also is preserved, i.e., $|\mathrm{sol}(\psi)| = \sum_i |\mathrm{sol}(\psi_i)|$.

▶ **Lemma 27** (Implicit from Section 3.2 in [16]). *Algorithm 1 takes as input a CNF formula ψ on n input variables, with m clauses, that is of width strictly greater than k and outputs a number of k-CNF formulas ψ_i each defined on at most n input variables and at most m clauses such that $|sol(\psi)| = \sum_i |sol(\psi_i)|$.*

[12] This algorithm can also be used to solve CNFSAT on n variables, m clauses in $\mathcal{O}(\mathrm{poly}(n) 2^{n(1-1/(1+\log m))})$ expected time.

In the full version of this paper [17] we present the proof for completeness. Using Lemma 27 and Lemma 5 in [16] we will now show the hardness of k-SAT and its counting variants when $k = \Theta(\log n)$ without introducing new conjectures.

▶ **Corollary 28.** *For each constant $\delta > 0$, there exists a constant c such that there is no bounded-error quantum algorithm that solves*

1. *$c \log n$-SAT in $\mathcal{O}(2^{(1-\delta)n/2})$ time unless Basic-QSETH (see Corollary 6) is false;*
2. *$\#c \log n$-SAT in $\mathcal{O}(2^{(1-\delta)n})$ time unless #QSETH (see Theorem 13) is false;*
3. *$\oplus c \log n$-SAT in $\mathcal{O}(2^{(1-\delta)n})$ time unless $\oplus$QSETH (see Corollary 14) is false;*
4. *$\oplus_q c \log n$-SAT in $\mathcal{O}(2^{(1-\delta)n})$ time unless $\#_q$QSETH (see Corollary 15) is false.*

Proof. We first prove the first item. Suppose that for each constant c, there is an algorithm $\mathcal{A}$ that solves $\#c \log n$-SAT in 2^{ns} for some constant $s := 1 - \delta < 1$. Let $k = c \log n$ for the rest of the proof. Consider the ReduceWidth$_k$ algorithm (Algorithm 1) with input CNF formula ψ. Let p be some path of length t in the tree T of recursive calls to ReduceWidth$_k(\psi)$. Let ψ_p be the output formula of width at most k at the leaf of p. Let l, r be the number of left, right branches respectively on path p. Every left branch in the path reduces the width of exactly 1 clause to k, therefore $l \leq m$. On the other hand, with additional poly(n, m) time, every right branch of path p reduces the number of variables by k, therefore $r \leq n/k$. As a result, the number of paths in tree T with r right branches is at most $\binom{m+r}{r}$ and each outputs a formula with $n - rk$ variables.

Using the same arguments as in [16, Lemma 5], one can see that $\mathcal{A}$ together with the ReduceWidth$_k$ subroutine can be used to solve #CNFSAT (ignoring poly(n) factors) in time at most

$$
\begin{aligned}
&\sum_{r=0}^{n/k} \binom{m+r}{r} 2^{s(n-rk)} + \binom{m+n/k}{n/k} \cdot \mathrm{poly}(m,n) \\
\leq &\sum_{r=0}^{n/k} \binom{m+\frac{n}{k}}{r} 2^{s(n-rk)} \\
= &2^{sn} \sum_{r=0}^{n/k} \binom{m+\frac{n}{k}}{r} \frac{1}{2^{srk}} \\
\leq &2^{sn} (1 + \frac{1}{2^{sk}})^{m+\frac{n}{k}} \\
\leq &2^{sn} e^{\frac{1}{2^{sk}}(m+\frac{n}{k})} \qquad \text{since } (1+x) \leq e^x \\
\leq &2^{sn + \frac{4m}{2^{sk}}},
\end{aligned}
$$

where the last equality holds because we can assume that $m \geq \frac{n}{k}$ without loss of generality (by appending dummy clauses). Therefore, for each c', there exist a constant c for $k = c \log n$ and δ' such that if $m \leq c' n^2$, then $s + \frac{4m}{n 2^{sk}} < 1 - \delta'$. As a result, a 2^{ns}-time algorithm for $\#c \log n$-SAT implies a $2^{n(1-\delta')}$-time algorithm for #CNFSAT (restricted to formulas with $m \leq c' n^2$), which would refute #-QSETH (Theorem 13). This proves the first item of the corollary. The same arguments hold for k-SAT, $\oplus$k-SAT, and $\oplus_q$k-SAT as well. ◀

Note that, we *cannot* extend the same arguments for the MAJORITY or *st*-MAJORITY or additive-error approximation of count because those properties are not count-preserving. However, these arguments do extend to the multiplicative-factor approximation of the count.

▶ **Corollary 29.** *Let* $\gamma \in \left[\frac{1}{2^n}, 0.4999\right)$*. For each constant* $\delta > 0$*, there exists constant* c *such that, there is no bounded-error quantum algorithm that solves* γ*-#*$c \log n$*-SAT in time*

1. $\mathcal{O}\left(\left(\frac{1}{\gamma}\sqrt{\frac{2^n - \hat{h}}{\hat{h}}}\right)^{1-\delta}\right)$ *if* $\gamma\hat{h} > 1$*, where* $\hat{h}$ *is the number of satisfying assignments;*
2. $\mathcal{O}(2^{n(1-\delta)})$ *otherwise,*

unless γ*-#QSETH (see Corollary 26, implied by Lemma 25) is false.*

4 Quantum strong simulation of quantum circuits

Combining results from [25] with our results from Section 3 we are able to comment on the exact complexity of the *strong simulation problem* which is defined as follows.[13]

▶ **Definition 30** (The strong simulation problem)**.** *Let* $p \in \mathbb{N}$*. Given a quantum circuit* C *on* n *qubits and* $x \in \{0,1\}^n$*, the goal of* strong simulation *with* p*-bit precision is to output the value of* $|\langle x|C|0^n\rangle| := 0.C_1C_2\ldots$ *up to* p*-bit precision. That is, output* $C_0.C_1\ldots C_{p-1}$*.*[14]

For a quantum circuit C, computing $|\langle x|C|0^n\rangle|$ exactly, to a precision of n bits, is #P-hard [19, 35]. This means even a *quantum* computer will likely require exponential time to strongly simulate another quantum circuit. In the full version of this paper [17], we prove a more *precise* quantum time bound for strongly simulating quantum circuits, both exactly and approximately; in the approximate case, we present complexity results for both multiplicative factor and additive error approximation. Our results extend the results by [25] in two directions: firstly, we give explicit (conditional) bounds proving that it is hard to strongly simulate quantum circuits using *quantum* computers as well. Secondly, we also address the open question posed by [25] on the (conditional) hardness of strong simulation up to accuracy $O(2^{-n/2})$, however, our results are based on a hardness assumption different from SETH or Basic QSETH.

Our results are based on two main components. Firstly, on the observation that the reduction from CNFSAT to the strong simulation problem given (in Theorem 3) by [25] encodes the count of the number of satisfying assignments; this fact allows us to use the same reduction to reduce other variants of CNFSAT, such as #CNFSAT or $\oplus$CNFSAT, to the strong simulation problem, moreover, the same reduction also allows us to reduce γ-#CNFSAT and Δ-ADD-#CNFSAT to analogous variants of the strong simulation problem, respectively. As the second main component, we use the quantum hardness of these variants of CNFSAT problem discussed in Section 3.

In the full version of this paper on arXiv, we first state the result of the exact quantum time complexity of the strong simulation problem and then use that result to later show how hard it is for a quantum computer to strongly simulate a quantum circuit with an additive error or a multiplicative factor approximation as stated in Table 2 - see [17] for details.

5 Quantum lower bound for lattice counting and q-count problems

In the full version of this paper on arXiv, we connect k-SAT to variants of lattice problems and then use the QSETH lower bound we have in Section 3.2 to give quantum fine-grained complexity for those lattice problems.

[13] Note that this is different from the *weak simulation* problem; a weak simulation *samples* from probability distribution $p(x) := |\langle 0^n|C|x\rangle|^2$.

[14] Though in some papers the strong simulation problem requires that we output $\langle x|C|0^n\rangle$ instead of $|\langle x|C|0^n\rangle|$, we use this definition because it is more comparable to the definition of the weak simulation problem. Also, the lower bound we present holds for both of these definitions.

Fine-grained complexity of lattice problems is quite widely studied in the classical case [11, 2, 3, 5, 12, 13]. Lots of variants of lattice problems have been considered before, and the most well-studied one is the closest vector problem (with respect to ℓ_p norm).

CVP_p is known to have a 2^n SETH lower bound for any $p \notin 2\mathbb{Z}$ [11, 2], and for even p, there seems a barrier for showing a fine-grained reduction from k-SAT to CVP [4]. Kannan gave a $n^{\mathcal{O}(n)}$-time algorithm for solving CVP_p for arbitrary $p \geq 1$ [29], while the best-known algorithm for solving CVP_p with noneven p is still n^{cn} for some constant c. To get a conditional quantum lower bound for CVP_p for noneven p, given there is already a classical reduction from k-SAT to CVP_p using $2^k \cdot \text{poly}(n)$ time (for noneven p) [11, 2], either one can directly use the QSETH framework by Aaronson et al. [1] to get a $2^{(0.5-\varepsilon)n}$ lower bound, or we can use Corollary 28 to get the same lower bound in our QSETH framework.[15]

A natural question is invoked here: Can we have a $2^{(0.5+\varepsilon)n}$ quantum SETH lower bound for any (variants of) lattice problems? The answer is yes by using the framework and the problems introduced in Section 3.2 and by considering the counting variant of lattice problems stated in Table 2. In the full version, we begin by introducing the (approximate) lattice counting problem and some other related problems - see [17] for all the relevant details.

6 Hardness of Counting/Parity of OV, Hitting Set, and Set-Cover

In the full version of this paper on arXiv we discuss the consequences of Corollary 26 and Corollary 28 for some well-motivated optimization problems: Orthogonal Vectors, Hitting Set and Set Cover as stated in Table 2 – see [17] for details.

7 Discussion and open questions

We believe that this paper opens up the possibility of concluding quantum time lower bounds for many other problems, both for other variants of CNFSAT and also for problems that are not immediately related to CNFSAT. While this is a natural broad future direction to explore, we also mention the following few directions for future work which are more contextual to this paper.

- One of the motivations to use AC^0_2-QSETH in this paper is so that we can "tie" certain conjectures, that would have otherwise been standalone conjectures, to one main conjecture. But in the process, we conjecture compression obliviousness of several properties. It would be nice if we could also have an "umbrella" conjecture that allows one to establish compression obliviousness of several properties. For e.g., it would be nice if we could show that compression obliviousness of a natural property like COUNT or PARITY implies compression obliviousness of say Δ-ADD-COUNT.
- It will be interesting to see if one can use the QSETH framework (or the AC^0_2-QSETH conjecture) to give a single exponential lower bound for #CVP in Euclidean norm.
- Using Boolean function Fourier analysis, we were able to show that the existence of (quantum-secure) PRFs imply that majority and parity are compression oblivious, whenever the input is given by a formula or circuit. This proof technique could plausibly be extended to larger sets of functions that have a similar structure, e.g., a natural candidate would be to show an equivalent statement for symmetric functions with non-negligible mass on high-degree Fourier coefficients.

[15] Basic QSETH assumption is weaker than the QSETH assumption in Aaronson et al [1], so our lower bound under basic QSETH assumption (Conjecture 2 and Corollary 6) will also imply a lower bound under their quantum SETH framework.

Additionally, extending this result to majority / parity for AC^0_2-QSETH, i.e. CNF or DNF input, would be another step towards grounding the (necessary) assumption that such properties are compression oblivious.

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. In *Proceedings of the 35th Computational Complexity Conference*, CCC '20, Dagstuhl, DEU, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2020.16.

2 Divesh Aggarwal, Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. Fine-grained hardness of CVP(P) - everything that we can prove (and nothing else). In *SODA*, pages 1816–1835. SIAM, 2021. doi:10.1137/1.9781611976465.109.

3 Divesh Aggarwal and Eldon Chung. A note on the concrete hardness of the shortest independent vector in lattices. *Inf. Process. Lett.*, 167:106065, 2021. doi:10.1016/j.ipl.2020.106065.

4 Divesh Aggarwal and Rajendra Kumar. Why we couldn't prove SETH hardness of the closest vector problem for even norms, and of the subset sum problem! *CoRR*, abs/2211.04385, 2022. doi:10.48550/arXiv.2211.04385.

5 Divesh Aggarwal and Noah Stephens-Davidowitz. (gap/s)eth hardness of SVP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 228–238. ACM, 2018. doi:10.1145/3188745.3188840.

6 Shyan Akmal and R. Ryan Williams. MAJORITY-3SAT (and related problems) in polynomial time. *CoRR*, abs/2107.02748, 2021. arXiv:2107.02748.

7 Andris Ambainis, Harry Buhrman, Koen Leijnse, Subhasree Patro, and Florian Speelman. Matching triangles and triangle collection: Hardness based on a weak quantum conjecture, 2022. doi:10.48550/arXiv.2207.11068.

8 Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Average-case fine-grained hardness. In *STOC*, pages 483–496. ACM, 2017. doi:10.1145/3055399.3055466.

9 Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Proofs of work from worst-case assumptions. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 789–819, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-96884-1_26.

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

11 Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. On the quantitative hardness of CVP. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 13–24. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.11.

12 Huck Bennett and Chris Peikert. Hardness of bounded distance decoding on lattices in ℓ_p norms. In *35th Computational Complexity Conference, CCC*, volume 169 of *LIPIcs*, pages 36:1–36:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.36.

13 Huck Bennett, Chris Peikert, and Yi Tang. Improved hardness of BDD and SVP under gap-(s)eth. In *13th Innovations in Theoretical Computer Science Conference, ITCS*, volume 215 of *LIPIcs*, pages 19:1–19:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.19.

14 Harry Buhrman, Bruno Loff, Subhasree Patro, and Florian Speelman. Limits of quantum speed-ups for computational geometry and other problems: Fine-grained complexity via quantum walks. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 31:1–31:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ITCS.2022.31.

15 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. doi:10.4230/LIPIcs.STACS.2021.19.

16 C. Calabro, R. Impagliazzo, and R. Paturi. A duality between clause width and clause density for sat. In *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, pages 7 pp.–260, 2006. doi:10.1109/CCC.2006.6.

17 Yanlin Chen, Yilei Chen, Rajendra Kumar, Subhasree Patro, and Florian Speelman. QSETH strikes again: Finer quantum lower bounds for lattice problem, strong simulation, hitting set problem, and more. arXiv:2309.16431, 2023. doi:10.48550/arXiv.2309.16431.

18 Yanlin Chen, Yilei Chen, Rajendra Kumar, Subhasree Patro, and Florian Speelman. Fine-grained complexity via quantum natural proofs. arXiv:2504.10363, 2025. doi:10.48550/arXiv.2504.10363.

19 Jordan S. Cotler, Hsin-Yuan Huang, and Jarrod R. McClean. Revisiting dequantization and quantum advantage in learning tasks. *ArXiv*, abs/2112.00811, 2021. arXiv:2112.00811.

20 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 Transactions on Algorithms (TALG)*, 12(3):1–24, 2016. doi:10.1145/2925416.

21 Mina Dalirrooyfard, Andrea Lincoln, and Virginia Vassilevska Williams. New techniques for proving fine-grained average-case hardness. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 774–785, 2020. doi:10.1109/FOCS46700.2020.00077.

22 Holger Dell and John Lapinskas. Fine-grained reductions from approximate counting to decision. *ACM Trans. Comput. Theory*, 13(2):8:1–8:24, 2021. doi:10.1145/3442352.

23 Oded Goldreich and Guy N. Rothblum. Counting t-cliques: Worst-case to average-case reductions and direct interactive proof systems. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 77–88. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00017.

24 Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996. doi:10.1145/237814.237866.

25 Cupjin Huang, Michael Newman, and Mario Szegedy. Explicit lower bounds on strong quantum simulation. *IEEE Transactions on Information Theory*, 66(9):5585–5600, 2020. doi:10.1109/TIT.2020.3004427.

26 Jeremy Ahrens Huang, Young Kun Ko, and Chunhao Wang. On the (classical and quantum) fine-grained complexity of log-approximate cvp and max-cut. *arXiv preprint arXiv:2411.04124*, 2024. doi:10.48550/arXiv.2411.04124.

27 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.

28 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.

29 Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo, and Joel I. Seiferas, editors, *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, pages 193–206, 1983. doi:10.1145/800061.808749.

30 Rio LaVigne, Andrea Lincoln, and Virginia Vassilevska Williams. Public-key cryptography in the fine-grained setting. In *CRYPTO (3)*, volume 11694 of *Lecture Notes in Computer Science*, pages 605–635. Springer, 2019. doi:10.1007/978-3-030-26954-8_20.

31 Ashwin Nayak and Felix Wu. The quantum query complexity of approximating the median and related statistics. In Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton, editors, *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 384–393. ACM, 1999. doi:10.1145/301250.301349.

32 Ramamohan Paturi. On the degree of polynomials that approximate symmetric boolean functions (preliminary version). In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, pages 468–474, New York, NY, USA, 1992. Association for Computing Machinery. doi:10.1145/129712.129758.

33 Daan Schoneveld. Quantum fine-grained complexity: Hitting-set and related problems. Bachelors thesis, Universiteit van Amsterdam, 2022.

34 Rainer Schuler. An algorithm for the satisfiability problem of formulas in conjunctive normal form. *J. Algorithms*, 54(1):40–44, January 2005. doi:10.1016/j.jalgor.2004.04.012.

35 Maarten Van Den Nes. Classical simulation of quantum computation, the gottesman-knill theorem, and slightly beyond. *Quantum Info. Comput.*, 10(3):258–271, March 2010. doi:10.26421/QIC10.3-4-6.

36 Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk). In *10th International Symposium on Parameterized and Exact Computation (IPEC 2015)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.IPEC.2015.17.

37 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the international congress of mathematicians: Rio de janeiro 2018*, pages 3447–3487. World Scientific, 2018.