Electronic Colloquium on Computational Complexity, Report No. 117 (2023)

Space-bounded quantum state testing
via space-efficient quantum singular value transformation

François Le Gall[*], Yupan Liu[†], Qisheng Wang[‡]

Graduate School of Mathematics, Nagoya University

Abstract

Driven by exploring the power of quantum computation with a limited number of qubits, we present a novel complete characterization for space-bounded quantum computation, which encompasses settings with one-sided error (unitary coRQL) and two-sided error (BQL), approached from a quantum (mixed) state testing perspective:

- The *first* family of natural complete problems for unitary coRQL, namely *space-bounded quantum state certification* for trace distance and Hilbert-Schmidt distance;
- A new family of (arguably simpler) natural complete problems for BQL, namely *space-bounded quantum state testing* for trace distance, Hilbert-Schmidt distance, and (von Neumann) entropy difference.

In the space-bounded quantum state testing problem, we consider two logarithmic-qubit quantum circuits (devices) denoted as Q_0 and Q_1, which prepare quantum states ρ_0 and ρ_1, respectively, with access to their "source code". Our goal is to decide whether ρ_0 is ϵ_1-close to or ϵ_2-far from ρ_1 with respect to a specified distance-like measure. Interestingly, unlike time-bounded state testing problems, which exhibit computational hardness depending on the chosen distance-like measure (either QSZK-complete or BQP-complete), our results reveal that the space-bounded state testing problems, considering all three measures, are computationally as easy as preparing quantum states.

Our results primarily build upon *a space-efficient variant* of the quantum singular value transformation (QSVT) introduced by Gilyén, Su, Low, and Wiebe (STOC 2019), which is of independent interest. Our technique provides a unified approach for designing space-bounded quantum algorithms. Specifically, we show that implementing QSVT for any bounded polynomial that approximates a piecewise-smooth function incurs only a constant overhead in terms of the space required for (special forms of) the projected unitary encoding.

[*]Email: legall@math.nagoya-u.ac.jp
[†]Email: yupan.liu.e6@math.nagoya-u.ac.jp
[‡]Email: QishengWang1994@gmail.com

ISSN 1433-8092

Contents

1 Introduction

In recent years, exciting experimental advancements in quantum computing have been achieved, but concerns about their scalability persist. It thus becomes essential to characterize the computational power of feasible models of quantum computation that operate under restricted resources, such as *time* (i.e., the number of gates in the circuit) and *space* (i.e., the number of qubits on which the circuit acts). This paper specifically focuses on the latter aspect: what is the computational power of quantum computation with a limited number of qubits?

Previous studies [Wat99, Wat03, vMW12] on complete problems of space-bounded quantum computation have primarily focused on well-conditioned versions of standard linear-algebraic problems [TS13, FL18, FR21] and have been limited to the two-sided error scenario. In contrast, we propose a novel family of complete problems that not only characterize the *one-sided error scenario (and extend to the two-sided scenario)* but also arise from a quantum property testing perspective. Our new complete problems are arguably more natural and simpler, driven by recent intriguing challenges of verifying the intended functionality of quantum devices.

Consider the situation where a quantum device is designed to prepare a quantum (mixed) state ρ_0, but a possibly malicious party could provide another quantum device that outputs a different n-qubit (mixed) state ρ_1, claiming that $\rho_0 \approx_\epsilon \rho_1$. The problem of testing whether ρ_0 is ϵ_1-close to or ϵ_2-far from ρ_1 with respect to a specified distance-like measure, given the ability to produce copies of ρ_0 and ρ_1, is known as *quantum state testing* [MdW16, Section 4]. Quantum state testing (resp., distribution testing) typically involves utilizing sample accesses to quantum states ρ_0 and ρ_1 (resp., distributions D_0 and D_1) and determining the number of samples required to test the closeness between quantum states (resp., distributions). This problem is a quantum (non-commutative) generalization of classical property testing, which is a fundamental problem in theoretical computer science (see [Gol17]), specifically (tolerant) distribution testing (see [Can20]). Moreover, this problem is an instance of the emerging field of quantum property testing (see [MdW16]), which aims at designing quantum testers for the properties of quantum objects.

In this paper, we investigate quantum state testing problems where quantum states ρ_0 and ρ_1 are preparable by *computationally constrained resources*, specifically state-preparation circuits (viewed as the "source code" of devices) that are *(log)space-bounded*. Our main result conveys a conceptual message that testing quantum states prepared in bounded space is (computationally) as *easy* as preparing these states in a space-bounded manner. Consequently, we can introduce the first family of natural $\mathsf{coRQ_UL}$-complete promise problems since Watrous [Wat01] introduced unitary RQL and coRQL (known as $\mathsf{RQ_UL}$ and $\mathsf{coRQ_UL}$, respectively) in 2001, as well as a new family of natural BQL-complete promise problems.

Our main technique is a *space-efficient variant* of the quantum singular value transformation (QSVT) [GSLW19], distinguishing itself from prior works primarily focused on time-efficient QSVT. As time-efficient QSVT provides a unified framework for designing time-efficient quantum algorithms [GSLW19, MRTC21], we believe our work indicates a unified approach to designing space-bounded quantum algorithms, potentially facilitating the discovery of new complete problems for BQL and its one-sided error variants. Subsequently, we will first state our main results and then provide justifications for the significance of our results from various perspectives.

1.1 Main results

We will commence by providing definitions for time- and space-bounded quantum circuits. We say that a quantum circuit Q is (*poly*)*time-bounded* if Q is polynomial-size and acts on $\mathrm{poly}(n)$ qubits. Likewise, we say that a quantum circuit Q is (*log*)*space-bounded* if Q is polynomial-size and acts on $O(\log n)$ qubits. It is worthwhile to note that primary complexity classes, e.g., BQL, $\mathsf{coRQ_UL}$, and BPL, mentioned in this paper correspond to *promise problems*.

Complete characterizations of quantum logspace from state testing. While prior works [TS13, FL18, FR21] on BQL-complete problems have mainly focused on well-conditioned versions of standard linear-algebraic problems (in DET^*), our work takes a different perspective by exploring *quantum property testing*. Specifically, we investigate the problem of *space-bounded quantum state testing*, which aims to test the closeness between two quantum states that are preparable by (log)space-bounded quantum circuits (devices), with access to the corresponding "source code" of these devices.

We begin by considering a computational problem that serves as a "white-box" space-bounded counterpart of *quantum state certification* [BOW19], equivalent to quantum state testing with one-sided error. Our first main theorem (Theorem 1.1) demonstrates the *first* family of natural $\mathsf{coRQ_UL}$-complete problems in the context of space-bounded quantum state certification with respect to the trace distance (td) and the squared Hilbert-Schmidt distance (HS^2).

Theorem 1.1 (Informal of Theorem 4.5)**.** *The following (log)space-bounded quantum state certification problems are* $\mathsf{coRQ_UL}$*-complete: for any* $\alpha(n) \geq 1/\operatorname{poly}(n)$*, decide whether*

(1) $\overline{\textsc{CertQSD}}_{\log}$: $\rho_0 = \rho_1$ *or* $\mathrm{td}(\rho_0, \rho_1) \geq \alpha(n)$*;*

(2) $\overline{\textsc{CertQHS}}_{\log}$: $\rho_0 = \rho_1$ *or* $\mathrm{HS}^2(\rho_0, \rho_1) \geq \alpha(n)$*;*

By extending the error requirement from one-sided to two-sided, we broaden the scope of space-bounded quantum state testing to include two more distance-like measures: the quantum entropy difference, denoted by $\mathrm{S}(\rho_0) - \mathrm{S}(\rho_1)$, and the quantum Jensen-Shannon divergence (QJS_2). As a result, we establish our second main theorem, introducing a new family of natural BQL-complete problems:

Theorem 1.2 (Informal of Theorem 4.6)**.** *The following (log)space-bounded quantum state testing problems are* BQL*-complete: for any* $\alpha(n)$ *and* $\beta(n)$ *such that* $\alpha(n) - \beta(n) \geq 1/\operatorname{poly}(n)$*, or for any* $g(n) \geq 1/\operatorname{poly}(n)$*, decide whether*

(1) $\textsc{GapQSD}_{\log}$: $\mathrm{td}(\rho_0, \rho_1) \geq \alpha(n)$ *or* $\mathrm{td}(\rho_0, \rho_1) \leq \beta(n)$*;*

(2) $\textsc{GapQED}_{\log}$: $\mathrm{S}(\rho_0) - \mathrm{S}(\rho_1) \geq g(n)$ *or* $\mathrm{S}(\rho_1) - \mathrm{S}(\rho_0) \geq g(n)$*;*

(3) $\textsc{GapQJS}_{\log}$: $\mathrm{QJS}_2(\rho_0, \rho_1) \geq \alpha(n)$ *or* $\mathrm{QJS}_2(\rho_0, \rho_1) \leq \beta(n)$*;*

(4) $\textsc{GapQHS}_{\log}$: $\mathrm{HS}^2(\rho_0, \rho_1) \geq \alpha(n)$ *or* $\mathrm{HS}^2(\rho_0, \rho_1) \leq \beta(n)$*;*

Notably, Theorem 1.2(1) demonstrates that our algorithm for $\textsc{GapQSD}_{\log}$ exhibits a *polynomial advantage* in space over the best-known classical algorithms [Wat02], since Watrous implicitly showed in [Wat02, Proposition 21] that $\textsc{GapQSD}_{\log}$ is contained in (classical) polylogarithmic space.[1]

Space-efficient quantum singular value transformation. Proving our main theorems mentioned above poses a significant challenge: establishing the containment in the relevant class (BQL or $\mathsf{coRQ_UL}$), which is also the difficult direction for showing the known family of BQL-complete problems [TS13, FL18, FR21].

Proving the containment for the one-sided error scenario is not an effortless task: such a task is not only already relatively complicated for $\overline{\textsc{CertQHS}}_{\log}$, but also additionally requires novel techniques for $\overline{\textsc{CertQSD}}_{\log}$. On the other hand, for two-sided error scenarios, while showing the containment is straightforward for $\textsc{GapQHS}_{\log}$, it still demands sophisticated techniques for all other problems, such as $\textsc{GapQSD}_{\log}$, $\textsc{GapQED}_{\log}$, and $\textsc{GapQJS}_{\log}$.

[1] Notably, our algorithm for $\textsc{GapQSD}_{\log}$ provides an alternating proof for the original statement that (α, β)-QSD is in PSPACE when $\alpha(n) - \beta(n) \geq \exp(-\operatorname{poly}(n))$. In particular, Watrous [Wat02] provided an algorithm in NC to solve the Trace Norm Approximation problem on estimating $\|X\|_1$ with polynomial precision, given that the polynomial-size matrix X enables evaluation of all entries in deterministic $O(\log n)$ space.

As explained in Section 1.4, our primary technical contribution and proof technique involve developing a space-efficient variant of the quantum singular value transformation (QSVT), which constitutes our third main theorem (Theorem 1.4).

1.2 Background on space-bounded quantum computation

Watrous [Wat99, Wat03] initiated research on space-bounded quantum computation and showed that fundamental properties, including closure under complement, hold for $\mathsf{BQSPACE}[s(n)]$ with $s(n) \geq \Omega(\log n)$. Watrous also investigated classical simulations of space-bounded quantum computation (with unbounded error), presenting deterministic simulations in $O(s^2(n))$ space and unbounded-error randomized simulations in $O(s(n))$ space. A decade later, van Melkebeek and Watson [vMW12] provided a simultaneous $\tilde{O}(t(n))$ time and $O(s(n) + \log t(n))$ space unbounded-error randomized simulation for a bounded-error quantum algorithm in $t(n)$ time and $s(n)$ space. The complexity class corresponding to space-bounded quantum computation with $s(n) = \Theta(\log(n))$ is known as BQL, or $\mathsf{BQ_UL}$ if only *unitary* gates are permitted.

Significantly, several developments over the past two decades have shown that BQL is well-defined, independent of the following factors in chronological order:

- **The choice of gateset**. The Solovey-Kitaev theorem [Kit97] establishes that most quantum classes are gateset-independent, given that the gateset is closed under adjoint and all entries in gates have reasonable precision. The work of [vMW12] presented a space-efficient counterpart of the Solovay-Kitaev theorem, implying that BQL is also *gateset-independent*.

- **Error reduction**. Repeating $\mathsf{BQ_UL}$ sequentially necessitates reusing the workspace, making it unclear how to reduce errors for $\mathsf{BQ_UL}$ as intermediate measurements are not allowed. To address this issue, the work of [FKL+16] adapted the witness-preserving error reduction for QMA [MW05] with several other ideas to the space-efficient setting.

- **Intermediate measurements**. In the space-bounded scenario, the principle of deferred measurement is not applicable since this approach leads to an exponential increase in space complexity. Initially, BQL appeared to be seemingly more powerful than $\mathsf{BQ_UL}$ since we cannot directly demonstrate that $\mathsf{BPL} \subseteq \mathsf{BQ_UL}$. Recently, Fefferman and Remscrim [FR21] (as well as [GRZ21, GR22]) proved the equivalence between BQL and $\mathsf{BQ_UL}$, indicating a space-efficient approach to eliminating intermediate measurements.

BQL-complete problems. Identifying natural complete problems for the class BQL (or $\mathsf{BQ_UL}$) is a crucial and intriguing question. Ta-Shma [TS13] proposed the first candidate BQL-complete problem, building upon the work of Harrow, Hassidim, and Lloyd [HHL09] which established a BQP-complete problem for inverting a (polynomial-size) well-conditioned matrix. Specifically, Ta-Shma showed that inverting a well-conditioned matrix with polynomial precision is in BQL. Similarly, computing eigenvalues of an Hermitian matrix is also in BQL. These algorithms offer a quadratic space advantage over the best-known classical algorithms that saturate the classical simulation bound [Wat99, Wat03, vMW12]. Fefferman and Lin [FL18] later improved upon this result to obtain the first natural $\mathsf{BQ_UL}$-complete problem by ingeniously utilizing amplitude estimation to avoid intermediate measurements.

More recently, Fefferman and Remscrim [FR21] further extended this natural $\mathsf{BQ_UL}$-complete problem (or BQL-complete, equivalently) to a *family* of natural BQL-complete problems. They showed that a well-conditioned version of standard DET^*-complete problems is BQL-complete, where DET^* denotes the class of problems that are NC^1 (Turing) reducible to INTDET, including well-conditioned integer determinant (DET), well-conditioned matrix powering (MATPOW), and well-conditioned iterative matrix product (ITMATPROD), among others.

$\mathsf{RQ_UL}$- and $\mathsf{coRQ_UL}$-complete problems. Watrous [Wat01] introduced the one-sided error counterpart of $\mathsf{BQ_UL}$, namely $\mathsf{RQ_UL}$ and $\mathsf{coRQ_UL}$, and developed error reduction techniques.

Moreover, Watrous proved that the undirected graph connectivity problem (USTCON) is in $\mathsf{RQ_UL} \cap \mathsf{coRQ_UL}$ whereas Reingold [Rei08] demonstrated that USTCON is in L several years later. Recently, Fefferman and Remscrim [FR21] proposed a "verification" version of the well-conditioned iterative matrix product problem (vITMATPROD) as a *candidate* coRQL-complete problem. However, although this problem is known to be coRQL-hard, its containment remains *unresolved*. Specifically, vITMATPROD requires to decide whether a single entry in the product of polynomially many well-conditioned matrices is equal to zero.

1.3 Time-bounded and space-bounded distribution and state testing

We summarize prior works and our main results for time-bounded[2] and space-bounded distribution and state testing with respect to ℓ_1 norm, entropy difference, and ℓ_2 norm in Table 1.

Interestingly, the sample complexity of testing the closeness of quantum states (resp., distributions) depends on the choice of distance-like measures,[3] including the one-sided error counterpart known as *quantum state certification* [BOW19]. In particular, for distance-like measures such as the ℓ_1 norm, called total variation distance in the case of distributions [CDVV14] and trace distance in the case of states [BOW19], as well as classical entropy difference [JVHW15, WY16] and its quantum analog [AISW20, OW21], the sample complexity of distribution and state testing is polynomial in the dimension N. However, for distance-like measures such as the ℓ_2 norm, called Euclidean distance in the case of distributions [CDVV14] and Hilbert-Schmidt distance in the case of states [BOW19], the sample complexity is *independent* of dimension N.

	ℓ_1 norm	ℓ_2 norm	Entropy
Classical Time-bounded	SZK-complete[5] [SV03, GSV98]	BPP-complete Folklore	SZK-complete [GV99, GSV98]
Quantum Time-bounded	QSZK-complete[6] [Wat02, Wat09]	BQP-complete [BCWdW01, RASW23]	QSZK-complete [BASTS10, Wat09]
Quantum Space-bounded	BQL-complete Theorem 1.2(1)	BQL-complete [BCWdW01] and Theorem 1.2(4)	BQL-complete Theorem 1.2(2)

Table 1: Time- and space-bounded distribution or state testing.

As depicted in Table 1, this phenomenon that the required sample complexity for distribution and state testing, with polynomial precision and exponential dimension, depends on the choice of distance-like measure has reflections on time-bounded quantum state testing:

- For ℓ_1 norm and entropy difference, the time-bounded scenario is *seemingly much harder than* preparing states or distributions since $\mathsf{QSZK} \subseteq \mathsf{BQP}$ and $\mathsf{SZK} \subseteq \mathsf{BPP}$ are unlikely.
- For ℓ_2 norm, the time-bounded scenario is *as easy as* preparing states or distributions.

However, interestingly, a similar phenomenon *does not appear* for space-bounded quantum state testing. Although no direct classical counterpart has been investigated before in a complexity-theoretic fashion, namely space-bounded distribution testing, there is another closely related model (a version of streaming distribution testing) that does not demonstrate an analogous phenomenon either, as we will discuss in Section 1.3.2.

[2]The problem of *time-bounded distribution (resp., state) testing* aims to test the closeness between two distributions (resp., states) that are preparable by (poly)time-bounded circuits (devices), with access to the corresponding "source code" of these devices.

[3]It is noteworthy that the quantum entropy difference is not a distance.

1.3.1 Time-bounded distribution and state testing

We review prior works on time-bounded state (resp., distribution) testing, with a particular focus on testing the closeness between states (resp., distributions) are preparable by (poly)time-bounded quantum (resp., classical) circuits (device), with access to the "source code" of corresponding devices. For time-bounded distribution testing, we also recommend a brief survey [GV11] by Goldreich and Vadhan.

ℓ_1 norm scenarios. Sahai and Vadhan [SV03] initiated the study of the time-bounded distribution testing problem, where distributions D_0 and D_1 are *efficiently samplable*, and the distance-like measure is the total variation distance. Their work named this problem Statistical Difference (SD). In particular, the promise problem (α, β)-SD asks whether D_0 is α-far from or β-close to D_1 with respect to $\|D_0 - D_1\|_{\mathrm{TV}}$. Although sampling from the distribution is in BPP,[4] testing the closeness between these distributions is SZK-complete [SV03, GSV98], where SZK is the class of promise problems possessing statistical zero-knowledge proofs. It is noteworthy that the SZK containment of (α, β)-SD for any $\alpha(n) - \beta(n) \geq 1/\operatorname{poly}(n)$ is currently unknown.[5] In addition, we note that SZK is contained in $\mathsf{AM} \cap \mathsf{coAM}$ [For87, AH91].

Following the pioneering work [SV03], Watrous [Wat02] introduced the time-bounded quantum state testing problem, where two quantum states ρ_0 and ρ_1 that are preparable by time-bounded quantum circuits Q_0 and Q_1, respectively, as well as the distance-like measure is the trace distance. This problem is known as the Quantum State Distinguishability (QSD), specifically, (α, β)-QSD asks whether ρ_0 is α-far from or β-close to ρ_1 with respect to $\mathrm{td}(\rho_0, \rho_1)$. Analogous to its classical counterpart, QSD is QSZK-complete [Wat02, Wat09], whereas the QSZK containment for any $\alpha(n) - \beta(n) \geq 1/\operatorname{poly}(n)$ remains an open question.[6] Additionally, it is worth noting that $\mathsf{QIP}(2)$ contains QSZK [Wat02, Wat09].

Entropy difference scenarios. Beyond ℓ_1 norm, another distance-like measure commonly considered in time-bounded quantum state testing (or distribution testing) is the (quantum) entropy difference, which also corresponds to the (quantum) Jensen-Shannon divergence. The promise problem Entropy Difference (ED), first introduced by Goldreich and Vadhan [GV99] following the work of [SV03], asks whether efficiently samplable distributions D_0 and D_1 satisfy $\mathrm{H}(D_0) - \mathrm{H}(D_1) \geq g$ or $\mathrm{H}(D_1) - \mathrm{H}(D_0) \geq g$ for $g = 1$. They demonstrated that ED is SZK-complete. Ben-Aroya, Schwartz, and Ta-Shma [BASTS10] further investigated the promise problem Quantum Entropy Difference (QED), which asks whether $\mathrm{S}(\rho_0) - \mathrm{S}(\rho_1) \geq g$ or $\mathrm{S}(\rho_1) - \mathrm{S}(\rho_0) \geq g$, for efficiently preparable quantum states ρ_0 and ρ_1 and $g = 1/2$. They showed that QED is QSZK-complete. Moreover, the SZK (resp., QSZK) containment for ED (resp., QED) automatically holds for any $g(n) \geq 1/\operatorname{poly}(n)$.

Furthermore, Berman, Degwekar, Rothblum, and Vasudevan [BDRV19] demonstrated that the Jensen-Shannon divergence problem (JSP), asking whether $\mathrm{JS}(D_0, D_1) \geq \alpha$ or $\mathrm{JS}(D_0, D_1) \leq \beta$ for efficiently samplable distributions D_0 and D_1, is SZK-complete. Their work accomplished this result by reducing the problem to ED, and this containment applies to $\alpha(n) - \beta(n) \geq 1/\operatorname{poly}(n)$. Recently, Liu [Liu23] showed a quantum counterpart, referred to as the Quantum Jensen-Shannon Divergence Problem (QJSP), is QSZK-complete. Notably, the quantum

[4] Rigorously speaking, as an instance in SD, sample-generating circuits are not necessarily (poly)time-uniform.

[5] The works of [SV03, GSV98] demonstrated that (α, β)-SD is in SZK for any constant $\alpha^2 - \beta > 0$. The same technique works for the parameter regime $\alpha^2(n) - \beta(n) \geq 1/O(\log n)$. However, further improvement of the parameter regime requires new ideas, as clarified in [Gol19]. Recently, the work of [BDRV19] improved the parameter regime to $\alpha^2(n) - \beta(n) \geq 1/\operatorname{poly}(n)$ by utilizing a series of tailor-made reductions. Currently, we only know that (α, β)-SD for $\alpha(n) - \beta(n) \geq 1/\operatorname{poly}(n)$ is also in $\mathsf{AM} \cap \mathsf{coAM}$ [BL13].

[6] Like SD and SZK, the techniques in [Wat02, Wat09] show that (α, β)-QSD is in QSZK for $\alpha^2(n) - \beta(n) \geq 1/O(\log n)$, and the same limitation also applies to the quantum settings. A recent result [Liu23] following the line of work of [BDRV19] improved the parameter regime to $\alpha^2(n) - \sqrt{2 \ln 2}\beta(n) \geq 1/\operatorname{poly}(n)$, but the differences between classical and quantum distances make it challenging to push the bound further. In [Wat02, Proposition 21], Watrous implicitly proved a PSPACE upper bound for the parameter regime $\alpha(n) - \beta(n) \geq \exp(-\operatorname{poly}(n))$.

Jensen-Shannon divergence is a special instance of the Holevo χ quantity [Hol73].[7]

ℓ_2 norm scenarios. For the quantum setting, it is straightforward that applying the SWAP test [BCWdW01][8] to efficiently preparable quantum states ρ_0 and ρ_1 can lead to a BQP containment, in particular, additive-error estimations of $\mathrm{Tr}(\rho_0^2)$, $\mathrm{Tr}(\rho_1^2)$, and $\mathrm{Tr}(\rho_0\rho_1)$ with polynomial precision. Recently, the work of [RASW23] observed that time-bounded quantum state testing with respect to the squared Hilbert-Schmidt distance is BQP-complete. For the classical setting, namely the squared Euclidean distance, the BPP-completeness is relatively effortless.[9]

1.3.2 Space-bounded distribution and state testing

To the best of our knowledge, no prior work has specifically focused on space-bounded distribution testing from a complexity-theoretic perspective. Instead, we will review prior works that are (closely) related to this computational problem. Afterward, we will delve into space-bounded quantum state testing, which constitutes the main contribution of our work.

Space-bounded distribution testing and related works. We focus on a computational problem involving two $\mathrm{poly}(n)$-size classical circuits C_0 and C_1, which generate samples from the distributions D_0 and D_1 respectively. Each circuit contains a read-once polynomial-length random-coins tape.[10] The input length and output length of the circuits are $O(\log n)$. The task is to decide whether D_0 is α-far from or β-close to D_1 with respect to some distance-like measure. Additionally, we can easily observe that space-bounded distribution testing with respect to the squared Euclidean distance (ℓ_2 norm) is BPL-complete, much like its time-bounded counterpart.

Several models related to space-bounded distribution testing have been investigated previously. Earlier streaming-algorithmic works [FKSV02, GMV06] utilize *entries* of the distribution as the data stream, with entries given in different orders for different models. On the other hand, a later work [CLM10] considered a data stream consisting of a sequence of i.i.d. samples drawn from distributions and studied low-space streaming algorithms for distribution testing.

Regarding (Shannon) entropy estimation, previous streaming algorithms considered worst-case ordered samples drawn from N-dimensional distributions and required $\mathrm{poly}\log(N/\epsilon)$ space, where ϵ is the additive error. Recently, Acharya, Bhadane, Indyk, and Sun [ABIS19] addressed the entropy estimation problem with i.i.d. samples drawn from distributions as the data stream and demonstrated the first $O(\log(N/\epsilon))$ space streaming algorithm. The sample complexity, viewed as the time complexity, was subsequently improved in [AMNW22].

However, for the total variation distance (ℓ_1 norm), previous works focused on the trade-off between the sample complexity and the space complexity (memory constraints), achieving only a nearly-log-squared space streaming algorithm [DGKR19].

Notably, the main differences between the computational and streaming settings lie in how we access the sampling devices.[11] In the computational problem, we have access to the "source code" of the devices and can potentially use them for purposes like "reverse engineering". Conversely, the streaming setting utilizes the sampling devices in a "black-box" manner, obtaining i.i.d. samples. As a result, a logspace streaming algorithm will result in a BPL containment.[12]

[7] In particular, the quantum Jensen-Shannon divergence coincides with the Holevo χ quantity on size-2 ensembles with a uniform distribution, which arises in the Holevo bound [Hol73]. See [NC02, Theorem 12.1].

[8] We note that the SWAP test also applies to mixed states, see Proposition 9 in [KMY09].

[9] Specifically, we achieve BPP containment by following the approach in [BCH+19, Theorem 7.1]. On the other hand, the BPP hardness owes to the fact that the squared Euclidean distance between the distribution $(p_{\mathrm{acc}}, 1-p_{\mathrm{acc}})$ from the output bit of any BPP algorithm and the distribution $(1,0)$ is $(1-p_{\mathrm{acc}})^2$.

[10] It is noteworthy that random coins are provided as *input* to classical circuits C_0 and C_1 for generating samples from the corresponding distributions in the time-bounded scenario, such as SD and ED.

[11] Of course, not all distributions can be described as a polynomial-size circuit (i.e., a succinct description).

[12] In particular, the sample-generating circuits C_0 and C_1 in space-bounded distribution testing can produce the i.i.d. samples in the data stream.

Space-bounded quantum state testing. Among the prior works on streaming distribution testing, particularly entropy estimation, the key takeaway is that the space complexity of the corresponding computational problem is $O(\log(N/\epsilon))$. This observation leads to a conjecture that the computational hardness of space-bounded distribution and state testing is *independent* of the choice of commonplace distance-like measures. Our work, in turn, provides a positive answer for space-bounded quantum state testing.

Space-bounded state testing with respect to the squared Hilbert-Schmidt distance (ℓ_2 norm) is BQL-complete, as shown in Theorem 1.2(4). Specifically, the BQL containment follows from the SWAP test [BCWdW01], similar to the time-bounded scenario. Moreover, proving BQL hardness, as well as $\mathsf{coRQ_U L}$-hardness for state certification, are also straightforward.[13]

Regarding space-bounded state testing with respect to the trace distance (ℓ_1 norm), we note that [Wat02, Proposition 21] implicitly established an NC containment. The BQL-hardness, as well as $\mathsf{coRQ_U L}$-hardness for state certification, is adapted from [RASW23]. Similarly, we derive the BQL-hardness for space-bounded state testing with respect to the quantum Jensen-Shannon divergence and the quantum entropy difference from previous works [Liu23].

Finally, we devote the remainder of this section to our main technique (Theorem 1.4), and consequently, we present BQL (resp., $\mathsf{coRQ_U L}$) containment for state testing (resp., certification) problems for other distance-like measures beyond the squared Hilbert-Schmidt distance.

1.4 Proof technique: Space-efficient quantum singular value transformation

The quantum singular value transformation (QSVT) [GSLW19] is a powerful and efficient framework for manipulating the singular values $\{\sigma_i\}_i$ of a linear operator A, using a corresponding projected unitary encoding U of $A = \tilde{\Pi} U \Pi$ for projectors $\tilde{\Pi}$ and Π. The singular value decomposition is $A = \sum_i \sigma_i |\tilde{\psi}_i\rangle\langle\psi_i|$ where $|\tilde{\psi}_i\rangle$ and $|\psi_i\rangle$ are left and right singular vectors, respectively. QSVT has numerous applications in quantum algorithm design, and is even considered a grand unification of quantum algorithms [MRTC21]. To implement the transformation $f^{(\mathrm{SV})}(A) = f^{(\mathrm{SV})}(\tilde{\Pi} U \Pi)$, we require a degree-$d$ polynomial $\hat{P}_d(x)$ that satisfies two conditions. Firstly, $\hat{P}_d$ well-approximates f on the interval of interest $\mathcal{I}$, with $\max_{x\in\mathcal{I}\setminus\mathcal{I}_\delta} |\hat{P}_d(x) - f(x)| \leq \epsilon$, where $\mathcal{I}_\delta \subseteq \mathcal{I} \subseteq [-1,1]$ and typically $\mathcal{I}_\delta := (-\delta,\delta)$. Secondly, $\hat{P}_d$ is bounded, with $\max_{x\in[-1,1]} |\hat{P}_d(x)| \leq 1$. The degree of $\hat{P}_d$ depends on the precision parameters δ and ϵ, with $d = O(\delta^{-1}\log\epsilon^{-1})$, and all coefficients of $\hat{P}_d$ can be computed efficiently.

According to [GSLW19], we can use an alternating phase modulation to implement $\hat{P}_d^{(\mathrm{SV})}(\tilde{\Pi} U \Pi)$,[14] which requires a sequence of rotation angles $\Phi \in \mathbb{R}^d$. For instance, consider $\hat{P}_d(x) = T_d(x)$ where $T_d(x)$ is the d-th Chebyshev polynomial (of the first kind), then we know that $\phi_1 = (1-d)\pi/2$ and $\phi_j = \pi/2$ for all $j \in \{2,3,\cdots,d\}$. QSVT techniques, including classical pre-processing and quantum circuit implementation, are generally *time-efficient*. Additionally, the quantum circuit implementation of QSVT is already *space-efficient* because implementing QSVT with a degree-d bounded polynomial for any $s(n)$-qubit projected unitary encoding requires $O(s(n))$ qubits, where $s(n) \geq \Omega(\log n)$. However, the classical pre-processing in the QSVT techniques is typically not space-efficient. Indeed, prior works on classical pre-processing for QSVT, specifically angle-finding algorithms in [Haa19, CDG+20, DMWL21], which have time complexity polynomially dependent on the degree d, do not consider the space-efficiency. Therefore, the use of previous angle-finding algorithms may lead to an *exponential* increase in space complexity. This raises a fundamental question on making the classical pre-processing space-efficient as well:

Problem 1.3 (Space-efficient QSVT)**.** Can we implement a degree-d QSVT for any $s(n)$-qubit

[13] In particular, considering any BQL circuit C_x that accepts with probability $p_{\mathrm{acc}} = \||1\rangle\langle 1|_{\mathrm{out}} C_x|\bar{0}\rangle\|_2^2$, we can construct a new circuit C'_x from C_x such that C'_x accepts with probability $\||\bar{0}\rangle\langle\bar{0}|C'_x|\bar{0}\rangle|2^2 = pacc^2 = \mathrm{Tr}(\rho_0\rho_1) = 1 - \mathrm{HS}^2(\rho_0,\rho_1)$, where pure states $\rho_0 = |\bar{0}\rangle\langle\bar{0}|$ and $\rho_1 = C'_x|\bar{0}\rangle\langle\bar{0}|{C'_x}^\dagger$. See Lemma 4.17 for details.

[14] This procedure is a generalization of quantum signal processing, as explained in [MRTC21, Section II.A].

projected unitary encoding with $d \leq 2^{O(s(n))}$, using only $O(s(n))$ space in both classical pre-processing and quantum circuit implementation?

QSVT via Chebyshev interpolation. Recently, Metger and Yuen [MY23] constructed bounded polynomial approximations of the sign and square root functions with exponential precision in polynomial space by utilizing Chebyshev interpolation, which offers a partial solution to Problem 1.3.[15] The key ingredient behind their approach is the *near-minimax approximation by Chebyshev interpolation* [Pow67]. More precisely, for any continuous function $f\colon [-1,1] \to \mathbb{R}$, if there is a degree-$d$ polynomial $\hat{P}_d$ satisfying $\max_{x\in[-1,1]} |f(x) - \hat{P}_d(x)| \leq \epsilon$, then we have a Chebyshev interpolation polynomial $P_d(x) := \frac{c_0}{2} + \sum_{k=1}^d c_k T_k$ where $c_k := \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}}\mathrm{d}x$ such that $\max_{x\in[-1,1]} |P_d(x) - f(x)| \leq O(\epsilon \log d)$. As the angles for any Chebyshev polynomial $T_k(x)$ are explicitly known, the implementation involves applying a Chebyshev polynomial to a *bitstring indexed encoding*, which additionally requires projectors $\tilde{\Pi}$ and Π span on the corresponding subset of $\{|0\rangle, |1\rangle\}^{\otimes s}$,[16] and implementing the Chebyshev interpolation polynomial by LCU techniques [BCC+15]. It is noteworthy that combining the aforementioned techniques causes a *super-quadratic dependence* of the degree d in the query complexity to U.

A refined analysis indicates that applying a Chebyshev interpolation polynomial to a bitstring indexed encoding for any $d \leq 2^{O(s(n))}$ and $\epsilon \geq 2^{-O(s(n))}$ requires $O(s(n))$ qubits and deterministic $O(s(n))$ space, provided that an evaluation oracle Eval_{P_d} estimates coefficients $\{c_k\}_{k=0}^d$ of the Chebyshev interpolation polynomial with $O(\log(\epsilon/d))$ precision. This result leads to the establishment of a space-efficient variant of QSVT:

Theorem 1.4 (Space-efficient QSVT, informal of Theorem 3.4)**.** *Let $f\colon \mathbb{R} \to \mathbb{R}$ be a continuous function bounded on $\mathcal{I} \subseteq [-1,1]$. If there exists a degree-d polynomial $\hat{P}_d$ that approximates $h\colon [-1,1] \to \mathbb{R}$, where h approximates f only on $\mathcal{I}$, such that $\max_{x\in[-1,1]} |h(x) - \hat{P}_d(x)| \leq \epsilon$, then Chebyshev interpolation yields another degree-d polynomial P_d satisfying the following conditions: $\max_{x\in\mathcal{I}} |f(x) - P_d(x)| \leq O(\epsilon \log d)$ and $\max_{x\in[-1,1]} |P_d(x)| \leq 1$.*
Furthermore, we have an algorithm $\mathcal{A}_f$ that computes any coefficient $\{c_k\}_{k=0}^d$ of the Chebyshev interpolation polynomial P_d space-efficiently. The algorithm is deterministic for bounded f, and bounded-error randomized for piecewise-smooth f. Additionally, for any $s(n)$-qubit bitstring indexed encoding U of $A = \tilde{\Pi}U\Pi$ with $d \leq 2^{O(s(n))}$, we can implement the quantum singular value transformation $P_d^{(\mathrm{SV})}(A)$ using $O(d^2\|\mathbf{c}\|_1)$ queries[17] In addition, $\|\mathbf{c}\|_1$ is generally upper-bounded by $O(d)$ for all piecewise-smooth functions. However, for specific functions, such as the sign function, we can improve the upper bound to $O(\log d)$. to U with $O(s(n))$ qubits.

Our techniques in Theorem 1.4 offer two advantages over the techniques proposed by [MY23]. Firstly, our techniques can handle any *piecewise-smooth function*, such as the normalized logarithmic function $\ln_\beta(x) := \frac{\ln(1/x)}{2\ln(2/\beta)}$ on the interval $\mathcal{I} = [\beta, 1]$ for any $\beta \geq 2^{-O(s(n))}$, whereas the techniques from [MY23] are restricted to functions that are bounded on the interval $\mathcal{I} = [-1,1]$. Secondly, our technique is *constant overhead* in terms of the space complexity of the bitstring indexed encoding U, while the techniques from [MY23] are only *poly-logarithmic overhead*.

In addition, it is noteworthy that applying the space-efficient QSVT with the sign function will imply a unified approach to error reduction for the classes $\mathsf{BQ_UL}$, $\mathsf{coRQ_UL}$ and $\mathsf{RQ_UL}$.

Computing the coefficients. We will implement the evaluation oracle Eval_{P_d} to prove Theorem 1.4. To estimate the coefficients $\{c_k\}_{k=0}^d$ resulting from Chebyshev interpolation for any

[15] To clarify, we can see from [MY23] that directly adapting their construction shows that implementing QSVT for any $s(n)$-qubit block-encoding with $O(s(n))$-bit precision requires $\mathrm{poly}(s(n))$ classical and quantum space for any $s(n) \geq \Omega(\log n)$. However, Problem 1.3 (space-efficient QSVT) seeks to reduce the dependence of $s(n)$ in the space complexity from *polynomial* to *linear*.

[16] To ensure that $\tilde{\Pi}U\Pi$ admits a matrix representation, we require the basis of projectors $\tilde{\Pi}$ and Π to have a well-defined order, leading us to focus exclusively on bitstring indexed encoding. Additionally, for simplicity, we assume no ancillary qubits are used here, and refer to Definition 3.1 for a formal definition.

[17] The dependence of $\|\mathbf{c}\|_1$ arises from renormalizing the bitstring indexed encoding via amplitude amplification.

function f that is bounded on the interval $\mathcal{I} = [-1, 1]$, we can use standard numerical integral techniques,[18] given that the integrand's second derivative in $\{c_k\}_{k=0}^{d}$ is bounded by $\mathrm{poly}(d)$.

However, implementing the evaluation oracle for piecewise-smooth functions f on an interval $\mathcal{I} \subsetneq [-1, 1]$ is relatively convoluted. We cannot simply apply Chebyshev interpolation to f. Instead, we consider a low-degree Fourier approximation g resulting from implementing smooth functions to Hamiltonians [vAGGdW20, Appendix B]. We then make the error vanish outside $\mathcal{I}$ by multiplying with a Gaussian error function, resulting in h which approximates f *only* on $\mathcal{I}$. Therefore, we can apply Chebyshev interpolation and our algorithm for bounded functions to h through a somewhat complicated calculation.

Finally, we need to compute the coefficients of the low-degree Fourier approximation g. Interestingly, this step involves the *stochastic matrix powering problem*, which lies at the heart of space-bounded derandomization, e.g., [SZ99, CDSTS23, PP23]. We utilize space-bounded random walks on a directed graph to estimate the power of a stochastic matrix. Consequently, we can only develop a bounded-error randomized algorithm $\mathcal{A}_f$ for piecewise-smooth functions.[19]

1.5 Proof overview: A general framework for quantum state testing

Our framework enables space-bounded quantum state testing, specifically for proving Theorem 1.1 and Theorem 1.2, and is based on the one-bit precision phase estimation [Kit95], also known as the *Hadamard test* [AJL09]. Prior works [TS13, FL18] have employed (one-bit precision) phase estimation in space-bounded quantum computation.

To address quantum state testing problems, we reduce them to estimating $\mathrm{Tr}(P_d(A)\rho)$, where ρ is a (mixed) quantum state prepared by a quantum circuit Q_ρ, A is an Hermitian operator block-encoded in a unitary operator U_A, and P_d is a space-efficiently computable degree-d polynomial. This approach has been applied in *time-bounded* quantum state testing, including fidelity estimation [GP22] and subsequently trace distance estimation [WZ23a].

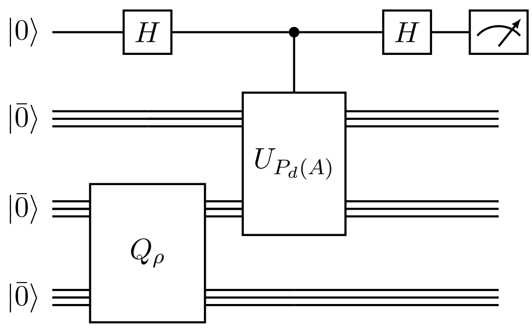

Figure 1: General framework for quantum state testing $\mathcal{T}(Q_\rho, U_A, P_d)$.

To implement a unitary operator $U_{P_d(A)}$ that (approximately) block-encodes $P_d(A)$ in a space-efficient manner, we require P_d to meet the conditions specified in Theorem 1.4. As illustrated in Figure 1, we denote the quantum circuit as $\mathcal{T}(Q_\rho, U_A, P_d)$, where we exclude the precision for simplicity. The measurement outcome of $\mathcal{T}(Q_\rho, U_A, P_d)$ will be 0 with a probability close to $\frac{1+\mathrm{Tr}(P_d(A)\rho)}{2}$. This property allows us to estimate $\mathrm{Tr}(P_d(A)\rho)$ within an additive error ϵ using $O(1/\epsilon^2)$ sequential repetitions, resulting in a BQL containment.

As an example of the application, $\mathcal{T}(Q_i, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}})$ is utilized in GapQSD, where $U_{\frac{\rho_0-\rho_1}{2}}$ is a block-encoding of $\frac{\rho_0-\rho_1}{2}$, and P_d^{sgn} is a space-efficient polynomial approximation of the

[18] We remark that using a more efficient numerical integral technique, such as the exponentially convergent trapezoidal rule, may improve the required space complexity for computing coefficients by a constant factor.

[19] The classical pre-processing in space-efficient QSVT is not part of the deterministic Turing machine producing the quantum circuit description in the BQL model (Definition 2.6). Instead, we treat it as a component of quantum computation, allowing the use of randomized algorithms since $\mathsf{BPL} \subseteq \mathsf{BQL}$ [FR21].

sign function. Similarly, $\mathcal{T}(Q_i, U_{\rho_i}, P_d^{\ln})$ is utilized in GAPQED, where U_{ρ_i} is a block-encoding of ρ_i for $i \in \{0,1\}$, and $P_d^{\ln}$ is a space-efficient polynomial approximation of the normalized logarithmic function. Both P_d^{sgn} and $P_d^{\ln}$ can be obtained by employing Theorem 1.4.[20]

Making the error one-sided. The main challenge is constructing a unitary U of interest, such as $\mathcal{T}(Q_\rho, U_A, P_d)$, that accepts with a certain fixed probability p for *yes* instances ($\rho_0 = \rho_1$), while having a probability that polynomially deviates from p for *no* instances. As an example, we consider $\overline{\textsc{CertQHS}}_{\log}$ and express $\mathrm{HS}^2(\rho_0, \rho_1)$ as a linear combination of $\mathrm{Tr}(\rho_0^2)$, $\mathrm{Tr}(\rho_1^2)$, and $\mathrm{Tr}(\rho_0\rho_1)$. We thus design a unitary quantum algorithm employing the LCU technique, which accepts with probability $\left(\frac{1}{2} + \frac{1}{4}\mathrm{HS}^2(\rho_0,\rho_1)\right)^2$, equal $1/4$ for *yes* instances. Applying the exact amplitude amplification [BBHT98, BHMT02], we achieve perfect completeness, and the analysis demonstrates that the acceptance probability polynomially deviates from 1 for *no* instances. By applying error reduction for $\mathsf{coRQ_UL}$, the resulting algorithm is indeed in $\mathsf{coRQ_UL}$.

Moving on to $\overline{\textsc{CertQSD}}_{\log}$, we consider the quantum circuit $U_i = \mathcal{T}(Q_i, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}})$ for $i \in \{0,1\}$. Let p_i be the probability that the measurement outcome of $U_i|\bar{0}\rangle$ in Figure 1 is 0. Since our space-efficient QSVT preserves parity, specifically the approximation polynomial P_d^{sgn} satisfies $P_d^{\mathrm{sgn}}(0) = 0$,[21] we obtain $p_0 = p_1 = 1/2$ for *yes* instances ($\rho_0 = \rho_1$). With a simple modification, U_0 and U_1 enable algorithm $\mathcal{A}$ to meet the condition of exact amplitude amplification for *yes* instances. Further analysis shows that $\mathcal{A}$ accepts with probability polynomially away from 1 for *no* instances. We thus can conclude a $\mathsf{coRQ_UL}$ containment similar to $\overline{\textsc{CertQHS}}_{\log}$.

1.6 Discussion and open problems

Since space-efficient quantum singular value transformation (QSVT) offers a unified framework for designing quantum logspace algorithms, it suggests a new direction to find applications of space-bounded quantum computation. An intriguing candidate is solving positive semidefinite programming (SDP) programs with constant precision [JY11, AZLO16]. A major challenge in achieving a BQL containment for this problem is that iteratively applying the space-efficient QSVT super-constantly many times may lead to a bitstring indexed encoding requiring $\omega(\log n)$ ancillary qubits, raising the question:

(i) Is it possible to have an approximation scheme (possibly under certain conditions) that introduces merely $O(1)$ additional ancillary qubits in the bitstring indexed encoding per iteration, such that applying space-efficient QSVT $\log n$ times results in a bitstring indexed encoding with at most $O(\log n)$ ancillary qubits?

Furthermore, as quantum distances investigated in this work are all instances of a quantum analog of symmetric f-divergence, there is a natural question on other instances:

(ii) Can we demonstrate that space-bounded quantum state testing problems with respect to other quantum distances are also BQL-complete, such as quantum analogs of squared Hellinger distance or quantum analogs of triangular discrimination [Liu23]?

In addition, there is a question on improving the efficiency of the space-efficient QSVT:

(iii) Can we improve the query complexity of U and $U^\dagger$ in the space-efficient QSVT implementation (e.g., for the sign function) from $O(d^2 \log d)$ to $O(d)$?

Notably, classical pre-processing in QSVT techniques usually involves finding the sequence of z-axis rotation angles, while our approach instead uses Chebyshev interpolation and the LCU technique. A solution thus involves developing a space-efficient angle-finding algorithm. An interesting direction from a recent work [MW23], which investigated QSVT with SU(2) rotations, may shed light on Question (iii) since finding SU(2) rotation angles appears easier.

[20] In particular, P_d^{sgn} is given in Corollary 3.7, as well as $P_d^{\ln}$ is given in Corollary 3.10.

[21] Let f be any odd function such that space-efficient QSVT associated with f can be implemented by Theorem 1.4. It follows that the corresponding approximation polynomial $P_d^{(f)}$ is also odd. See Remark 3.12.

1.7 Related works: more on quantum state testing problems

Testing the spectrum of quantum states was studied in [OW21]: for example, whether a quantum state is maximally mixed or ϵ-far away in trace distance from mixed states can be tested using $\Theta(N/\epsilon^2)$ samples. Later, it was generalized in [BOW19] to quantum state certification with respect to fidelity and trace distance. Estimating distinguishability measures of quantum states [RASW23] is another topic, including the estimation of fidelity [FL11, WZC$^+$23, GP22] and trace distance [WGL$^+$22, WZ23a].

Entropy estimation of quantum states has been widely studied in the literature. Given quantum purified access, it was shown in [GL20] that the von Neumann entropy $\mathrm{S}(\rho)$ can be estimated within additive error ϵ with query complexity $\tilde{O}(N/\epsilon^{1.5})$. If we know the reciprocal κ of the minimum non-zero eigenvalue of ρ, then $\mathrm{S}(\rho)$ can be estimated with query complexity $\tilde{O}(\kappa^2/\epsilon)$ [CLW20]. We can estimate $\mathrm{S}(\rho)$ within multiplicative error ϵ with query complexity $\tilde{O}(n^{\frac{1}{2}+\frac{1+\eta}{2\epsilon^2}})$ [GHS21], provided that $\mathrm{S}(\rho) = \Omega(\epsilon + 1/\eta)$. If ρ is of rank r, then $\mathrm{S}(\rho)$ can be estimated with query complexity $\tilde{O}(r^2/\epsilon^2)$ [WGL$^+$22]. Estimating the Rényi entropy $S_\alpha(\rho)$ given quantum purified access was first studied in [GHS21], and then was improved in [WGL$^+$22, LWZ22]. In addition, the work of [GH20] investigates the (conditional) hardness of GapQED with logarithmic depth or constant depth.

Paper organization. Our paper begins by introducing key concepts in Section 2, including quantum distance and divergences, space-bounded quantum computation, Chebyshev polynomials and interpolation, and a toolkit for space-bounded randomized and quantum computation. In Section 3, we demonstrate our space-efficient variant of quantum singular value transformation (Theorem 1.4) and offer examples for bounded functions and piecewise-smooth functions. We also provide a simple proof of space-efficient error reduction for unitary quantum computation. Then, in Section 4, we formally define space-bounded quantum state testing problems with four distance-like measures, and present the first family of natural $\mathsf{coRQ_UL}$-complete problems (Theorem 1.1), as well as a novel family of natural BQL-complete problems (Theorem 1.2).

2 Preliminaries

We assume that the reader is familiar with quantum computation and the theory of quantum information. For an introduction, the textbooks by [NC02] and [dW19] provide a good starting point, while for a more comprehensive survey on quantum complexity theory, refer to [Wat08].

In addition, we adopt the convention that the logarithmic function log has a base of 2, denoted by $\log(x) := \log_2(x)$ for any $x \in \mathbb{R}^+$. For the purpose of clarity, we will denote the operator norm as $\|A\| := \|A\|_{2\to 2}$. Moreover, for the sake of simplicity, we utilize the notation $|\bar{0}\rangle$ to represent $|0\rangle^{\otimes a}$ with $a > 1$.

2.1 Distances and divergences for quantum states

We will provide an overview of relevant quantum distances and divergences, along with useful inequalities among different quantum distance-like measures. Additionally, we recommend [BOW19, Section 3.1] for a nice survey on quantum distance and divergences.

Definition 2.1 (Quantum distances and divergences)**.** For any quantum states ρ_0 and ρ_1, we define several distance-like measures and relevant quantities:

- **Trace distance**. $\mathrm{td}(\rho_0,\rho_1) := \frac{1}{2}\mathrm{Tr}|\rho_0-\rho_1| = \frac{1}{2}\mathrm{Tr}(((\rho_0-\rho_1)^\dagger(\rho_0-\rho_1))^{1/2})$.
- **(Uhlmann) Fidelity**. $\mathrm{F}(\rho_0,\rho_1) := \mathrm{Tr}|\sqrt{\rho_0}\sqrt{\rho_1}|$.
- **Squared Hilbert-Schmidt distance**. $\mathrm{HS}^2(\rho_0,\rho_1) := \frac{1}{2}\mathrm{Tr}(\rho_0-\rho_1)^2$.

- **von Neumann entropy**. $\mathrm{S}(\rho) := -\mathrm{Tr}(\rho \ln \rho)$ for any quantum state ρ.

- **Quantum Jensen-Shannon divergence**. $\mathrm{QJS}(\rho_0, \rho_1) := \mathrm{S}\big(\frac{\rho_0+\rho_1}{2}\big) - \frac{\mathrm{S}(\rho_0)+\mathrm{S}(\rho_1)}{2}$.

The trace distance and the squared Hilbert-Schmidt distance reach the minimum of 0 when ρ_0 equals ρ_1, while the fidelity attains a maximum value of 1. Additionally, there are two equalities when at least one of the two states is a pure state:

- For a pure state ρ_0 and a mixed state ρ_1, $\mathrm{F}^2(\rho_0, \rho_1) = \mathrm{Tr}(\rho_0\rho_1)$.

- For two pure states ρ_0 and ρ_1, $\mathrm{Tr}(\rho_0\rho_1) = 1 - \mathrm{HS}^2(\rho_0, \rho_1)$.

Moreover, we have $\mathrm{HS}^2(\rho_0, \rho_1) = \frac{1}{2}(\mathrm{Tr}(\rho_0^2) + \mathrm{Tr}(\rho_1^2)) - \mathrm{Tr}(\rho_0\rho_1)$. Additionally, Fuchs and van de Graaf [FvdG99] showed a well-known inequality between the trace distance and the fidelity:

Lemma 2.2 (Trace distance vs. fidelity, adapted from [FvdG99]). *For any states ρ_0 and ρ_1,*

$$1 - \mathrm{F}(\rho_0, \rho_1) \leq \mathrm{td}(\rho_0, \rho_1) \leq \sqrt{1 - \mathrm{F}^2(\rho_0, \rho_1)}.$$

The joint entropy theorem (Lemma 2.3) enhances our understanding of entropy in classical-quantum states and is necessary for our usages of the von Neumann entropy.

Lemma 2.3 (Joint entropy theorem, adapted from Theorem 11.8(5) in [NC02]). *Suppose p_i are probabilities corresponding to a distribution D, $|i\rangle$ are orthogonal state of a system A, and $\{\rho_i\}_i$ is any set of density operators for another system B. Then $\mathrm{S}\big(\sum_i p_i|i\rangle\langle i|\otimes\rho_i\big) = \mathrm{H}(D) + \sum_i p_i \mathrm{S}(\rho_i)$.*

Let us now turn our attention to the quantum Jensen-Shannon divergence, which is defined in [MLP05]. For simplicity, we define $\mathrm{QJS}_2(\rho_0, \rho_1) := \mathrm{QJS}(\rho_0, \rho_1)/\ln 2$ using the base-2 (matrix) logarithmic function. Notably, when considering size-2 ensembles with a uniform distribution, the renowned Holevo bound [Hol73] (see Theorem 12.1 in [NC02]) indicates that the *quantum Shannon distinguishability* studied in [FvdG99] is at most the quantum Jensen-Shannon divergence. Consequently, this observation yields inequalities between the trace distance and the quantum Jensen-Shannon divergence.[22]

Lemma 2.4 (Trace distance vs. quantum Jensen-Shannon divergence, adapted from [FvdG99, Hol73, BH09]). *For any quantum states ρ_0 and ρ_1, we have*

$$1 - \mathrm{H}_2\left(\frac{1-\mathrm{td}(\rho_0,\rho_1)}{2}\right) \leq \mathrm{QJS}_2(\rho_0, \rho_1) \leq \mathrm{td}(\rho_0, \rho_1).$$

Here, the binary entropy $\mathrm{H}_2(p) := -p\log(p) - (1-p)\log(1-p)$.

2.2 Space-bounded quantum computation

We say that a function $s(n)$ is *space-constructible* if there exists a deterministic space $s(n)$ Turing machine that takes 1^n as an input and output $s(n)$ in the unary encoding. Moreover, we say that a function $f(n)$ is $s(n)$-*space computable* if there exists a deterministic space $s(n)$ Turing machine that takes 1^n as an input and output $f(n)$. Our definitions of space-bounded quantum computation are formulated in terms of *quantum circuits*, whereas many prior works focused on *quantum Turing machines* [Wat09, Wat03, vMW12]. For a discussion on the equivalence between space-bounded quantum computation using *quantum circuits* and *quantum Turing machines*, we refer readers to [FL18, Appendix A] and [FR21, Section 2.2].

We begin by defining time-bounded and space-bounded quantum circuit families, and then proceed to the corresponding complexity class $\mathsf{BQ_USPACE}[s(n)]$. It is worth noting that we use the abbreviated notation C_x to denote that the circuit $C_{|x|}$ takes input x.

[22] For a detailed proof of these inequalities, please refer to [Liu23, Appendix B].

Definition 2.5 (Time- and space-bounded quantum circuit families)**.** A (unitary) quantum circuit is a sequence of quantum gates, each of which belongs to some fixed gateset that is universal for quantum computation, such as $\{\textsc{Hadamard}, \text{CNOT}, \text{T}\}$. For a promise problem $\mathcal{L} = (\mathcal{L}_{\text{yes}}, \mathcal{L}_{\text{no}})$, we say that a family of quantum circuits $\{C_x : x \in \mathcal{L}\}$ is $t(n)$-time-bounded if there is a deterministic Turing machine that, on any input $x \in \mathcal{L}$, runs in time $O(t(|x|))$, and outputs a description of C_x such that C_x accepts (resp., rejects) if $x \in \mathcal{L}_{\text{yes}}$ (resp., $x \in \mathcal{L}_{\text{no}}$). Similarly, we say that a family of quantum circuits $\{C_x : x \in \mathcal{L}\}$ is $s(n)$-space-bounded if there is a deterministic Turing machine that, on any input $x \in \mathcal{L}$, runs in space $O(s(|x|))$ (and hence time $2^{O(s(|x|))}$), and outputs a description of C_x such that C_x accepts (resp., rejects) if $x \in \mathcal{L}_{\text{yes}}$ (resp., $x \in \mathcal{L}_{\text{no}}$), as well as C_x is acting on $O(s(|x|))$ qubits and has $2^{O(s(|x|)}$ gates..

Definition 2.6 ($\mathsf{BQ_USPACE}[s(n), a(n), b(n)]$, adapted from Definition 5 in [FR21])**.** Let $s \colon \mathbb{N} \to \mathbb{N}$ be a space-constructible function such that $s(n) \geq \Omega(\log n)$. Let $a(n)$ and $b(n)$ be functions that are computable in deterministic space $s(n)$. A promise problem $(\mathcal{L}_{\text{yes}}, \mathcal{L}_{\text{no}})$ is in $\mathsf{BQ_USPACE}[s(n), a(n), b(n)]$ if there exists a family of $s(n)$-space-bounded (unitary) quantum circuits $\{C_x\}_{x \in \mathcal{L}}$, where $n = |x|$, satisfying the following:

- The output qubit is measured in the computational basis after applying C_x. We say that C_x *accepts* x if the measurement outcome is 1, whereas C_x *rejects* x if the outcome is 0.
- $\Pr[C_x \text{ accepts } x] \geq a(|x|)$ if $x \in \mathcal{L}_{\text{yes}}$, whereas $\Pr[C_x \text{ accepts } x] \leq b(|x|)$ if $x \in \mathcal{L}_{no}$.

We remark that Definition 2.6 is *gateset-independent*, given that the gateset is closed under adjoint and all entries in chosen gates have reasonable precision. This property is due to the space-efficient Solovay-Kitaev theorem presented in [vMW12]. Moreover, we can achieve error reduction for $\mathsf{BQ_USPACE}[s(n), a(n), b(n)]$ as long as $a(n) - b(n) \geq 2^{-O(s(n))}$, which follows from [FKL+16] or our space-efficient QSVT-based construction in Section 3.4. We thereby define $\mathsf{BQ_USPACE}[s(n)] := \mathsf{BQ_USPACE}[s(n), 2/3, 1/3]$ to represent (two-sided) bounded-error unitary quantum space, and $\mathsf{BQ_UL} := \mathsf{BQ_USPACE}[O(\log n)]$ to denote unitary quantum logspace.

We next consider general space-bounded quantum computation, which allows *intermediate quantum measurements*. As indicated in [AKN98, Section 4.1], for any quantum channel Φ mapping from density matrices on k_1 qubits to density matrices on k_2 qubits, we can exactly simulate this quantum channel Φ by a unitary quantum circuit acting on $2k_1 + k_2$ qubits. Therefore, we extend Definition 2.5 to *general quantum circuits*, which allows local operations, such as intermediate measurements in the computational basis, resetting qubits to their initial states, and tracing out qubits. Now we proceed with a definition on $\mathsf{BQSPACE}[s(n)]$.

Definition 2.7 ($\mathsf{BQSPACE}[s(n), a(n), b(n)]$, adapted from Definition 7 in [FR21])**.** Let $s \colon \mathbb{N} \to \mathbb{N}$ be a space-constructible function such that $s(n) \geq \Omega(\log n)$. Let $a(n)$ and $b(n)$ be functions that are computable in deterministic space $s(n)$. A promise problem $(\mathcal{L}_{\text{yes}}, \mathcal{L}_{\text{no}})$ is in $\mathsf{BQSPACE}[s(n), a(n), b(n)]$ if there exists a family of $s(n)$-space-bounded general quantum circuits $\{\Phi_x\}_{x \in \mathcal{L}}$, where $n = |x|$, satisfying the following holds:

- The output qubit is measured in the computational basis after applying Φ_x. We say that Φ_x *accepts* x if the measurement outcome is 1, whereas Φ_x *rejects* x if the outcome is 0.
- $\Pr[\Phi_x \text{ accepts } x] \geq a(|x|)$ if $x \in \mathcal{L}_{\text{yes}}$, whereas $\Pr[\Phi_x \text{ accepts } x] \leq b(|x|)$ if $x \in \mathcal{L}_{no}$.

It is noteworthy that unitary quantum circuits, which correspond to unitary channels, are a specific instance of general quantum circuits that correspond to quantum channels. we thus infer that $\mathsf{BQ_USPACE}[s(n)] \subseteq \mathsf{BQSPACE}[s(n)]$ for any $s(n) \geq \Omega(\log n)$. However, the opposite direction was a long-standing open problem. Recently, Fefferman and Remscrim [FR21] demonstrated a remarkable result that $\mathsf{BQSPACE}[s(n)] \subseteq \mathsf{BQ_USPACE}[O(s(n))]$. In addition, it is evident that $\mathsf{BQSPACE}[s(n)]$ can achieve error reduction since it admits sequential repetition simply by resetting working qubits. Therefore, we define $\mathsf{BQSPACE}[s(n)] := \mathsf{BQSPACE}[s(n), 2/3, 1/3]$

to represent (two-sided) bounded-error general quantum space, and denote general quantum logspace by $\mathsf{BQL} := \mathsf{BQSPACE}[O(\log n)]$.

We now turn our attention to *one-sided* bounded-error unitary quantum space $\mathsf{RQ_U SPACE}[s(n)]$ and $\mathsf{coRQ_U SPACE}[s(n)]$ for $s(n) \geq \Omega(\log n)$. These complexity classes were first introduced by Watrous [Wat01] and have been further discussed in [FR21]. We proceed with the definitions:

- $\mathsf{RQ_U SPACE}[s(n), a(n)] := \mathsf{BQ_U SPACE}[s(n), a(n), 0]$;
- $\mathsf{coRQ_U SPACE}[s(n), b(n)] := \mathsf{BQ_U SPACE}[s(n), 1, b(n)]$.

Note that $\mathsf{RQ_U SPACE}[s(n), a(n)]$ and $\mathsf{coRQ_U SPACE}[s(n), b(n)]$ can achieve error reduction, as shown in [Wat01] or our space-efficient QSVT-based construction in Section 3.4. We define

$$\mathsf{RQ_U SPACE}[s(n)] := \mathsf{BQ_U SPACE}\left[s(n), \tfrac{1}{2}, 0\right] \text{ and } \mathsf{coRQ_U SPACE}[s(n)] := \mathsf{BQ_U SPACE}\left[s(n), 1, \tfrac{1}{2}\right]$$

to represent one-sided bounded-error unitary quantum space, as well as logspace counterparts

$$\mathsf{RQ_U L} := \mathsf{RQ_U SPACE}[O(\log n)] \text{ and } \mathsf{coRQ_U L} := \mathsf{coRQ_U SPACE}[O(\log n)].$$

Remark 2.8 ($\mathsf{RQ_U L}$ and $\mathsf{coRQ_U L}$ are gateset-dependent). We observe that changing the gateset in space-efficient Solovay-Kitaev theorem [vMW12] can cause errors, revealing the *gateset-dependence* of unitary quantum space classes with one-sided bounded-error. To address this issue, we adopt a larger gateset $\mathcal{G}$ for $\mathsf{RQ_U SPACE}[s(n)]$ and $\mathsf{coRQ_U SPACE}[s(n)]$, which includes any single-qubit gates whose amplitudes can be computed in deterministic $O(s(n))$ space.

2.3 Near-minimax approximation by Chebyshev interpolation

We will define Chebyshev polynomials and introduce Chebyshev interpolation, which is notable for providing *near-minimax approximations*. These concepts are essential to our space-efficient quantum singular value transformation techniques (Section 3).

Definition 2.9 (Chebyshev polynomials)**.** The Chebyshev polynomials (of the first kind) $T_k(x)$ are defined via the following recurrence relation: $T_0(x) := 1$, $T_1(x) := x$, and $T_{k+1}(x) := 2xT_k(x) - T_{k-1}(x)$. For $x \in [-1,1]$, an equivalent definition is $T_k(\cos\theta) = \cos(k\theta)$.

In order to use Chebyshev polynomials for interpolation, we first need to define an inner product between two functions, f and g, as long as the following integral exists:

$$\langle f, g\rangle := \frac{2}{\pi}\int_{-1}^{1}\frac{f(x)g(x)}{\sqrt{1-x^2}}\mathrm{d}x = \frac{2}{\pi}\int_{-\pi}^{0} f(\cos\theta)g(\cos\theta)\mathrm{d}\theta.$$

The Chebyshev polynomials form an orthonormal basis in this inner product space induced by $\langle\cdot,\cdot\rangle$. As a result, any degree-d polynomial P_d can be represented as a linear combination of Chebyshev polynomials using a technique called *Chebyshev interpolation*, see [MH02, Section 6.5] for the details. In particular, $P_d = \frac{1}{2}\langle T_0, P_d\rangle + \sum_{k=1}^{d}\langle T_k, P_d\rangle T_k$. It is noteworthy that Lemma 2.10 is first proven in [Pow67].

Lemma 2.10 (Near-minimax approximation by Chebyshev interpolation, adapted from Theorem 6.13 in [MH02])**.** *For any continuous function $f\colon [-1,1] \to \mathbb{R}$, if there exists an explicit degree-d polynomial $\hat{P}_d \in \mathbb{R}[x]$ such that $\max_{x\in[-1,1]} |f(x) - \hat{P}_d(x)| \leq \epsilon$, then we know that $P_d = \frac{1}{2}\langle T_0, f\rangle + \sum_{k=1}^{d}\langle T_k, f\rangle T_k$ satisfies $\max_{x\in[-1,1]} |f(x) - P_d(x)| \leq O(\epsilon \log d)$.*

2.4 Tools for space-bounded randomized and quantum algorithms

Our convention assumes that for any algorithm $\mathcal{A}$ in bounded-error randomized time $t(n)$ and space $s(n)$, $\mathcal{A}$ outputs the correct value with probability at least $2/3$ (viewed as "success probability"). We first proceed with space-efficient success probability estimation.

Lemma 2.11 (Space-efficient success probability estimation by sequential repetitions)**.** *Let $\mathcal{A}$ be a randomized (resp., quantum) algorithm that outputs the correct value with probability p, has time complexity $t(n)$, and space complexity $s(n)$. We can obtain an additive-error estimation $\hat{p}$ such that $|p - \hat{p}| \leq \epsilon$, where $\epsilon \geq 2^{-O(s(n))}$. Moreover, this estimation can be computed in bounded-error randomized (resp., quantum) time $O(\epsilon^{-2} t(n))$ and space $O(s(n))$.*

Proof. Consider a m-time sequential repetition of the algorithm $\mathcal{A}$, and let X_i be a random variable indicating whether the i-th repetition succeeds, then we obtain a random variable $X = \frac{1}{m}\sum_{i=1}^{m} X_i$ such that $\mathbb{E}[X] = p$. Now let $\hat{X} = \frac{1}{m}\sum_{i=1}^{m} \hat{X}_i$ be the additive-error estimation, where $\hat{X}_i$ is the outcome of $\mathcal{A}$ in the i-th repetition. By the Chernoff-Hoeffding bound (e.g., Theorem 4.12 in [MU17]), we know that $\Pr\left[|\hat{X} - p| \geq \epsilon\right] \leq 2\exp(-2m\epsilon^2)$. By choosing $m = 2\epsilon^{-2}$, this choice of m ensures that each run of $\mathcal{A}$ succeeds with probability at least $2/3$.

Furthermore, the space complexity of our algorithm is $O(s(n))$ since we can simply reuse the workspace. Also, the time complexity is $m \cdot t(n) = O(\epsilon^{-2} t(n))$ as desired. □

Notably, when applying Lemma 2.11 to a quantum algorithm, we introduce intermediate measurements to retain space complexity through reusing working qubits. While space-efficient success probability estimation without intermediate measurements is possible,[23] we will use Lemma 2.11 for convenience, given that $\mathsf{BQL} = \mathsf{BQ_UL}$ [FR21].

The SWAP test was originally proposed for pure states in [BCWdW01]. Subsequently, in [KMY09], it was demonstrated that the SWAP test can also be applied to mixed states.

Lemma 2.12 (SWAP test for mixed states, adapted from [KMY09, Proposition 9])**.** *Suppose ρ_0 and ρ_1 are two n-qubit mixed quantum states. There is a $(2n+1)$-qubit quantum circuit that outputs 0 with probability $\frac{1+\mathrm{Tr}(\rho_0\rho_1)}{2}$, using 1 sample of each ρ_0 and ρ_1 and $O(n)$ one- and two-qubit quantum gates.*

A matrix B is said to be *sub-stochastic* if all its entries are non-negative and the sum of entries in each row (respectively, column) is strictly less than 1. Moreover, a matrix B is *row-stochastic* if all its entries are non-negative and the sum of entries in each row is equal to 1.

Lemma 2.13 (Sub-stochastic matrix powering in bounded space)**.** *Let B be an $l \times l$ sub-stochastic matrix, where each entry of B requires at most ℓ-bit precision. Then, there exists an explicit randomized algorithm that computes the matrix power $B^k[s,t]$ in $\log(l+1)$ space and $O(\ell k)$ time. Specifically, the algorithm accepts with probability $B^k[s,t]$.*

Proof. Our randomized algorithm leverages the equivalence between space-bounded randomized computation and Markov chains, see [Sak96, Section 2.4] for a detailed introduction.

First, we construct a row-stochastic matrix $\hat{B}$ from B by adding an additional column and row. Let $\hat{B}[i,j]$ denote the entry at the i-th column and the j-th row of $\hat{B}$. Specifically,

$$\hat{B}[i,j] := \begin{cases} B[i,j], & \text{if } 1 \leq i,j \leq l; \\ 1 - \sum_{j=1}^{l+1-i} b_j^{(1)}, & \text{if } i = l+1 \text{ and } 1 \leq j \leq l+1; \\ 0, & \text{if } 1 \leq i \leq l \text{ and } j = l+1. \end{cases}$$

Next, we view $\hat{B}$ as a transition matrix of a Markov chain since $\hat{B}$ is row-stochastic. We consequently have a random walk on the directed graph $G = (V, E)$ where $V = \{1, 2, \cdots, l\} \cup \{\perp\}$ and $(u, v) \in E$ iff $\hat{B}(u, v) > 0$. In particular, the probability that a k-step random walk starting at node s and ending at node t is exactly $\hat{B}^k[s,t] = B^k[s,t]$. This is because the walker who visits the dummy node $\perp$ will not reach other nodes.

[23] Fefferman and Lin [FL18] noticed that one can achieve space-efficient success probability estimation for quantum algorithms without intermediate measurements via quantum amplitude estimation [BHMT02].

Finally, note that $\hat{B}$ is a $(l+1)\times(l+1)$ matrix, the matrix powering of $\hat{B}^k$ can be computed in $\log(l)$ space. In addition, the overall time complexity is $O(\ell k)$ since we simulate the dyadic rationals (with ℓ-bit precision) of a single transition exactly by ℓ coin flips. $\square$

3 Space-efficient quantum singular value transformations

We begin by defining the *projected unitary encoding* and its special forms, viz. the bitstring indexed encoding and the block-encoding, as well as notations on *singular value decomposition* and *singular value transformation*.

Definition 3.1 (Projected unitary encoding and its special forms, adapted from [GSLW19])**.** Let U be an (α, a, ϵ)-projected unitary encoding of a linear operator A if $\|A - \alpha\tilde{\Pi}U\Pi\| \leq \epsilon$, where U and orthogonal projectors $\tilde{\Pi}$ and Π act on $s+a$ qubits, and both $\mathrm{rank}(\tilde{\Pi})$ and $\mathrm{rank}(\Pi)$ are at least 2^a (a is viewed as the number of ancillary qubits). Furthermore, we are interested in two special forms of the projected unitary encoding:

- **Bitstring indexed encoding.** We say that a projected unitary encoding is a *bitstring indexed encoding* if both orthogonal projectors $\tilde{\Pi}$ and Π span on $\tilde{S}, S \subseteq \{|0\rangle, |1\rangle\}^{\otimes(a+s)}$, respectively.[24] In particular, for any $|\tilde{s}_i\rangle \in \tilde{S}$ and $|s_j\rangle \in S$, we have a matrix representation $A_{\tilde{S},S}(i,j) := \langle \tilde{s}_i|U|s_j\rangle$ of A.
- **Block-encoding.** We say that a projected unitary encoding is a block-encoding if both orthogonal projectors are of the form $\Pi = \tilde{\Pi} = |0\rangle\langle 0|^{\otimes a} \otimes I_s$. We use the shorthand $A = (\langle\bar{0}| \otimes I_s)U(|\bar{0}\rangle \otimes I_s)$ for convenience.

Definition 3.2 (Singular value decomposition of a projected unitary, adapted from Definition 7 in [GSLW19])**.** Given a projected unitary encoding of A, denoted by U, associated with orthogonal projectors Π and $\tilde{\Pi}$ on a finite-dimensional Hilbert space $\mathcal{H}_U$. Namely, $A = \tilde{\Pi}U\Pi$. Then there exists orthonormal bases of Π and $\tilde{\Pi}$ such that Π: $\{|\psi_i\rangle : i \in [d]\}$, where $d := \mathrm{rank}(\Pi)$, of a subspace $\mathrm{Img}(\Pi) = \mathrm{span}\{|\psi_i\rangle\}$; $\tilde{\Pi}$: $\big\{|\tilde{\psi}_i\rangle : i \in [\tilde{d}]\big\}$, where $\tilde{d} := \mathrm{rank}(\tilde{\Pi})$, of a subspace $\mathrm{Img}(\tilde{\Pi}) = \mathrm{span}\big\{|\tilde{\psi}_i\rangle\big\}$. These bases ensure that the singular value decomposition $\tilde{\Pi}U\Pi = \sum_{i=1}^{\min\{d,\tilde{d}\}} \sigma_i |\tilde{\psi}_i\rangle\langle\psi_i|$ where singular values $\sigma_i > \sigma_j$ for any $i < j \in [\min\{d,\tilde{d}\}]$.

Definition 3.3 (Singular value transformation by even or odd functions, adapted from Definition 9 in [GSLW19])**.** Let $f\colon \mathbb{R} \to \mathbb{C}$ be an even or odd function. We consider a linear operator $A \in \mathbb{C}^{\tilde{d}\times d}$ satisfying the singular value decomposition $A = \sum_{i=1}^{\min\{d,\tilde{d}\}} \sigma_i |\tilde{\psi}_i\rangle\langle\tilde{\psi}_i|$. We define the *singular value transformation* corresponding to f as follows:

$$f^{(\mathrm{SV})}(A) := \begin{cases} \sum_{i=1}^{\min\{d,\tilde{d}\}} f(\sigma_i)|\tilde{\psi}_i\rangle\langle\psi_i|, & \text{for odd } f, \\ \sum_{i=1}^{d} f(\sigma_i)|\psi_i\rangle\langle\psi_i|, & \text{for even } f. \end{cases}$$

Here, for $i \in \{\min\{d,\tilde{d}\}+1, \cdots, d-1, d\}$, we define $\sigma_i := 0$.

It is worth noting that $f^{(\mathrm{SV})}(A) = f(A)$ when A is an Hermitian matrix.

With these definitions in place, we present the main (informal) theorem in this section:

Theorem 3.4 (Space-efficient QSVT)**.** *Let $f\colon \mathbb{R} \to \mathbb{R}$ be a continuous function bounded on the closed interval of interest $\mathcal{I} \subseteq [-1,1]$. If there exists a degree-d polynomial $\hat{P}_d$ that approximates $h\colon [-1,1] \to \mathbb{R}$, where h approximates f only on $\mathcal{I}$, such that* $\max_{x\in[-1,1]} |h(x) - \hat{P}_d(x)| \leq \epsilon$, *then Chebyshev interpolation yields another degree-d polynomial P_d satisfying the following conditions:* $\max_{x\in\mathcal{I}} |f(x) - P_d(x)| \leq O(\epsilon \log d)$ *and* $\max_{x\in[-1,1]} |P_d(x)| \leq 1$.

[24] Typically, to ensure these orthogonal projectors coincide with space-bounded quantum computation, we additionally require the corresponding subsets $\tilde{S}$ and S admit space-efficient set membership, namely deciding the membership of these subsets is in deterministic $O(s+a)$ space.

Moreover, we have space-efficient classical algorithms for computing any entry in the coefficient vector $\mathbf{c}$ *of the Chebyshev interpolation polynomial* P_d*:*

- *If* f *is a bounded function,*[25] *then any entry in the coefficient vector* $\mathbf{c}$ *can be computed in deterministic* $O(\log d)$ *space;*
- *If* f *is a piecewise-smooth function, then any entry in the coefficient vector* $\mathbf{c}$ *can be computed in bounded-error randomized* $O(\log d)$ *space.*

Furthermore, for any $(1, a, 0)$*-bitstring indexed encoding* U *of* $A = \tilde{\Pi} U \Pi$*, acting on* $s + a$ *qubits where* $a(n) \leq s(n)$*, and any* P_d *with* $d \leq 2^{O(s(n))}$*, we can implement the quantum singular value transformation* $P_d^{(\mathrm{SV})}(A)$ *that acts on* $O(s(n))$ *qubits by using* $O(d^2 \|\mathbf{c}\|_1)$ *queries to* U*.*

We remark that we can apply Theorem 3.4 to general forms of the projected unitary encoding U with orthogonal projectors Π and $\tilde{\Pi}$, as long as such an encoding meets the conditions: (1) The basis of Π and $\tilde{\Pi}$ admits a well-defined order; (2) Both controlled-Π and controlled-$\tilde{\Pi}$ admit computationally efficient implementation. We note that bitstring indexed encoding defined in Definition 3.1 trivially meets the first condition, and a sufficient condition for the second condition is that the corresponding subsets S and $\tilde{S}$ have space-efficient set membership.

Specifically, we elaborate on three main technical contributions that culminate in our space-efficient quantum singular value transformations (Theorem 3.4):

- We provide deterministic space-efficient polynomial approximations for *bounded* functions (Lemma 3.5), including the sign function (Corollary 3.7). Our approach leads to a simple proof of space-efficient error reduction for unitary quantum computations (Section 3.4).
- We present bounded-error randomized space-efficient polynomial approximations for *piecewise-smooth* functions (Theorem 3.8), such as the normalized logarithmic function (Corollary 3.10).
- We propose QSVT implementations using Chebyshev interpolation polynomials (Theorem 3.11), including those for the sign function (Corollary 3.16) and the normalized logarithmic function (Corollary 3.17).

3.1 Space-efficient bounded polynomial approximations

We provide a systematic approach for constructing *space-efficient* polynomial approximations of real-valued piecewise-smooth functions, which is a space-efficient counterpart of Corollary 23 in [GSLW19]. It is worth mentioning that our algorithm (Lemma 3.5) is *deterministic* for continuous functions that are bounded on the interval $[-1, 1]$. However, for general piecewise-smooth functions, we only introduce a *randomized* algorithm (Theorem 3.8). In addition, please refer to Section 2.3 as a brief introduction to Chebyshev polynomials and Chebyshev interpolation.

3.1.1 Bounded functions

We propose a space-efficient algorithm for computing the coefficients of a polynomial approximation with high accuracy for bounded functions. Our approach uses Chebyshev interpolation and numerical integration, building upon the methodology outlined in Lemma 2.10 of [MY23] with meticulous analysis.

Lemma 3.5 (Space-efficient polynomial approximations for bounded functions)**.** *Consider a continuous function* f*, and let* $\hat{P}_d^{(f)}$ *be a degree-*d *polynomial with the same parity as* f*, such that* $\max_{x \in [-1,1]} |f(x) - \hat{P}_d^{(f)}(x)| \leq \epsilon$*, where* f *is bounded with* $\max_{x \in [-1,1]} |f(x)| \leq B$*. By using*

[25] This conclusion also applies to a linear combination of bounded functions, provided that the coefficients are bounded and can be computed deterministically and space-efficiently.

Chebyshev interpolation, we can obtain another degree-d polynomial $P_d^{(f)}$ that has the same parity as $\hat{P}_d^{(f)}$ and satisfies $\max_{x\in[-1,1]} |f(x) - P_d^{(f)}(x)| \leq O(\epsilon \log d)$. This polynomial $P_d^{(f)}$ is defined as a linear combination of Chebyshev polynomials $T_k(\cos\theta) = \cos(k\theta)$:

$$P_d^{(f)}(x) = \frac{c_0}{2} + \sum_{k=1}^{d} c_k T_k(x) \text{ where } c_k := \frac{2}{\pi}\int_{-\pi}^{0} F_k(\theta)\mathrm{d}\theta \text{ and } F_k(\theta) := \cos(k\theta) f(\cos\theta).$$

If the integrand $F_k(\theta)$ satisfies $\max_{\xi\in[-\pi,0]} |F_k''(\xi)| \leq O(d^\gamma)$ for some constant γ, then any entry of the coefficient vector $\mathbf{c} = (c_0, \cdots, c_d)$ can be computed in deterministic time $O(d^{(\gamma+1)/2}\epsilon^{-1/2} t(\ell))$ and space $O(\log(d^{\gamma+1}\epsilon^{-1}B))$, where evaluating $F(\theta)$ in ℓ-bit precision is in deterministic time $t(\ell)$ and space $O(\ell)$ for $\ell = O(\log(d^{(\gamma+1)/2}\epsilon^{-3/2}))$. Furthermore, the coefficient vector $\mathbf{c}$ has a norm bounded by $\|\mathbf{c}\|_1 \leq O(Bd)$.

Proof. To apply Chebyshev interpolation to a bounded continuous function $f(x)$, we begin with a degree-d polynomial $\hat{P}_d^{(f)}$ such that $\max_{x\in[-1,1]} |f(x) - \hat{P}_d^{(f)}(x)| \leq \epsilon$. By utilizing Lemma 2.10, we can construct a degree-d Chebyshev interpolation of $f(x)$ denoted as $P_d^{(f)}$. This interpolation is expressed as $P_d^{(f)} = c_0/2 + \sum_{k=1}^{d} c_k T_k$, where $c_k = \frac{2}{\pi}\int_{-\pi}^{0} F_k(\theta)\mathrm{d}\theta$ and $F_k(\theta) := \cos(k\theta) f(\cos\theta)$, and additionally satisfies the error bound: $\max_{x\in[-1,1]} \left|f(x) - P_d^{(f)}(x)\right| \leq O(\epsilon \log d)$.

Computing the coefficients. It is left to compute the coefficients c_k for $0 \leq k \leq d$. We can estimate the numerical integration using *the composite trapezium rule*, as described in [SM03, Section 7.5]. The application of this method yields the following result:

$$\int_{-\pi}^{0} F_k(x)\mathrm{d}x \approx \frac{\pi}{m}\Big(\frac{F_k(x_0)}{2} + \sum_{l=1}^{m} F_k(x_l) + \frac{F_k(x_m)}{2}\Big) \text{ where } x_l := -\pi + \frac{\pi l}{m} \text{ for } l = 0, 1, \cdots, m. \tag{3.1}$$

Moreover, we know the upper bound on the numerical errors for computing the coefficient c_k:

$$\varepsilon_{d,k}^{(f)} := \sum_{l=1}^{m} \Big| \int_{x_{i-1}}^{x_i} F_k(x)\mathrm{d}x - \frac{\pi}{2m}\cdot (F_k(x_{i-1}) + F_k(x_i)) \Big| \leq \frac{\pi^3}{12m^2} \max_{\xi\in[-\pi,\pi]} \left|F_k''(\xi)\right|. \tag{3.2}$$

To obtain an upper bound on the number of intervals m, we need to ensure that the error of the numerical integration is within $\varepsilon_d^{(f)} = \sum_{k=1}^{d} \varepsilon_{d,k}^{(f)} \leq \epsilon$. Plugging the assumption $|F_k''(x)| \leq O(d^\gamma)$ into Equation (3.2), by choosing an appropriate value of $m = O(\epsilon^{-1/2} d^{(\gamma+1)/2})$, we establish that $\varepsilon_d^{(f)} \leq O(d^{\gamma+1}/m^2) \leq O(\epsilon)$. Moreover, to guarantee that the accumulated error is $O(\epsilon)$ in Equation (3.1), we need to evaluate the integrand $F(\theta)$ with ℓ-bit precision, where $\ell = O(\log(m/\epsilon)) = O(\log(\epsilon^{-3/2} d^{(\gamma+1)/2}))$. In addition, note that $c_k = \frac{2}{\pi}\int_{-\pi}^{0} F_k(\theta)\mathrm{d}\theta \leq 2\cdot \max_{x\in[-1,1]} |f(x)| \leq 2B$, we know that the coefficient vector $\mathbf{c}$ satisfies $\|\mathbf{c}\|_1 = \sum_{k=0}^{d} |c_i| \leq O(Bd)$.

Analyzing time and space complexity. The presented numerical integration algorithm is deterministic, and therefore, the time complexity for computing the integral is $O(mt(\ell))$, where $t(\ell)$ is the time complexity for evaluating the integrand $F_k(\theta)$ within $2^{-\ell}$ accuracy (i.e., ℓ-bit precision) in $O(\ell)$ space. The space complexity required for computing the numerical integration is the number of bits required to index the integral intervals and represent the resulting coefficients. To be specific, the space complexity is

$$\max\big\{O(\log m), O\big(\log\tfrac{m}{\epsilon}\big), \log\|\mathbf{c}\|_\infty\big\} \leq O\big(\max\big\{\log\big(\epsilon^{-\frac{3}{2}} d^{-\frac{\gamma+1}{2}}\big), \log B\big\}\big) \leq O\big(\log\big(\epsilon^{-\frac{3}{2}} d^{-\frac{\gamma+1}{2}} B\big)\big).$$

Here, $\|\mathbf{c}\|_\infty = \max_{0\leq k\leq d} \frac{2}{\pi}|\int_{-\pi}^{0} \cos(k\theta) f(\cos\theta)\mathrm{d}\theta| \leq \max_{0\leq k\leq d}\max_{-\pi\leq\theta\leq 0} O(|f(\cos\theta)|) \leq O(B)$, and the last inequality is due to the fact that $\Theta(\max\{\log A, \log B\}) = \Theta(\log(AB))$ for any $A, B > 0$. □

It is worth noting that evaluating a large family of functions, called holonomic functions, with ℓ-bit precision requires only *deterministic* $O(\ell)$ space:

Remark 3.6 (Space-efficient evaluation of holonomic functions). Holonomic functions encompass several commonly used functions,[26] such as polynomials, rational functions, sine and cosine functions (but not other trigonometric functions such as tangent or secant), exponential functions, logarithms (to any base), the Gaussian error function, and the normalized binomial coefficients. In [CGKZ05, Mez12], these works have demonstrated that evaluating a holonomic function with ℓ-bit precision is achievable in deterministic time $\tilde{O}(\ell)$ and space $O(\ell)$. Prior works achieved the same time complexity, but with a space complexity of $O(\ell \log \ell)$.

We now present an example of bounded functions, specifically the sign function.

Corollary 3.7 (Space-efficient approximation to the sign function). *For any $\delta, \epsilon > 0$, there is an explicit odd polynomial $P_d^{\mathrm{sgn}} := \frac{c_0}{2} + \sum_{k=1}^{d} c_k T_k \in \mathbb{R}[x]$ of degree $d \leq \tilde{C}_{\mathrm{sgn}} \delta^{-1} \log \epsilon^{-1}$, where $\tilde{C}_{\mathrm{sgn}}$ is a universal constant. Any entry of the coefficient vector $\mathbf{c} := (c_0, \cdots, c_d)$ can be computed in deterministic time $\tilde{O}\big(\epsilon^{-1/2} d^2\big)$ and space $O(\log(\epsilon^{-1} d^4))$. Furthermore, the polynomial P_d^{sgn} satisfies the following conditions:*

$$\forall x \in [-1,1] \setminus [-\delta, \delta], \left|\mathrm{sgn}(x) - P_d^{\mathrm{sgn}}(x)\right| \leq C_{\mathrm{sgn}} \epsilon \log d, \text{ where } C_{\mathrm{sgn}} \text{ is a universal constant,}$$
$$\forall x \in [-1,1], \left|P_d^{\mathrm{sgn}}(x)\right| \leq 1.$$

Additionally, the coefficient vector $\mathbf{c}$ has a norm bounded by $\|\mathbf{c}\|_1 \leq \hat{C}_{\mathrm{sgn}} \log d$, where $\hat{C}_{\mathrm{sgn}}$ is another universal constant. Without loss of generality. we assume that all constants C_{sgn}, $\hat{C}_{\mathrm{sgn}}$, and $\tilde{C}_{\mathrm{sgn}}$ are at least 1.

Proof. We start from a degree-d polynomial $\hat{P}_d^{\mathrm{sgn}}$ that well-approximates $\mathrm{sgn}(x)$.

Proposition 3.7.1 (Polynomial approximation of the sign function, adapted from Lemma 10 and Corollary 4 in [LC17]). *For any $\delta > 0$, $x \in \mathbb{R}$, $\epsilon \in (0, \sqrt{2e\pi})$. Let $\kappa = \frac{2}{\delta} \log^{1/2}\left(\frac{\sqrt{2}}{\sqrt{\pi}\epsilon}\right)$, Then*

$$g_{\delta,\epsilon}(x) := \mathrm{erf}(\kappa x) \text{ satisfies that } |g_{\delta,\epsilon}(x)| \leq 1 \text{ and } \max_{|x| \geq \delta/2} |g_{\delta,\epsilon}(x) - \mathrm{sgn}(x)| \leq \epsilon.$$

Moreover, there is an explicit odd polynomial $\hat{P}_d^{\mathrm{sgn}} \in \mathbb{R}[x]$ of degree $d = O(\sqrt{(\kappa^2 + \log \epsilon^{-1}) \log \epsilon^{-1}})$ such that $\max_{x \in [-1,1]} \left|\hat{P}_d^{\mathrm{sgn}}(x) - \mathrm{erf}(\kappa x)\right| \leq \epsilon$

By applying Proposition 3.7.1, we obtain a polynomial $\hat{P}_d^{\mathrm{sgn}}$ that well approximates the function $\mathrm{erf}(\kappa x)$ where $\kappa = O(\delta^{-1}\sqrt{\log \epsilon^{-1}})$. Consequently, this polynomial $\hat{P}_d^{\mathrm{sgn}}$ has a degree of $d \leq \tilde{C}_{\mathrm{sgn}} \delta^{-1} \log \epsilon^{-1}$, where $\tilde{C}_{\mathrm{sgn}}$ is a universal constant. Note that the Gaussian error function is bounded, namely $|\,\mathrm{erf}(\kappa x)| \leq 1$ for any x. To utilize Lemma 3.5, it suffices to upper bound $\max_{\xi \in [-\pi, 0]} |F_k''(\xi)|$ for any $0 \leq k \leq d$, as specified in Fact 3.7.2 and the proof is deferred to Appendix A.1.1.

Fact 3.7.2. *Let $F_k(\theta) := \mathrm{erf}(\kappa \cos\theta) \cos(k\theta)$, $\max\limits_{0 \leq k \leq d} \max\limits_{\xi \in [-\pi, 0]} |F_k''(\xi)| \leq \frac{2}{\sqrt{\pi}} \kappa + k^2 + \frac{4}{\sqrt{\pi}} \kappa^3 + \frac{4}{\sqrt{\pi}} k\kappa.$*

Note that $\kappa \leq O(d)$ and $k \leq d$, Fact 3.7.2 indicates that $\max_{\xi \in [-\pi,0]} |F_k''(\xi)| \leq O(d^3)$ for any $0 \leq k \leq d$. Hence, we result in an approximation polynomial $\tilde{P}_d^{\mathrm{sgn}}$ by Lemma 3.5 satisfies that $\max_{x \in [-1,1]} |\,\mathrm{erf}(\kappa x) - \tilde{P}_d^{\mathrm{sgn}}(x)| \leq O(\epsilon \log d)$, which additionally derives that

$$\max_{x \in [-1,1]} |\mathrm{sgn}(x) - \tilde{P}_d^{\mathrm{sgn}}(x)| \leq \epsilon + \max_{x \in [-1,1]} |\,\mathrm{erf}(\kappa x) - \tilde{P}_d^{\mathrm{sgn}}(x)| \leq C_{\mathrm{sgn}} \epsilon \log d.$$

Here, C_{sgn} is a universal constant. Moreover, we specify the bound of $\|\tilde{\mathbf{c}}^{\mathrm{sgn}}\|_1$ in Fact 3.7.3, and the proof is deferred to Appendix A.1.1:

Fact 3.7.3 (Implicit in [MY23, Lemma 2.10]). *For the coefficient vector $\tilde{\mathbf{c}}^{\mathrm{sgn}}$ corresponding to a degree-d polynomial $\tilde{P}_d^{\mathrm{sgn}}$, we have $\|\tilde{\mathbf{c}}^{\mathrm{sgn}}\|_1 \leq \hat{C}_{\mathrm{sgn}} \log d$ where $\hat{C}_{\mathrm{sgn}}$ is a universal constant.*

[26] For a more detailed introduction, please refer to [BZ10, Section 4.9.2].

In addition, the coefficient vector $\tilde{\mathbf{c}}^{\text{sgn}}$ can be computed in deterministic space $O(\log(d\epsilon^{-1}))$. As the evaluation of the integrand $F(\theta)$ requires ℓ-bit precision where $\ell = O(\log(\epsilon^{-3/2}d^2))$, together with Remark 3.6, $\tilde{\mathbf{c}}^{\text{sgn}}$ can be computed in deterministic time $\tilde{O}(\epsilon^{-1/2}d^2)$.

Finally, we obtain that $|\tilde{P}_d^{\text{sgn}}(x)| \leq 1+\epsilon$ for any $x \in [-1,1]$ since $|\text{sgn}(x)| \leq 1$ for any x. We finish the proof by normalizing $\tilde{P}_d^{\text{sgn}}$, in particular, considering $P_d^{\text{sgn}}(x) := (1+\epsilon)^{-1}\tilde{P}_d^{\text{sgn}}$. It is evident to verify that P_d^{sgn} is an odd polynomial that satisfies all desired requirements. □

3.1.2 Piecewise-smooth functions

We present a randomized algorithm for constructing bounded polynomial approximations of piecewise-smooth functions, which can be seen as a *space-efficient* alternative to Corollary 23 in [GSLW19], as described in Theorem 3.8. Our algorithm leverages Lemma 3.5 and Lemma 3.9.

Theorem 3.8 (Taylor series based space-efficient bounded polynomial approximations). *Consider a real-valued function $f\colon [-x_0-r-\delta, x_0+r+\delta] \to \mathbb{R}$ such that $f(x_0+x) = \sum_{l=0}^{\infty} a_l x^l$ for all $x \in [-r-\delta, r+\delta]$, where $x_0 \in [-1,1]$, $r \in (0,2]$, $\delta \in (0,r]$. Assume that $\sum_{l=0}^{\infty}(r+\delta)^l |a_l| \leq B$ where $B > 0$. Let $\epsilon \in (0, \frac{1}{2B}]$ such that $B > \epsilon$, then there is a polynomial $P \in \mathbb{R}[x]$ of degree $O(\delta^{-1}\log(\epsilon^{-1}B))$, such that any entry of the coefficient vector $\mathbf{c}^{(P)}$ can be computed in bounded-error randomized time $\tilde{O}(\max\{(\delta')^{-5}\epsilon^{-2}B^2, d^2\epsilon^{-1/2}B\})$ and space $O(\log(d^4(\delta')^{-4}\epsilon^{-1}B))$ where $\delta' := \frac{\delta}{2(r+\delta)}$, such that*

$$\|f(x) - P(x)\|_{[x_0-r,x_0+r]} \leq O(\epsilon \log d),$$
$$\|P(x)\|_{[-1,1]} \leq O(\epsilon \log d) + \|f(x)\|_{[x_0-r-\delta/2, x_0+r+\delta/2]} \leq O(\epsilon \log d) + B,$$
$$\|P(x)\|_{[-1,1]\setminus[x_0-r-\delta/2, x_0+r+\delta/2]} \leq O(\epsilon \log d).$$

Furthermore, the coefficient vector $\mathbf{c}^{(P)}$ of P has a norm bounded by $\|\mathbf{c}^{(P)}\|_1 \leq O(Bd)$.

The main ingredient, and the primary challenge, for demonstrating Theorem 3.8 is to construct a low-weight approximation using Fourier series, as shown in Lemma 37 of [vAGGdW20], which requires computing the powers of sub-stochastic matrices in bounded space (Lemma 2.13).

Lemma 3.9 (Space-efficient low-weight approximation by Fourier series). *Let $0 < \delta, \epsilon < 1$ and $f\colon \mathbb{R} \to \mathbb{R}$ be a real-valued function such that $|f(x) - \sum_{k=0}^{K} a_k x^k| \leq \epsilon/4$ for all $x \in \mathcal{I}_\delta$, the interval $\mathcal{I}_\delta := [-1+\delta, 1-\delta]$ and $\|\mathbf{a}\|_1 \leq O(\max\{\epsilon^{-1}, \delta^{-1}\})$. Then there is a coefficient vector $\mathbf{c} \in \mathbb{C}^{2M+1}$ such that*

- *For even functions, $\left|f(x) - \sum_{m=-M}^{M} c_m^{(\text{even})}\cos(\pi x m)\right| \leq \epsilon$ for any $x \in \mathcal{I}_\delta$;*
- *For odd functions, $\left|f(x) - \sum_{m=-M}^{M} c_m^{(\text{odd})}\sin\left(\pi x\left(m+\frac{1}{2}\right)\right)\right| \leq \epsilon$ for any $x \in \mathcal{I}_\delta$;*
- *Otherwise, $\left|f(x) - \sum_{m=-M}^{M}\left(c_m^{(\text{even})}\cos(\pi x m) + c_m^{(\text{odd})}\sin\left(\pi x\left(m+\frac{1}{2}\right)\right)\right)\right| \leq \epsilon$ for any $x \in \mathcal{I}_\delta$.*

Here $M := \max\left(2\lceil\delta^{-1}\ln(4\|a\|_1\epsilon^{-1})\rceil, 0\right)$ and $\|\mathbf{c}\|_1 \leq \|\mathbf{a}\|_1$. Moreover, the coefficient vector $\mathbf{c}$ can be computed in bounded-error randomized time $\tilde{O}(\delta^{-5}\epsilon^{-2})$ and space $O(\log(\delta^{-4}\epsilon^{-1}))$.

Proof. We begin by defining $\|f\|_\infty := \sup\{|f(x)| : x \in [-1+\delta, 1-\delta]\}$. It is worth noting that the truncation error of $\sum_{k=0}^{K} a_k x^k$, as shown in [SM03, Theorem A.4], is $(1-\delta)^{k+1} \leq e^{-\delta(k+1)} \leq \epsilon$, implying that $K \geq \Omega(\delta^{-1}\ln\epsilon^{-1})$. Without loss of generality, we can assume that $\|\mathbf{a}\|_1 \geq \epsilon/2$.[27]

Construction of polynomial approximations. Our construction involves three approximations, as described in Lemma 37 of [vAGGdW20]. We defer the detailed proofs of all three approximations to Appendix A.1.2.

The first approximation combines the assumed $\sum_{k=0}^{K} a_k x^k$ with $\arcsin(x)$'s Taylor series.

[27] This is because if $\|\mathbf{a}\|_1 < \epsilon/2$, then $\|f\|_\infty \leq \|f(x) - \sum_{k=0}^{K} a_k x^k\|_\infty + \|\sum_{k=0}^{K} a_k x^k\|_\infty \leq \epsilon/4 + \|\mathbf{a}\|_1 < \epsilon$, implying that $M = 0$ and $\mathbf{c} = 0$.

Proposition 3.9.1 (First approximation)**.** *Let* $\hat{f}_1(x) := \sum_{k=0}^K a_k x^k$ *such that* $\|f - \hat{f}_1\|_\infty \le \epsilon/4$. *Then we know that* $\hat{f}_1(x) = \sum_{k=0}^K a_k \sum_{l=0}^\infty b_l^{(k)} \sin^l\left(\frac{x\pi}{2}\right)$ *where the coefficients* $b_l^{(k)}$ *satisfy that*

$$b_l^{(k+1)} = \sum_{l'=0}^{l} b_{l'}^{(k)} b_{l-l'}^{(1)} \text{ where } b_l^{(1)} = \begin{cases} 0 & \text{if } l \text{ is even,} \\ \binom{l-1}{\frac{l-1}{2}} \frac{2^{-l+1}}{l} \cdot \frac{2}{\pi} & \text{if } l \text{ is odd.} \end{cases} \tag{3.3}$$

Furthermore, the coefficients $\{b_l^{(k)}\}$ *satisfies the following: (1)* $\|\mathbf{b}^{(k)}\|_1 = 1$ *for all* $k \ge 1$*; (2)* $\mathbf{b}^{(k)}$ *is entry-wise non-negative for all* $k \ge 1$*; (3)* $b_l^{(k)} = 0$ *if* l *and* k *have different parities.*

The second approximation truncates the series at $l = L$, and bounds the truncation error.

Proposition 3.9.2 (Second approximation)**.** *Let* $\hat{f}_2(x) := \sum_{k=0}^K a_k \sum_{l=0}^L b_l^{(k)} \sin^l\left(\frac{x\pi}{2}\right)$ *where* $L := \lceil \delta^{-2} \ln(4\|\mathbf{a}\|_1 \epsilon^{-1}) \rceil$*, then we have that* $\|\hat{f}_1 - \hat{f}_2\|_\infty \le \epsilon/4$.

The third approximation approximates the functions $\sin^l(x)$ in $\hat{f}_2(x)$ using a tail bound of the binomial distribution. Notably, this construction not only quadratically improves the dependence on δ, but also ensures that the integrand's second derivative is *bounded* when combined with Lemma 3.5.

Proposition 3.9.3 (Third approximation)**.** *Let* $\hat{f}_3(x)$ *be polynomial approximations of* f *that depends on the parity of* f *such that* $\|\hat{f}_2 - \hat{f}_3\| \le \epsilon/2$ *and* $M = \lfloor \delta^{-1} \ln(4\|\mathbf{a}\|_1 \epsilon^{-1}) \rfloor$*, then we have*

$$\hat{f}_3^{(\text{even})}(x) := \sum_{k=0}^K a_k \sum_{\hat{l}=0}^{L/2} (-1)^{\hat{l}} 2^{-2\hat{l}} b_{2\hat{l}}^{(k)} \sum_{m'=\hat{l}-M}^{\hat{l}+M} (-1)^{m'} \binom{2\hat{l}}{m'} \cos(\pi x(m' - \hat{l})),$$

$$\hat{f}_3^{(\text{odd})}(x) := \sum_{k=0}^K a_k \sum_{\hat{l}=0}^{(L-1)/2} (-1)^{\hat{l}+1} 2^{-2\hat{l}-1} b_{2\hat{l}+1}^{(k)} \sum_{m'=\hat{l}+1-M}^{\hat{l}+1+M} (-1)^{m'} \binom{2\hat{l}+1}{m'} \sin\left(\pi x\left(m' - \hat{l} - \tfrac{1}{2}\right)\right).$$

Therefore, we have that $\hat{f}_3(x) := \hat{f}_3^{(\text{even})}(x)$ *if* f *is even, whereas* $\hat{f}_3(x) := \hat{f}_3^{(\text{odd})}(x)$ *if* f *is odd. In addition, if* f *is neither even or odd, then* $\hat{f}_3(x) := \hat{f}_3^{(\text{even})}(x) + \hat{f}_3^{(\text{odd})}(x)$.

We adopt the third approximation as our construction by rearranging the summations and introducing a new parameter m. The value of m is defined as $m := m' - \hat{l}$ if f is even and $m := m' - \hat{l} - 1$ if f is odd. Moreover, the definition of m depends on the parity of $l = 2\hat{l} + 1$[28] if f is neither even nor odd. By applying this approach, we can derive the following:

$$\begin{aligned} \hat{f}_3^{(\text{even})}(x) &= \sum_{m=-M}^{M} c_m^{(\text{even})} \cos(\pi x m) \text{ where } c_m^{(\text{even})} := (-1)^m \sum_{k=0}^K a_k \sum_{\hat{l}=0}^{L/2} b_{2\hat{l}}^{(k)} \binom{2\hat{l}}{m+\hat{l}} 2^{-2\hat{l}}; \\ \hat{f}_3^{(\text{odd})}(x) &= \sum_{m=-M}^{M} c_m^{(\text{odd})} \sin\left(\pi x\left(m + \tfrac{1}{2}\right)\right) \text{ where } c_m^{(\text{odd})} := (-1)^m \sum_{k=0}^K a_k \sum_{\hat{l}=0}^{(L-1)/2} b_{2\hat{l}+1}^{(k)} \binom{2\hat{l}+1}{m+\hat{l}+1} 2^{-2\hat{l}-1}. \end{aligned} \tag{3.4}$$

We then notice that the rearrangement of terms in Equation (3.4) can be directly applied to the definition of $\hat{f}_3(x)$ in Proposition 3.9.3. As a consequence, we obtain the following bound on the accumulative error: $\|f - \hat{f}_3\|_\infty \le \|f - \hat{f}_1\|_\infty + \|\hat{f}_1 - \hat{f}_2\|_\infty + \|\hat{f}_2 - \hat{f}_3\|_\infty \le \epsilon$. Additionally, we remark that $\|\mathbf{c}\|_1 \le \|\mathbf{a}\|_1$, since $\|\mathbf{b}^{(k)}\|_1 = 1$ (see Proposition 3.9.1) and $\sum_{m=0}^{l} \binom{l}{m} = 2^l$.

Analyzing time and space complexity. To evaluate the bounded polynomial approximation $\hat{f}_3(x)$ with ϵ accuracy, it is necessary to approximate the summand with ℓ-bit precision, where $\ell = O(\log(KLM\epsilon^{-1})) = O(\log(\delta^{-4}\epsilon^{-1}))$. Since the summand is a product of a constant number of holonomic functions, approximating $b_l^{(k)}$ with ℓ-bit precision is sufficient. Other quantities in the summand can be evaluated with the desired accuracy in deterministic time $\tilde{O}(\ell)$ and space $O(\ell)$ as stated in Remark 3.6.

[28] In particular, the summand in $\hat{f}_3(x)$ is $c_m^{(\text{even})} \cos(\pi x m) + c_m^{(\text{odd})} \sin\left(\pi x\left(m + \frac{1}{2}\right)\right)$ if f is neither even nor odd.

We now present a bounded-error randomized algorithm for estimating $b_l^{(k)}$. As $\mathbf{b}^{(1)}$ is entry-wise non-negative and $\sum_{i=1}^{l} b_i^{(1)} < \|\mathbf{b}^{(1)}\|_1 = 1$ following Proposition 3.9.1, we can express the recursive formula in Equation (3.3) as the matrix powering of a sub-stochastic matrix B_1:

$$B_1^k := \begin{pmatrix} b_1^{(1)} & b_2^{(1)} & \cdots & b_{l-1}^{(1)} & b_l^{(1)} \\ 0 & b_1^{(1)} & \cdots & b_{l-2}^{(1)} & b_{l-1}^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_1^{(1)} & b_2^{(1)} \\ 0 & 0 & \cdots & 0 & b_1^{(1)} \end{pmatrix}^k = \begin{pmatrix} b_1^{(k)} & b_2^{(k)} & \cdots & b_{l-1}^{(k)} & b_l^{(k)} \\ 0 & b_1^{(k)} & \cdots & b_{l-2}^{(k)} & b_{l-1}^{(k)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_1^{(k)} & b_2^{(k)} \\ 0 & 0 & \cdots & 0 & b_1^{(k)} \end{pmatrix} := B_k.$$

In addition, we approximate the sub-stochastic matrix B_1 by dyadic rationals with ℓ-bit precision, denoted as $\hat{B}_1$. Utilizing Lemma 2.13, we can compute any entry $\hat{B}_1^k[s,t]$ with a randomized algorithm that runs in $O(\ell k)$ time and $\log(l+1)$ space with acceptance probability $\hat{B}_1^k[s,t]$. To evaluate $\hat{B}_1^k[s,t]$ with an additive error of ϵ, we use the sequential repetitions outlined in Lemma 2.11. Specifically, we repeat the algorithm $m = 2\epsilon^{-2}\ln(KLM) = O(\epsilon^{-2}\log(\delta^{-4}))$ times, and each turn succeeds with probability at least $1 - 1/(3KLM)$. Note that the number of the evaluation of $b_l^{(k)}$ for computing $\hat{f}_3(x)$ is $O(KLM)$, and by the union bound, we can conclude that the success probability of evaluating all coefficients in $\mathbf{c}$ is at least $2/3$.

Finally, we complete the proof by analyzing the overall computational complexity. It is evident that our algorithm utilizes $O(\ell + \log m) = O(\log(\delta^{-4}\epsilon^{-3}))$ space because indexing m repetitions requires additional $O(\log m)$ bits. Moreover, since there are $O(KLM)$ summands in $\hat{f}_3(x)$, and evaluating $b_l^{(k)}$ takes m repetitions with time complexity $O(\ell K)$ for a single turn, the overall time complexity is $O(KLM \cdot \ell K \cdot \epsilon^{-2}\log(KLM)) = \tilde{O}(\delta^{-5}\epsilon^{-2})$. □

Now we present the proof of Theorem 3.8, which is a space-efficient and randomized algorithm for constructing bounded polynomial approximations of piecewise-smooth functions.

Proof of Theorem 3.8. Our approach is based on Theorem 40 in [vAGGdW20] and Corollary 23 in [GSLW19]. Firstly, we obtain a Fourier approximation $\hat{f}(x)$ of the given function $f(x)$ by truncating it using Lemma 3.9. Next, we ensure that $\hat{f}(x)$ is negligible outside the interval $[-x_0 - r, x_0 + r]$ by multiplying it with a suitable rectangle function, denoted as $h(x)$. Finally, we derive a space-efficient polynomial approximation $\hat{h}(x)$ of $h(x)$ by applying Lemma 3.5.

Construction of a bounded function. Let us begin by defining a linear transformation $L(x) := \frac{x-x_0}{r+\delta}$ that maps $[x_0 - r - \delta, x_0 + r + \delta]$ to $[-1, 1]$. For convenience, we denote $g(y) := f(L^{-1}(y))$ and $b_l := a_l(r+\delta)^l$, then it is evident that $g(y) := \sum_{l=0}^{\infty} b_l y^l$ for any $y \in [-1, 1]$.

To construct a Fourier approximation by Lemma 3.9, we need to bound the truncation error $\varepsilon_J^{(g)}$. We define $\delta' := \frac{\delta}{2(r+\delta)}$ and $J := \lceil (\delta')^{-1}\log(12B\epsilon^{-1}) \rceil$. This ensures that the truncation error $\varepsilon_J^{(g)} := \left|g(y) - \sum_{j=0}^{J-1} b_j y^j\right|$ for any $y \in [-1+\delta', 1-\delta']$ satisfies the following:

$$\varepsilon_J^{(g)} = \Big|\sum_{j=J}^{\infty} b_j y^j\Big| \leq \sum_{j=J}^{\infty} \left|b_j(1-\delta')^j\right| \leq (1-\delta')^J \sum_{j=J}^{\infty} |b_j| \leq (1-\delta')^J B \leq e^{-\delta' J} B \leq \frac{\epsilon}{12} := \frac{\epsilon'}{4}.$$

Afterward, let $\hat{\mathbf{b}} := (b_0, b_1, \cdots, b_{J-1})$, then we know that $\|\hat{\mathbf{b}}\|_1 \leq \|\mathbf{b}\|_1 \leq B$ by the assumption. Now we utilize Lemma 3.9 and obtain the Fourier approximation $\hat{g}(y)$:

$$\hat{g}(y) := \begin{cases} \sum_{m=-M}^{M} c_m^{(\text{even})} \cos(\pi y m), & \text{if } f \text{ is even} \\ \sum_{m=-M}^{M} c_m^{(\text{odd})} \sin\!\big(\pi y (m+\frac{1}{2})\big), & \text{if } f \text{ is odd} \\ \sum_{m=-M}^{M} \Big(c_m^{(\text{even})}\cos(\pi y m) + c_m^{(\text{odd})}\sin\big(\pi y (m+\frac{1}{2})\big)\Big), & \text{otherwise} \end{cases}. \tag{3.5}$$

By appropriately choosing $M = O\big((\delta')^{-1}\log\big(\|\hat{\mathbf{b}}\|_1/\epsilon'\big)\big) = O\big(r\delta^{-1}\log\big(B/\epsilon\big)\big)$, we obtain that the vectors of coefficients $\mathbf{c}^{(\text{even})}$ and $\mathbf{c}^{(\text{odd})}$ satisfy $\|\mathbf{c}^{(\text{even})}\|_1 \le \|\hat{\mathbf{b}}\|_1 \le \|\mathbf{b}\|_1 \le B$ and similarly $\|\mathbf{c}^{(\text{odd})}\|_1 \le B$. Plugging $f(x) = g(L(x))$ into Equation (3.5), we conclude that $\hat{f}(x) = \hat{g}(L(x))$ is a Fourier approximation of f with an additive error of $\epsilon/3$ on the interval $[x_0-r-\delta/2, x_0+r+\delta/2]$:

$$\hat{f}(x) = \hat{g}\Big(\frac{x-x_0}{r+\delta}\Big) = \begin{cases} \sum\limits_{m=-M}^{M} c_m^{(\text{even})}\cos\big(\pi m\big(\frac{x-x_0}{r+\delta}\big)\big), & \text{if } f \text{ is even} \\ \sum\limits_{m=-M}^{M} c_m^{(\text{odd})}\sin\big(\pi\big(m+\frac{1}{2}\big)\big(\frac{x-x_0}{r+\delta}\big)\big), & \text{if } f \text{ is odd} \\ \sum\limits_{m=-M}^{M} c_m^{(\text{even})}\cos\big(\pi m\big(\frac{x-x_0}{r+\delta}\big)\big) + c_m^{(\text{odd})}\sin\big(\pi\big(m+\frac{1}{2}\big)\big(\frac{x-x_0}{r+\delta}\big)\big), & \text{otherwise} \end{cases}.$$

Making the error negligible outside the interval. Subsequently, we define the function $h(x) = \hat{f}(x)\cdot R(x)$ such that it becomes negligible outside the interval of interest, i.e., $[x_0 - r - \delta/2, x_0 + r + \delta/2]$. Here, the approximate rectangle function $R(x)$ is $\tilde{\epsilon}$-close to 1 on the interval $[x_0 - r, x_0 + r]$, and is $\tilde{\epsilon}$-close to 0 on the interval $[-1, 1] \setminus [x_0 - r - 2\tilde{\delta}, x_0 + r + 2\tilde{\delta}]$, where $\tilde{\epsilon} := \epsilon/(3B)$ and $\tilde{\delta} := \delta/4$. Moreover, $|R(x)| \le 1$ for any $x \in [-1, 1]$. Similar to Lemma 29 in [GSLW19], $R(x)$ can be expressed as a linear combination of Gaussian error functions:

$$R(x) := \frac{1}{2}\Big[\operatorname{erf}\big(\kappa(x-x_0+r+\delta')\big) - \operatorname{erf}\big(\kappa(x-x_0-r-\delta')\big)\Big] \text{where } \kappa := \frac{2}{\delta'}\log^{\frac{1}{2}}\frac{\sqrt{2}}{\sqrt{\pi}\epsilon'} = \frac{8}{\delta}\log^{\frac{1}{2}}\frac{\sqrt{18}B}{\sqrt{\pi}\epsilon}. \quad (3.6)$$

Bounded polynomial approximation with Chebyshev interpolation. We here present an algorithmic, space-efficient, randomized polynomial approximation method using Chebyshev interpolation to approximate the function $h(x) := \hat{f}(x)\cdot R(x)$. As suggested in Proposition 3.8.1, we use an explicit polynomial approximation $\hat{P}(x)$ of the bounded function $h(x)$ of degree $d = O(\delta^{-1}\log(B\epsilon^{-1}))$ that satisfies the conditions specified in Equation (3.7).

Proposition 3.8.1 (Bounded polynomial approximations based on a local Taylor series, adapted from [GSLW19, Corollary 23])**.** *Let $x_0 \in [-1,1]$, $r \in (0,2]$, $\delta \in (0,r]$ and let $f\colon [-x_0-r-\delta, x_0+r+\delta] \to \mathbb{R}$ and be such that $f(x_0+x) := \sum_{l=0}^{\infty} a_l x^l$ for all $x \in [-r-\delta, r+\delta]$. Suppose $B > 0$ is such that $\sum_{l=0}^{\infty}(r+\delta)^l|a_l| \le B$. Let $\epsilon \in \big(0, \frac{1}{2B}\big]$, there is a $\epsilon/3$-precise Fourier approximation $\tilde{f}(x)$ of $f(x)$ on the interval $[x_0 - r+\delta/2, x_0+r+\delta/2]$, where $\hat{f}(x) := \sum_{m=-M}^{M}\mathrm{Re}\Big[\tilde{c}_m e^{-\frac{i\pi m}{2(r+\delta)}x_0} e^{\frac{i\pi m}{2(r+\delta)}x}\Big]$ and $\|\tilde{\mathbf{c}}\|_1 \le B$. We have a time-efficient polynomial $P^* \in \mathbb{R}[x]$ of degree $O(\delta^{-1}\log(B\epsilon^{-1}))$ s.t.*

$$\begin{aligned} &\|\hat{f}(x)R(x) - P^*(x)\|_{[x_0-r,x_0+r]} \le \epsilon, \\ &\|P^*(x)\|_{[-1,1]} \le \epsilon + \|\hat{f}(x)R(x)\|_{[x_0-r-\delta/2,x_0+r+\delta/2]} \le \epsilon + B, \quad (3.7) \\ &\|P^*(x)\|_{[-1,1]\setminus[x_0-r-\delta/2,x_0+r+\delta/2]} \le \epsilon. \end{aligned}$$

To utilize Lemma 3.5, we need to bound the second derivative $\max_{\xi\in[-\pi,0]}|F_k''(\xi)|$, where the integrand $F_k(\cos\theta) := \cos(k\theta)h(\cos\theta)$ for any $0 \le k \le d$. We will calculate this upper bound directly in Fact 3.8.2, and the proof is deferred to Appendix A.1.3.

Fact 3.8.2. *Consider the integrand $F_k(\theta) = \sum_{m=-M}^{M}\frac{c_m}{2}\big(H_{k,m}^{(+)} - H_{k,m}^{(-)}\big)$ for any function f which is either even or odd. If f is even, we have that $c_m = c_m^{(\text{even})}$ defined in Lemma 3.9, and*

$$H_{k,m}^{(\pm)}(\theta) := \cos\Big(\pi m\Big(\frac{\cos\theta - x_0}{r+\delta}\Big)\Big)\cdot\cos(k\theta)\cdot\operatorname{erf}\Big(\kappa\Big(\cos\theta - x_0 \pm r \pm \frac{\delta}{4}\Big)\Big). \quad (3.8)$$

Likewise, if f is odd, we know that $c_m = c_m^{(\text{odd})}$ defined in Lemma 3.9, and

$$H_{k,m}^{(\pm)}(\theta) := \sin\Big(\pi\Big(m+\frac{1}{2}\Big)\Big(\frac{\cos\theta - x_0}{r+\delta}\Big)\Big)\cdot\cos(k\theta)\cdot\operatorname{erf}\Big(\kappa\Big(\cos\theta - x_0 \pm r \pm \frac{\delta}{4}\Big)\Big). \quad (3.9)$$

Moreover, the integrand is $F_k(\theta) = \sum_{m=-M}^{M}\Big(\frac{c_m^{(\text{even})}}{2}\big(\hat{H}_{k,m}^{(+)} - \hat{H}_{k,m}^{(-)}\big) + \frac{c_m^{(\text{odd})}}{2}\big(\tilde{H}_{k,m}^{(+)} - \tilde{H}_{k,m}^{(-)}\big)\Big)$ when f is neither even nor odd, where $\hat{H}_{k,m}^{(\pm)}$ and $\tilde{H}_{k,m}^{(\pm)}$ follow from Equation (3.8) and Equation (3.9), *respectively. Regardless of the parity of f, we have that the second derivative $F''_k(\theta) \leq O(Bd^3)$.*

Together with Fact 3.8.2, we are ready to apply Lemma 3.5 to $h(x) = \hat{f}(x)R(x)$, resulting in a degree-d polynomial $P(x)$. Since $P(x)$ is the minimax approximation of $P^*(x)$ by Chebyshev interpolation and satisfies Equation (3.7), we can define intervals $\mathcal{I}_{\text{int}} := [x_0 - r, x_0 + r]$ and $\mathcal{I}_{\text{ext}} := [x_0 - r - \delta/2, x_0 + r + \delta/2]$ to obtain:

$$\begin{aligned} \|f(x) - P(x)\|_{\mathcal{I}_{\text{int}}} &\leq \|f(x) - h(x)\|_{\mathcal{I}_{\text{int}}} + \|h(x) - P(x)\|_{\mathcal{I}_{\text{int}}} \leq \epsilon + O(\epsilon \log d) = O(\epsilon \log d), \\ \|P(x) - 0\|_{\mathcal{I}_{\text{ext}}} &\leq \|P(x) - h(x)\|_{\mathcal{I}_{\text{ext}}} + \|h(x) - 0\|_{\mathcal{I}_{\text{ext}}} \leq O(\epsilon \log d) + O(\epsilon) \leq O(\epsilon \log d). \end{aligned} \tag{3.10}$$

We can achieve the desired error bound by observing that Equation (3.10) implies $|P(x)|_{[-1,1]} \leq O(\epsilon \log d) + |P(x)|_{[-1,1]\setminus\mathcal{I}_{\text{ext}}} \leq O(\epsilon \log d) + B$. Moreover, we note that the norm of the coefficient vector $\mathbf{c}^{(P)}$ of the polynomial $P(x)$ is bounded by $|\mathbf{c}^{(P)}| \leq O(Bd) \cdot (1 + O(\epsilon \log d)) = O(Bd)$, which follows directly from our utilization of Lemma 3.5.

Analyzing time and space complexity. The construction of $\hat{f}(x)$ can be implemented in bounded-error randomized time $\tilde{O}((\delta')^{-5}\epsilon^{-2}B^2)$ and space $O(\log((\delta')^{-4}\epsilon^{-1}B))$, given that this construction uses Lemma 3.9 with $\delta' = \frac{\delta}{2(r+\delta)} \in (0, \frac{1}{2}]$ and $\epsilon' = \frac{\epsilon}{3B}$. Having $\hat{f}(x)$, we can construct a bounded polynomial approximation $\hat{h}(x)$ deterministically using Lemma 3.5. This construction can be implemented in deterministic time $\tilde{O}(d^2\epsilon^{-1/2}B)$ and space $O(\log(d^4\epsilon^{-1}B))$ since the integrand $F_k(\theta)$ is a product of a constant number of (compositions of) holonomic functions (Remark 3.6). Therefore, our construction can be implemented in bounded-error randomized time $\tilde{O}(\max\left\{(\delta')^{-5}\epsilon^{-2}B^2, d^2\epsilon^{-1/2}B\right\})$ and space $O(\log(d^4(\delta')^{-4}\epsilon^{-1}B))$. □

With the aid of Theorem 3.8, we can provide a space-efficient polynomial approximation to the normalized logarithmic function utilized in Lemma 11 of [GL20].

Corollary 3.10 (Space-efficient polynomial approximation to the normalized logarithmic function)**.** *Let $\beta \in (0, 1]$ and $\epsilon \in (0, 1/2)$, there is an even polynomial P of degree $d \leq \tilde{C}_{\ln}\beta^{-1}\log \epsilon^{-1}$ where $\tilde{C}_{\ln}$ is a universal constant such that*

$$\forall x \in [\beta, 1], \left|P(x) - \frac{\ln(1/x)}{2\ln(2/\beta)}\right| \leq C_{\ln}\epsilon \log d, \text{ where } C_{\ln} \text{ is a universal constant,}$$
$$\forall x \in [-1, 1], |P(x)| \leq 1.$$

Moreover, the coefficient vector $\mathbf{c}^{(P)}$ of P has a norm bounded by $\|\mathbf{c}^{(P)}\|_1 \leq \hat{C}_{\ln}d$, where $\hat{C}_{\ln}$ is another universal constant. In addition, any entry of the coefficient vector $\mathbf{c}^{(P)}$ can be computed in bounded-error randomized time $\tilde{O}(\max\{\beta^{-5}\epsilon^{-2}, d^2\epsilon^{-1/2}\})$ and space $O(\log(d^4\beta^{-4}\epsilon^{-1}))$. Without loss of generality, we assume that all constants $C_{\ln}$, $\hat{C}_{\ln}$, and $\tilde{C}_{\ln}$ are at least 1.

Proof. Consider the function $f(x) := \frac{\ln(1/x)}{2\ln(2/\beta)}$. We apply Theorem 3.8 to $f(x)$ by choosing the same parameters as in Lemma 11 of [GL20], specifically $\epsilon' = \epsilon/2$, $x_0 = 1$, $r = 1 - \beta$, $\delta = \beta/2$, and $B = 1/2$.[29] This results in a space-efficient randomized polynomial approximation $\tilde{P} \in \mathbb{R}[x]$ of degree $d = O(\delta^{-1}\log(\epsilon^{-1}B)) \leq \tilde{C}_{\ln}\beta^{-1}\log \epsilon^{-1}$, where $\tilde{C}_{\ln}$ is a universal constant. By appropriately choosing $\eta \leq 1/2$ such that $C'_{\ln}\epsilon \log d = \eta/4$ for a universal constant $C'_{\ln}$, the approximation guarantees the following inequalities:

$$\begin{aligned} \|f(x) - \tilde{P}(x)\|_{[\beta,2-\beta]} &\leq C'_{\ln}\epsilon \log d = \tfrac{\eta}{4} \\ \|\tilde{P}(x)\|_{[-1,1]} &\leq B + C'_{\ln}\epsilon \log d \leq \tfrac{1}{2} + C'_{\ln}\epsilon \log d = \tfrac{1}{2} + \tfrac{\eta}{4} \\ \|\tilde{P}(x)\|_{[-1,\beta/2]} &\leq C'_{\ln}\epsilon \log d = \tfrac{\eta}{4}. \end{aligned} \tag{3.11}$$

[29] As indicated in Lemma 11 of [GL20], since the Taylor series of $f(x)$ at $x = 1$ is $\frac{1}{2\ln(2/\beta)}\sum_{l=1}^{\infty}\frac{(-1)^l x^l}{l}$, we obtain that $B = f(\frac{\beta}{2} - 1) = \frac{1}{2\ln(2/\beta)}\sum_{l=1}^{\infty}\frac{(1-\beta/2)^l}{l} = -\frac{1}{2\ln(2/\beta)}\sum_{l=1}^{\infty}\frac{(-1)^{l-1}}{l}(\beta/2-1)^l = -\frac{1}{2\ln(2/\beta)}\ln\frac{\beta}{2} = \frac{1}{2}$.

Additionally, the coefficient vector $\mathbf{c}^{(\tilde{P})}$ of $\tilde{P}$ satisfies that $\|\mathbf{c}^{(\tilde{P})}\|_1 \leq O(Bd) \leq \hat{C}_{\ln} d$ where $\hat{C}_{\ln}$ is a universal constant. Notice that $\delta' = \frac{\delta}{2(r+\delta)} = \frac{\beta/2}{2(1-\beta+\beta/2)} = \frac{\beta}{4(1-\beta/2)} = \Theta(\beta)$, our utilization of Theorem 3.8 yields a bounded-error randomized algorithm that requires $O(\log(d^4(\delta')^{-4}\epsilon^{-1}B)) = O(\log(d^4\beta^{-4}\epsilon^{-1}))$ space and $\tilde{O}(\max\{(\delta')^{-5}\epsilon^{-2}B^2, d^2\epsilon^{-1/2}B\}) = \tilde{O}(\max\{\beta^{-5}\epsilon^{-2}, d^2\epsilon^{-1/2}\})$ time.

Furthermore, note that the real-valued function $f(x)$ only defines when $x > 0$, then $\tilde{P}(x)$ is not an even polynomial in general. Instead, we consider $P(x) := (1+\eta)^{-1}(\tilde{P}(x) + \tilde{P}(-x))$ for all $x \in [-1,1]$. Together with Equation (3.11), we have derived that:

$$\begin{aligned}
\|f(x) - P(x)\|_{[\beta,1]} &\leq \left\|f(x) - \tfrac{1}{1+\eta}\tilde{P}(x)\right\|_{[\beta,1]} + \left\|\tfrac{1}{1+\eta}\tilde{P}(-x)\right\|_{[\beta,1]}\\
&\leq \left\|f(x) - \tilde{P}(x)\right\|_{[\beta,1]} + \left\|\tilde{P}(x) - \tfrac{1}{1+\eta}\tilde{P}(x)\right\|_{[\beta,1]} + \left\|\tfrac{1}{1+\eta}\tilde{P}(-x)\right\|_{[\beta,1]}\\
&\leq \tfrac{\eta}{4} + \tfrac{\eta}{1+\eta}\cdot\left(\tfrac{1}{2}+\tfrac{\eta}{4}\right) + \tfrac{1}{1+\eta}\cdot\tfrac{\eta}{4}\\
&= \tfrac{\eta}{4} + \tfrac{\eta}{1+\eta}\cdot\tfrac{1+\eta}{4} + \tfrac{1}{1+\eta}\cdot\tfrac{\eta}{2}\\
&\leq \eta.
\end{aligned} \tag{3.12}$$

Here, the last line owes to the fact that $\eta > 0$. Consequently, Equation (3.12) implies that $\|f(x) - P(x)\|_{[\beta,1]} \leq 4C'_{\ln}\epsilon\log d := C_{\ln}\epsilon\log d$ for another universal constant $C_{\ln}$. Notice $P(x)$ is an even polynomial with $\deg(P) \leq \tilde{C}_{\ln}\beta^{-1}\log\epsilon^{-1}$, Equation (3.11) yields that:

$$\|P(x)\|_{[-1,1]} = \|P(x)\|_{[0,1]} \leq \|\tfrac{1}{1+\eta}\tilde{P}(x)\|_{[0,1]} + \|\tfrac{1}{1+\eta}\tilde{P}(x)\|_{[-1,0]} \leq \tfrac{1}{1+\eta}\cdot\tfrac{1+\eta}{2} + \tfrac{1}{1+\eta}\cdot\tfrac{\eta}{2} \leq 1.$$

We now complete the proof by noticing $\eta \leq 1/2$. □

3.2 Applying Chebyshev interpolation to bitstring indexed encodings

Equipped with space-efficient bounded polynomial approximations of piecewise-smooth functions, it suffices to implement Chebyshev interpolation on bitstring indexed encodings, as specified in Theorem 3.11. The proof follows from combining Lemma 3.13 and Lemma 3.14.

Theorem 3.11 (Chebyshev interpolation applied to bitstring indexed encodings). *Let A be an Hermitian matrix acting on s qubits, and let U be a $(1,a,\epsilon_1)$-bitstring indexed encoding of A that acts on $s+a$ qubits. For any degree-d polynomial $P_d(x) = \frac{c_0}{2} + \sum_{k=1}^d c_k T_k(x)$ where $d \leq 2^{O(s(n))}$ and T_k is the k-th Chebyshev polynomial (of the first kind), equipped with an evaluation oracle* Eval *that returns $\hat{c}_k$ with precision $\varepsilon := O(\epsilon_2^2/d)$, then we have a $(1, a', 144d\sqrt{\epsilon_1}\|\mathbf{c}\|_1^2 + 36\epsilon_2\|\mathbf{c}\|_1)$-bitstring indexed encoding V of $P_d(A)$ that acts on $s+a'$ qubits where $a' := a + \lceil\log d\rceil + 3$. This implementation requires $O(d^2\|\mathbf{c}\|_1)$ uses of U, $U^\dagger$,* $\mathrm{C}_\Pi\mathrm{NOT}$, $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$, *and $O(d^2\|\mathbf{c}\|_1)$ multi-controlled single-qubit gates.*[30] *Moreover, we can compute the description of the resulting quantum circuit in deterministic time $\tilde{O}(d^2\|\mathbf{c}\|_1\log(d/\epsilon_2))$ and space $O(\max\{s(n), \log(d/\epsilon_2^2)\})$*[31]*, also $O(d^2\|\mathbf{c}\|_1)$ oracle calls to* Eval *with precision ε.*
Furthermore, our construction straightforwardly extends to any linear (possibly non-Hermitian) operator A by simply replacing $P_d(A)$ with $P_d^{(\mathrm{SV})}(A)$ defined in Definition 3.3.

Remark 3.12 (QSVT implementations of Chebyshev interpolation preserve the parity). As shown in Proposition 3.13.1, we can implement the quantum singular value transformation $T_k(A)$ *exactly* for any Hermitian matrix that admits a bitstring indexed encoding, because we observe that the rotation angles corresponding to the k-th Chebyshev polynomials are either $\pi/2$ or $(1-k)\pi/2$, indicating that $T_k(0) = 0$ for any odd k. We then implement the QSVT corresponding to the Chebyshev interpolation polynomial $P_d(x) = \sum_{l=0}^{(d-1)/2} c_{2l+1}T_{2l+1}(x)$, as described in Theorem 3.11, although the actual implementation results in a slightly different polynomial, $\hat{P}_d(x) = \sum_{l=0}^{(d-1)/2} \hat{c}_{2l+1}T_{2l+1}(x)$. However, we still have $\hat{P}_d(0) = 0 = P_d(0)$, indicating that the implementations in Theorem 3.11 preserve the parity.

[30] As indicated in Figure 3(c) of [GSLW19] (see also Lemma 19 in [GSLW18]), we replace the single-qubit gates used in Lemma 3.13 with multi-controlled (or "multiply controlled") single-qubit gates.

We first demonstrate an approach, based on Lemma 3.12 in [MY23], that constructs Chebyshev polynomials of bitstring indexed encodings in a space-efficient manner.

Lemma 3.13 (Chebyshev polynomials applied to bitstring indexed encodings)**.** *Let A be a linear operator acting on s qubits, and let U be a $(1,a,\epsilon)$-bitstring indexed encoding of A that acts on $s+a$ qubits. Then, for the k-th Chebyshev polynomial (of the first kind) $T_k(x)$ of degree $k \leq 2^{O(s)}$, there exists a new $(1,a+1,4k\sqrt{\epsilon})$-bitstring indexed encoding V of $T_k^{(\mathrm{SV})}(A)$ that acts on $s+a+1$ qubits. This implementation requires k uses of U, $U^\dagger$, $\mathrm{C}_\Pi\mathrm{NOT}$, $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$, and k single-qubit gates. Moreover, we can compute the description of the resulting quantum circuit in deterministic time k and space $O(s)$.*
Furthermore, consider $A' := \tilde{\Pi} U \Pi$, where $\tilde{\Pi}$ and Π are the corresponding orthogonal projectors of the bitstring indexed encoding U. If A and A' satisfy the conditions $\|A - A'\| + \left\|\frac{A+A'}{2}\right\|^2 \leq 1$ and $\left\|\frac{A+A'}{2}\right\|^2 \leq \zeta$, then V is a $\left(1, a+1, \frac{\sqrt{2}}{\sqrt{1-\zeta}} k\epsilon\right)$-bitstring indexed encoding of $T_k^{(\mathrm{SV})}(A)$.

Proof. As specified in Proposition 3.13.1, we first notice that we can derive the sequence of rotation angles corresponding to Chebyshev polynomials $T_k(x)$ by directly factorizing them.

Proposition 3.13.1 (Chebyshev polynomials in quantum signal processing, adapted from Lemma 6 in [GSLW19])**.** *Let $T_k \in \mathbb{R}[x]$ be the k-th Chebyshev polynomial (of the first kind). Consider the corresponding sequence of rotation angles $\Phi \in \mathbb{R}^k$ such that $\phi_1 := (1-k)\pi/2$, and $\phi_j := \pi/2$ for all $j \in [k] \setminus \{1\}$, then we know that $\prod_{j=1}^{k} \left[\begin{pmatrix} \exp(i\phi_j) & 0 \\ 0 & \exp(-i\phi_j) \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix}\right] = \begin{pmatrix} T_k & \cdot \\ \cdot & \cdot \end{pmatrix}$.*

Then we implement the quantum singular value transformation $T_k^{(\mathrm{SV})}(A)$, utilizing an alternating phase modulation (Proposition 3.13.2) with the aforementioned sequence of rotation angles, denoted by V.

Proposition 3.13.2 (QSVT by alternating phase modulation, adapted from Theorem 10 and Figure 3 in [GSLW19])**.** *Suppose $P \in \mathbb{C}[x]$ is a polynomial, and let $\Phi \in \mathbb{R}^n$ be the corresponding sequence of rotation angles. We can construct $P^{(\mathrm{SV})}(\tilde{\Pi} U \Pi) = \begin{cases} \tilde{\Pi} U_\Phi \Pi, & \text{if } n \text{ is odd} \\ \Pi U_\Phi \Pi, & \text{if } n \text{ is even} \end{cases}$ with a single ancillary qubit. Moreover, this implementation in [GSLW19, Figure 3] makes k uses of U, $U^\dagger$, $\mathrm{C}_\Pi\mathrm{NOT}$, $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$, and single-qubit gates.*

Owing to the robustness of QSVT (Lemma 22 in [GSLW18], full version of [GSLW19]), we have that $\|T_k^{(\mathrm{SV})}(U) - T_k^{(\mathrm{SV})}(U')\| \leq 4k\sqrt{\|A-A'\|} = 4k\sqrt{\epsilon}$, where U' is a $(1,a,0)$-bitstring indexed encoding of A. Moreover, with a tighter bound for A and A', namely $\|A - A'\| + \left\|\frac{A+A'}{2}\right\|^2 \leq 1$, we can deduce that $\|T_k^{(\mathrm{SV})}(U) - T_k^{(\mathrm{SV})}(U')\| \leq k\frac{\sqrt{2}}{\sqrt{1-\|(A+A')/2\|^2}}\|A-A'\| \leq \frac{\sqrt{2}}{\sqrt{1-\zeta}}k\epsilon$ following [GSLW18, Lemma 23], indicating an improved dependence of ϵ. Finally, we can compute the description of the resulting quantum circuits in $O(\log k) = O(s(n))$ space and $O(k)$ times because of the implementation specified in Proposition 3.13.2. □

We then proceed by presenting a linear combination of bitstring indexed encodings, which adapts the LCU technique proposed by Berry, Childs, Cleve, Kothari, and Somma in [BCC+15], and incorporates a space-efficient state preparation operator. We say that $P_{\mathbf{y}}$ is an ϵ-state preparation operator for $\mathbf{y}$ if $P_{\mathbf{y}}|\bar{0}\rangle := \sum_{i=1}^{m}\sqrt{\hat{y}_i}|i\rangle$ for some $\hat{\mathbf{y}}$ such that $\|\mathbf{y}/\|\mathbf{y}\|_1 - \hat{\mathbf{y}}\|_1 \leq \epsilon$.

Lemma 3.14 (Linear combinations of bitstring indexed encodings, adapted from Lemma 29 in [GSLW19])**.** *Given a matrix $A = \sum_{i=0}^{m-1} y_i A_i$ such that each linear operator A_i $(1 \leq i \leq m)$ acts on s qubits with the corresponding $(\|\mathbf{y}\|_1, a, \epsilon_1)$-bitstring indexed encoding U_i acting on $s+a$ qubits associated with projections $\tilde{\Pi}_i$ and Π_i. Also each y_i $(1 \leq i \leq m)$ can be expressed in $O(s(n))$ bits with an evaluation oracle* Eval *that returns $\hat{y}_i$ with precision $\varepsilon := O(\epsilon_2^2/m)$. Then utilizing an ϵ_2-state preparation operator $P_{\mathbf{y}}$ for $\mathbf{y}$ acting on $O(\log m)$ qubits, and a $(s+a+$*

$\lceil \log m \rceil$)*-qubit unitary* $W = \sum_{i=0}^{m-1} |i\rangle\langle i| \otimes U_i + \left(I - \sum_{i=0}^{m-1} |i\rangle\langle i|\right) \otimes I$*, we can implement a* $(\|\mathbf{y}\|_1, a + \lceil \log m \rceil, \epsilon_1 \|\mathbf{y}\|_1^2 + \epsilon_2 \|\mathbf{y}\|_1)$*-bitstring indexed encoding of* A *acting on* $s + a + \lceil \log m \rceil$ *qubits with a single use of* W*,* $P_{\mathbf{y}}$*,* $P_{\mathbf{y}}^\dagger$*. In addition, the classical pre-processing can be implemented in deterministic time* $\tilde{O}(m^2 \log(m/\epsilon_2))$ *and space* $O(\log(m/\epsilon_2^2))$*,*[31] *as well as* m^2 *oracle calls to* Eval *with precision* ε*.*

Proof. For the ϵ_2-state preparation operator $P_{\mathbf{y}}$ such that $P_{\mathbf{y}}|\bar{0}\rangle = \sum_{i=1}^{m} \sqrt{\hat{y}_i}|i\rangle$, we utilize a scheme introduced by Zalka [Zal98] (also independently rediscovered in [GR02] and [KM01]). We make an additional analysis of the required classical computational complexity, and the proof can be found in Appendix A.2.

Proposition 3.14.1 (Space-efficient state preparation, adapted from [Zal98, KM01, GR02])**.** *Given an* l*-qubit quantum state* $|\psi\rangle := \sum_{i=1}^{m} \sqrt{\hat{y}_i}|i\rangle$*, where* $l = \lceil \log m \rceil$ *and* $\hat{y}_i$ *are real amplitudes associated with an evaluation oracle* $\mathrm{Eval}(i, \varepsilon)$ *that returns* $\hat{y}_i$ *up to accuracy* ε *we can prepare* $|\psi\rangle$ *up to accuracy* ϵ *in deterministic time* $\tilde{O}(m^2 \log(m/\epsilon))$ *and space* $O(\log(m/\epsilon^2))$*, together with* m^2 *evaluation oracle calls with precision* $\varepsilon := O(\epsilon^2/m)$*.*

Now consider the bitstring indexed encoding $\left(P_{\mathbf{y}}^\dagger \otimes I_s\right) W \left(P_{\mathbf{y}} \otimes I_s\right)$ of A acting on $s+a+\lceil \log m \rceil$ qubits. Let $y_i' := y_i/\|\mathbf{y}\|_1$, then we obtain the implementation error:

$$
\begin{aligned}
&\left\| A - \|\mathbf{y}\|_1 \left(|\bar{0}\rangle\langle\bar{0}| \otimes \tilde{\Pi}\right)\left(P_y^\dagger \otimes I_s\right) W \left(P_y \otimes I_s\right) \left(|\bar{0}\rangle\langle\bar{0}| \otimes \Pi\right) \right\| \\
=&\left\| A - \|\mathbf{y}\|_1 \textstyle\sum_{i=0}^{m-1} \hat{y}_i \tilde{\Pi}_i U_i \Pi_i \right\| \\
\leq&\left\| A - \|\mathbf{y}\|_1 \textstyle\sum_{i=0}^{m-1} y_i' \tilde{\Pi}_i U_i \Pi_i \right\| + \|\mathbf{y}\|_1 \textstyle\sum_{i=0}^{m-1} (y_i' - \hat{y}_i) \|\tilde{\Pi}_i U_i \Pi_i\| \\
\leq&\|\mathbf{y}\|_1 \textstyle\sum_{i=0}^{m-1} y_i' \|A_i - \tilde{\Pi}_i U_i \Pi_i\| + \epsilon_2 \|\mathbf{y}\|_1 \\
\leq&\epsilon_1 \|\mathbf{y}\|_1^2 + \epsilon_2 \|\mathbf{y}\|_1.
\end{aligned}
$$

Here, the third line is due to the triangle inequality, the fourth line owes to Proposition 3.14.1, and the fifth line is because U_i is a $(1, a, \epsilon_1)$-bitstring indexed encoding of A_i for $0 \leq i < m$. □

To make the resulting bitstring indexed encoding from Lemma 3.14 with $\alpha = 1$, we need to perform *a renormalization procedure* to construct a new encoding with the desired α. We achieve this by extending the proof strategy outlined by Gilyen [Gil19, Page 52] for block-encodings to bitstring indexed encodings. The renormalization procedure is provided in Lemma 3.15, and the complete proof is available in Appendix A.2. Additionally, similar results have been established in [MY23, Lemma 3.10] and [WZ23b, Corollary 2.8].

Lemma 3.15 (Renormalizing bitstring indexed encoding)**.** *Let* U *be an* (α, a, ϵ)*-bitstring indexed encoding of* A*, where* $\alpha > 1$ *and* $0 < \epsilon < 1$*, and* A *is a linear operator acting on* $s(n)$ *qubits. We can implement a quantum circuit* V*, serving as a normalization of* U*, such that* V *is a* $(1, a+2, 36\epsilon)$*-bitstring indexed encoding of* A*. This implementation requires* $O(\alpha)$ *uses of* U*,* $U^\dagger$*,* $\mathrm{C}_\Pi\mathrm{NOT}$*,* $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$*, and* $O(\alpha)$ *single-qubit gates. Moreover, the description of the resulting quantum circuit can be computed in deterministic time* $O(\alpha)$ *and space* $O(s)$*.*

Finally, we combine Lemma 3.14 and Lemma 3.13 to proceed with the proof of Theorem 3.11.

Proof of Theorem 3.11. By using Lemma 3.13, we have $P_d(A) = \frac{c_0}{2} + \sum_{k=1}^{d} c_k T_k(A)$ where $T_k(A)$ corresponding to a $(1, a+1, 4k\sqrt{\epsilon_1})$-bitstring indexed encoding V_k. Employing Lemma 3.14, we result in a $(\|\mathbf{c}\|_1, \hat{a}, 4k\sqrt{\epsilon_1}\|\mathbf{c}\|_1^2 + \epsilon_2\|\mathbf{c}\|_1)$-bitstring indexed encoding $\hat{V}$ where $\hat{a} := a + \lceil \log d \rceil + 1$. Moreover, by utilizing Lemma 3.15, we obtain a $(1, a', 144k\sqrt{\epsilon_1}\|\mathbf{c}\|_1^2 + 36\epsilon_2\|\mathbf{c}\|_1)$-bitstring indexed encoding V acts on $s + a'$ qubits where $a' := \hat{a} + 2 = a + \lceil \log d \rceil + 3$. A direct calculation demonstrates that this implementation makes $\sum_{k=1}^{d} k \cdot O(\|\mathbf{c}\|_1) = O(d^2 \|\mathbf{c}\|_1)$ uses of U, $U^\dagger$, $\mathrm{C}_\Pi\mathrm{NOT}$,

[31] It is noteworthy that we define $\tilde{O}(f) := O(f \operatorname{poly}\log(f))$.

$\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$, also $\sum_{k=0}^{d} k \cdot O(\|\mathbf{c}\|_1) = O(d^2\|\mathbf{c}\|_1)$ uses of multi-controlled single-qubit gates. In addition, the descriptions of quantum circuits $\{V_k\}_{k=0}^{d}$ can be computed in $O(s(n))$ space and $\sum_{k=0}^{d} k \cdot O(\|\mathbf{c}\|_1) = O(d^2\|\mathbf{c}\|_1)$ time. Therefore, the description of the quantum circuit V can be computed in deterministic time $\max\{\tilde{O}(d^2\|\mathbf{c}\|_1 \log(d/\epsilon_2)), O(d^2\|\mathbf{c}\|_1)\} = \tilde{O}(d^2\|\mathbf{c}\|_1 \log(d/\epsilon_2))$ and space $O(\max\{s(n), d/\epsilon_2^2\})$, as well as $O(d^2\|\mathbf{c}\|_1)$ oracle calls to Eval with precision ε.

Finally, we can extend our construction to any linear operator A by replacing $P_d(A)$ with $P_d^{(\mathrm{SV})}$ as defined in Definition 3.3, taking into account that the Chebyshev polynomial (of the first kind) T_k is either an even or an odd function. □

3.3 Examples: the sign function and the normalized logarithmic function

In this subsection, we provide explicit examples that illustrate the usage of the space-efficient quantum singular value transformation (QSVT) technique. We define two functions:

$$\mathrm{sgn}(x) := \begin{cases} 1, & x > 0 \\ -1, & x < 0 \\ 0, & x = 0 \end{cases} \quad \text{and} \quad \ln_\beta(x) := \frac{\ln(1/x)}{2\ln(2/\beta)}.$$

In particular, the sign function is a bounded function, and we derive the corresponding bitstring indexed encoding with *deterministic* space-efficient classical pre-processing in Corollary 3.7. On the other hand, the logarithmic function is a piecewise-smooth function that is bounded by 1, and we deduce the corresponding bitstring indexed encoding with *randomized* space-efficient classical pre-processing in Corollary 3.10.

Corollary 3.16 (Sign polynomial with space-efficient coefficients applied to bitstring indexed encodings)**.** *Let A be an Hermitian matrix that acts on s qubits, where $s(n) \geq \Omega(\log(n))$. Let U be a $(1, a, \epsilon_1)$-bitstring indexed encoding of A that acts on $s+a$ qubits. Then, for any $d \leq 2^{O(s(n))}$ and $\epsilon_2 \geq 2^{-O(s(n))}$, we have an $\big(1, a+\lceil \log d\rceil+3, 144\hat{C}_{\mathrm{sgn}}^2 d\epsilon_1^{1/2}\log^2(d)+36\hat{C}_{\mathrm{sgn}}\epsilon_2\log(d)\big)$-bitstring indexed encoding V of $P_d^{\mathrm{sgn}}(A)$, where P_d^{sgn} is a space-efficient bounded polynomial approximation of the sign function specified in Corollary 3.7, and $\hat{C}_{\mathrm{sgn}}$ is a universal constant. This implementation requires $O(d^2\log d)$ uses of U, $U^\dagger$, $\mathrm{C}_\Pi\mathrm{NOT}$, $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$, and $O(d^2\log d)$ multi-controlled single-qubit gates[30]. Moreover, we can compute the description of V in deterministic time $\tilde{O}(\epsilon_2^{-1}d^{9/2})$ and space $O(s(n))$.*
Furthermore, our construction straightforwardly extends to any non-Hermitian (but linear) matrix A by simply replacing $P_d^{\mathrm{sgn}}(A)$ with $P_{\mathrm{sgn},d}^{(\mathrm{SV})}(A)$ defined in the same way as Definition 3.3.

Proof. In Corollary 3.7, we can express $P_d^{\mathrm{sgn}}(x)$ as $\frac{c_0}{2} + \sum_{k=1}^{d} c_k T_k(x)$, where $d = O(\delta^{-1}\log\epsilon^{-1})$. For all $x \in [-1,1]\setminus[-\delta,\delta]$, we have $|\mathrm{sgn}(x) - P_d^{\mathrm{sgn}}(x)| \leq O(\epsilon\log d) := \epsilon_2$. To implement Eval with precision ε, we can compute the corresponding entry c_i of the coefficient vector, which requires $O(\log(\varepsilon^{-1}d^4)) = O(\log(\epsilon_2^{-2}d^5))$ space and $\tilde{O}(\varepsilon^{-1/2}d^2) = \tilde{O}(\epsilon_2^{-1}d^{5/2})$ time. Using Theorem 3.11, we can conclude that P_d^{sgn} has a $\big(1, a', 144dC_{\mathrm{sgn}}^2\epsilon_1^{1/2}\log^2(d) + 36\hat{C}_{\mathrm{sgn}}\epsilon_2\log d\big)$-bitstring indexed encoding V that acts on $s + a'$ qubits, where $a' := a + \lceil\log d\rceil + 3$ and $\|\mathbf{c}\|_1 \leq \hat{C}_{\mathrm{sgn}}\log d$.

Furthermore, the quantum circuit of V makes $O(d^2\log d)$ uses of U, $U^\dagger$, $\mathrm{C}_\Pi\mathrm{NOT}$, and $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$ as well as $O(d^2\log d)$ multi-controlled single-qubit gates. We note that $d \leq 2^{O(s(n))}$ and $\epsilon_2 \geq 2^{-O(s(n))}$. Moreover, we can compute the description of V in $O(s(n))$ space since each oracle call to Eval with precision ε can be computed in $O(\log(\epsilon_2^{-2}d^5))$ space. Additionally, the time complexity for computing the description of V is

$$\max\{\tilde{O}(d^2\log d\log(d/\epsilon_2)), d^2\log d\cdot\tilde{O}(\epsilon_2^{-1}d^{5/2})\} = \tilde{O}(\epsilon_2^{-1}d^{9/2}).$$

□

Corollary 3.17 (Log polynomial with space-efficient coefficients applied to bitstring indexed encodings)**.** *Let A be an Hermitian matrix that acts on s qubits, where $s(n) \geq \Omega(\log(n))$. Let U be a $(1, a, \epsilon_1)$-bitstring indexed encoding of A that acts on $s+a$ qubits. Then, for any $d \leq 2^{O(s(n))}$,*

$\epsilon_2 \geq 2^{-O(s(n))}$, and $\beta \geq 2^{-O(s(n))}$, we have a $(1, a+\lceil \log d \rceil+3, 144\hat{C}_{\ln}\epsilon_1^{1/2}d^3+36\hat{C}_{\ln}\epsilon_2 d)$-bitstring indexed encoding V of $P_d^{\ln}(A)$, where $P_d^{\ln}$ is a space-efficient bounded polynomial approximation of the normalized log function specified in Corollary 3.10, and $\hat{C}_{\ln}$ is a universal constant. This implementation requires $O(d^3)$ uses of U, $U^\dagger$, $\mathrm{C}_\Pi\mathrm{NOT}$, $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$, and $O(d^3)$ multi-controlled single-qubit gates[30]. Moreover, we can compute the description of the resulting quantum circuit in bounded-error randomized time $\tilde{O}(\max\{\beta^{-5}\epsilon_2^{-4}d^5, \epsilon_2^{-1}d^{11/2}\})$ and space $O(s(n))$.

Proof. In Corollary 3.10, we can express $P_d^{\ln}(x)$ as $\frac{c_0}{2}+\sum_{k=1}^{d} c_k T_k(x)$, where $d = O(\delta^{-1}\log\epsilon^{-1})$. For any $\ln_\beta(x)$, we have $|\ln_\beta(x) - P_d^{\ln}(x)| \leq O(\epsilon \log d) := \epsilon_2$ for all $x \in [\beta, 1]$. To implement Eval with precision ε, we can compute the corresponding entry c_i of the coefficient vector by a bounded-error randomized algorithm. This requires $O(\log(\varepsilon^{-1}d^4\beta^{-4})) = O(\log(\beta^{-4}\epsilon_2^{-2}d^5))$ space and $\tilde{O}(\max\{\beta^{-5}\varepsilon^{-2}, \varepsilon^{-1/2}d^2\}) = \tilde{O}(\max\{\beta^{-5}\epsilon_2^{-4}d^2, \epsilon_2^{-1}\}d^{5/2})$ time. Using Theorem 3.11, we conclude that $P_d^{\ln}$ has a $(1, a', 144C_{\ln}^2\epsilon_1^{1/2}d^3 + 36\hat{C}_{\ln}\epsilon_2 d)$-bitstring indexed encoding V that acts on $s+a'$ qubits, where $a' := a + \lceil \log d \rceil + 3$ and $\|\mathbf{c}\|_1 \leq \hat{C}_{\ln} d$.

Furthermore, the quantum circuit of V makes $O(d^3)$ uses of U, $U^\dagger$, $\mathrm{C}_\Pi\mathrm{NOT}$, and $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$ as well as $O(d^3)$ multi-controlled single-qubit gates. We note that $d \leq 2^{O(s(n))}$, $\epsilon_2 \geq 2^{-O(s(n))}$, and $\beta \geq 2^{-O(s(n))}$. Additionally, we can compute the description of V in $O(s(n))$ space since each oracle call to Eval with precision ε can be computed in $O(\log(\beta^{-4}\epsilon_2^{-2}d^5))$ space. The time complexity for computing the description of V is given by:

$$\max\{\tilde{O}(d^3\log(d/\epsilon_2)), d^3 \cdot \tilde{O}(\max\{\beta^{-5}\epsilon_2^{-4}d^2, \epsilon_2^{-1}d^{5/2}\}) = \tilde{O}(\max\{\beta^{-5}\epsilon_2^{-4}d^5, \epsilon_2^{-1}d^{11/2}\}). \quad (3.13)$$

Finally, to guarantee that the probability that all $O(d^3)$ oracle calls to Eval succeed is at least $2/3$, we use a $(4\ln d)$-time sequential repetition of Eval for each oracle call. Together with the Chernoff-Hoeffding bound and the union bound, the resulting randomized algorithm succeeds with probability at least $1 - d^3 \cdot 2\exp(-4\ln d) \geq 2/3$. We further note that the time complexity specified in Equation (3.13) only increase by a $4\ln d$ factor. □

3.4 Application: space-efficient error reduction for unitary quantum computations

We provide a unified space-efficient error reduction for unitary quantum computations. In particular, one-sided error scenarios (e.g., $\mathsf{RQ_UL}$ and $\mathsf{coRQ_UL}$) have been proven in [Wat01], and the two-sided error scenario (e.g., $\mathsf{BQ_UL}$) has been demonstrated in [FKL+16].

Theorem 3.18 (Space-efficient error reduction for unitary quantum computations)**.** *Let $s(n)$ be a space-constructible function, and let $a(n)$, $b(n)$, and $l(n)$ be deterministic $O(s(n))$ space computable functions such that $a(n) - b(n) \geq 2^{-O(s(n))}$, we know that for any $l(n) \leq O(s(n))$, there is $d := l(n)/\max\{\sqrt{a}-\sqrt{b}, \sqrt{1-b}-\sqrt{1-a}\}$ such that*

$$\mathsf{BQ_USPACE}[s(n), a(n), b(n)] \subseteq \mathsf{BQ_USPACE}\big[s(n) + \lceil \log d \rceil + 1, 1 - 2^{-l(n)}, 2^{-l(n)}\big].$$

Furthermore, for one-sided error scenarios, we have that for any $l(n) \leq 2^{O(s(n))}$:

$\mathsf{RQ_USPACE}[s(n), a(n)] \subseteq \mathsf{RQ_USPACE}\big[s(n) + \lceil \log d_0 \rceil + 1, 1 - 2^{-l(n)}\big]$ *where* $d_0 := \frac{l(n)}{\max\{\sqrt{a}, 1-\sqrt{1-a}\}}$,

$\mathsf{coRQ_USPACE}[s(n), b(n)] \subseteq \mathsf{coRQ_USPACE}\big[s(n) + \lceil \log d_1 \rceil + 1, 2^{-l(n)}\big]$ *where* $d_1 := \frac{l(n)}{\max\{1-\sqrt{b}, \sqrt{1-b}\}}$.

By choosing $s(n) = \Theta(\log(n))$, we derive error reduction for logarithmic-space quantum computation in a unified approach:

Corollary 3.19 (Error reduction for $\mathsf{BQ_UL}$, $\mathsf{RQ_UL}$, and $\mathsf{coRQ_UL}$)**.** *For deterministic logspace computable functions $a(n)$, $b(n)$, and $l(n)$ satisfying $a(n) - b(n) \geq 1/\operatorname{poly}(n)$ and $l(n) \leq$*

$O(\log n)$, *we have the following inclusions:*

$$\mathsf{BQ_UL}[a(n),b(n)] \subseteq \mathsf{BQ_UL}[1-2^{-l(n)}, 2^{-l(n)}],$$
$$\mathsf{RQ_UL}[a(n)] \subseteq \mathsf{RQ_UL}[1-2^{-l(n)}],$$
$$\mathsf{coRQ_UL}[b(n)] \subseteq \mathsf{coRQ_UL}[2^{-l(n)}].$$

The construction specified in Theorem 3.18 crucially relies on Lemma 3.20. And the proof of Lemma 3.20 directly follows from Theorem 20 in [GSLW19], which is deferred to Appendix A.3.

Lemma 3.20 (Space-efficient singular value discrimination)**.** *Let* $0 \leq \alpha < \beta \leq 1$ *and* $A := \tilde{\Pi} U \Pi$ *be a* $(1,0,0)$*-bitstring indexed encoding where* U *acts on* s *qubits and* $s(n) \geq \Omega(\log n)$*. Consider an unknown quantum state* $|\psi\rangle$*, with the promise that it is a right singular vector of* A *with a singular value either above* α *or below* β*. We can distinguish the two cases with error probability at most* $\varepsilon := O(\epsilon \log d)$ *using a degree-*d *quantum singular value transformation where* $d = \frac{\log 1/\epsilon}{\max\{\beta-\alpha, \sqrt{1-\alpha^2}-\sqrt{1-\beta^2}\}}$*. Moreover, we can make the error one-sided if* $\alpha = 0$ *or* $\beta = 1$*. In particular, the implementation requires* $O(d^2 \log d)$ *uses of* U*,* $U^\dagger$*,* $\mathrm{C}_\Pi\mathrm{NOT}$*,* $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$*, and* $O(d^2 \log d)$ *multi-controlled single-qubit gates. Also, we can compute the description of the implementation in deterministic time* $\tilde{O}(\varepsilon^{-1} d^{9/2})$ *and space* $O(s(n))$*.*

Finally, we provide the proof of Theorem 3.18, which closely relates to Theorem 38 in [GSLW18] (the full version of [GSLW19]).

Proof of Theorem 3.18. It suffices to amplify the promise gap by QSVT. Note that the probability that a $\mathsf{BQ_USPACE}[s(n)]$ circuit C_x accepts is $\Pr[C_x \text{ accepts}] = \||1\rangle\langle 1|_{\text{out}} C_x |0^{k+m}\rangle\|_2^2 \geq a$ for *yes* instances, whereas $\Pr[C_x \text{ accepts }] = \||1\rangle\langle 1|_{\text{out}} C_x |0^{k+m}\rangle\|_2^2 \leq b$ for *no* instances. Then consider a $(1,0,0)$-bitstring indexed encoding $M_x := \Pi_{\text{out}} C_x \Pi_{\text{in}}$ such that $\|M_x\| \geq \sqrt{a}$ for *yes* instances while $\|M_x\| \leq \sqrt{b}$ for *no* instances, where $\Pi_{\text{in}} := |0\rangle\langle 0|^{\otimes k+m}$ and $\Pi_{\text{out}} := |1\rangle\langle 1|_{\text{out}} \otimes I_{m+k-1}$. Since $\|M_x\| = \sigma_{\max}(M_x)$ where $\sigma_{\max}(M_x)$ is the largest singular value of M_x, it suffices to distinguish the largest singular value of M_x are either above $\sqrt{a}$ or below $\sqrt{b}$. By setting $\alpha := \sqrt{a}$, $\beta := \sqrt{b}$ and $\varepsilon := 2^{-l(n)}$, this task is a direct corollary of Lemma 3.20. ☐

4 Space-bounded quantum state testing

We begin by defining the problem of quantum state testing in a space-bounded manner:

Definition 4.1 (Space-bounded Quantum State Testing)**.** Given polynomial-size quantum circuits (devices) Q_0 and Q_1 that act on $O(\log n)$ qubits and have a succinct description (the "source code" of devices), with $r(n)$ specified output qubits, where $r(n)$ is a deterministic logspace computable function such that $0 < r(n) \leq O(\log(n))$. Let ρ_i denote the mixed state obtained by running Q_i on the all-zero state $|\bar{0}\rangle$ and tracing out the non-output qubits.
We define a *space-bounded quantum state testing* problem, with respect to a specified distance-like measure, to decide whether ρ_0 and ρ_1 are easily distinguished or almost indistinguishable. Likewise, we also define a *space-bounded quantum state certification* problem to decide whether ρ_0 and ρ_1 are easily distinguished or *exactly* indistinguishable.

We remark that space-bounded quantum state certification, defined in Definition 4.1, represents a "white-box" (log)space-bounded counterpart of quantum state certification [BOW19].

Remark 4.2 (Lifting to exponential-size instances by succinct encodings)*.* For $s(n)$ space-uniform quantum circuits Q_0 and Q_1 acting on $O(s(n))$ qubits, if these circuits admit a succinct encoding,[32] namely there is a deterministic $O(s(n))$-space Turing machine with time complexity

[32]For instance, the construction in [FL18, Remark 11], or [PY86, BLT92] in general.

$\mathrm{poly}(s(n))$ can uniformly generate the corresponding gate sequences, then Definition 4.1 can be extended to any $s(n)$ satisfying $\Omega(\log n) \leq s(n) \leq \mathrm{poly}(n)$.[33]

Next, we define space-bounded quantum state testing problems, based on Definition 4.1, with respect to four commonplace distance-like measures.

Definition 4.3 (Space-bounded Quantum State Distinguishability Problem, $\textsc{GapQSD}_{\log}$)**.** Consider deterministic logspace computable functions $\alpha(n)$ and $\beta(n)$, satisfying $0 \leq \beta(n) < \alpha(n) \leq 1$ and $\alpha(n) - \beta(n) \geq 1/\mathrm{poly}(n)$. Then the promise is that one of the following holds:

- *Yes* instances: A pair of quantum circuits (Q_0, Q_1) such that $\mathrm{td}(\rho_0, \rho_1) \geq \alpha(n)$;
- *No* instances: A pair of quantum circuits (Q_0, Q_1) such that $\mathrm{td}(\rho_0, \rho_1) \leq \beta(n)$.

Moreover, we also define the certification counterpart of $\textsc{GapQSD}_{\log}$, referred to as $\textsc{CertQSD}_{\log}$, given that $\beta = 0$. Specifically, $\textsc{CertQSD}_{\log}[\alpha(n)] := \textsc{GapQSD}_{\log}[\alpha(n), 0]$.

Likewise, we can define $\textsc{GapQJS}_{\log}$ and $\textsc{GapQHS}_{\log}$, also the certification version $\overline{\textsc{CertQHS}}_{\log}$, in a similar manner to Definition 4.3 by replacing the distance-like measure accordingly:

- $\textsc{GapQJS}_{\log}[\alpha(n), \beta(n)]$: Decide whether $\mathrm{QJS}_2(\rho_0, \rho_1) \geq \alpha(n)$ or $\mathrm{QJS}_2(\rho_0, \rho_1) \leq \beta(n)$;
- $\textsc{GapQHS}_{\log}[\alpha(n), \beta(n)]$: Decide whether $\mathrm{HS}^2(\rho_0, \rho_1) \geq \alpha(n)$ or $\mathrm{HS}^2(\rho_0, \rho_1) \leq \beta(n)$.

Furthermore, we use the notation $\overline{\textsc{CertQSD}}_{\log}$ to indicate the *complement* of $\textsc{CertQSD}_{\log}$ with respect to the chosen parameter $\alpha(n)$, and so does $\overline{\textsc{CertQHS}}_{\log}$.

Definition 4.4 (Space-bounded Quantum Entropy Difference Problem, $\textsc{GapQED}_{\log}$)**.** Consider a deterministic logspace computable function $g : \mathbb{N} \to \mathbb{R}^+$, satisfying $g(n) \geq 1/\mathrm{poly}(n)$. Then the promise is that one of the following cases holds:

- *Yes* instance: A pair of quantum circuits (Q_0, Q_1) such that $\mathrm{S}(\rho_0) - \mathrm{S}(\rho_1) \geq g(n)$;
- *No* instance: A pair of quantum circuits (Q_0, Q_1) such that $\mathrm{S}(\rho_1) - \mathrm{S}(\rho_0) \geq g(n)$.

Novel complete characterizations for space-bounded quantum computation. We now present the main theorems in this section and the paper. Theorem 4.5 establishes the first family of natural $\mathsf{coRQ_UL}$-complete problems. By relaxing the error requirement from one-sided to two-sided, Theorem 4.6 identifies a new family of natural BQL-complete problems on space-bounded quantum state testing.

Theorem 4.5. *The computational hardness of the following (log)space-bounded quantum state certification problems, for any deterministic logspace computable $\alpha(n) \geq 1/\mathrm{poly}(n)$, is as follows:*

(1) $\overline{\textsc{CertQSD}}_{\log}[\alpha(n)]$ *is* $\mathsf{coRQ_UL}$*-complete;*

(2) $\overline{\textsc{CertQHS}}_{\log}[\alpha(n)]$ *is* $\mathsf{coRQ_UL}$*-complete.*

Theorem 4.6. *The computational hardness of the following (log)space-bounded quantum state testing problems, where $\alpha(n) - \beta(n) \geq 1/\mathrm{poly}(n)$ or $g(n) \geq 1/\mathrm{poly}(n)$ as well as $\alpha(n)$, $\beta(n)$, $g(n)$ can be computed in deterministic logspace, is as follows:*

(1) $\textsc{GapQSD}_{\log}[\alpha(n), \beta(n)]$ *is* BQL*-complete;*

(2) $\textsc{GapQED}_{\log}[g(n)]$ *is* BQL*-complete;*

(3) $\textsc{GapQJS}_{\log}[\alpha(n), \beta(n)]$ *is* BQL*-complete;*

[33] It is noteworthy that Definition 4.1 (mostly) coincides with the case of $s(n) = \Theta(O(\log n))$ and directly takes the corresponding gate sequence of Q_0 and Q_1 as an input.

(4) $\textsc{GapQHS}_{\log}[\alpha(n), \beta(n)]$ *is* BQL*-complete.*

It is noteworthy that we can naturally extend Theorem 4.5 and Theorem 4.6 to their exponential-size up-scaling counterparts with $2^{-O(s(n))}$-precision, employing the extended version of Definition 4.1 outlined in Remark 4.2, thus achieving the complete characterizations for $\mathsf{coRQ_U SPACE}[s(n)]$ and $\mathsf{BQPSPACE}[s(n)]$, respectively.

In the remainder of this section, we first address problems with two-sided errors. Specifically, by employing a general framework for space-bounded quantum state testing demonstrated in Section 4.1, we demonstrate the BQL containment of $\textsc{GapQSD}_{\log}$ in Section 4.2, as well as the BQL containment of $\textsc{GapQED}_{\log}$ and $\textsc{GapQJS}_{\log}$ in Section 4.3. Subsequently, in Section 4.4, we focus on making the error one-sided and establish the $\mathsf{coRQ_U L}$ containment of $\overline{\textsc{CertQSD}}_{\log}$ and $\overline{\textsc{CertQHS}}_{\log}$. Additionally, we show the BQL containment of $\textsc{GapQHS}_{\log}$ in Appendix B. The corresponding hardness proof for all these problems is provided in Section 4.5.

4.1 Space-bounded quantum state testing: a general framework

In this subsection, we introduce a general framework for quantum state testing that utilizes a quantum tester $\mathcal{T}$. Specifically, the space-efficient tester $\mathcal{T}$ succeeds (outputting the value "0") with probability x, which is linearly dependent on some quantity closely related to the distance-like measure of interest. Consequently, we can obtain an additive-error estimation $\widetilde{x}$ of x with high probability through sequential repetition (Lemma 2.11).

To construct $\mathcal{T}$, we combine the one-bit precision phase estimation [Kit95], commonly known as the Hadamard test [AJL09], for block-encodings (Lemma 4.9), with our space-efficient quantum singular value transformation (QSVT) technique, which we describe in Section 3.

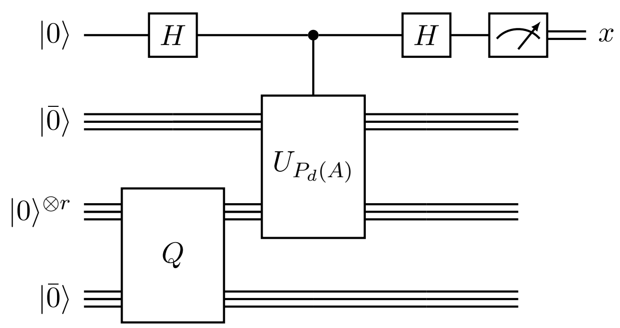

Figure 2: Quantum tester $\mathcal{T}(Q, U_A, P_d, \epsilon)$: the circuit implementation.

Constructing a space-efficient quantum tester. We now provide a formal definition and the detailed construction of the quantum tester $\mathcal{T}$. The quantum circuit shown in Figure 2 defines the quantum tester $\mathcal{T}(Q, U_A, P_d, \epsilon)$ using the following parameters with $s(n) = \Theta(\log n)$:

- A $s(n)$-qubit quantum circuit Q prepares the purification of an $r(n)$-qubit quantum state ρ where ρ is the quantum state of interest;
- U_A is a $(1, s-r, 0)$-block-encoding of an $r(n)$-qubit Hermitian operator A where A relates to the quantum states of interest and $r(n) \leq s(n)$;
- P_d is a degree-d bounded polynomial with a particular form $P_d = \frac{c_0}{2} + \sum_{k=1}^{d} c_k T_k \in \mathbb{R}[x]$ where T_k is the k-th Chebyshev polynomial, with $d \leq 2^{O(s(n))}$, such that the coefficients $\mathbf{c} := (c_0, \cdots, c_d)$ can be computed in bounded-error randomized space $O(s(n))$;
- ϵ is the precision parameter used in the estimation of x, with $\epsilon \geq 2^{-O(s(n))}$.

Moreover, we define the corresponding estimation procedure, denoted as $\hat{\mathcal{T}}(Q, U_A, P_d, \epsilon, \epsilon_H, \delta)$, namely a quantum algorithm that computes an additive-error estimation $\widetilde{x}$ of the output x from the tester $\mathcal{T}(Q, U_A, P_d, \epsilon)$. Technically speaking, $\hat{\mathcal{T}}$ outputs $\widetilde{x}$ such that $|x-\widetilde{x}| \leq \epsilon\|\mathbf{c}\|_1+\epsilon_H$ with probability at least $1-\delta$. Now we will demonstrate that both the tester $\mathcal{T}$ and the corresponding estimation procedure $\hat{\mathcal{T}}$ are space-efficient:

Lemma 4.7 (Quantum tester $\mathcal{T}$ and estimation procedure $\hat{\mathcal{T}}$ are space-efficient)**.** *The quantum tester $\mathcal{T}(Q, U_A, P_d, \epsilon)$, as specified in Figure 2, accepts (outputting the value "0") with probability $\frac{1}{2}(1+\mathrm{Re}(\mathrm{Tr}(P_d(A)\rho))) \pm \frac{1}{2}\epsilon\|\mathbf{c}\|_1$. Moreover, we can compute the quantum circuit description of $\mathcal{T}$ in deterministic space $O(s+\log(1/\epsilon))$ given the coefficient vector $\mathbf{c}$ of P_d. Furthermore, we can implement the corresponding estimation procedure $\hat{\mathcal{T}}(Q, U_A, P_d, \epsilon, \epsilon_H, \delta)$ in bounded-error quantum space $O(s+\log(1/\epsilon)+\log(1/\epsilon_H)+\log\log(1/\delta))$.*

We first provide two useful lemmas for implementing our quantum tester $\mathcal{T}$. It is noteworthy that Lemma 4.8 originates from [LC19], as well as Lemma 4.9 is a specific version of one-bit precision phase estimation (or the Hadamard test) [Kit95, AJL09].

Lemma 4.8 (Purified density matrix, [GSLW19, Lemma 25])**.** *Suppose ρ is an s-qubit density operator and U is an $(a+s)$-qubit unitary operator such that $U|0\rangle^{\otimes a}|0\rangle^{\otimes s} = |\rho\rangle$ and $\rho = \mathrm{Tr}_a(|\rho\rangle\langle\rho|)$. Then, we can construct an $O(a+s)$-qubit quantum circuit $\widetilde{U}$ that is an $(O(a+s), 0)$-block-encoding of ρ, using $O(1)$ queries to U and $O(a+s)$ one- and two-qubit quantum gates.*

Lemma 4.9 (Hadamard test for block-encodings, adapted from [GP22, Lemma 9])**.** *Suppose U is an $(a+s)$-qubit unitary operator that is a block-encoding of $s(n)$-qubit operator A. We can implement an $O(a+s)$-qubit quantum circuit that, on input $s(n)$-qubit quantum state ρ, outputs 0 with probability $\frac{1+\mathrm{Re}(\mathrm{Tr}(A\rho))}{2}$.*

Finally, we proceed with the actual proof of Lemma 4.7.

Proof of Lemma 4.7. By applying Chebyshev interpolation on U_A (Theorem 3.11 with the choice of $\epsilon_1 = 0$ and $\epsilon_2 = \epsilon/36$), we can implement an $O(s(n))$-qubit quantum circuit $U_{P_d(A)}$ that is a $(1, a, 0)$-block-encoding of A'_{P_d} using $O(d^2\|\mathbf{c}\|_1)$ queries to U_A, where $a = s - r + \lceil \log d \rceil + 3$ and A'_{P_d} is specified in Theorem 3.11 satisfying $\|P_d(A) - A'_{P_d}\| \leq \epsilon\|\mathbf{c}\|_1$. Additionally, we can compute the quantum circuit description of $U_{P_d(A)}$ in deterministic space $O(s+\log(1/\epsilon))$ given the coefficient vector $\mathbf{c}$ of P_d. As the quantum tester $\mathcal{T}(Q, U_A, P_d, \epsilon)$ is mainly based on the Hadamard test, by employing Lemma 4.9, we have that $\mathcal{T}$ outputs 0 with probability

$$\Pr[x=0] = \frac{1}{2}\big(1+\mathrm{Re}(\mathrm{Tr}(A'_{P_d}\rho))\big) = \frac{1}{2}(1+\mathrm{Re}(\mathrm{Tr}(P_d(A)\rho))) \pm \frac{1}{2}\epsilon\|\mathbf{c}\|_1.$$

It is left to construct the estimation procedure $\hat{\mathcal{T}}$. As detailed in in Lemma 2.11, we can obtain an estimation $\widetilde{x}$ by sequentially repeating the quantum tester $\mathcal{T}(Q, U_A, P_d, \epsilon)$ for $O(1/\epsilon_H^2)$ times. This repetition ensures that $|\widetilde{x} - \mathrm{Re}(\mathrm{Tr}(A'_{P_d}\rho))| \leq \epsilon_H$ holds with probability at least $\Omega(1)$, and derives an further implication on $P_d(A)$:

$$\Pr[|\widetilde{x} - \mathrm{Re}(\mathrm{Tr}(P_d(A)\rho))| \leq \epsilon\|\mathbf{c}\|_1 + \epsilon_H] \geq \Omega(1).$$

We thus conclude that construction of the estimation procedure $\hat{\mathcal{T}}(Q, U_A, P_d, \epsilon, \epsilon_H, \delta)$ by utilizing $O(\log(1/\delta)/\epsilon_H^2)$ sequential repetitions of $\mathcal{T}(Q, U_A, P_d, \epsilon)$. Similarly following Lemma 2.11, $\hat{\mathcal{T}}(Q, U_A, P_d, \epsilon, \epsilon_H, \delta)$ outputs an estimation $\widetilde{x}$ satisfies the following condition:

$$\Pr[|\widetilde{x} - \mathrm{Re}(\mathrm{Tr}(P_d(A)\rho))| \leq \epsilon\|\mathbf{c}\|_1 + \epsilon_H] \geq 1-\delta.$$

In addition, a direct calculation indicates that we can implement $\hat{\mathcal{T}}(Q, U_A, P_d, \epsilon, \epsilon_H, \delta)$ in quantum space $O(s+\log(1/\epsilon)+\log(1/\epsilon_H)+\log\log(1/\delta))$ as desired. □

4.2 GapQSD$_{\log}$ is in BQL

In this subsection, we demonstrate Theorem 4.10 by constructing a quantum algorithm that incorporates testers $\mathcal{T}(Q_i, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}}, \epsilon)$ for $i \in \{0,1\}$, where the construction of testers utilizes the space-efficient QSVT associated with the sign function.

Theorem 4.10. *For any functions $\alpha(n)$ and $\beta(n)$ that can be computed in deterministic logspace and satisfy $\alpha(n)-\beta(n) \geq 1/\operatorname{poly}(n)$, we have that* $\textsc{GapQSD}_{\log}[\alpha(n),\beta(n)]$ *is in* BQL.

Proof. Inspired by time-efficient algorithms for the low-rank variant of $\textsc{GapQSD}$ [WZ23a], we devise a space-efficient algorithm for $\textsc{GapQSD}_{\log}$, present formally in Algorithm 1.

Algorithm 1: Space-efficient algorithm for $\textsc{GapQSD}_{\log}$.

Input : Quantum circuits Q_i that prepares the purification of ρ_i for $i \in \{0,1\}$.

Output: An additive-error estimation of $\mathrm{td}(\rho_0,\rho_1)$.

Params: $\varepsilon := \frac{\alpha-\beta}{4}$, $\delta := \frac{\varepsilon}{2^{r+3}}$, $\epsilon := \frac{\varepsilon}{2(\hat{C}_{\mathrm{sgn}}+2C_{\mathrm{sgn}})\tilde{C}_{\mathrm{sgn}}2^{r+3}} \cdot \frac{1}{2\log\left(2(\hat{C}_{\mathrm{sgn}}+2C_{\mathrm{sgn}})\tilde{C}_{\mathrm{sgn}}2^{r+3}/\varepsilon\right)}$, $d := \tilde{C}_{\mathrm{sgn}}\delta^{-1}\log\epsilon^{-1}$, $\varepsilon_H := \frac{\varepsilon}{4}$.

1. Construct block-encodings of ρ_0 and ρ_1, denoted by U_{ρ_0} and U_{ρ_1}, respectively, using $O(1)$ queries to Q_0 and Q_1 and $O(s(n))$ ancillary qubits by Lemma 4.8;
2. Construct a block-encoding of $\frac{\rho_0-\rho_1}{2}$, denoted by $U_{\frac{\rho_0-\rho_1}{2}}$, using $O(1)$ queries to U_{ρ_0} and U_{ρ_1} and $O(s(n))$ ancillary qubits by Lemma 3.14;

Let P_d^{sgn} be the degree-d polynomial specified in Corollary 3.7 with parameters δ and ϵ such that its Chebyshev coefficients are computable in deterministic space $O(\log(d/\epsilon))$;

3. Set $x_0 := \hat{\mathcal{T}}(Q_0, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}}, \epsilon, \epsilon_H, 1/10)$, $x_1 := \hat{\mathcal{T}}(Q_1, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}}, \epsilon, \epsilon_H, 1/10)$;
4. Compute $x = (x_0 - x_1)/2$. Return "yes" if $x > (\alpha+\beta)/2$, and "no" otherwise.

Let us demonstrate the correctness of Algorithm 1 and analyze the computational complexity. We focus on the setting with $s(n) = \Theta(\log n)$. We set $\varepsilon := (\alpha-\beta)/4 \geq 2^{-O(s)}$ and assume that Q_0 and Q_1 are $s(n)$-qubit quantum circuits that prepare the purifications of ρ_0 and ρ_1, respectively. According to Lemma 4.8, we can construct $O(s)$-qubit quantum circuits U_{ρ_0} and U_{ρ_1} that encode ρ_0 and ρ_1 as $(1, O(s), 0)$-block-encodings, using $O(1)$ queries to Q_0 and Q_1 as well as $O(1)$ one- and two-qubit quantum gates. Next, we apply Lemma 3.14 to construct a $(1, O(s), 0)$-block-encoding $U_{\frac{\rho_0-\rho_1}{2}}$ of $\frac{\rho_0-\rho_1}{2}$, using $O(1)$ queries to Q_{ρ_0} and Q_{ρ_1}, as well as $O(1)$ one- and two-qubit quantum gates.

Let $\delta := \frac{\varepsilon}{2^{r+3}}$, $\epsilon := \frac{\varepsilon}{2(\hat{C}_{\mathrm{sgn}}+2C_{\mathrm{sgn}})\tilde{C}_{\mathrm{sgn}}2^{r+3}} \cdot \frac{1}{2\log\left(2(\hat{C}_{\mathrm{sgn}}+2C_{\mathrm{sgn}})\tilde{C}_{\mathrm{sgn}}2^{r+3}/\varepsilon\right)}$ and $d := \tilde{C}_{\mathrm{sgn}}\delta^{-1}\log\epsilon^{-1} = 2^{O(s)}$ where $\tilde{C}_{\mathrm{sgn}}$ comes from Corollary 3.7. Let $P_d^{\mathrm{sgn}} \in \mathbb{R}[x]$ be the polynomial specified in Corollary 3.7. Let $\epsilon_H = \varepsilon/4$. By employing Corollary 3.16 and the corresponding estimation procedure $\hat{\mathcal{T}}(Q_i, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}}, \epsilon, \epsilon_H, 1/10)$ from Lemma 4.7, we obtain the values x_i for $i \in \{0,1\}$, ensuring the following inequalities:

$$\Pr\left[\left|x_i - \operatorname{Tr}\left(P_d^{\mathrm{sgn}}\left(\frac{\rho_0-\rho_1}{2}\right)\rho_i\right)\right| \leq \hat{C}_{\mathrm{sgn}}\epsilon\log d + \epsilon_H\right] \geq \frac{9}{10} \text{ for } i \in \{0,1\}. \tag{4.1}$$

Here, the implementation uses $O(d^2\log d)$ queries to $U_{\frac{\rho_0-\rho_1}{2}}$ and $O(d^2\log d)$ multi-controlled single-qubit gates. Moreover, the circuit descriptions of $\hat{\mathcal{T}}(Q_i, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}}, \epsilon, \epsilon_H, 1/10)$ can be computed in deterministic time $\tilde{O}(d^{9/2}/\epsilon)$ and space $O(s)$.

Now let $x := (x_0 - x_1)/2$. We will finish the correctness analysis of Algorithm 1 by showing $\Pr[|x - \mathrm{td}(\rho_0,\rho_1)| \leq \varepsilon] > 0.8$ through Equation (4.1). By considering the approximation error of P_d^{sgn} in Corollary 3.7 and the QSVT implementation error in Corollary 3.16, we derive the following inequality in Proposition 4.10.1, and the proof is deferred to Appendix B.1:

Proposition 4.10.1. $\Pr\left[|x - \mathrm{td}(\rho_0,\rho_1)| \leq \hat{C}_{\mathrm{sgn}}\epsilon\log d + \epsilon_H + 2C_{\mathrm{sgn}}\epsilon\log d + 2^{r+1}\delta\right] > 0.8.$

Consequently, it is left to show that $\hat{C}_{\rm sgn}\epsilon\log d + \epsilon_H + 2C_{\rm sgn}\epsilon\log d + 2^{r+1}\delta \leq \varepsilon$ for the aforementioned choice of δ, ϵ, and ϵ_H. Note that $\epsilon_H = \varepsilon/4$ and $2^{r+1}\delta = \varepsilon/4$, we complete the correctness analysis by choosing $\epsilon := \delta'/2\log(\delta'^{-1})$ with $\delta' := \delta/\big(2(\hat{C}_{\rm sgn}+2C_{\rm sgn})\tilde{C}_{\rm sgn}\big) \leq 1/2$ and subsequently deriving the following inequality:

$$(\hat{C}_{\rm sgn}+2C_{\rm sgn})\epsilon\log d \leq (\hat{C}_{\rm sgn}+2C_{\rm sgn})\epsilon\log(\delta'^{-1}\log(\epsilon^{-1})) \leq (\hat{C}_{\rm sgn}+2C_{\rm sgn})\delta' \leq \varepsilon/2.$$

Here, the second inequality results from the fact that $\gamma\log(\varepsilon^{-1}\log\gamma^{-1}) \leq \varepsilon$ for $0 < \varepsilon \leq 1/2$, with $\gamma := \varepsilon/2\log(\varepsilon^{-1})$, and the last inequality owes to the chosen δ', along with the facts that $\delta := \epsilon/2^{r+3} \leq \varepsilon$ and $\tilde{C}_{\rm sgn} \geq 1$.

Finally, we analyze the computational resources required for Algorithm 1. According to Lemma 4.7, we can compute x in BQL, with the resulting algorithm requiring $O(d^2\log d/\epsilon_H^2) = \tilde{O}(2^{2r}/\varepsilon^4)$ queries to Q_0 and Q_1. In addition, its circuit description can be computed in deterministic time $\tilde{O}(d^{9/2}/\varepsilon) = \tilde{O}(2^{4.5r}/\varepsilon^{5.5})$. □

4.3 GapQED$_{\log}$ and GapQJS$_{\log}$ are in BQL

In this subsection, we will demonstrate Theorem 4.11 by devising a quantum algorithm that encompasses testers $\mathcal{T}(Q_i, U_{\rho_i}, P_d^{\ln}, \epsilon)$ for $i \in \{0,1\}$, where the construction of testers employs the space-efficient QSVT associated with the normalized logarithmic function. Consequently, we can deduce that GapQJS$_{\log}$ is in BQL via a reduction from GapQJS$_{\log}$ to GapQED$_{\log}$.

Theorem 4.11. *For any deterministic logspace computable function $g(n)$ that satisfies $g(n) \geq 1/\mathrm{poly}(n)$, we have that* GapQED$_{\log}[g(n)]$ *is in* BQL.

Proof. We begin with a formal algorithm in Algorithm 2.

Algorithm 2: Space-efficient algorithm for GapQED$_{\log}$.

Input : Quantum circuits Q_i that prepares the purification of ρ_i for $i \in \{0,1\}$.

Output: An additive-error estimation of $\mathrm{S}(\rho_0) - \mathrm{S}(\rho_1)$.

Params: $\varepsilon := \frac{g}{4}$, $\beta := \min\{\frac{\varepsilon}{2^{r+5}\ln(2^{r+4}/\varepsilon)}, \frac{1}{4}\}$, $d := \tilde{C}_{\ln}\cdot\frac{1}{\beta}\log\frac{1}{\epsilon}$,
$\epsilon := \frac{\beta\varepsilon}{4\tilde{C}_{\ln}(\hat{C}_{\ln}+C_{\ln})\ln(1/\beta)} \cdot \frac{1}{4\log\left(4\tilde{C}_{\ln}(\hat{C}_{\ln}+C_{\ln})\ln(1/\beta)/(\beta\varepsilon)\right)}$, $\epsilon_H := \frac{\varepsilon}{8\ln(1/\beta)}$.

1. Construct block-encodings of ρ_0 and ρ_1, denoted by U_{ρ_0} and U_{ρ_1}, respectively, using $O(1)$ queries to Q_0 and Q_1 and $O(s(n))$ ancillary qubits by Lemma 4.8;

Let $P_d^{\ln}$ be the degree-d polynomial specified in Corollary 3.10 with parameters δ and ϵ such that its Chebyshev coefficients are computable in bounded-error randomized space $O(\log(d/\epsilon))$;

2. Set $x_0 := \hat{\mathcal{T}}(Q_0, U_{\rho_0}, P_d^{\ln}, \epsilon, \epsilon_H, 1/10)$, $x_1 := \hat{\mathcal{T}}(Q_1, U_{\rho_1}, P_d^{\ln}, \epsilon, \epsilon_H, 1/10)$;
3. Compute $x = (x_0 - x_1)\ln(2/\beta)$. Return "yes" if $x > 0$, and "no" otherwise.

Let us now demonstrate the correctness and computational complexity of Algorithm 2. We concentrate on the scenario with $s(n) = \Theta(\log n)$ and $\varepsilon = g/4 \geq 2^{-O(s)}$. Our strategy is to estimate the entropy of each of ρ_0 and ρ_1, respectively. We assume that Q_0 and Q_1 are s-qubit quantum circuits that prepare the purifications of ρ_0 and ρ_1, respectively. By Lemma 4.8, we can construct $(1, O(s), 0)$-block-encodings U_{ρ_0} and U_{ρ_1} of ρ_0 and ρ_1, respectively, using $O(1)$ queries to Q_0 and Q_1 as well as $O(1)$ one- and two-qubit quantum gates.

Let $\beta = \min\{\frac{\varepsilon}{2^{r+5}\ln(2^{r+4}/\varepsilon)}, \frac{1}{4}\}$, $\epsilon := \frac{\beta\varepsilon}{4\tilde{C}_{\ln}(\hat{C}_{\ln}+C_{\ln})\ln(1/\beta)} \cdot \frac{1}{4\log\left(4\tilde{C}_{\ln}(\hat{C}_{\ln}+C_{\ln})\ln(1/\beta)/(\beta\varepsilon)\right)}$ and $d = \tilde{C}_{\ln}\beta^{-1}\log\epsilon^{-1} = 2^{O(s(n))}$ where $\tilde{C}_{\ln}$ comes from Corollary 3.10. Let $P_d^{\ln} \in \mathbb{R}[x]$ be the polynomial specified in Corollary 3.10. Let $\epsilon_H = \frac{\varepsilon}{8\ln(1/\beta)}$. By utilizing Corollary 3.17 and the corresponding estimation procedure $\hat{\mathcal{T}}(Q_i, U_{\rho_i}, P_d^{\ln}, \epsilon, \epsilon_H, 1/10)$ from Lemma 4.7, we obtain the values x_i for $i \in \{0,1\}$, ensuring the following inequalities:

$$\Pr\left[\left|x_i - \operatorname{Tr}\left(P_d^{\ln}(\rho_i)\,\rho_i\right)\right| \leq \hat{C}_{\ln}\epsilon d + \epsilon_H\right] \geq \frac{9}{10} \text{ for } i \in \{0,1\}. \tag{4.2}$$

Here, the implementation uses $O(d^3)$ queries to U_{ρ_0} and $O(d^3)$ multi-controlled single-qubit gates. Moreover, the circuit descriptions of $\hat{\mathcal{T}}(Q_i, U_{\rho_i}, P_d^{\ln}, \epsilon, \epsilon_H, 1/10)$ can be computed in bounded-error time $\tilde{O}(d^9/\epsilon^4)$ and space $O(s)$.

We will finish the correctness analysis of Algorithm 2 by demonstrating $\Pr[|x_i - \mathrm{S}(\rho_i)| \leq \varepsilon] \geq 0.9$ through Equation (4.2). By considering the approximation error of $P_d^{\ln}$ in Corollary 3.10 and the QSVT implementation error in Corollary 3.17, we derive the following inequality in Proposition 4.11.1, and the proof is deferred to Appendix B.1:

Proposition 4.11.1. *The following inequality holds for $i \in \{0,1\}$:*

$$\Pr\left[\left|x_i \ln\left(\tfrac{2}{\beta}\right) - \mathrm{S}(\rho_i)\right| \leq 2\ln\left(\tfrac{1}{\beta}\right)\left(\hat{C}_{\ln}\epsilon d + \epsilon_H + C_{\ln}\epsilon \log d + 2^{r+1}\beta\right)\right] \geq \tfrac{9}{10}.$$

Consequently, it is left to show that $2\ln\left(\frac{1}{\beta}\right)\left(\hat{C}_{\ln}\epsilon d + \epsilon_H + C_{\ln}\epsilon \log d + 2^{r+1}\beta\right) \leq \varepsilon$ for the aforementioned choice of β, ϵ, and ϵ_H. Note that $2\ln(1/\beta)\epsilon_H = \varepsilon/4$ and $2\ln(1/\beta) \cdot 2^{r+1}\beta \leq \varepsilon/4$, we complete the correctness analysis by choosing $\epsilon := \delta/4 \cdot \log(1/\delta)$ with $\delta := \frac{\beta\varepsilon}{4\tilde{C}_{\ln}(\hat{C}_{\ln}+C_{\ln})\ln(1/\beta)} \leq 1/2$, and subsequently deriving the following inequality:

$$\begin{aligned}
2\ln(\beta^{-1})(\hat{C}_{\ln}\epsilon d + C_{\ln}\epsilon\log(d)) &\leq 2\ln(\beta^{-1})(\hat{C}_{\ln} + C_{\ln})\epsilon d \\
&= 2\ln(\beta^{-1})(\hat{C}_{\ln} + C_{\ln})\tilde{C}_{\ln}\beta^{-1}\epsilon\log(\epsilon^{-1}) \\
&\leq 2\ln(\beta^{-1})(\hat{C}_{\ln} + C_{\ln})\tilde{C}_{\ln}\beta^{-1}\delta \\
&= \varepsilon/2.
\end{aligned}$$

Here, the first line is because of $\log(d) \leq d$, the third line owes to the fact that $\epsilon\log(\epsilon^{-1}) \leq \delta$, and the last line is due to choice of δ.

Finally, we analyze the computational resources required for Algorithm 2. As per Lemma 4.7, we can compute x in BQL, with the resulting algorithm requiring $O(d^3/\epsilon_H^2) = \tilde{O}(2^{3r}/\varepsilon^4)$ queries to Q_0 and Q_1. Furthermore, its circuit description can be computed in bounded-error randomized time $\tilde{O}(d^{11}/\varepsilon^4) = \tilde{O}(2^{11r}/\varepsilon^{15})$. □

$\textsc{GapQJS}_{\log}$ is in BQL. It is noteworthy that we can achieve $\textsc{GapQJS}_{\log} \in \mathsf{BQL}$ by employing the estimation procedure $\hat{\mathcal{T}}$ in Algorithm 2 for *three according states*, given that the quantum Jensen-Shannon divergence $\mathrm{QJS}(\rho_0, \rho_1)$ is a linear combination of $\mathrm{S}(\rho_0), \mathrm{S}(\rho_1)$, and $\mathrm{S}\left(\frac{\rho_0+\rho_1}{2}\right)$. Nevertheless, the log-space Karp reduction from $\textsc{GapQJS}_{\log}$ to $\textsc{GapQED}_{\log}$ (Corollary 4.12) allows us to utilize $\hat{\mathcal{T}}$ for only *two* states. Furthermore, our construction is adapted from the time-bounded scenario [Liu23, Lemma 4.3].

Corollary 4.12. *For any functions $\alpha(n)$ and $\beta(n)$ that can be computed in deterministic logspace and satisfy $\alpha(n) - \beta(n) \geq 1/\operatorname{poly}(n)$, we have that $\textsc{GapQJS}_{\log}[\alpha(n), \beta(n)]$ is in BQL.*

Proof. Let Q_0 and Q_1 be the given $s(n)$-qubit quantum circuits where $s(n) = \Theta(\log n)$. Consider a classical-quantum mixed state on a classical register B and a quantum register Y, denoted by $\rho_1' := \frac{1}{2}|0\rangle\langle 0| \otimes \rho_0 + \frac{1}{2}|1\rangle\langle 1| \otimes \rho_1$, where ρ_0 and ρ_1 are the state obtained by running Q_0 and Q_1, respectively, and tracing out the non-output qubits. We utilize our reduction to output classical-quantum mixed states ρ_0' and ρ_1', which are the output of $(s(n)+2)$-qubit quantum circuits Q_0'

and Q_1',[34] respectively, where $\rho_0' := (p_0|0\rangle\langle 0| + p_1|1\rangle\langle 1|) \otimes (\frac{1}{2}\rho_0 + \frac{1}{2}\rho_1)$ and $\mathsf{B}' := (p_0, p_1)$ is an independent random bit with entropy $\mathrm{H}(\mathsf{B}') = 1 - \frac{1}{2}[\alpha(n) + \beta(n)]$. Let $\mathrm{S}_2(\rho) := \mathrm{S}(\rho)/\ln 2$ for any quantum state ρ, we then have derived that:

$$
\begin{aligned}
\mathrm{S}_2(\rho_0') - \mathrm{S}_2(\rho_1') &= \mathrm{S}_2(\mathsf{B}', \mathsf{Y})_{\rho_0'} - \mathrm{S}_2(\mathsf{B}, \mathsf{Y})_{\rho_1'} \\
&= [\mathrm{H}(\mathsf{B}') + \mathrm{S}_2(\mathsf{Y}|\mathsf{B}')_{\rho_0'}] - [\mathrm{H}(\mathsf{B}) + \mathrm{S}_2(\mathsf{Y}|\mathsf{B})_{\rho_1'}] \\
&= \mathrm{S}_2(\mathsf{Y})_{\rho_0'} - \mathrm{S}_2(\mathsf{Y}|\mathsf{B})_{\rho_1'} + \mathrm{H}(\mathsf{B}') - \mathrm{H}(\mathsf{B}) \\
&= \mathrm{S}_2(\mathsf{Y})_{\rho_0'} - \mathrm{S}_2(\mathsf{Y}|\mathsf{B})_{\rho_1'} - \tfrac{1}{2}[\alpha(n) + \beta(n)] \\
&= \mathrm{S}_2\left(\tfrac{1}{2}\rho_0 + \tfrac{1}{2}\rho_1\right) - \tfrac{1}{2}(\mathrm{S}_2(\rho_0) + \mathrm{S}_2(\rho_1)) - \tfrac{1}{2}[\alpha(n) + \beta(n)] \\
&= \mathrm{QJS}_2(\rho_0, \rho_1) - \tfrac{1}{2}[\alpha(n) + \beta(n)].
\end{aligned}
\tag{4.3}
$$

Here, the second line derives from the definition of quantum conditional entropy and acknowledges that both B and B' are classical registers. The third line owes to the independence of B' as a random bit. Furthermore, the fifth line relies on the Joint entropy theorem (Lemma 2.3).

By plugging Equation (4.3) into the promise of $\textsc{GapQJS}_{\log}[\alpha(n), \beta(n)]$, we can define $g(n) := \frac{\ln 2}{2}\big(\alpha(n-1) - \beta(n-1)\big)$ and conclude that:

- If $\mathrm{QJS}_2(\rho_0, \rho_1) \geq \alpha(n)$, then $\mathrm{S}(\rho_0') - \mathrm{S}(\rho_1') \geq \frac{\ln 2}{2}\big(\alpha(n) - \beta(n)\big) = g(n+1)$;
- If $\mathrm{QJS}_2(\rho_0, \rho_1) \leq \beta(n)$, then $\mathrm{S}(\rho_0') - \mathrm{S}(\rho_1') \leq -\frac{\ln 2}{2}\big(\alpha(n) - \beta(n)\big) = -g(n+1)$.

As ρ_1' and ρ_0' are $r'(n)$-qubit states where $r'(n) := r(n)+1$, the output length of the corresponding space-bounded quantum circuits Q_0' and Q_1' is $r'(n)$. Therefore, $\textsc{GapQJS}_{s(n)}[\alpha(n), \beta(n)]$ is logspace Karp reducible to $\textsc{GapQED}_{s+1}[g(n)]$ by mapping (Q_0, Q_1) to (Q_0', Q_1'). □

4.4 $\overline{\textsc{CertQSD}_{\log}}$ and $\overline{\textsc{CertQHS}_{\log}}$ are in $\mathsf{coRQ_UL}$

To make the error one-sided, we adapt the Grover search when the number of solutions is one quarter [BBHT98], also known as the exact amplitude amplification [BHMT02].

Lemma 4.13 (Exact amplitude amplification, adapted from [BHMT02, Equation 8])**.** *Suppose U is a unitary of interest such that $U|\bar{0}\rangle = \sin(\theta)|\psi_0\rangle + \cos(\theta)|\psi_1\rangle$, where $|\psi_0\rangle$ and $|\psi_1\rangle$ are normalized pure states and $\langle\psi_0|\psi_1\rangle = 0$. Let $G = -U(I - 2|\bar{0}\rangle\langle\bar{0}|)U^\dagger(I - 2|\psi_0\rangle\langle\psi_0|)$ be the Grover operator. Then, for every integer $j \geq 0$, we have $G^j U|\bar{0}\rangle = \sin((2j+1)\theta)|\psi_0\rangle + \cos((2j+1)\theta)|\psi_1\rangle$. In particular, with a single application of G, we obtain $GU|\bar{0}\rangle = \sin(3\theta)|\psi_0\rangle + \cos(3\theta)|\psi_1\rangle$, signifying that $GU|\bar{0}\rangle = |\psi_0\rangle$ when $\sin(\theta) = 1/2$.*

Notably, when dealing with the unitary of interest with the property specified in Lemma 4.13, which is typically a quantum algorithm with acceptance probability linearly dependent on the chosen distance-like measure (e.g., a tester $\mathcal{T}$ from Lemma 4.7), Lemma 4.13 guarantees that the resulting algorithm $\mathcal{A}$ accepts with probability exactly 1 for *yes* instances ($\rho_0 = \rho_1$). However, achieving $\mathcal{A}$ to accept with probability polynomially deviating from 1 for *no* instances requires additional efforts, leading to the $\mathsf{coRQ_UL}$ containment established through error reduction for $\mathsf{coRQ_UL}$ (Corollary 3.19). In a nutshell, demonstrating $\mathsf{coRQ_UL}$ containment entails satisfying the desired property, which is achieved differently for $\overline{\textsc{CertQSD}_{\log}}$ and $\overline{\textsc{CertQHS}_{\log}}$.

[34] To construct Q_1', we follow these steps: We start by applying a $\textsc{Hadamard}$ gate on B followed by a $\mathrm{CNOT}_{\mathsf{B}\to\mathsf{R}}$ gate where B and R are single-qubit quantum registers initialized on $|0\rangle$. Next, we apply the controlled-Q_1 gate on the qubits from B to S, where $\mathsf{S} = (\mathsf{Y}, \mathsf{Z})$ is an $s(n)$-qubit register initialized on $|\bar{0}\rangle$. We then apply X gate on B followed by the controlled-Q_0 gate on the qubits from B to S, and we apply X gate on B again. Finally, we obtain ρ_1' by tracing out R and the qubits in Z. In addition, we can construct Q_0' similarly.

4.4.1 $\overline{\textsc{CertQSD}}_{\log}$ is in $\mathsf{coRQ_UL}$

Our algorithm in Theorem 4.14 relies on the quantum tester $\mathcal{T}(Q_i, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}}, \epsilon)$ specified in Algorithm 1. Note that the exact implementation of the space-efficient QSVT associated with odd polynomials preserves the original point (Remark 3.12). Consequently, $\mathcal{T}(Q_i, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}}, \epsilon)$ outputs 0 with probability exactly 1/2 when $\rho_0 = \rho_1$, enabling us to derive the $\mathsf{coRQ_UL}$ containment through a relatively involved analysis for cases when $\mathrm{td}(\rho_0, \rho_1) \geq \alpha$:

Theorem 4.14. *For any deterministic logspace computable function $\alpha(n) \geq 1/\mathrm{poly}(n)$, we have that $\overline{\textsc{CertQSD}}_{\log}[\alpha(n)]$ is in $\mathsf{coRQ_UL}$.*

Proof. We first present a formal algorithm in Algorithm 3:

Algorithm 3: Space-efficient algorithm for $\overline{\textsc{CertQSD}}_{\log}$.

Input : Quantum circuits Q_i that prepares the purification of ρ_i for $i \in \{0,1\}$.

Output: Return "yes" if $\rho_0 = \rho_1$, and "no" otherwise.

Params: $\varepsilon := \frac{\alpha}{2}$, $\delta := \frac{\varepsilon}{2^{r+3}}$, $\epsilon := \frac{\varepsilon}{2(\hat{C}_{\mathrm{sgn}}+2C_{\mathrm{sgn}})\tilde{C}_{\mathrm{sgn}}2^{r+3}} \cdot \frac{1}{2\log\left(2(\hat{C}_{\mathrm{sgn}}+2C_{\mathrm{sgn}})\tilde{C}_{\mathrm{sgn}}2^{r+3}/\varepsilon\right)}$, $d := \tilde{C}_{\mathrm{sgn}}\delta^{-1}\log\epsilon^{-1}$.

1. Construct block-encodings of ρ_0 and ρ_1, denoted by U_{ρ_0} and U_{ρ_1}, respectively, using $O(1)$ queries to Q_0 and Q_1 and $O(s(n))$ ancillary qubits by Lemma 4.8;
2. Construct a block-encoding of $\frac{\rho_0-\rho_1}{2}$, denoted by $U_{\frac{\rho_0-\rho_1}{2}}$, using $O(1)$ queries to U_{ρ_0} and U_{ρ_1} and $O(s(n))$ ancillary qubits by Lemma 3.14;

Let P_d^{sgn} be the degree-d odd *polynomial specified in Corollary 3.7 with parameters δ and ϵ such that its Chebyshev coefficients are computable in deterministic space $O(\log(d/\epsilon))$;*

3. Let $U_0 := \mathcal{T}(Q_0, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}}, \epsilon)$ and $U_1 := \mathcal{T}(Q_1, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}}, \epsilon)$;
4. Let $G_i := -(H \otimes U_i)(I - 2|\bar{0}\rangle\langle\bar{0}|)(H \otimes U_i^\dagger)(I - 2\Pi_0)$ for $i \in \{0,1\}$, where Π_0 is the projector onto the subspace spanned by $\{|0\rangle|0\rangle|\varphi\rangle\}$ over all $|\varphi\rangle$;
5. Measure the first two qubits of $G_i(H \otimes U_i)|0\rangle|0\rangle|\bar{0}\rangle$, and let x_{i0} and x_{i1} be the outcomes, respectively. Return "yes" if $x_{00} = x_{01} = x_{10} = x_{11} = 0$, and "no" otherwise.

Constructing the unitary of interest via the space-efficient QSVT. We consider the setting with $s(n) = \Theta(\log n)$ and $\varepsilon = \alpha/2$. Suppose Q_0 and Q_1 are $s(n)$-qubit quantum circuits that prepare the purifications of ρ_0 and ρ_1, respectively. Similar to Algorithm 1, we first construct an $O(s)$-qubit quantum circuit $U_{\frac{\rho_0-\rho_1}{2}}$ that is a $(1, O(s), 0)$-block-encoding of $\frac{\rho_0-\rho_1}{2}$, using $O(1)$ queries to Q_{ρ_0} and Q_{ρ_1} and $O(1)$ one- and two-qubit quantum gates.

Let $\delta = \frac{\varepsilon}{2^{r+3}}$, $\epsilon := \frac{\varepsilon}{2(\hat{C}_{\mathrm{sgn}}+2C_{\mathrm{sgn}})\tilde{C}_{\mathrm{sgn}}2^{r+3}} \cdot \frac{1}{2\log\left(2(\hat{C}_{\mathrm{sgn}}+2C_{\mathrm{sgn}})\tilde{C}_{\mathrm{sgn}}2^{r+3}/\varepsilon\right)}$ and $d := \tilde{C}_{\mathrm{sgn}}\delta^{-1}\log\epsilon^{-1} = 2^{O(s)}$ where $\tilde{C}_{\mathrm{sgn}}$ comes from Corollary 3.7. Let $P_d^{\mathrm{sgn}} \in \mathbb{R}[x]$ be the odd polynomial specified in Corollary 3.7. Let $U_i := \mathcal{T}(Q_i, U_{\frac{\rho_0-\rho_1}{2}}, P_d^{\mathrm{sgn}}, \epsilon)$ for $i \in \{0,1\}$, then we have the following equalities with $0 \leq p_0, p_1 \leq 1$:

$$\begin{aligned} U_0|0\rangle|\bar{0}\rangle &= \sqrt{p_0}|0\rangle|\psi_0\rangle + \sqrt{1-p_0}|1\rangle|\psi_1\rangle, \\ U_1|0\rangle|\bar{0}\rangle &= \sqrt{p_1}|0\rangle|\phi_0\rangle + \sqrt{1-p_1}|1\rangle|\phi_1\rangle. \end{aligned}$$

Let H be the Hadamard gate, then we derive the following equality for $i \in \{0,1\}$:

$$(H \otimes U_i)|0\rangle|0\rangle|\bar{0}\rangle = \sqrt{\frac{p_i}{2}}|0\rangle|0\rangle|\psi_0\rangle + \underbrace{\sqrt{\frac{p_i}{2}}|0\rangle|1\rangle|\psi_0\rangle + \sqrt{\frac{1-p_i}{2}}|1\rangle|0\rangle|\psi_1\rangle + \sqrt{\frac{1-p_i}{2}}|1\rangle|1\rangle|\psi_1\rangle}_{\sqrt{1-\frac{p_i}{2}}|\perp_i\rangle}.$$

Making the error one-sided by exact amplitude amplification. Consider the Grover operator $G_i := -(H \otimes U_i)(I - 2|\bar{0}\rangle\langle\bar{0}|)(H \otimes U_i^\dagger)(I - 2\Pi_0)$, where Π_0 is the projector onto the subspace spanned by $\{|0\rangle|0\rangle|\varphi\rangle\}$ over all $|\varphi\rangle$. By employing the exact amplitude amplification (Lemma 4.13), we can obtain that:

$$G_i(H\otimes U_i)|0\rangle|0\rangle|\bar{0}\rangle = \sin(3\theta_i)|0\rangle|0\rangle|\psi_0\rangle + \cos(3\theta_i)|\perp_i\rangle \text{ where } \sin^2(\theta_i) = \tfrac{p_i}{2} \text{ when } \theta_i \in \left[0, \tfrac{\pi}{4}\right]. \quad (4.4)$$

Let x_{i0} and x_{i1} be the measurement outcomes of the first two qubits of $G_i(H \otimes U_i)|0\rangle|0\rangle|\bar{0}\rangle$ for $i \in \{0,1\}$. Algorithm 3 returns "yes" if $x_{00} = x_{01} = x_{10} = x_{11} = 0$, and "no" otherwise. We will show the correctness of our algorithm as follows:

- For *yes* instances ($\rho_0 = \rho_1$), $U_{P_d^{\mathrm{sgn}}\left(\frac{\rho_0-\rho_1}{2}\right)}$ is a $(1, O(s), 0)$-block-encoding of the zero operator, following from Remark 3.12. Consequently, $\mathcal{T}(Q_i, U_{\frac{\rho_0-\rho_1}{2}}, P_d, \epsilon)$ outputs 0 with probability 1/2 for $i \in \{0,1\}$, i.e., $p_0 = p_1 = 1/2$. As a result, we have $\theta_0 = \theta_1 = \pi/6$ and $\sin^2(3\theta_0) = \sin^2(3\theta_1) = 1$. Substituting these values into Equation (4.4), we can conclude that $x_{00} = x_{01} = x_{10} = x_{11} = 0$ with certainty, which completes the analysis.

- For *no* instances ($\mathrm{td}(\rho_0, \rho_1) \geq \alpha$), $U_{P_d^{\mathrm{sgn}}\left(\frac{\rho_0-\rho_1}{2}\right)}$ is a $(1, O(s), 0)$-block-encoding of A satisfying $\left\|A - P_d^{\mathrm{sgn}}\left(\frac{\rho_0-\rho_1}{2}\right)\right\| \leq \hat{C}_{\mathrm{sgn}}\epsilon \log d$. Let p_i be the probability that $\mathcal{T}(Q_i, U_{\frac{\rho_0-\rho_1}{2}}, P_d, \epsilon)$ outputs 0 for $i \in \{0,1\}$, then $p_i = \frac{1}{2}\big(1 + \mathrm{Re}(\mathrm{Tr}(\rho_i A))\big)$ following from Lemma 4.7. A direct calculation similar to Proposition 4.10.1 indicates that:
$$|(p_0 - p_1) - \mathrm{td}(\rho_0, \rho_1)| \leq \hat{C}_{\mathrm{sgn}}\epsilon \log d + 2C_{\mathrm{sgn}}\epsilon \log d + 2^{r+1}\delta.$$
Under the choice of δ, ϵ, and d in the proof of Theorem 4.11, we obtain that $|(p_0 - p_1) - \mathrm{td}(\rho_0, \rho_1)| \leq \varepsilon$ which yields that $\max\{|p_0 - 1/2|, |p_1 - 1/2|\} \geq \varepsilon/2$.[35]
Note that $\Pr[x_{i0} = x_{i1} = 0] = \sin^2(3\theta_i)$ for $i \in \{0,1\}$, Algorithm 3 will return "yes" with probability $p_{\mathrm{yes}} = \sin^2(3\theta_0)\sin^2(3\theta_1)$. We provide an upper bound for p_{yes} in Proposition 4.14.1, with the proof deferred to Appendix B.2:
Proposition 4.14.1. *Let $f(\theta_0, \theta_1) := \sin^2(3\theta_0)\sin^2(3\theta_1)$ be a function such that $\sin^2(\theta_i) = p_i/2$ for $i \in \{0,1\}$ and $\max\{|p_0 - 1/2|, |p_1 - 1/2|\} \geq \varepsilon/2$, then $f(\theta_0, \theta_1) \leq 1 - \varepsilon^2/4$.*
Consequently, we finish the analysis by noticing $p_{\mathrm{yes}} = f(\theta_0, \theta_1) \leq 1 - \varepsilon^2/4 = 1 - \alpha^2/16$.

Now we analyze the complexity of Algorithm 3. Following Lemma 4.7, we can compute $x_{00}, x_{01}, x_{10}, x_{11}$ in BQL. The quantum circuit that computes $x_{00}, x_{01}, x_{10}, x_{11}$ takes $O(d^2 \log d) = \tilde{O}(2^{2r}/\alpha^2)$ queries to Q_0 and Q_1, and its circuit description can be computed in deterministic time $\tilde{O}(d^{9/2}/\alpha) = \tilde{O}(2^{4.5r}/\alpha^{5.5})$. Finally, we conclude the $\mathsf{coRQ_UL}$ containment of $\overline{\textsc{CertQSD}_{\log}}$ by applying error reduction for $\mathsf{coRQ_UL}$(Corollary 3.19) to Algorithm 3. ☐

4.4.2 $\overline{\textsc{CertQHS}_{\log}}$ is in $\mathsf{coRQ_UL}$

Our algorithm in Theorem 4.15 is based on the observation that by expressing $\mathrm{HS}^2(\rho_0, \rho_1)$ as a summation of $\frac{1}{2}\mathrm{Tr}(\rho_0^2)$, $\frac{1}{2}\mathrm{Tr}(\rho_1^2)$, and $-\mathrm{Tr}(\rho_0\rho_1)$, we can devise a hybrid algorithm with *two random coins* using the SWAP test. However, to ensure *unitary*, we design another algorithm employing the LCU technique, which serves as the unitary of interest with the desired property.

Theorem 4.15. *For any deterministic logspace computation function $\alpha(n) \geq 1/\mathrm{poly}(n)$, we have that $\overline{\textsc{CertQHS}_{\log}}[\alpha(n)]$ is in $\mathsf{coRQ_UL}$.*

Proof. We first provide a formal algorithm in Algorithm 4.

[35] This inequality is because $|p_0 - p_1| \geq \mathrm{td}(\rho_0, \rho_1) - \varepsilon \geq 2\varepsilon - \varepsilon = \varepsilon$.

Algorithm 4: Space-efficient algorithm for $\overline{\textsc{CertQHS}}_{\log}$.

Input : Quantum circuits Q_i that prepares the purification of ρ_i for $i \in \{0,1\}$.

Output: Return "yes" if $\rho_0 = \rho_1$, and "no" otherwise.

1. Construct subroutines $T_{ij} := \mathrm{SWAP}(\rho_i, \rho_j)$ for $(i,j) \in \{(0,0),(1,1),(0,1)\}$, which output 0 with probability p_{ij}. The subroutine $\mathrm{SWAP}(\rho_i,\rho_j)$ involves applying Q_i and Q_j to prepare quantum states ρ_i and ρ_j, respectively, and then employing the SWAP test (Lemma 2.12) on these states ρ_i and ρ_j;
2. Construct a block-encoding of $\varrho\big(\frac{1}{2} + \frac{\mathrm{HS}^2(\rho_0,\rho_1)}{4}\big)$ where $\varrho(p) := p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|$, denoted by U, using $O(1)$ queries to T_{00}, T_{11}, and T_{01} by Lemma 3.14;
3. Let $G := -U(I - 2|\bar{0}\rangle\langle\bar{0}|)U^\dagger(I - 2|\bar{0}\rangle\langle\bar{0}|)$;
4. Measure all qubits of $GU|\bar{0}\rangle$ in the computational basis. Return "yes" if the measurement outcome is an all-zero string, and "no" otherwise.

Constructing the unitary of interest via the SWAP test. We consider the setting with $s(n) = \Theta(s(n))$. Our main building block is the circuit implementation of the SWAP test (Lemma 2.12). Specifically, we utilize the subroutine $\mathrm{SWAP}(\rho_i,\rho_j)$ for $i,j \in \{0,1\}$, which involves applying Q_i and Q_j to prepare quantum states ρ_i and ρ_j, respectively, and then employing the SWAP test on these states ρ_i and ρ_j. We denote by p_{ij} the probability that $\mathrm{SWAP}(\rho_i,\rho_j)$ outputs 0 based on the measurement outcome of the control qubit in the SWAP test. Following Lemma 2.12, we have $p_{ij} = \frac{1}{2}\big(1 + \mathrm{Tr}(\rho_i\rho_j)\big)$ for $i,j \in \{0,1\}$.

We define $T_{ij} := \mathrm{SWAP}(\rho_i,\rho_j)$ for $(i,j) \in \mathcal{I} := \{(0,0),1,1,0,1\}$, with the control qubit in $\mathrm{SWAP}(\rho_i,\rho_j)$ serving as the output qubit of T_{ij}. By introducing another ancillary qubit, we construct $T'_{ij} := \mathrm{CNOT}(I \otimes T_{ij})$ for $(i,j) \in \mathcal{I}$, where CNOT is controlled by the output qubit of T_{ij} and targets on the new ancillary qubit. It is effortless to see that T'_{ij} prepares the purification of $\varrho(p_{ij})$ with $\varrho(p_{ij}) := p_{ij}|0\rangle\langle 0| + (1-p_{ij})|1\rangle\langle 1|$ for $(i,j) \in \mathcal{I}$.

By applying Lemma 4.8, we can construct quantum circuits T''_{ij} for $(i,j) \in \mathcal{I}$ that serve as $(1, O(s), 0)$-block-encoding of $\varrho(p_{ij})$, using $O(1)$ queries to T'_{ij} and $O(1)$ one- and two-qubit quantum gates. Notably, $(X \otimes I)T''_{01}$, with X acting on the qubit of $\varrho(p_2)$, prepares the purification of $X\varrho(p_{01})X^\dagger = p_{01}|1\rangle\langle 1| + (1-p_{01})|0\rangle\langle 0| = \varrho(1-p_{01})$, leading to the equality:

$$\varrho(\rho_0,\rho_1) := \frac{1}{4}\varrho(p_{00}) + \frac{1}{4}\varrho(p_{11}) + \frac{1}{2}\varrho(1-p_{01}) = \varrho\left(\frac{1}{2} + \frac{\mathrm{HS}^2(\rho_0,\rho_1)}{4}\right).$$

Consequently, we employ Lemma 3.14 to construct a unitary quantum circuit U that is a $(1,m,0)$-block-encoding of $\varrho\big(\frac{1}{2} + \frac{\mathrm{HS}^2(\rho_0,\rho_1)}{4}\big)$ using $O(1)$ queries to T''_{00}, T''_{11}, $(X \otimes I)T''_{01}$, and $O(1)$ one- and two-qubit quantum gates, where $m := O(s)$. The construction ensures the following:

$$U|0\rangle|0\rangle^{\otimes m} = \underbrace{\left(\frac{1}{2} + \frac{\mathrm{HS}^2(\rho_0,\rho_1)}{4}\right)}_{\sin(\theta)}|0\rangle|0\rangle^{\otimes m} + \cos(\theta)|\perp\rangle, \text{ where } \langle 0|\langle 0|^{\otimes m}|\perp\rangle = 0. \tag{4.5}$$

Making the error one-sided. Let us consider the Grover operator $G := -U(I - 2|\bar{0}\rangle\langle\bar{0}|)U^\dagger(I - 2|\bar{0}\rangle\langle\bar{0}|)$. By applying Lemma 4.13, we derive that $GU|0\rangle|0\rangle^{\otimes m} = \sin(3\theta)|0\rangle|0\rangle^{\otimes m} + \cos(3\theta)|\perp\rangle$. Subsequently, we measure all qubits of $GU|0\rangle|0\rangle^{\otimes m}$ in the computational basis, represented as $x \in \{0,1\}^{m+1}$. Hence, Algorithm 4 returns "yes" if the outcome x is 0^{m+1} and "no" otherwise. Algorithm 4 accepts with probability $\sin^2(3\theta)$. Now we analyze the correctness of the algorithm:

- For *yes* instances ($\rho_0 = \rho_1$), we have $\mathrm{HS}^2(\rho_0,\rho_1) = 0$. Following Equation (4.5), we obtain $\sin(\theta) = 1/2$ and thus $\sin^2(3\theta) = 1$. We conclude that Algorithm 4 will always return "yes".

- For *no* instances, we have $\mathrm{HS}^2(\rho_0,\rho_1)\geq\alpha$. According to Equation (4.5), we derive that:
$$\sin(\theta)=\frac{1}{2}+\frac{\mathrm{HS}^2(\rho_0,\rho_1)}{4}\geq\frac{1}{2}+\frac{\alpha}{4}\text{ and }\frac{1}{4}\leq\sin^2(\theta)=\left(\frac{1}{2}+\frac{\mathrm{HS}^2(\rho_0,\rho_1)}{4}\right)^2\leq\left(\frac{1}{2}+\frac{1}{4}\right)^2=\frac{9}{16}. \quad (4.6)$$
As a result, considering the fact that $\sin^2(3\theta)=f(\sin^2(\theta))$ where $f(x):=16x^3-24x^2+9x$, we require Proposition 4.15.1 and the proof is deferred to Appendix B.2:

Proposition 4.15.1. *The polynomial function $f(x):=16x^3-24x^2+9x$ is monotonically decreasing in $[1/4,9/16]$. Moreover, we have $f\big((\frac{1}{2}+\frac{\alpha}{4})^2\big)\leq 1-\frac{\alpha^2}{2}$ for any $0\leq\alpha\leq 1$.*

Combining Equation (4.6) and Proposition 4.15.1, we have that $\sin^2(3\theta)=f(\sin^2(\theta))\leq f\big((\frac{1}{2}+\frac{\alpha}{4})^2\big)\leq 1-\frac{\alpha^2}{2}$. Hence, Algorithm 4 will return "no" with probability at least $\alpha^2/2$.

Regarding the computational complexity of Algorithm 4, this algorithm requires $O(s(n))$ qubits and performs $O(1)$ queries to Q_0 and Q_1. Finally, we finish the proof by applying error reduction from $\mathsf{coRQ_UL}$ (Corollary 3.19) to Algorithm 3. □

4.5 BQL- and $\mathsf{coRQ_UL}$-hardness for space-bounded state testing problems

We will prove that space-bounded state testing problems mentioned in Theorem 4.6 are $\mathsf{BQ_UL}$-hard, which implies their BQL-hardness since $\mathsf{BQL}{=}\mathsf{BQ_UL}$ [FR21]. Similarly, all space-bounded state certification problems mentioned in Theorem 4.5 are $\mathsf{coRQ_UL}$-hard.

4.5.1 Hardness results for $\textsc{GapQSD}_{\log}$, $\textsc{GapQHS}_{\log}$, and their certification version

Employing analogous constructions, we can establish the $\mathsf{BQ_UL}$-hardness of both $\textsc{GapQSD}_{\log}$ and $\textsc{GapQHS}_{\log}$. The former involves a single-qubit pure state and a single-qubit mixed state, while the latter involves two pure states.

Lemma 4.16 ($\overline{\textsc{GapQSD}}_{\log}$ is $\mathsf{BQ_UL}$-hard)**.** *For any deterministic logspace computable functions $a(n)$ and $b(n)$ such that $a(n)-b(n)\geq 1/\mathrm{poly}(n)$, we have that $\overline{\textsc{GapQSD}}_{\log}[1-\sqrt{a(n)},\sqrt{1-b(n)}]$ is $\mathsf{BQ_UL}[a(n),b(n)]$-hard.*

Proof. Consider a promise problem $(\mathcal{L}_{yes},\mathcal{L}_{no})\in\mathsf{BQ_UL}[a(n),b(n)]$, then we know that the acceptance probability $\Pr[C_x\text{ accepts}]\geq a(n)$ if $x\in\mathcal{L}_{yes}$, whereas $\Pr[C_x\text{ accepts}]\leq b(n)$ if $x\in\mathcal{L}_{no}$. Now we notice that the acceptance probability is the fidelity between a single-qubit pure state ρ_0 and a single-qubit mixed state ρ_1 that generates by two logarithmic-qubit quantum circuits Q_0 and Q_1, respectively:
$$\begin{aligned}\Pr[C_x\text{ accepts}]=&\||1\rangle\langle 1|_{\mathrm{out}}C_x|\bar{0}\rangle\|_2^2\\ =&\mathrm{Tr}\left(|1\rangle\langle 1|_{\mathrm{out}}\mathrm{Tr}_{\overline{\mathrm{out}}}\left(C_x|\bar{0}\rangle\langle\bar{0}|C_x^\dagger\right)\right)\\ =&\mathrm{F}^2\left(|1\rangle\langle 1|_{\mathrm{out}},\mathrm{Tr}_{\overline{\mathrm{out}}}\left(C_x|\bar{0}\rangle\langle\bar{0}|C_x^\dagger\right)\right)\\ :=&\mathrm{F}^2(\rho_0,\rho_1).\end{aligned}\tag{4.7}$$
In particular, the corresponding Q_0 is simply flipping the designated output qubit, as well as the corresponding Q_1 is exactly the circuit C_x, then we prepare ρ_0 and ρ_1 by tracing out all non-output qubits. By utilizing Lemma 2.2, we have derived that:

- For *yes* instances, $\mathrm{F}^2(\rho_0,\rho_1)\geq a(n)$ deduces that $\mathrm{td}(\rho_0,\rho_1)\leq 1-\sqrt{a(n)}$;
- For *no* instances, $\mathrm{F}^2(\rho_0,\rho_1)\leq b(n)$ yields that $\mathrm{td}(\rho_0,\rho_1)\geq\sqrt{1-b(n)}$

Therefore, we demonstrate that $\overline{\textsc{GapQSD}}_{\log}[1-\sqrt{a(n)},\sqrt{1-b(n)}]$ is $\mathsf{BQL}[a(n),b(n)]$-hard. □

To construct pure states, adapted from the construction in Lemma 4.16, we replace the final measurement in the BQL circuit C_x with a quantum gate (CNOT) and design a new algorithm based on C_x with the final measurement on *all* qubits in the computational basis.

Lemma 4.17 ($\overline{\text{GAPQHS}}_{\log}$ is $\mathsf{BQ_UL}$-hard)**.** *For any deterministic logspace computable functions $a(n)$ and $b(n)$ such that $a(n)-b(n) \geq 1/\operatorname{poly}(n)$, we have that $\overline{\text{GAPQHS}}_{\log}[1-a^2(n), 1-b^2(n)]$ is $\mathsf{BQ_UL}[a(n),b(n)]$-hard.*

Proof. For any promise problem $(\mathcal{L}_{yes}, \mathcal{L}_{no}) \in \mathsf{BQ_UL}[a(n),b(n)]$, we have that the acceptance probability $\Pr[C_x \text{ accepts}] \geq a(n)$ if $x \in \mathcal{L}_{yes}$, whereas $\Pr[C_x \text{ accepts}] \leq b(n)$ if $x \in \mathcal{L}_{no}$. For convenience, let the output qubit be the register O. Now we construct a new quantum circuit C'_x with an additional ancillary qubit on the register F initialized to zero:

$$C'_x := C_x^\dagger X_{\mathsf{O}}^\dagger \text{CNOT}_{\mathsf{O}\to\mathsf{F}} X_{\mathsf{O}} C_x.$$

And we say that C'_x accepts if the measurement outcome of all qubits (namely the working qubit of C_x and F) are all zero. Through a direct calculation, we obtain:

$$\begin{aligned}
\Pr\left[C'_x \text{ accepts}\right] &= \left\|(|\bar{0}\rangle\langle\bar{0}| \otimes |0\rangle\langle 0|_{\mathsf{F}}) C_x^\dagger X_{\mathsf{O}} \text{CNOT}_{\mathsf{O}\to\mathsf{F}} X_{\mathsf{O}} C_x (|\bar{0}\rangle \otimes |0\rangle_{\mathsf{F}})\right\|_2^2 \\
&= \left|(\langle\bar{0}| \otimes \langle 0|_{\mathsf{F}}) C_x^\dagger (|1\rangle\langle 1|_{\mathsf{O}} \otimes I_{\mathsf{F}} + |0\rangle\langle 0|_{\mathsf{O}} \otimes X_{\mathsf{F}}) C_x (|\bar{0}\rangle \otimes |0\rangle_{\mathsf{F}})\right|^2 \\
&= \left|\langle\bar{0}| C_x^\dagger |1\rangle\langle 1|_{\mathsf{O}} C_x |\bar{0}\rangle\right|^2 \\
&= \Pr{}^2\left[C_x \text{ accepts}\right].
\end{aligned} \tag{4.8}$$

Here, the second line owes to $\text{CNOT}_{\mathsf{O}\to\mathsf{F}} = |0\rangle\langle 0|_{\mathsf{O}} \otimes I_{\mathsf{F}} + |1\rangle\langle 1|_{\mathsf{O}} \otimes X_{\mathsf{F}}$, and the last line is because of Equation (4.7). Interestingly, by defining two pure states $\rho_0 := |\bar{0}\rangle\langle\bar{0}| \otimes |0\rangle\langle 0|_{\mathsf{F}}$ and $\rho_1 := C'_x(|\bar{0}\rangle\langle\bar{0}| \otimes |0\rangle\langle 0|_{\mathsf{F}}) {C'}_x^\dagger$ corresponding to $Q_0 = I$ and $Q_1 = C'_x$, respectively, we deduce the following from Equation (4.8):

$$\Pr\left[C'_x \text{ accepts}\right] = \operatorname{Tr}(\rho_0\rho_1) = 1 - \operatorname{HS}^2(\rho_0, \rho_1). \tag{4.9}$$

Combining Equation (4.8) and Equation (4.9), we conclude that:

- For *yes* instances, $\Pr[C_x \text{ accepts}] \geq a(n)$ implies that $\operatorname{HS}^2(\rho_0,\rho_1) \leq 1 - a^2(n)$;
- For *no* instances, $\Pr[C_x \text{ accepts}] \leq b(n)$ yields that $\operatorname{HS}^2(\rho_0,\rho_1) \geq 1 - b^2(n)$.

We thus complete the proof of $\overline{\text{GAPQHS}}_{\log}[1-a^2(n), 1-b^2(n)]$ is $\mathsf{BQ_UL}[a(n),b(n)]$-hard. ☐

Our constructions in the proof of Lemma 4.16 and Lemma 4.17 are somewhat analogous to Theorem 12 and Theorem 13 in [RASW23]. Then we proceed with a few direct corollaries of Lemma 4.16 and Lemma 4.17.

Corollary 4.18 ($\mathsf{BQ_UL}$- and $\mathsf{coRQ_UL}$-hardness)**.** *For any functions $a(n)$ and $b(n)$ are computable in deterministic logspace such that $a(n) - b(n) \geq 1/\operatorname{poly}(n)$, the following holds for some polynomial $p(n)$ which can be computed in deterministic logspace:*

1. $\text{GAPQSD}_{\log}[\alpha(n), \beta(n)]$ *is $\mathsf{BQ_UL}$-hard for $\alpha \leq 1 - 1/p(n)$ and $\beta \geq 1/p(n)$;*
2. $\overline{\text{CERTQSD}}_{\log}[\gamma(n)]$ *is $\mathsf{coRQ_UL}$-hard for $\gamma \leq 1 - 1/p(n)$;*
3. $\text{GAPQHS}_{\log}[\alpha(n), \beta(n)]$ *is $\mathsf{BQ_UL}$-hard for $\alpha \leq 1 - 1/p(n)$ and $\beta \geq 1/p(n)$;*
4. $\overline{\text{CERTQHS}}_{\log}[\gamma(n)]$ *is $\mathsf{coRQ_UL}$-hard for $\gamma \leq 1 - 1/p(n)$.*

Proof. Firstly, it is important to note that $\mathsf{BQ_UL}$ is closed under complement, as demonstrated in [Wat99, Corollary 4.8]. By combining error reduction for $\mathsf{BQ_UL}$ (Corollary 3.19) and Lemma 4.16 (resp., Lemma 4.17), we can derive the first statement (resp., the third statement).

Moreover, to obtain the second statement (resp., the fourth statement), we can utilize error reduction for $\mathsf{coRQ_UL}$ (Corollary 3.19) and set $a = 1$ in Lemma 4.16 (resp., Lemma 4.17). ☐

4.5.2 Hardness results for $\textsc{GapQJS}_{\log}$ and $\textsc{GapQED}_{\log}$

We demonstrate the $\mathsf{BQ_UL}$-hardness of $\textsc{GapQJS}_{\log}$ by reducing $\textsc{GapQSD}_{\log}$ to $\textsc{GapQJS}_{\log}$, following a similar approach as shown in [Liu23, Lemma 4.11].

Lemma 4.19 ($\textsc{GapQJS}_{\log}$ is $\mathsf{BQ_UL}$-hard)**.** *For any functions $\alpha(n)$ and $\beta(n)$ are computable in deterministic logspace, we have $\textsc{GapQJS}_{\log}[\alpha(n),\beta(n)]$ is $\mathsf{BQ_UL}$-hard for $\alpha(n) \leq 1-\sqrt{2}/\sqrt{p(n)}$ and $\beta(n) \geq 1/p(n)$, where $p(n)$ is some deterministic logspace computable polynomial.*

Proof. By employing Corollary 4.18, it suffices to reduce $\textsc{GapQSD}_{\log}[1-1/p(n), 1/p(n)]$ to $\textsc{GapQJS}_{\log}[\alpha(n),\beta(n)]$. Consider logarithmic-qubit quantum circuits Q_0 and Q_1, which is an instance of $\textsc{GapQSD}_{\log}$. We will obtain ρ_k by performing Q_k on $|0^n\rangle$ and tracing out the non-output qubits for $k \in \{0,1\}$. We then have the following:

- If $\mathrm{td}(\rho_0,\rho_1) \geq 1-1/p(n)$, then Lemma 2.4 yields that
$$\mathrm{QJS}_2(\rho_0,\rho_1) \geq 1 - \mathrm{H}_2\left(\tfrac{1-\mathrm{td}(\rho_0,\rho_1)}{2}\right) \geq 1-\mathrm{H}_2\left(\tfrac{1}{2p(n)}\right) \geq 1-\tfrac{\sqrt{2}}{\sqrt{p(n)}} \geq \alpha(n),$$
where the third inequality owing to $\mathrm{H}_2(x) \leq 2\sqrt{x}$ for all $x \in [0,1]$.
- If $\mathrm{td}(\rho_0,\rho_1) \leq 1/p(n)$, then Lemma 2.4 indicates that
$$\mathrm{QJS}_2(\rho_0,\rho_1) \leq \mathrm{td}(\rho_0,\rho_1) \leq \tfrac{1}{p(n)} \leq \beta(n).$$

Therefore, we can utilize the same quantum circuits Q_0 and Q_1, along with their corresponding quantum states ρ_0 and ρ_1, respectively, to establish a logspace Karp reduction from $\textsc{GapQSD}_{\log}[1-1/p(n),1/p(n)]$ to $\textsc{GapQJS}_{\log}[\alpha(n),\beta(n)]$, as required. □

By combining the reduction from $\textsc{GapQSD}_{\log}$ to $\textsc{GapQJS}_{\log}$ (Lemma 4.19) and the reduction from $\textsc{GapQJS}_{\log}$ to $\textsc{GapQED}_{\log}$ (Corollary 4.12), we will demonstrate that the $\mathsf{BQ_UL}$-hardness for $\textsc{GapQED}_{\log}$ through reducing $\textsc{GapQSD}_{\log}$ to $\textsc{GapQED}_{\log}$.

Corollary 4.20 ($\textsc{GapQED}_{\log}$ is $\mathsf{BQ_UL}$-hard)**.** *For any function $g(n)$ are computable in deterministic logspace, we have $\textsc{GapQED}_{\log}[g(n)]$ is $\mathsf{BQ_UL}$-hard for $g(n) \leq \frac{\ln 2}{2}\big(1-\frac{\sqrt{2}}{\sqrt{p(n-1)}}-\frac{1}{p(n-1)}\big)$, where $p(n)$ is some polynomial that can be computed in deterministic logspace.*

Proof. By combining Corollary 4.18 and Lemma 4.19, we establish that $\textsc{GapQJS}_{\log}[\alpha(n),\beta(n)]$ is $\mathsf{BQ_UL}$-hard for $\alpha(n) \leq 1-\sqrt{2}/\sqrt{p(n)}$ and $\beta(n) \geq 1/p(n)$, where $p(n)$ is some deterministic logspace computable polynomial. The hard instances specified in Corollary 4.18 consist of $s(n)$-qubit quantum circuits Q_0 and Q_1 that prepares a purification of $r(n)$-qubit (mixed) quantum states ρ_0 and ρ_1, respectively, where $1 \leq r(n) \leq s(n) = \Theta(\log n)$.

Subsequently, by employing Corollary 4.12, we construct $(s+1)$-qubit quantum circuits Q_0' and Q_1' that prepares a purification of $(r+1)$-qubit quantum states $\rho_0' = \big(p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|\big) \otimes \big(\frac{1}{2}\rho_0 + \frac{1}{2}\rho_1\big)$ satisfying $\mathrm{H}_2(p) = 1-\frac{1}{2}\big(\alpha(n)+\beta(n)\big)$ and $\rho_1' = \frac{1}{2}|0\rangle\langle 0|\otimes\rho_0 + \frac{1}{2}|1\rangle\langle 1|\otimes\rho_1$, respectively. Following Corollary 4.12, $\textsc{GapQED}_{\log}[g(n)]$ is $\mathsf{BQ_UL}$-hard as long as
$$g(n) = \tfrac{\ln 2}{2}\big(\alpha(n-1)-\beta(n-1)\big) \leq \tfrac{\ln 2}{2}\Big(1-\tfrac{\sqrt{2}}{\sqrt{p(n-1)}}-\tfrac{1}{p(n-1)}\Big).$$
Therefore, $\textsc{GapQSD}_{s(n)}[\alpha(n),\beta(n)]$ is logspace Karp reducible to $\textsc{GapQED}_{s+1}[g(n)]$ by mapping (Q_0,Q_1) to (Q_0',Q_1'). □

Acknowledgments

This work was partially supported by MEXT Q-LEAP grant No. JPMXS0120319794. FLG was also supported by JSPS KAKENHI grants Nos. JP19H04066, JP20H05966, JP20H00579, JP20H04139, and JP21H04879. YL was also supported by JST, the establishment of University fellowships towards the creation of science technology innovation, Grant No. JPMJFS2125. We express our gratitude to anonymous reviewers for providing detailed suggestions on the space-efficient quantum singular value transformation and for suggesting to add discussion on space-bounded distribution testing. Circuit diagrams were drawn by the Quantikz package [Kay18].

References

[ABIS19] Jayadev Acharya, Sourbh Bhadane, Piotr Indyk, and Ziteng Sun. Estimating entropy of distributions in constant space. *Advances in Neural Information Processing Systems*, 32, 2019. 6

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, 1991. 5

[AISW20] Jayadev Acharya, Ibrahim Issa, Nirmal V. Shende, and Aaron B. Wagner. Estimating quantum entropy. *IEEE Journal on Selected Areas in Information Theory*, 1(2):454–468, 2020. 4

[AJL09] Dorit Aharonov, Vaughan Jones, and Zeph Landau. A polynomial quantum algorithm for approximating the Jones polynomial. *Algorithmica*, 55(3):395–421, 2009. 9, 32, 33

[AKN98] Dorit Aharonov, Alexei Kitaev, and Noam Nisan. Quantum circuits with mixed states. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 20–30, 1998. 13

[AMNW22] Maryam Aliakbarpour, Andrew McGregor, Jelani Nelson, and Erik Waingarten. Estimation of entropy in constant space with improved sample complexity. *Advances in Neural Information Processing Systems*, 35:32474–32486, 2022. 6

[AS16] Noga Alon and Joel H Spencer. *The Probabilistic Method*. John Wiley & Sons, 2016. 52

[AZLO16] Zeyuan Allen-Zhu, Yin Tat Lee, and Lorenzo Orecchia. Using optimization to obtain a width-independent, parallel, simpler, and faster positive SDP solver. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1824–1831. SIAM, 2016. 10

[BASTS10] Avraham Ben-Aroya, Oded Schwartz, and Amnon Ta-Shma. Quantum expanders: Motivation and construction. *Theory of Computing*, 6(1):47–79, 2010. 4, 5

[BBHT98] 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. 10, 37

[BCC+15] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical Review Letters*, 114(9):090502, 2015. 8, 26

[BCH+19] Adam Bouland, Lijie Chen, Dhiraj Holden, Justin Thaler, and Prashant Nalini Vasudevan. On the power of statistical zero knowledge. *SIAM Journal on Computing*, 49(4):FOCS17–1, 2019. 6

[BCWdW01] Harry Buhrman, Richard Cleve, John Watrous, and Ronald de Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902, 2001. 4, 6, 7, 15

[BDRV19] Itay Berman, Akshay Degwekar, Ron D Rothblum, and Prashant Nalini Vasudevan. Statistical difference beyond the polarizing regime. In *Theory of Cryptography Conference*, pages 311–332. Springer, 2019. 5

[BH09] Jop Briët and Peter Harremoës. Properties of classical and quantum Jensen-Shannon divergence. *Physical review A*, 79(5):052311, 2009. 12

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Quantum Computation and Information*, 305:53–74, 2002. 10, 15, 37

[BL13] Andrej Bogdanov and Chin Ho Lee. Limits of provable security for homomorphic encryption. In *Annual Cryptology Conference*, pages 111–128. Springer, 2013. 5

[BLT92] José L Balcázar, Antoni Lozano, and Jacobo Torán. The complexity of algorithmic problems on succinct instances. In *Computer science: research and applications*, pages 351–377. Springer, 1992. 30

[BOW19] Costin Bădescu, Ryan O'Donnell, and John Wright. Quantum state certification. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 503–514, 2019. 2, 4, 11, 30

[BZ10] Richard P Brent and Paul Zimmermann. *Modern computer arithmetic*, volume 18. Cambridge University Press, 2010. 19

[Can20] Clément L Canonne. A survey on distribution testing: Your data is big. but is it blue? *Theory of Computing*, pages 1–100, 2020. 1

[CDG+20] Rui Chao, Dawei Ding, Andras Gilyen, Cupjin Huang, and Mario Szegedy. Finding angles for quantum signal processing with machine precision. *arXiv preprint arXiv:2003.02831*, 2020. 7

[CDSTS23] Gil Cohen, Dean Doron, Ori Sberlo, and Amnon Ta-Shma. Approximating iterated multiplication of stochastic matrices in small space. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 35–45, 2023. 9

[CDVV14] Siu-On Chan, Ilias Diakonikolas, Paul Valiant, and Gregory Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1193–1203. SIAM, 2014. 4

[CGKZ05] Howard Cheng, Barry Gergel, Ethan Kim, and Eugene Zima. Space-efficient evaluation of hypergeometric series. *ACM SIGSAM Bulletin*, 39(2):41–52, 2005. 19

[CLM10] Steve Chien, Katrina Ligett, and Andrew McGregor. Space-efficient estimation of robust statistics and distribution testing. In *ICS*, pages 251–265, 2010. 6

[CLW20] Anirban N. Chowdhury, Guang Hao Low, and Nathan Wiebe. A variational quantum algorithm for preparing quantum Gibbs states. ArXiv e-prints, 2020. 11

[DGKR19] Ilias Diakonikolas, Themis Gouleakis, Daniel M Kane, and Sankeerth Rao. Communication and memory efficient testing of discrete distributions. In *Conference on Learning Theory*, pages 1070–1106. PMLR, 2019. 6

[DMWL21] Yulong Dong, Xiang Meng, K Birgitta Whaley, and Lin Lin. Efficient phase-factor evaluation in quantum signal processing. *Physical Review A*, 103(4):042419, 2021. 7

[dW19] Ronald de Wolf. Quantum computing: Lecture notes. *arXiv preprint arXiv:1907.09415*, 2019. 11

[FKL+16] Bill Fefferman, Hirotada Kobayashi, Cedric Yen-Yu Lin, Tomoyuki Morimae, and Harumichi Nishimura. Space-efficient error reduction for unitary quantum computations. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016. 3, 13, 29

[FKSV02] Joan Feigenbaum, Sampath Kannan, Martin J. Strauss, and Mahesh Viswanathan. An approximate l_1-difference algorithm for massive data streams. *SIAM Journal on Computing*, 32(1):131–151, 2002. 6

[FL11] Steven T. Flammia and Yi-Kai Liu. Direct fidelity estimation from few Pauli measurements. *Physical Review Letters*, 106(23):230501, 2011. 11

[FL18] Bill Fefferman and Cedric Yen-Yu Lin. A complete characterization of unitary quantum space. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018. 1, 2, 3, 9, 12, 15, 30

[For87] Lance Fortnow. The complexity of perfect zero-knowledge. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 204–209, 1987. 5

[FR21] Bill Fefferman and Zachary Remscrim. Eliminating intermediate measurements in space-bounded quantum computation. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1343–1356, 2021. 1, 2, 3, 4, 9, 12, 13, 14, 15, 41

[FvdG99] Christopher A Fuchs and Jeroen van de Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, 1999. 12

[GH20] Alexandru Gheorghiu and Matty J Hoban. Estimating the entropy of shallow circuit outputs is hard. *arXiv preprint arXiv:2002.12814*, 2020. 11

[GHS21] Tom Gur, Min-Hsiu Hsieh, and Sathyawageeswar Subramanian. Sublinear quantum algorithms for estimating von Neumann entropy. ArXiv e-prints, 2021. 11

[Gil19] András Gilyén. *Quantum singular value transformation & its algorithmic applications*. PhD thesis, University of Amsterdam, 2019. 27

[GL20] András Gilyén and Tongyang Li. Distributional property testing in a quantum world. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. 11, 24

[GMV06] Sudipto Guha, Andrew McGregor, and Suresh Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 733–742, 2006. 6

[Gol17] Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017. 1

[Gol19] Oded Goldreich. Errata (3-Feb-2019). http://www.wisdom.weizmann.ac.il/~/oded/entropy.html, 2019. 5

[GP22] András Gilyén and Alexander Poremba. Improved quantum algorithms for fidelity estimation. *arXiv preprint arXiv:2203.15993*, 2022. 9, 11, 33

[GR02] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002. 27, 54

[GR22] Uma Girish and Ran Raz. Eliminating intermediate measurements using pseudorandom generators. In *13th Innovations in Theoretical Computer Science Conference, ITCS 2022*, page 76. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik, 2022. 3

[GRZ21] Uma Girish, Ran Raz, and Wei Zhan. Quantum logspace algorithm for powering matrices with bounded norm. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, page 73. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021. 3

[GSLW18] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. *arXiv preprint arXiv:1806.01838*, 2018. 25, 26, 30, 55, 56

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019. 1, 7, 16, 17, 20, 22, 23, 25, 26, 30, 33, 55, 56

[GSV98] Oded Goldreich, Amit Sahai, and Salil Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 399–408, 1998. 4, 5

[GV99] Oded Goldreich and Salil Vadhan. Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In *Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference)(Cat. No. 99CB36317)*, pages 54–73. IEEE, 1999. 4, 5

[GV11] Oded Goldreich and Salil Vadhan. On the complexity of computational problems regarding distributions. *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, pages 390–405, 2011. 5

[Haa19] Jeongwan Haah. Product decomposition of periodic functions in quantum signal processing. *Quantum*, 3:190, 2019. 7

[HHL09] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009. 3

[Hol73] Alexander Semenovich Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973. 6, 12

[JVHW15] Jiantao Jiao, Kartik Venkat, Yanjun Han, and Tsachy Weissman. Minimax estimation of functionals of discrete distributions. *IEEE Transactions on Information Theory*, 61(5):2835–2885, 2015. 4

[JY11] Rahul Jain and Penghui Yao. A parallel approximation algorithm for positive semidefinite programming. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 463–471. IEEE, 2011. 10

[Kay18] Alastair Kay. Tutorial on the quantikz package. *arXiv preprint arXiv:1809.03842*, 2018. 44

[Kit95] Alexei Yu Kitaev. Quantum measurements and the Abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995. 9, 32, 33

[Kit97] Alexei Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191, 1997. 3

[KM01] Phillip Kaye and Michele Mosca. Quantum networks for generating arbitrary quantum states. In *International Conference on Quantum Information*. Optica Publishing Group, 2001. 27, 54

[KMY09] Hirotada Kobayashi, Keiji Matsumoto, and Tomoyuki Yamakami. Quantum Merlin-Arthur proof systems: Are multiple Merlins more helpful to Arthur? *Chicago Journal of Theoretical Computer Science*, 2009:3, 2009. 6, 15

[LC17] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by uniform spectral amplification. *arXiv preprint arXiv:1707.05391*, 2017. 19

[LC19] Guang Hao Low and Isaac L. Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019. 33

[Liu23] Yupan Liu. Quantum state testing beyond the polarizing regime and quantum triangular discrimination. *arXiv preprint arXiv:2303.01952*, 2023. 5, 7, 10, 12, 36, 43

[LWZ22] Tongyang Li, Xinzhao Wang, and Shengyu Zhang. A unified quantum algorithm framework for estimating properties of discrete probability distributions. ArXiv e-prints, 2022. 11

[MdW16] Ashley Montanaro and Ronald de Wolf. A survey of quantum property testing. *Theory of Computing*, pages 1–81, 2016. 1

[Mez12] Marc Mezzarobba. A note on the space complexity of fast D-finite function evaluation. In *Computer Algebra in Scientific Computing: 14th International Workshop, CASC 2012, Maribor, Slovenia, September 3-6, 2012. Proceedings 14*, pages 212–223. Springer, 2012. 19

[MH02] John C Mason and David C Handscomb. *Chebyshev polynomials*. Chapman and Hall/CRC, 2002. 14

[MLP05] Ana P Majtey, Pedro W Lamberti, and Domingo P Prato. Jensen-Shannon divergence as a measure of distinguishability between mixed quantum states. *Physical Review A*, 72(5):052310, 2005. 12

[MP16] Ashley Montanaro and Sam Pallister. Quantum algorithms and the finite element method. *Physical Review A*, 93(3):032324, 2016. 54

[MRTC21] John M Martyn, Zane M Rossi, Andrew K Tan, and Isaac L Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2(4):040203, 2021. 1, 7

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017. 15

[MW05] Chris Marriott and John Watrous. Quantum Arthur–Merlin games. *Computational Complexity*, 14(2):122–152, 2005. 3

[MW23] Danial Motlagh and Nathan Wiebe. Generalized quantum signal processing. *arXiv preprint arXiv:2308.01501*, 2023. 10

[MY23] Tony Metger and Henry Yuen. $\mathsf{stateQIP} = \mathsf{statePSPACE}$. *arXiv preprint arXiv:2301.07730*, 2023. 8, 17, 19, 26, 27, 51

[NC02] Michael A Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. American Association of Physics Teachers, 2002. 6, 11, 12

[OW21] Ryan O'Donnell and John Wright. Quantum spectrum testing. *Communications in Mathematical Physics*, 387(1):1–75, 2021. 4, 11

[Pow67] Michael JD Powell. On the maximum errors of polynomial approximations defined by interpolation and by least squares criteria. *The Computer Journal*, 9(4):404–407, 1967. 8, 14

[PP23] Aaron Putterman and Edward Pyne. Near-optimal derandomization of medium-width branching programs. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 23–34, 2023. 9

[PY86] Christos H Papadimitriou and Mihalis Yannakakis. A note on succinct representations of graphs. *Information and Control*, 71(3):181–185, 1986. 30

[RASW23] Soorya Rethinasamy, Rochisha Agarwal, Kunal Sharma, and Mark M Wilde. Estimating distinguishability measures on quantum computers. *Physical Review A*, 108(1):012409, 2023. 4, 6, 7, 11, 42

[Rei08] Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM (JACM)*, 55(4):1–24, 2008. 4

[Sak96] Michael Saks. Randomization and derandomization in space-bounded computation. In *Proceedings of Computational Complexity (Formerly Structure in Complexity Theory)*, pages 128–149. IEEE, 1996. 15

[SM03] Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003. 18, 20

[SV03] Amit Sahai and Salil Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM (JACM)*, 50(2):196–249, 2003. 4, 5

[SZ99] Michael Saks and Shiyu Zhou. $\mathrm{BP_HSPACE}(s) \subseteq \mathrm{DSPACE}(s^{3/2})$. *Journal of computer and system sciences*, 58(2):376–403, 1999. 9

[TS13] Amnon Ta-Shma. Inverting well conditioned matrices in quantum logspace. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 881–890, 2013. 1, 2, 3, 9 [vAGGdW20] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum SDP-solvers: Better upper and lower bounds. *Quantum*, 4:230, 2020. 9, 20, 22

[vMW12] Dieter van Melkebeek and Thomas Watson. Time-space efficient simulations of quantum computations. *Theory of Computing*, 8(1):1–51, 2012. 1, 3, 12, 13, 14

[Wat99] John Watrous. Space-bounded quantum complexity. *Journal of Computer and System Sciences*, 59(2):281–326, 1999. 1, 3, 42

[Wat01] John Watrous. Quantum simulations of classical random walks and undirected graph connectivity. *Journal of computer and system sciences*, 62(2):376–391, 2001. 1, 3, 14, 29

[Wat02] John Watrous. Limits on the power of quantum statistical zero-knowledge. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 459–468. IEEE, 2002. 2, 4, 5, 7

[Wat03] John Watrous. On the complexity of simulating space-bounded quantum computations. *computational complexity*, 12:48–84, 2003. 1, 3, 12

[Wat08] John Watrous. Quantum computational complexity. *arXiv preprint arXiv:0804.3401*, 2008. 11

[Wat09] John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009. 4, 5, 12

[WGL+22] Qisheng Wang, Ji Guan, Junyi Liu, Zhicheng Zhang, and Mingsheng Ying. New quantum algorithms for computing quantum entropies and distances. *arXiv preprint arXiv:2203.13522*, 2022. 11

[WY16] Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6):3702–3720, 2016. 4

[WZ23a] Qisheng Wang and Zhicheng Zhang. Fast quantum algorithms for trace distance estimation. *arXiv preprint arXiv:2301.06783*, 2023. 9, 11, 34

[WZ23b] Qisheng Wang and Zhicheng Zhang. Quantum lower bounds by sample-to-query lifting. *arXiv preprint arXiv:2308.01794*, 2023. 27

[WZC+23] Qisheng Wang, Zhicheng Zhang, Kean Chen, Ji Guan, Wang Fang, Junyi Liu, and Mingsheng Ying. Quantum algorithm for fidelity estimation. *IEEE Transactions on Information Theory*, 69(1):273–282, 2023. 11

[Zal98] Christof Zalka. Simulating quantum systems on a quantum computer. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):313–322, 1998. 27, 54

A Omitted proofs in space-efficient QSVT

In this section, we will present all Omitted proofs in Section 3.

A.1 Space-efficient bounded polynomial approximations

A.1.1 Omitted proofs in Corollary 3.7

Fact 3.7.2. *Let* $F_k(\theta) := \operatorname{erf}(\kappa\cos\theta)\cos(k\theta)$*,* $\max_{0\le k\le d}\max_{\xi\in[-\pi,0]}|F_k''(\xi)| \le \frac{2}{\sqrt{\pi}}\kappa + k^2 + \frac{4}{\sqrt{\pi}}\kappa^3 + \frac{4}{\sqrt{\pi}}k\kappa$*.*

Proof. Through a straightforward calculation, we have derived that

$$\begin{aligned}|F_k''(\theta)| =& \frac{2}{\sqrt{\pi}}\left|\kappa\exp(-\kappa^2\cos^2\theta)\cos\theta\cos(\kappa\theta)\right| + \left|k^2\cos(k\theta)\operatorname{erf}(\kappa\cos(\theta))\right| \\ &+ \frac{4}{\sqrt{\pi}}\left|\kappa^3\exp(-\kappa^2\cos^2\theta)\cos\theta\cos(k\theta)\sin^2\theta\right| \\ &+ \frac{4}{\sqrt{\pi}}\left|k\kappa\exp(-\kappa^2\cos^2\theta)\sin\theta\sin(k\theta)\right| \\ \le& \frac{2}{\sqrt{\pi}}\kappa + k^2 + \frac{4}{\sqrt{\pi}}\kappa^3 + \frac{4}{\sqrt{\pi}}k\kappa.\end{aligned} \tag{A.1}$$

Here the last line owes to the facts that $|\operatorname{erf}(x)|\le 1$, $\exp(-x^2)\le 1$, $|\sin x|\le 1$, and $|\cos x|\le 1$ for any x. We thus complete the proof by noting that Equation (A.1) holds for any $0\le k\le d$. ◻

Fact 3.7.3 (Implicit in [MY23, Lemma 2.10])**.** *For the coefficient vector* $\tilde{\mathbf{c}}^{\mathrm{sgn}}$ *corresponding to a degree-*d *polynomial* $\tilde{P}_d^{\mathrm{sgn}}$*, we have* $\|\tilde{\mathbf{c}}^{\mathrm{sgn}}\|_1 \le \hat{C}_{\mathrm{sgn}}\log d$ *where* $\hat{C}_{\mathrm{sgn}}$ *is a universal constant.*

Proof. Consider $c'_k := \langle T_k, \operatorname{sgn}\rangle$, and a direct integration yields $c'_k = (-1)^{(k-1)/2}\frac{4}{\pi k}$ for odd k which implies that $\|\mathbf{c}'\|_1 = \sum_{l=1}^{2l\le d+1}|c'_{2l-1}| = O(\log d)$, as per the Euler–Maclaurin formula. By observing that $|\operatorname{erf}(\kappa x) - \operatorname{sgn}(x)| \le 1$ on the interval $[-\delta,\delta]$, and $|\operatorname{erf}(\kappa x)-\operatorname{sgn}(x)|\le\epsilon$ elsewhere, it follows that $|\tilde{c}_k - c'_k| \le O(\max\{\delta,\epsilon\})$. This observation leads us to the conclusion that $|\|\tilde{\mathbf{c}}\|_1 - \|\mathbf{c}'\|_1| \le O(d\max\{\delta,\epsilon\})$, which implies that $\|\tilde{\mathbf{c}}\|_1 \le \hat{C}_{\mathrm{sgn}}\log d$ for some universal constant $\hat{C}_{\mathrm{sgn}}$. ◻

A.1.2 Omitted proofs in Lemma 3.9

Proposition 3.9.1 (First approximation)**.** *Let* $\hat{f}_1(x) := \sum_{k=0}^K a_k x^k$ *such that* $\|f-\hat{f}_1\|_\infty \le \epsilon/4$*. Then we know that* $\hat{f}_1(x) = \sum_{k=0}^K a_k \sum_{l=0}^\infty b_l^{(k)} \sin^l\left(\frac{x\pi}{2}\right)$ *where the coefficients* $b_l^{(k)}$ *satisfy that*

$$b_l^{(k+1)} = \sum_{l'=0}^{l} b_{l'}^{(k)} b_{l-l'}^{(1)} \text{ where } b_l^{(1)} = \begin{cases} 0 & \text{if } l \text{ is even,} \\ \binom{l-1}{\frac{l-1}{2}}\frac{2^{-l+1}}{l}\cdot\frac{2}{\pi} & \text{if } l \text{ is odd.}\end{cases} \tag{3.3}$$

Furthermore, the coefficients $\{b_l^{(k)}\}$ *satisfies the following: (1)* $\|\mathbf{b}^{(k)}\|_1 = 1$ *for all* $k\ge 1$*; (2)* $\mathbf{b}^{(k)}$ *is entry-wise non-negative for all* $k \ge 1$*; (3)* $b_l^{(k)} = 0$ *if* l *and* k *have different parities.*

Proof. We construct a Fourier series by a linear combination of the power of sines. We first note that $x = \frac{2}{\pi}\cdot\arcsin\left(\sin\left(\frac{x\pi}{2}\right)\right)$ for all $x\in[-1,1]$, and plug it into $\hat{f}_1(x) := \sum_{k=0}^K a_k x^k$, which deduces that $\|f-\hat{f}_1\|\le\epsilon/4$ by the assumption. Let $\mathbf{b}^{(k)}$ be the coefficients of $\left(\frac{\arcsin y}{\pi/2}\right)^k = \sum_{l=0}^\infty b_l^{(k)} y^l$ for all $y\in[-1,1]$, then we result in our first approximation. Moreover, we observe that $\frac{\pi}{2}\cdot\mathbf{b}^{(1)}$ is exactly the Taylor series of arcsin, whereas we know that $\left(\frac{\arcsin y}{\pi/2}\right)^{k+1} = \left(\frac{\arcsin y}{\pi/2}\right)^k\cdot\left(\sum_{l=0}^\infty b_l^{(1)} y^l\right)$ for $k>1$, which derives Equation (3.3) by comparing the coefficients. In addition, notice that $\|\mathbf{b}^{(k)}\|_1 = \sum_{l=0}^\infty b_l^{(k)} 1^l = \left(\frac{\arcsin 1}{\pi/2}\right)^k = 1$, together with straightforward reasoning follows from Equation (3.3), we deduce the desired property for $\{b_l^{(k)}\}$. ◻

Proposition 3.9.2 (Second approximation)**.** *Let* $\hat{f}_2(x) := \sum_{k=0}^K a_k \sum_{l=0}^L b_l^{(k)}\sin^l\left(\frac{x\pi}{2}\right)$ *where* $L := \lceil\delta^{-2}\ln(4\|\mathbf{a}\|_1\epsilon^{-1})\rceil$*, then we have that* $\|\hat{f}_1 - \hat{f}_2\|_\infty \le \epsilon/4$*.*

Proof. We truncate the summation over l in $f_1(x)$ at $l = L$, and it suffices to bound the truncation error. For all $k \in \mathbb{N}$ and $x \in [-1+\delta, 1-\delta]$, we obtain the error bound:

$$\left|\sum_{l=\lfloor L\rfloor}^{\infty} b_l^{(k)} \sin^l\left(\tfrac{x\pi}{2}\right)\right| \le \sum_{l=\lfloor L\rfloor}^{\infty} b_l^{(k)}\left|\sin^l\left(\tfrac{x\pi}{2}\right)\right| \le \sum_{l=\lfloor L\rfloor}^{\infty} b_l^{(k)} |1-\delta^2|^l \le (1-\delta^2)^L \sum_{l=\lfloor L\rfloor}^{\infty} b_l^{(k)} \le (1-\delta^2)^L.$$

Here, the second inequality owing to $\forall\delta \in [0,1]$, $\sin\big((1-\delta)\frac{\pi}{2}\big) \le 1-\delta^2$, and the last inequality is due to$\|\mathbf{b}^{(k)}\|_1 = 1$ in Proposition 3.9.1. By appropriately choosing $L := \delta^{-2}\ln(4\|\mathbf{a}\|_1\epsilon^{-1})$, we obtain that $\|\hat{f}_1 - \hat{f}_2\|_\infty \le \sum_{k=0}^{K} a_k(1-\delta^2)^L \le \|\mathbf{a}\|_1 \cdot \exp(-\delta^2 L) \le \epsilon/4$. $\square$

Proposition 3.9.3 (Third approximation)**.** *Let $\hat{f}_3(x)$ be polynomial approximations of f that depends on the parity of f such that $\|\hat{f}_2 - \hat{f}_3\| \le \epsilon/2$ and $M = \lfloor \delta^{-1}\ln(4\|\mathbf{a}\|_1\epsilon^{-1})\rfloor$, then we have*

$$\hat{f}_3^{(\mathrm{even})}(x) := \sum_{k=0}^{K} a_k \sum_{\hat{l}=0}^{L/2} (-1)^{\hat{l}} 2^{-2\hat{l}} b_{2\hat{l}}^{(k)} \sum_{m'=\hat{l}-M}^{\hat{l}+M} (-1)^{m'} \binom{2\hat{l}}{m'} \cos(\pi x(m'-\hat{l})),$$

$$\hat{f}_3^{(\mathrm{odd})}(x) := \sum_{k=0}^{K} a_k \sum_{\hat{l}=0}^{(L-1)/2} (-1)^{\hat{l}+1} 2^{-2\hat{l}-1} b_{2\hat{l}+1}^{(k)} \sum_{m'=\hat{l}+1-M}^{\hat{l}+1+M} (-1)^{m'} \binom{2\hat{l}+1}{m'} \sin\big(\pi x\big(m'-\hat{l}-\tfrac{1}{2}\big)\big).$$

Therefore, we have that $\hat{f}_3(x) := \hat{f}_3^{(\mathrm{even})}(x)$ if f is even, whereas $\hat{f}_3(x) := \hat{f}_3^{(\mathrm{odd})}(x)$ if f is odd. In addition, if f is neither even or odd, then $\hat{f}_3(x) := \hat{f}_3^{(\mathrm{even})}(x) + \hat{f}_3^{(\mathrm{odd})}(x)$.

Proof. We upper-bound $\sin^l(x)$ in $\hat{f}_2(x)$ defined in Proposition 3.9.2 using a tail bound of binomial coefficients. We obtain that $\sin^l(z) = \big(\frac{e^{-\mathrm{i}z}-e^{\mathrm{i}z}}{-2\mathrm{i}}\big)^l = \big(\frac{\mathrm{i}}{2}\big)^l \sum_{m=0}^{l} \exp(\mathrm{i}z(2m-l))$ by a direct calculation, which implies the counterpart for real-valued functions:

$$\sin^l(z) = \begin{cases} 2^{-l}(-1)^{(l+1)/2} \sum_{m'=0}^{l} (-1)^{m'} \binom{l}{m'} \sin(z(2m'-l)), & \text{if } l \text{ is odd;} \\ 2^{-l}(-1)^{l/2} \sum_{m'=0}^{l} (-1)^{m'} \binom{l}{m'} \cos(z(2m'-l)), & \text{if } l \text{ is even.} \end{cases} \tag{A.2}$$

Recall that the Chernoff bound (e.g., Corollary A.1.7 [AS16]) which corresponds a tail bound of binomial coefficients, and assume that $l \le L$, we have derived that:

$$\sum_{m'=0}^{\lfloor l/2\rfloor - M} 2^{-l}\binom{l}{m'} = \sum_{m'=\lceil l/2\rceil + M}^{l} 2^{-l}\binom{l}{m'} \le e^{-\frac{2M^2}{l}} \le e^{-\frac{2M^2}{L}} \le \left(\frac{\epsilon}{4\|a\|_1}\right)^2 \le \frac{\epsilon}{4\|a\|_1}. \tag{A.3}$$

Here, we choose $M = \lceil \delta^{-1}\ln(4\|\mathbf{a}\|_1\epsilon^{-1})\rceil$, and the last inequality is because of the assumption $\epsilon \le 2\|\mathbf{a}\|_1$. As stated in Proposition 3.9.1, $b_l^{(k)} = 0$ if k and l have different parities. Consequently, we only need to consider all odd (resp., even) $l \le L$ for odd (resp., even) functions. If the function f is neither even nor odd, we must consider all $l \le L$. Plugging Equation A.3 into Equation A.2, we can derive that:

$$\begin{aligned} &\text{If } l \text{ is odd, } \left\|\sin^l(z) - 2^{-l}(-1)^{(l+1)/2} \sum_{m'=(l+1)/2-M}^{(l+1)/2+M} (-1)^{m'}\binom{l}{m'} \sin(z(2m'-l))\right\|_\infty \le \frac{\epsilon}{2\|\mathbf{a}\|_1}; \\ &\text{If } l \text{ is even, } \left\|\sin^l(z) - 2^{-l}(-1)^{l/2} \sum_{m'=l/2-M}^{l/2+M} (-1)^{m'}\binom{l}{m'} \cos(z(2m'-l))\right\|_\infty \le \frac{\epsilon}{2\|\mathbf{a}\|_1}; \end{aligned} \tag{A.4}$$

Plugging Equation (A.4) into $\hat{f}_2(x)$, and substituting $z = x\pi/2$, this equation leads to $\hat{f}_3(x)$ as desired. In addition, combining $\sum_{k=0}^{K} |a_k| \sum_{l=0}^{\lfloor L\rfloor} |b_l^{(k)}| \le \sum_{k=0}^{K} |a_k| = \|\mathbf{a}\|_1$ with Equation (A.4), we achieve that $\|\hat{f}_2 - \hat{f}_3\|_\infty \le \epsilon/2$. $\square$

A.1.3 Omitted proof in Theorem 3.8

Fact 3.8.2. *Consider the integrand* $F_k(\theta)=\sum_{m=-M}^{M}\frac{c_m}{2}\big(H_{k,m}^{(+)}-H_{k,m}^{(-)}\big)$ *for any function* f *which is either even or odd. If* f *is even, we have that* $c_m=c_m^{(\mathrm{even})}$ *defined in Lemma 3.9, and*

$$H_{k,m}^{(\pm)}(\theta):=\cos\Big(\pi m\Big(\frac{\cos\theta-x_0}{r+\delta}\Big)\Big)\cdot\cos(k\theta)\cdot\operatorname{erf}\Big(\kappa\Big(\cos\theta-x_0\pm r\pm\frac{\delta}{4}\Big)\Big). \tag{3.8}$$

Likewise, if f *is odd, we know that* $c_m=c_m^{(\mathrm{odd})}$ *defined in Lemma 3.9, and*

$$H_{k,m}^{(\pm)}(\theta):=\sin\Big(\pi\Big(m+\frac{1}{2}\Big)\Big(\frac{\cos\theta-x_0}{r+\delta}\Big)\Big)\cdot\cos(k\theta)\cdot\operatorname{erf}\Big(\kappa\Big(\cos\theta-x_0\pm r\pm\frac{\delta}{4}\Big)\Big). \tag{3.9}$$

Moreover, the integrand is $F_k(\theta)=\sum_{m=-M}^{M}\Big(\frac{c_m^{(\mathrm{even})}}{2}\big(\hat{H}_{k,m}^{(+)}-\hat{H}_{k,m}^{(-)}\big)+\frac{c_m^{(\mathrm{odd})}}{2}\big(\tilde{H}_{k,m}^{(+)}-\tilde{H}_{k,m}^{(-)}\big)\Big)$ *when* f *is neither even nor odd, where* $\hat{H}_{k,m}^{(\pm)}$ *and* $\tilde{H}_{k,m}^{(\pm)}$ *follow from Equation (3.8) and Equation (3.9), respectively. Regardless of the parity of* f*, we have that the second derivative* $F_k''(\theta)\leq O(Bd^3)$*.*

Proof. We begin by deriving an upper bound of the second derivative of the integrand $F_k(\theta)$:

$$|F_k''(\theta)|\leq\sum_{m=-M}^{M}\frac{c_m}{2}\Big|\frac{\mathrm{d}^2}{\mathrm{d}\theta^2}H_{k,m}^{(+)}(\theta)-\frac{\mathrm{d}^2}{\mathrm{d}\theta^2}H_{k,m}^{(-)}(\theta)\Big|\leq\frac{\|\mathbf{c}\|}{2}\max_{-\pi\leq\theta\leq0}\Big(\Big|\frac{\mathrm{d}^2}{\mathrm{d}\theta^2}H_{k,m}^{(+)}(\theta)\Big|+\Big|\frac{\mathrm{d}^2}{\mathrm{d}\theta^2}H_{k,m}^{(-)}(\theta)\Big|\Big). \tag{A.5}$$

By a straightforward calculation, we have the second derivatives of $H_{k,m}^{\pm}(\theta)$ if f is even:

$$\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}\theta^2}H_{k,m}^{(\pm)}(\theta)=&-k^2\cos(k\theta)\cos\big(\tfrac{\pi m(\cos\theta-x_0)}{\delta+r}\big)\operatorname{erf}\big(\kappa\big(\cos\theta-x_0\mp r\mp\tfrac{\delta}{4}\big)\big)\\
&-\tfrac{\pi^2m^2}{(\delta+r)^2}\sin^2(\theta)\cos(k\theta)\cos\big(\tfrac{\pi m(\cos\theta-x_0)}{\delta+r}\big)\operatorname{erf}\big(\kappa\big(\cos\theta-x_0\mp r\mp\tfrac{\delta}{4}\big)\big)\\
&+\tfrac{\pi m}{\delta+r}\cos\theta\cos(k\theta)\sin\big(\tfrac{\pi m(\cos\theta-x_0)}{\delta+r}\big)\operatorname{erf}\big(\kappa\big(\cos\theta-x_0\mp r\mp\tfrac{\delta}{4}\big)\big)\\
&-\tfrac{2\pi km}{\delta+r}\sin(\theta)\sin(k\theta)\sin\big(\tfrac{\pi m(\cos\theta-x_0)}{\delta+r}\big)\operatorname{erf}\big(\kappa\big(\cos\theta-x_0\mp r\mp\tfrac{\delta}{4}\big)\big)\\
&-\tfrac{2\kappa}{\sqrt{\pi}}\cos\theta\cos(k\theta)\cos\big(\tfrac{\pi m(\cos\theta-x_0)}{\delta+r}\big)e^{-\kappa^2\left(\cos\theta-x_0\mp r\mp\frac{\delta}{4}\right)^2}\\
&-\tfrac{4\sqrt{\pi}\kappa m}{\delta+r}\sin^2(\theta)\cos(k\theta)\sin\big(\tfrac{\pi m(\cos\theta-x_0)}{\delta+r}\big)e^{-\kappa^2\left(\cos\theta-x_0\mp r\mp\frac{\delta}{4}\right)^2}\\
&+\tfrac{4\kappa k}{\sqrt{\pi}}\sin(\theta)\sin(k\theta)\cos\big(\tfrac{\pi m(\cos\theta-x_0)}{\delta+r}\big)e^{-\kappa^2\left(\cos\theta-x_0\mp r\mp\frac{\delta}{4}\right)^2}\\
&-\tfrac{4\kappa^3}{\sqrt{\pi}}\sin^2(\theta)\cos(k\theta)\cos\big(\tfrac{\pi m(\cos\theta-x_0)}{\delta+r}\big)\big(\cos\theta-x_0\mp r\mp\frac{\delta}{4}\big)e^{-\kappa^2\left(\cos\theta-x_0\mp r\mp\frac{\delta}{4}\right)^2}.
\end{aligned}$$

Note that all functions appear in $\frac{\mathrm{d}^2}{\mathrm{d}\theta^2}H_{k,m}^{(\pm)}(\theta)$, viz. $\sin x$, $\cos x$, $\exp(-x^2)$, and $\operatorname{erf}(x)$, are at most 1, as well as $|x_0\pm r\pm\delta/4|\leq 7/2$, then we obtain that

$$\begin{aligned}
\big|\tfrac{\mathrm{d}^2}{\mathrm{d}\theta^2}H_{k,m}^{(\pm)}(\theta)\big|&\leq k^2+\tfrac{2\kappa}{\sqrt{\pi}}+\tfrac{4\kappa k}{\sqrt{\pi}}+\tfrac{18\kappa^3}{\sqrt{\pi}}+m\cdot\big(\tfrac{\pi}{\delta+r}+\tfrac{2\pi k}{\delta+r}+\tfrac{4\sqrt{\pi}\kappa}{\delta+r}\big)+m^2\cdot\tfrac{\pi^2}{(\delta+r)^2}\\
&\leq d^2+O(d)+O(d^2)+O(d^3)+\tfrac{M}{\delta+r}\cdot(O(1)+O(d)+O(d))+M^2\cdot\tfrac{O(1)}{(\delta+r)^2}\\
&=O(d^3).
\end{aligned} \tag{A.6}$$

Here, the second line according to $k\leq d$ and $\kappa\leq O(d)$, also the last line is due to facts that $M\leq O(rd)$ and $1/2\leq r/(\delta+r)\leq 1$ if $0<\delta\leq r$ and $0<r\leq 2$. Additionally, a similar argument shows that the upper bound in Equation (A.6) applies to odd functions and functions that are neither even nor odd as well. This is because a direct computation yields the second

derivatives of $H^{(\pm)}_{k,m}(\theta)$ when f is odd:

$$
\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}\theta^2}H^{(\pm)}_{k,m}(\theta) = &- k^2\cos(kx)\sin\Big(\tfrac{\pi\left(m+\frac{1}{2}\right)(\cos(x)-x_0)}{\delta+r}\Big)\,\mathrm{erf}\left(\kappa\Big(\cos(x)-x_0\mp r\mp\tfrac{\delta}{4}\Big)\right)\\
&- \tfrac{\pi\left(m+\frac{1}{2}\right)}{\delta+r}\cos(x)\cos(kx)\cos\Big(\tfrac{\pi\left(m+\frac{1}{2}\right)(\cos(x)-x_0)}{\delta+r}\Big)\,\mathrm{erf}\left(\kappa\Big(\cos(x)-x_0\mp r\mp\tfrac{\delta}{4}\Big)\right)\\
&- \tfrac{\pi^2\left(m+\frac{1}{2}\right)^2}{(\delta+r)^2}\sin^2(x)\cos(kx)\sin\Big(\tfrac{\pi\left(m+\frac{1}{2}\right)(\cos(x)-x_0)}{\delta+r}\Big)\,\mathrm{erf}\left(\kappa\Big(\cos(x)-x_0\mp r\mp\tfrac{\delta}{4}\Big)\right)\\
&+ \tfrac{2\pi k\left(m+\frac{1}{2}\right)}{\delta+r}\sin(x)\sin(kx)\cos\Big(\tfrac{\pi\left(m+\frac{1}{2}\right)(\cos(x)-x_0)}{\delta+r}\Big)\,\mathrm{erf}\left(\kappa\Big(\cos(x)-x_0\mp r\mp\tfrac{\delta}{4}\Big)\right)\\
&+ \tfrac{4\sqrt{\pi}\kappa\left(m+\frac{1}{2}\right)}{\delta+r}\sin^2(x)\cos(kx)\cos\Big(\tfrac{\pi\left(m+\frac{1}{2}\right)(\cos(x)-x_0)}{\delta+r}\Big)e^{-\kappa^2\left(\cos(x)-x_0\mp r\mp\frac{\delta}{4}\right)^2}\\
&- \tfrac{2\kappa}{\sqrt{\pi}}\cos(x)\cos(kx)\sin\Big(\tfrac{\pi\left(m+\frac{1}{2}\right)(\cos(x)-x_0)}{\delta+r}\Big)e^{-\kappa^2\left(\cos(x)-x_0\mp r\mp\frac{\delta}{4}\right)^2}\\
&+ \tfrac{4\kappa k}{\sqrt{\pi}}\sin(x)\sin(kx)\sin\Big(\tfrac{\pi\left(m+\frac{1}{2}\right)(\cos(x)-x_0)}{\delta+r}\Big)e^{-\kappa^2\left(\cos(x)-x_0\mp r\mp\frac{\delta}{4}\right)^2}\\
&- \tfrac{4\kappa^3}{\sqrt{\pi}}\sin^2(x)\cos(kx)\Big(\cos(x)-x_0\mp r\mp\tfrac{\delta}{4}\Big)\sin\Big(\tfrac{\pi\left(m+\frac{1}{2}\right)(\cos(x)-x_0)}{\delta+r}\Big)e^{-\kappa^2\left(\cos(x)-x_0\mp r\mp\frac{\delta}{4}\right)^2}.
\end{aligned}
$$

Substituting Equation (A.6) into Equation (A.5), and noticing that the coefficient vector $\|\mathbf{c}^{(\text{even})}+\mathbf{c}^{(\text{odd})}\|_1 \leq B$ regardless of the parity of f, we conclude that $|F''_k(\theta)| \leq O(Bd^3)$. □

A.2 Applying arbitrary polynomials of bitstring indexed encodings

Proposition 3.14.1 (Space-efficient state preparation, adapted from [Zal98, KM01, GR02]). *Given an l-qubit quantum state $|\psi\rangle := \sum_{i=1}^m \sqrt{\hat{y}_i}|i\rangle$, where $l = \lceil \log m \rceil$ and $\hat{y}_i$ are real amplitudes associated with an evaluation oracle* $\mathrm{Eval}(i,\varepsilon)$ *that returns $\hat{y}_i$ up to accuracy ε we can prepare $|\psi\rangle$ up to accuracy ϵ in deterministic time $\tilde{O}(m^2\log(m/\epsilon))$ and space $O(\log(m/\epsilon^2))$, together with m^2 evaluation oracle calls with precision $\varepsilon := O(\epsilon^2/m)$.*

Proof. We follow the analysis presented in [MP16, Section III.A], with a particular focus on the classical computational complexity required for this state preparation procedure. The algorithm for preparing the state $|\psi\rangle$ expresses the weight W_x as a telescoping product, given by

$$\forall x \in \{0,1\}^l,\ W_x = W_{x_1}\cdot\frac{W_{x_1x_2}}{W_{x_1}}\cdot\frac{W_{x_1x_2x_3}}{W_{x_1x_2}}\cdots\frac{W_x}{W_{x_1\cdots x_{n-1}}}\ \text{where } W_x := \textstyle\sum_{y\in\{0,1\}^{l-|x|}}|\langle xy|\psi\rangle|^2. \tag{A.7}$$

To estimate $|\psi\rangle$ up to accuracy ϵ in the ℓ_2 norm, it suffices to approximate each weight W_x up to additive error $\varepsilon := O(\epsilon^2/m)$, as indicated in [MP16, Section III.A]. To compute $W_{x'}$, we need $2^{l-|x'|}$ oracle calls to $\mathrm{Eval}(\cdot,\varepsilon)$. Evaluating all terms in Equation (A.7) requires computing $W_{x_1}, W_{x_1x_2}, \cdots, W_x$ for any $x \in \{0,1\}^l$, which can be achieved by $2^{l-1}+2^{l-2}+\cdots+1 = 2^l$ oracle calls to $\mathrm{Eval}(\cdot,\varepsilon)$. As we need to compute Equation (A.7) for all $x\in\{0,1\}^l$, the overall number of oracle calls to $\mathrm{Eval}(\cdot,\varepsilon)$ is $2^{2l} = m^2$. The remaining computation can be achieved in deterministic time $\tilde{O}(m^2\log(m/\epsilon))$ and space $O(\log(m/\epsilon))$ where the time complexity is because of the iterated integer multiplication. □

Lemma 3.15 (Renormalizing bitstring indexed encoding). *Let U be an (α,a,ϵ)-bitstring indexed encoding of A, where $\alpha > 1$ and $0 < \epsilon < 1$, and A is a linear operator acting on $s(n)$ qubits. We can implement a quantum circuit V, serving as a normalization of U, such that V is a $(1, a+2, 36\epsilon)$-bitstring indexed encoding of A. This implementation requires $O(\alpha)$ uses of U, $U^\dagger$, $\mathrm{C}_\Pi\mathrm{NOT}$, $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$, and $O(\alpha)$ single-qubit gates. Moreover, the description of the resulting quantum circuit can be computed in deterministic time $O(\alpha)$ and space $O(s)$.*

Proof. Following Definition 3.1, we have $\|A-\alpha\tilde{\Pi}U\Pi\| \leq \epsilon$, where $\tilde{\Pi}$ and Π are the corresponding orthogonal projectors. Because U is a $(1,a,\epsilon/\alpha)$-bitstring indexed encoding A/α, we obtain that $\|A/\alpha\| \leq \|U\| + \epsilon/\alpha = 1+\epsilon/\alpha$, equivalently $\|A\| \leq \alpha+\epsilon$.

Adjusting the encoding through a single-qubit rotation. Consider an odd integer $k := 2\lceil \pi(\alpha+1)/2 \rceil + 1 \leq 9\alpha = O(\alpha)$ and $\gamma := (\alpha+\epsilon)\sin(\pi/2k) \leq 1$. We define new orthogonal projectors $\tilde{\Pi}' := \tilde{\Pi} \otimes |0\rangle\langle 0|$ and $\Pi' := \Pi \otimes |0\rangle\langle 0|$, and combine them with $U' = U \otimes R_\gamma$, where $R_\gamma = \begin{pmatrix} \gamma & -\sqrt{1-\gamma^2} \\ \sqrt{1-\gamma^2} & \gamma \end{pmatrix}$. By noting that $\tilde{\Pi}' U' \Pi' = \gamma \tilde{\Pi} U \Pi \otimes |0\rangle\langle 0|$, we deduce that U' is a $(1, a+1, \gamma\epsilon/\alpha)$-bitstring indexed encoding of $\gamma A/\alpha \otimes |0\rangle\langle 0|$, which is consequently a $(1, a+1, 2\gamma\epsilon/\alpha)$-bitstring indexed encoding of $\sin(2\pi/k) \cdot (A \otimes |0\rangle\langle 0|)$. An error bound follows:

$$\left\| \frac{\gamma}{\alpha} A - \sin\left(\frac{\pi}{2k}\right) A \right\| = \left\| \frac{\epsilon}{\alpha} \sin\left(\frac{\pi}{2k}\right) A \right\| \leq \frac{\epsilon}{\alpha} \sin\left(\frac{\pi}{2k}\right)(\alpha+\epsilon) = \frac{\gamma\epsilon}{\alpha}.$$

Renormalizing the encoding via robust oblivious amplitude amplification. We follow the construction in [GSLW18, Theorem 28], full version of [GSLW19], and perform a meticulous analysis on the computational resources. We observe that it suffices to consider $k \geq 3$, as for U' is already a $(1, a+1, 2\gamma\epsilon/\alpha)$-bitstring indexed encoding of $A \otimes |0\rangle\langle 0|$ when $k=1$. Let $\varepsilon := 2\gamma\epsilon/\alpha$, and for simplicity, we first start by considering the case with $\varepsilon = 0$. By Definition 3.1, we have $\tilde{\Pi}' U' \Pi' = \alpha \sin\left(\frac{\pi}{2k}\right) \tilde{\Pi} U \Pi \otimes |0\rangle\langle 0|$. Let $T_k \in \mathbb{R}[x]$ be the degree-k Chebyshev polynomial (of the first kind). By employing Lemma 3.13, we can apply the QSVT associated with T_k to the bitstring indexed encoding U', yielding:

$$\tilde{\Pi}' T_k^{(\mathrm{SV})}(U') \Pi' = \alpha T_k\Big(\sin\left(\frac{\pi}{2k}\right)\Big) \tilde{\Pi} U \Pi \otimes |0\rangle\langle 0| = \cos\left(\frac{k-1}{2}\pi\right) A \otimes |0\rangle\langle 0| = A \otimes |0\rangle\langle 0|.$$

Here, the second equality is due to $T_k\big(\sin\left(\frac{\pi}{2k}\right)\big) = T_k\big(\cos\left(\frac{\pi}{2} - \frac{\pi}{2k}\right)\big) = \cos\left(\frac{k-1}{2}\pi\right)$, and the last equality holds because k is odd.

Next, we move on the case with $\varepsilon > 0$ and restrict it to $\varepsilon \leq 1/3$.[36] Let $A' := \tilde{\Pi}' U' \Pi'$ and $\hat{A} := \gamma A \otimes |0\rangle\langle 0|$, then we have $\|A' - \hat{A}\| \leq \varepsilon$, indicating that $\left\| \frac{A'+\hat{A}}{2} \right\|^2 \leq \frac{4}{9} := \zeta$[37] and $\|A' - \hat{A}\| + \left\| \frac{A'+\hat{A}}{2} \right\|^2 \leq \frac{1}{3} + \frac{4}{9} < 1$. By employing Lemma 3.13, as well as the facts that $\frac{\sqrt{2}}{\sqrt{1-\zeta}} < 2$ and $2k\varepsilon = 4k\gamma\epsilon/\alpha \leq 36\epsilon$, we can construct a $(1, a+2, 36\epsilon)$-bitstring indexed encoding of A, denoted by V.

Finally, we provide the computational resources required for implementing V. As shown in Lemma 3.13, the implementation of V requires $O(\alpha)$ uses of U, $U^\dagger$, $\mathrm{C}_\Pi\mathrm{NOT}$, $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$, and $O(\alpha)$ single-qubit gates. Furthermore, the description of the resulting quantum circuit can be computed in deterministic time $O(\alpha)$ and space $O(s)$. □

A.3 Application: space-efficient error reduction for unitary quantum computations

Lemma 3.20 (Space-efficient singular value discrimination). *Let $0 \leq \alpha < \beta \leq 1$ and $A := \tilde{\Pi} U \Pi$ be a $(1,0,0)$-bitstring indexed encoding where U acts on s qubits and $s(n) \geq \Omega(\log n)$. Consider an unknown quantum state $|\psi\rangle$, with the promise that it is a right singular vector of A with a singular value either above α or below β. We can distinguish the two cases with error probability at most $\varepsilon := O(\epsilon \log d)$ using a degree-d quantum singular value transformation where $d = \frac{\log 1/\epsilon}{\max\{\beta-\alpha, \sqrt{1-\alpha^2}-\sqrt{1-\beta^2}\}}$. Moreover, we can make the error one-sided if $\alpha = 0$ or $\beta = 1$.*
In particular, the implementation requires $O(d^2 \log d)$ uses of U, $U^\dagger$, $\mathrm{C}_\Pi\mathrm{NOT}$, $\mathrm{C}_{\tilde{\Pi}}\mathrm{NOT}$, and $O(d^2 \log d)$ multi-controlled single-qubit gates. Also, we can compute the description of the implementation in deterministic time $\tilde{O}(\varepsilon^{-1} d^{9/2})$ and space $O(s(n))$.

Proof of Lemma 3.20. Given a $(1,0,0)$-bitstring indexed encoding $\tilde{\Pi} U \Pi$ with a singular value decomposition $W\Sigma V^\dagger$. Utilizing Corollary 3.16, it suffices to construct an even polynomial P

[36] If $\varepsilon > 1/3$, then $\|\tilde{\Pi}' U' \Pi' - A \otimes |0\rangle\langle 0|\| \leq 2 = 2 \cdot 3 \cdot \frac{1}{3}$ always holds, implying that we can directly use U' as V.
[37] This is because $\|A' + \hat{A}\| \leq \|A'\| + \|A'\| + \|A' - \hat{A}\| \leq 2\sin(\pi/2k) + \varepsilon \leq 2\sin(\pi/6) + 1/3 = 4/3$.

associated a $(1,\epsilon)$-bitstring indexed encoding U_P such that

$$\begin{aligned}&\Big\|\Pi'_{\geq t+\delta} U_P \Pi_{\geq t+\delta} - I \otimes \sum_{i:\sigma_i \geq t+\delta} |\tilde{\psi}_i\rangle\langle\psi_i|\Big\| \leq \epsilon \\ &\big\|(\langle +| \otimes \Pi'_{\leq t-\delta}) U_P (|+\rangle \otimes \Pi_{\leq t-\delta}) - 0\big\| \leq \epsilon.\end{aligned} \quad \text{where } \Pi' := \begin{cases} \tilde{\Pi}, & \text{if } \beta - \alpha \geq \sqrt{1-\alpha^2} - \sqrt{1-\beta^2} \\ I - \tilde{\Pi}, & \text{otherwise.}\end{cases} \tag{A.8}$$

Here singular value threshold projectors are defined as $\Pi_{\geq\delta} := \Pi V \Sigma_{\geq\delta} V^\dagger \Pi$, so does $\Pi_{\leq\delta}$. Likewise, $\Pi'_{\geq\delta} := \Pi' U \sigma_{\geq\delta} U^\dagger \Pi'$ and so does $\Pi'_{\leq\delta}$. In addition, the definition of Π' in Equation (A.8) in accordance with the proof presented in [GSLW19, Theorem 20].[38]

With the construction of the resulting bitstring indexed encoding $\Pi' U_P \Pi$ for an odd polynomial P, we then apply an ϵ-approximate singular value projector by choosing $t = (\alpha+\beta)/2$ and $\delta = (\beta-\alpha)/2$. Then, we measure $|+\rangle\langle+| \otimes \Pi'$: If the final state is in $\mathrm{Img}(|+\rangle\langle+| \otimes \Pi')$, there exists a singular value σ_i above α (resp., $\sqrt{1-\beta^2}$); Otherwise, all singular value σ_i must be below β (resp., $\sqrt{1-\alpha^2}$). Furthermore, we make the error one-sided since an *odd* quantum singular value transformation always preserves 0 singular values. It is left to implement singular value threshold projectors for an odd polynomial P.

Implementing singular value threshold projectors. We begin by constructing an odd polynomial $P \in \mathbb{R}[x]$ of degree $m = O(\delta^{-1}\log\epsilon^{-2})$, which approximates an odd function $Q(x) := \frac{1}{2}[(1-\epsilon)\cdot\mathrm{sgn}(x+t) + (1-\epsilon)\cdot\mathrm{sgn}(x-t) + 2\epsilon\cdot\mathrm{sgn}(x)]$ on the interval $[-1,1]\backslash(-t-\delta,-t+\delta)\cup(t-\delta,t+\delta)$ with $\epsilon^2/4$ precision. By leveraging the space-efficient odd degree-d polynomial approximation $P_d^{\mathrm{sgn}}(x)$ of the sign function, as specified in Corollary 3.7, we then obtain:

$$P(x) = \tfrac{1}{2}\big[(1-\epsilon)\cdot P_d^{\mathrm{sgn}}(x+t) + (1-\epsilon)\cdot P_d^{\mathrm{sgn}}(x-t) + 2\epsilon \cdot P_d^{\mathrm{sgn}}(x)\big]. \tag{A.9}$$

Hence, we ensure that $|P(x)| \leq 1$ for any $-1 \leq x \leq 1$, and $(-1)^z P(x) \in [0,\epsilon]$ if $(-1)^z x \in [0, t-\delta]$, as well as $(-1)^z P(x) \in [1-\epsilon, 1]$ if $(-1)^z \in [t+\delta, 1]$ for $z \in \{0,1\}$. To achieve the resulting bitstring indexed encoding U_P of $P^{(SV)}(\tilde{\Pi} U \Pi)$ with the desired precision, we apply Corollary 3.16 to $P(x)$ described in Equation (A.9). And then the implementation error of U_P is evidently upper-bounded by $\varepsilon = O(\epsilon \log d)$. □

B Omitted proofs in space-bounded quantum state testing

Theorem B.1. *For any functions $\alpha(n)$ and $\beta(n)$ that can be computed in deterministic logspace and satisfy $\alpha(n) - \beta(n) \geq 1/\mathrm{poly}(n)$, we have that* $\textsc{GapQHS}_{\log}[\alpha(n), \beta(n)]$ *is in* BQL.

Proof. Note that $\mathrm{HS}^2(\rho_0,\rho_1) = \frac{1}{2}\left(\mathrm{Tr}(\rho_0^2) + \mathrm{Tr}(\rho_1^2)\right) - \mathrm{Tr}(\rho_0\rho_1)$. Let $\varepsilon := (\alpha-\beta)/100$. According to Lemma 2.12, we can use the SWAP test to estimate $\mathrm{Tr}(\rho_0^2)$, $\mathrm{Tr}(\rho_1^2)$, and $\mathrm{Tr}(\rho_0\rho_1)$, and hence $\mathrm{HS}^2(\rho_0,\rho_1)$, within additive error ε with high probability by performing $O(1/\varepsilon^2)$ sequential repetitions. Therefore, we can conclude that $\textsc{GapQHS}_{\log}[\alpha(n),\beta(n)]$ is in BQL. □

B.1 Omitted proofs in BQL containments

Proposition 4.10.1. $\Pr\Big[|x - \mathrm{td}(\rho_0,\rho_1)| \leq \hat{C}_{\mathrm{sgn}}\epsilon\log d + \epsilon_H + 2C_{\mathrm{sgn}}\epsilon\log d + 2^{r+1}\delta\Big] > 0.8.$

Proof of Proposition 4.10.1. By applying the triangle inequality, we have obtained the following

[38] By applying [GSLW18, Definition 12] (the full version of [GSLW19]) to $\Pi' := I - \tilde{\Pi}$, we know that $|\psi\rangle$ is a singular vector of $\Pi' U \Pi$ with a singular value of at least $\sqrt{1-a^2}$ in the first case, or with a singular value of at most $\sqrt{1-b^2}$ in the second case. Additionally, in one-sided error scenarios, if $a=0$, then $b - a = b \geq 1 - \sqrt{1-b^2} = \sqrt{1-a^2} - \sqrt{1-b^2}$; while if $b = 1$, then $b - a = 1 - a \leq \sqrt{1-a^2} = \sqrt{1-a^2} - \sqrt{1-b^2}$.

inequality from Equation (4.1), which holds with probability at least $0.9^2 > 0.8$:

$$\begin{aligned}\left|\frac{x_0-x_1}{2}-\mathrm{td}(\rho_0,\rho_1)\right| &= \left|\frac{x_0-x_1}{2}-\mathrm{Tr}\Big(\frac{\rho_0-\rho_1}{2}\mathrm{sgn}\Big(\frac{\rho_0-\rho_1}{2}\Big)\Big)\right|\\ &\leq \left|\frac{x_0-x_1}{2}-\mathrm{Tr}\Big(\frac{\rho_0-\rho_1}{2}P_d^{\mathrm{sgn}}\Big(\frac{\rho_0-\rho_1}{2}\Big)\Big)\right|\\ &\quad+\left|\mathrm{Tr}\Big(\frac{\rho_0-\rho_1}{2}P_d^{\mathrm{sgn}}\Big(\frac{\rho_0-\rho_1}{2}\Big)\Big)-\mathrm{Tr}\Big(\frac{\rho_0-\rho_1}{2}\mathrm{sgn}\Big(\frac{\rho_0-\rho_1}{2}\Big)\Big)\right|.\end{aligned}$$

For the first term, by noting the QSVT implementation error in Corollary 3.16, we know that

$$\left|\frac{x_0-x_1}{2}-\mathrm{Tr}\Big(\frac{\rho_0-\rho_1}{2}P_d^{\mathrm{sgn}}\Big(\frac{\rho_0-\rho_1}{2}\Big)\Big)\right| \leq \hat{C}_{\mathrm{sgn}}\epsilon\log d+\epsilon_H. \tag{B.1}$$

For the second term, let $\frac{\rho_0-\rho_1}{2}=\sum_j\lambda_j|\psi_j\rangle\langle\psi_j|$, where $\{|\psi_j\rangle\}$ is an orthonormal basis. Then,

$$\left|\mathrm{Tr}\Big(\frac{\rho_0-\rho_1}{2}P_d^{\mathrm{sgn}}\Big(\frac{\rho_0-\rho_1}{2}\Big)\Big)-\mathrm{Tr}\Big(\frac{\rho_0-\rho_1}{2}\mathrm{sgn}\Big(\frac{\rho_0-\rho_1}{2}\Big)\Big)\right| \leq \sum_j\left|\lambda_jP_d^{\mathrm{sgn}}(\lambda_j)-\lambda_j\mathrm{sgn}(\lambda_j)\right|. \tag{B.2}$$

We split the summation over j into three separate summations: $\sum_j=\sum_{\lambda_j<-\delta}+\sum_{\lambda_j>\delta}+\sum_{-\delta\leq\lambda_j\leq\delta}$. By noticing the approximation error of P_d^{sgn} in Corollary 3.7, we can then obtain the following results for each of the three summations:

$$\begin{aligned}\sum_{\lambda>\delta}\left|\lambda_jP_d^{\mathrm{sgn}}(\lambda_j)-\lambda_j\mathrm{sgn}(\lambda_j)\right| &= \sum_{\lambda>\delta}|\lambda_j|\left|P_d^{\mathrm{sgn}}(\lambda_j)-1\right| \leq \sum_{\lambda>\delta}|\lambda_j|C_{\mathrm{sgn}}\epsilon\log d\leq C_{\mathrm{sgn}}\epsilon\log d,\\ \sum_{\lambda<-\delta}\left|\lambda_jP_d^{\mathrm{sgn}}(\lambda_j)-\lambda_j\mathrm{sgn}(\lambda_j)\right| &= \sum_{\lambda<-\delta}|\lambda_j|\left|P_d^{\mathrm{sgn}}(\lambda_j)+1\right| \leq \sum_{\lambda<-\delta}|\lambda_j|C_{\mathrm{sgn}}\epsilon\log d\leq C_{\mathrm{sgn}}\epsilon\log d,\end{aligned}$$

$$\sum_{-\delta\leq\lambda_j\leq\delta}\left|\lambda_jP_d^{\mathrm{sgn}}(\lambda_j)-\lambda_j\mathrm{sgn}(\lambda_j)\right| \leq \sum_{-\delta\leq\lambda_j\leq\delta}2|\lambda_j|\leq 2^{r+1}\delta.$$

Hence, we derive the following inequality by summing over the aforementioned three inequalities:

$$\sum_j\left|\lambda_jP_d^{\mathrm{sgn}}(\lambda_j)-\lambda_j\mathrm{sgn}(\lambda_j)\right| \leq 2^{r+1}\delta+2C_{\mathrm{sgn}}\epsilon\log d. \tag{B.3}$$

By combining Equation (B.1), Equation (B.2), and Equation (B.3), we conclude that

$$\left|\frac{x_0-x_1}{2}-\mathrm{td}(\rho_0,\rho_1)\right| \leq \hat{C}_{\mathrm{sgn}}\epsilon\log d+\epsilon_H+2C_{\mathrm{sgn}}\epsilon\log d+2^{r+1}\delta. \qquad \square$$

Proposition 4.11.1. *The following inequality holds for $i\in\{0,1\}$:*

$$\Pr\Big[\left|x_i\ln\left(\tfrac{2}{\beta}\right)-\mathrm{S}(\rho_i)\right| \leq 2\ln\left(\tfrac{1}{\beta}\right)\Big(\hat{C}_{\ln}\epsilon d+\epsilon_H+C_{\ln}\epsilon\log d+2^{r+1}\beta\Big)\Big] \geq \tfrac{9}{10}.$$

Proof of Proposition 4.11.1. We only prove the case with $i=0$ while the case with $i=1$ follows straightforwardly. By applying the triangle inequality on Equation (4.2) with $i=0$, we have:

$$|x_0\ln(2/\beta)-\mathrm{S}(\rho_0)| = \left|x_0\ln(2/\beta)-\ln(2/\beta)\mathrm{Tr}\left(P_d^{\ln}(\rho_0)\,\rho_0\right)\right|+\left|\ln(2/\beta)\mathrm{Tr}\left(P_d^{\ln}(\rho_0)\,\rho_0\right)-\mathrm{S}(\rho_0)\right|.$$

For the first term, by noting the QSVT implementation error in Corollary 3.17, we have

$$\left|x_0\ln(2/\beta)-\ln(2/\beta)\mathrm{Tr}\left(P_d^{\ln}(\rho_0)\,\rho_0\right)\right| \leq 2\ln(1/\beta)\left(\hat{C}_{\ln}\epsilon d+\epsilon_H\right). \tag{B.4}$$

For the second term, let $\rho_0=\sum_j\lambda_j|\psi_j\rangle\langle\psi_j|$, where $\{|\psi_j\rangle\}$ is an orthonormal basis. Then,

$$\left|\ln(2/\beta)\mathrm{Tr}\left(P_d^{\ln}(\rho_0)\,\rho_0\right)-\mathrm{S}(\rho_0)\right| \leq \sum_j\left|\lambda_jP_d^{\ln}(\lambda_j)\ln(2/\beta)-\lambda_j\ln(1/\lambda_j)\right|. \tag{B.5}$$

We split the summation over j into two separate summations: $\sum_j=\sum_{\lambda_j>\delta}+\sum_{\lambda\leq\delta}$. By noticing the approximation error of $P_d^{\ln}$ in Corollary 3.10, e can then obtain the following results for each of the two summations:

$$\begin{aligned}\sum_{\lambda_j>\delta}\left|\lambda_j P_d^{\ln}(\lambda_j)\ln(2/\beta)-\lambda_j\ln(1/\lambda_j)\right| &= \sum_{\lambda_j>\delta}|\lambda_j|\left|P_d^{\ln}(\lambda_j)\ln(2/\beta)-\ln(1/\lambda_j)\right|\\ &\leq \sum_{\lambda_j>\delta}|\lambda_j| C_{\ln}\epsilon\log d\ln(2/\beta)\\ &\leq 2C_{\ln}\epsilon\log d\ln(1/\beta),\end{aligned}$$

$$\sum_{\lambda_j\leq\delta}\left|\lambda_j P_d^{\ln}(\lambda_j)\ln(2/\beta)-\lambda_j\ln(1/\lambda_j)\right| \leq \sum_{\lambda_j\leq\delta}\left(|\lambda_j|\ln(2/\beta)+|\lambda_j|\ln(1/\beta)\right)\leq 2^{r+2}\delta\ln(1/\beta).$$

Hence, we have derived the following inequality by summing over the aforementioned inequalities:

$$\sum_{j}\left|\lambda_j P_d^{\ln}(\lambda_j)\ln(2/\beta)-\lambda_j\ln(1/\lambda_j)\right| \leq 2C_{\ln}\epsilon\log d\ln(1/\beta)+2^{r+2}\delta\ln(1/\beta). \tag{B.6}$$

By combining Equation (B.4), Equation (B.5), and Equation (B.6), we conclude that

$$|x_0\ln(2/\beta)-\mathrm{S}(\rho_0)|\leq 2\ln(1/\beta)\left(\hat{C}_{\ln}\epsilon d+\epsilon_H+C_{\ln}\epsilon\log d+2^{r+1}\beta\right). \qquad \square$$

B.2 Omitted proofs in $\mathsf{coRQ_UL}$ containments

Proposition 4.14.1. *Let $f(\theta_0,\theta_1):=\sin^2(3\theta_0)\sin^2(3\theta_1)$ be a function such that $\sin^2(\theta_i)=p_i/2$ for $i\in\{0,1\}$ and $\max\{|p_0-1/2|,|p_1-1/2|\}\geq\varepsilon/2$, then $f(\theta_0,\theta_1)\leq 1-\varepsilon^2/4$.*

Proof. We begin by stating the facts that $\sin^2(\theta_i)=p_i/2$ for $i\in\{0,1\}$ and $\sin^2(3\theta)=\sin^6(\theta)-6\cos^2(\theta)\sin^4(\theta)+9\cos^4(\theta)\sin^2(\theta)$. Then we notice that $0\leq p_0,p_1\leq 1$ and complete the proof by a direct calculation:

$$\begin{aligned}f(\theta_0,\theta_1)&=\left(2p_0^3-6p_0^2+\tfrac{9}{2}p_0\right)\left(2p_1^3-6p_1^2+\tfrac{9}{2}p_1\right)\\&\leq\left(1-\left(p_0-\tfrac{1}{2}\right)^2\right)\left(1-\left(p_1-\tfrac{1}{2}\right)^2\right)\\&\leq 1-\left(\max\left\{\left|p_0-\tfrac{1}{2}\right|,\left|p_1-\tfrac{1}{2}\right|\right\}\right)^2\\&\leq 1-\tfrac{\varepsilon^2}{4}.\end{aligned}$$

$\square$

Proposition 4.15.1. *The polynomial function $f(x):=16x^3-24x^2+9x$ is monotonically decreasing in $[1/4,9/16]$. Moreover, we have $f\left(\left(\frac{1}{2}+\frac{\alpha}{4}\right)^2\right)\leq 1-\frac{\alpha^2}{2}$ for any $0\leq\alpha\leq 1$.*

Proof. Through a direct calculation, we have $f'(x)=48x^2-48x+9\leq 0$ for $x\in[1/4,3/4]$, then $f(x)$ is monotonically decreasing in $[1/4,9/16]\subseteq[1/4,3/4]$. Moreover, it is left to show that:

$$f\left(\left(\tfrac{1}{2}+\tfrac{\alpha}{4}\right)^2\right)=\tfrac{\alpha^6}{256}+\tfrac{3\alpha^5}{64}+\tfrac{9\alpha^4}{64}-\tfrac{\alpha^3}{8}-\tfrac{3\alpha^2}{4}+1\leq 1-\tfrac{\alpha^2}{2}.$$

Equivalently, it suffices to show that $g(x):=-\frac{x^4}{256}-\frac{3x^3}{64}-\frac{9x^2}{64}+\frac{x}{8}+\frac{1}{4}\geq 0$ for $0\leq x\leq 1$. We first compute the first derivative of $g(x)$, which is $g'(x)=-\frac{x^3}{64}-\frac{9x^2}{64}-\frac{9x}{32}+\frac{1}{8}$. Setting $g'(x)$ equal to zero, we obtain three roots: $x_1=-4$, $x_2=\frac{1}{2}(-\sqrt{33}-5)<0$, and $x_3=\frac{1}{2}(\sqrt{33}-5)\in(0,1)$.

Since $g'(0)=1/8>0$ and $g'(1)=-5/16<0$, we conclude that $g(x)$ is monotonically increasing in $[0,x_3]$ and monotonically decreasing in $[x_3,1]$. Therefore, we can determine the minimum value of $g(x)$ by evaluating $g(0)=\frac{1}{4}$ and $g(1)=\frac{47}{256}$. Since both values are greater than zero, we conclude that $\min\{g(0),g(1)\}=\left\{\frac{1}{4},\frac{47}{256}\right\}>0$, as desired. $\square$

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il