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 $\rho_0$ and $\rho_1$, respectively, with access to their “source code”. Our goal is to decide whether $\rho_0$ is $\epsilon_1$-close to or $\epsilon_2$-far from $\rho_1$ 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: QishengWang1994@gmail.com
Contents

1 Introduction .................................................. 1
   1.1 Main results ................................................. 1
   1.2 Background on space-bounded quantum computation ............... 3
   1.3 Time-bounded and space-bounded distribution and state testing .... 4
      1.3.1 Time-bounded distribution and state testing .................. 5
      1.3.2 Space-bounded distribution and state testing ................. 6
   1.4 Proof technique: Space-efficient quantum singular value transformation .... 7
   1.5 Proof overview: A general framework for quantum state testing .......... 9
   1.6 Discussion and open problems .................................. 10
   1.7 Related works: more on quantum state testing problems .............. 11

2 Preliminaries ................................................ 11
   2.1 Distances and divergences for quantum states .................... 11
   2.2 Space-bounded quantum computation ................................ 12
   2.3 Near-minimax approximation by Chebyshev interpolation .......... 14
   2.4 Tools for space-bounded randomized and quantum algorithms ........ 14

3 Space-efficient quantum singular value transformations .................... 16
   3.1 Space-efficient bounded polynomial approximations .................. 17
      3.1.1 Bounded functions ......................................... 17
      3.1.2 Piecewise-smooth functions .................................. 20
   3.2 Applying Chebyshev interpolation to bitstring indexed encodings ...... 25
   3.3 Examples: the sign function and the normalized logarithmic function .... 28
   3.4 Application: space-efficient error reduction for unitary quantum computations ... 29

4 Space-bounded quantum state testing ................................ 30
   4.1 Space-bounded quantum state testing: a general framework .......... 32
   4.2 GAPQSDlog is in BQL ......................................... 33
   4.3 GAPQEDlog and GAPQJSlog are in BQL .......................... 35
   4.4 CERTQSDlog and CERTQHSlog are in coRQUL ...................... 37
      4.4.1 CERTQSDlog is in coRQUL .................................. 38
      4.4.2 CERTQHSlog is in coRQUL .................................. 39
   4.5 BQL- and coRQUL-hardness for space-bounded state testing problems ... 41
      4.5.1 Hardness results for GAPQSDlog, GAPQHSSlog, and their certification version 41
      4.5.2 Hardness results for GAPQJSlog and GAPQEDlog ............... 43

A Omitted proofs in space-efficient QSVT .................................. 50

B Omitted proofs in space-bounded quantum state testing ...................... 56
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 $\rho_0$, but a possibly malicious party could provide another quantum device that outputs a different $n$-qubit (mixed) state $\rho_1$, claiming that $\rho_0 \approx_\epsilon \rho_1$. The problem of testing whether $\rho_0$ is $\epsilon_1$-close to or $\epsilon_2$-far from $\rho_1$ with respect to a specified distance-like measure, given the ability to produce copies of $\rho_0$ and $\rho_1$, is known as quantum state testing [MdW16, Section 4]. Quantum state testing (resp., distribution testing) typically involves utilizing sample accesses to quantum states $\rho_0$ and $\rho_1$ (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 $\rho_0$ and $\rho_1$ 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 coRQUL-complete promise problems since Watrous [Wat01] introduced unitary RQL and coRQL (known as RQL and coRQL, 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 $\text{poly}(n)$ 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, coRQL, 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 coRQUL-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 coRQUL-complete: for any α(n) ≥ 1/poly(n), decide whether

1. \( \text{CERTQSD}_{\log}^{}: \rho_0 = \rho_1 \) or \( \text{td}(\rho_0, \rho_1) \geq \alpha(n) \);
2. \( \text{CERTQHS}_{\log}^{}: \rho_0 = \rho_1 \) or \( \text{HS}^2(\rho_0, \rho_1) \geq \alpha(n) \);

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 \( S(\rho_0) - S(\rho_1) \), 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 \( \alpha(n) - \beta(n) \geq 1/poly(n) \), or for any \( g(n) \geq 1/poly(n) \), decide whether

1. \( \text{GAPQSD}_{\log}^{}: \text{td}(\rho_0, \rho_1) \geq \alpha(n) \) or \( \text{td}(\rho_0, \rho_1) \leq \beta(n) \);
2. \( \text{GAPQED}_{\log}^{}: S(\rho_0) - S(\rho_1) \geq g(n) \) or \( S(\rho_1) - S(\rho_0) \geq g(n) \);
3. \( \text{GAPQJS}_{\log}^{}: \text{QJS}_2(\rho_0, \rho_1) \geq \alpha(n) \) or \( \text{QJS}_2(\rho_0, \rho_1) \leq \beta(n) \);
4. \( \text{GAPQHS}_{\log}^{}: \text{HS}^2(\rho_0, \rho_1) \geq \alpha(n) \) or \( \text{HS}^2(\rho_0, \rho_1) \leq \beta(n) \);

Notably, Theorem 1.2(1) demonstrates that our algorithm for GAPQSD_{\log}^{} exhibits a polynomial advantage in space over the best-known classical algorithms [Wat02], since Watrous implicitly showed in [Wat02, Proposition 21] that GAPQSD_{\log}^{} is contained in (classical) poly-logarithmic 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 coRQUL), 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 CERTQHS_{\log}^{}, but also additionally requires novel techniques for CERTQSD_{\log}^{}. On the other hand, for two-sided error scenarios, while showing the containment is straightforward for GAPQHS_{\log}^{}, it still demands sophisticated techniques for all other problems, such as GAPQSD_{\log}^{}, GAPQED_{\log}^{}, and GAPQJS_{\log}^{}.

¹Notably, our algorithm for GAPQSD_{\log}^{} provides an alternating proof for the original statement that \( (\alpha, \beta) \)-QSD is in PSPACE when \( \alpha(n) - \beta(n) \geq \exp(-\text{poly}(n)) \). 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 \( \text{BQSPACE}[s(n)] \) 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 \( \text{BQL} \), or \( \text{BQL} \) if only unitary gates are permitted.

Significantly, several developments over the past two decades have shown that \( \text{BQL} \) 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 \( \text{BQL} \) is also gateset-independent.

- **Error reduction.** Repeating \( \text{BQL} \) sequentially necessitates reusing the workspace, making it unclear how to reduce errors for \( \text{BQL} \) 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, \( \text{BQL} \) appeared to be seemingly more powerful than \( \text{BQL} \); however, we cannot directly demonstrate that \( \text{BPL} \subseteq \text{BQL} \). Recently, Fefferman and Remscrim [FR21] (as well as [GRZ21, GR22]) proved the equivalence between \( \text{BQL} \) and \( \text{BQL} \), indicating a space-efficient approach to eliminating intermediate measurements.

**BQL-complete problems.** Identifying natural complete problems for the class \( \text{BQL} \) is a crucial and intriguing question. Ta-Shma [TS13] proposed the first candidate \( \text{BQL} \)-complete problem, building upon the work of Harrow, Hassidim, and Lloyd [HHL09] which established a \( \text{BQP} \)-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 \( \text{BQL} \). Similarly, computing eigenvalues of an Hermitian matrix is also in \( \text{BQL} \), establishing 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 \( \text{BQL} \)-complete problem by ingeniously utilizing amplitude estimation to avoid intermediate measurements.

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

**RQL and coRQL-complete problems.** Watrous [Wat01] introduced the one-sided error counterpart of \( \text{BQL} \), namely \( \text{RQL} \) and \( \text{coRQL} \), and developed error reduction techniques.
Moreover, Watrous proved that the undirected graph connectivity problem (USTCON) is in \( \text{RQUL} \cap \text{coRQUL} \) whereas Reingold [Rei08] demonstrated that USTCON is in \( \text{L} \) several years later. Recently, Fefferman and Remscrim [FR21] proposed a “verification” version of the well-conditioned iterative matrix product problem (vITMATPROD) as a candidate \( \text{coRQL} \)-complete problem. However, although this problem is known to be \( \text{coRQL} \)-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 and space-bounded distribution and state testing with respect to \( \ell_1 \) norm, entropy difference, and \( \ell_2 \) 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, including the one-sided error counterpart known as quantum state certification [BOW19]. In particular, for distance-like measures such as the \( \ell_1 \) 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 \( \ell_2 \) 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 \( N \).

<table>
<thead>
<tr>
<th>( \ell_1 ) norm</th>
<th>( \ell_2 ) norm</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical</td>
<td>SZK-complete(^5)</td>
<td>BPP-complete</td>
</tr>
<tr>
<td>Time-bounded</td>
<td>[SV03, GSV98]</td>
<td>Folklore</td>
</tr>
<tr>
<td>Quantum</td>
<td>QSZK-complete(^6)</td>
<td>BQP-complete</td>
</tr>
<tr>
<td>Time-bounded</td>
<td>[Wat02, Wat09]</td>
<td>[BCWdW01, RASW23]</td>
</tr>
<tr>
<td>Quantum</td>
<td>BQL-complete</td>
<td>BQL-complete</td>
</tr>
<tr>
<td>Space-bounded</td>
<td>Theorem 1.2(1)</td>
<td>[BCWdW01] and Theorem 1.2(4)</td>
</tr>
</tbody>
</table>

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 \( \ell_1 \) norm and entropy difference, the time-bounded scenario is seemingly much harder than preparing states or distributions since QSZK \( \subseteq \) BQP and SZK \( \subseteq \) BPP are unlikely.

- For \( \ell_2 \) 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.

\( \ell_1 \) 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 \( (\alpha, \beta) \)-SD asks whether \( D_0 \) is \( \alpha \)-far from or \( \beta \)-close to \( D_1 \) with respect to \( \|D_0 - D_1\|_{TV} \). Although sampling from the distribution is in \( \text{BPP} \),\(^4\) testing the closeness between these distributions is \( \text{SZK} \)-complete [SV03, GSV98], where \( \text{SZK} \) is the class of promise problems possessing statistical zero-knowledge proofs. It is noteworthy that the \( \text{SZK} \) containment of \( (\alpha, \beta) \)-SD for any \( \alpha(n) - \beta(n) \geq 1/\text{poly}(n) \) is currently unknown.\(^5\) In addition, we note that \( \text{SZK} \) is contained in \( \text{AM} \cap \text{coAM} \) [For87, AH91].

Following the pioneering work [SV03], Watrous [Wat02] introduced the time-bounded quantum state testing problem, where two quantum states \( \rho_0 \) and \( \rho_1 \) 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, \( (\alpha, \beta) \)-QSD asks whether \( \rho_0 \) is \( \alpha \)-far from or \( \beta \)-close to \( \rho_1 \) with respect to \( \text{td}(\rho_0, \rho_1) \). Analogous to its classical counterpart, QSD is \( \text{QSZK} \)-complete [Wat02, Wat09], whereas the QSZK containment for any \( \alpha(n) - \beta(n) \geq 1/\text{poly}(n) \) remains an open question.\(^6\) Additionally, it is worth noting that \( \text{QIP}(2) \) contains \( \text{QSZK} \) [Wat02, Wat09].

Entropy difference scenarios. Beyond \( \ell_1 \) 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 \( H(D_0) - H(D_1) \geq g \) or \( H(D_1) - H(D_0) \geq g \) for \( g = 1 \). They demonstrated that ED is \( \text{SZK} \)-complete. Ben-Aroya, Schwartz, and Ta-Shma [BASTS10] further investigated the promise problem QUANTUM ENTROPY DIFFERENCE (QED), which asks whether \( S(\rho_0) - S(\rho_1) \geq g \) or \( S(\rho_1) - S(\rho_0) \geq g \), for efficiently preparable quantum states \( \rho_0 \) and \( \rho_1 \) and \( g = 1/2 \). They showed that QED is \( \text{QSZK} \)-complete. Moreover, the \( \text{SZK} \) (resp., \( \text{QSZK} \)) containment for ED (resp., QED) automatically holds for any \( g(n) \geq 1/\text{poly}(n) \).

Furthermore, Berman, Degwekar, Rothblum, and Vasudevan [BDRV19] demonstrated that the Jensen-Shannon divergence problem (JSP), asking whether \( \text{JS}(D_0, D_1) \geq \alpha \) or \( \text{JS}(D_0, D_1) \leq \beta \) for efficiently samplable distributions \( D_0 \) and \( D_1 \), is \( \text{SZK} \)-complete. Their work accomplished this result by reducing the problem to ED, and this containment applies to \( \alpha(n) - \beta(n) \geq 1/\text{poly}(n) \). Recently, Liu [Liu23] showed a quantum counterpart, referred to as the QUANTUM JENSEN-SHANNON DIVERGENCE PROBLEM (QJSP), is \( \text{QSZK} \)-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 \( (\alpha, \beta) \)-SD is in \( \text{SZK} \) 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 \( \alpha^2(n) - \beta(n) \geq 1/\text{poly}(n) \) by utilizing a series of tailor-made reductions. Currently, we only know that \( (\alpha, \beta) \)-SD for \( \alpha(n) - \beta(n) \geq 1/\text{poly}(n) \) is also in \( \text{AM} \cap \text{coAM} \) [BL13].

\(^6\)Like SD and \( \text{SZK} \), the techniques in [Wat02, Wat09] show that \( (\alpha, \beta) \)-QSD is in \( \text{QSZK} \) 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 \( \alpha^2(n) - \sqrt{2\ln \beta(n)} \geq 1/\text{poly}(n) \), but the differences between classical and quantum distances make it challenging to push the bound further. In [Wat02, Proposition 21], Watrous implicitly proved a \( \text{PSPACE} \) upper bound for the parameter regime \( \alpha(n) - \beta(n) \geq \exp(-\text{poly}(n)) \).
Jensen-Shannon divergence is a special instance of the Holevo $\chi$ quantity \cite{Hol73}.

**$\ell_2$ norm scenarios.** For the quantum setting, it is straightforward that applying the SWAP test \cite{BCWdW01} to efficiently preparable quantum states $\rho_0$ and $\rho_1$ can lead to a BQP containment, in particular, additive-error estimations of $\text{Tr}(\rho_0^2)$, $\text{Tr}(\rho_1^2)$, and $\text{Tr}(\rho_0 \rho_1)$ with polynomial precision. Recently, the work of \cite{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.\footnote{In particular, the quantum Jensen-Shannon divergence coincides with the Holevo $\chi$ quantity on size-2 ensembles with a uniform distribution, which arises in the Holevo bound \cite{Hol73}. See \cite[Theorem 12.1]{NC02}.}

### 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 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-coin tape.\footnote{We note that the SWAP test also applies to mixed states, see Proposition 9 in \cite{KMY09}.} The input length and output length of the circuits are $O(\log n)$. The task is to decide whether $D_0$ is $\alpha$-far from or $\beta$-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 ($\ell_2$ 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 \cite{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 \cite{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 $\text{poly} \log(N/\epsilon)$ space, where $\epsilon$ is the additive error. Recently, Acharya, Bhadane, Indyk, and Sun \cite{ABIS19} addressed the entropy estimation problem with i.i.d. samples drawn from distributions as the data stream and demonstrated the first $O(\log(\log(N/\epsilon))$ space streaming algorithm. The sample complexity, viewed as the time complexity, was subsequently improved in \cite{AMNW22}.

However, for the total variation distance ($\ell_1$ 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 \cite{DGKR19}.

Notably, the main differences between the computational and streaming settings lie in how we access the sampling devices.\footnote{Of course, not all distributions can be described as a polynomial-size circuit (i.e., a succinct description).} 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.\footnote{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 ($\ell_2$ 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 coRQ$_L$-hardness for state certification, are also straightforward.13

Regarding space-bounded state testing with respect to the trace distance ($\ell_1$ norm), we note that [Wat02, Proposition 21] implicitly established an NC containment. The BQL-hardness, as well as coRQ$_L$-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 [Lin23].

Finally, we devote the remainder of this section to our main technique (Theorem 1.4), and consequently, we present BQL (resp., coRQ$_L$) 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 $\{\sigma_i\}$ of a linear operator $A$, using a corresponding projected unitary encoding $U$ of $A = \Pi U \Pi$ for projectors $\Pi$ and $\Pi$. The singular value decomposition is $A = \sum \sigma_i |\psi_i\rangle\langle \psi_i|$, where $|\psi_i\rangle$ 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 $f^{(SV)}(A) = f^{(SV)}(\Pi U \Pi)$, we require a degree-$d$ polynomial $\tilde{P}_d(x)$ that satisfies two conditions. Firstly, $\tilde{P}_d$ well-approximates $f$ on the interval of interest $I$, with $\max_{x \in I \setminus I_0} |\tilde{P}_d(x) - f(x)| \leq \epsilon$, where $I_0 \subseteq I \subseteq [-1,1]$ and typically $I_0 := (-\delta,\delta)$. Secondly, $\tilde{P}_d$ is bounded, with $\max_{x \in [-1,1]} |\tilde{P}_d(x)| \leq 1$. The degree of $\tilde{P}_d$ depends on the precision parameters $\delta$ and $\epsilon$, with $d = O(\delta^{-1} \log \epsilon^{-1})$, and all coefficients of $\tilde{P}_d$ can be computed efficiently.

According to [GSLW19], we can use an alternating phase modulation to implement $\tilde{P}_d^{(SV)}(\Pi U \Pi)$,14 which requires a sequence of rotation angles $\Phi \in \mathbb{R}^d$. For instance, consider $\tilde{P}_d(x) = T_d(x)$ where $T_d(x)$ is the $d$-th Chebyshev polynomial (of the first kind), then we know that $\phi_1 = (1 - d/2$ 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

---

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

14This 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 \( f : [-1, 1] \to \mathbb{R} \), if there is a degree-\( d \) polynomial \( P_d \) satisfying \( \max_{x \in [-1, 1]} |f(x) - P_d(x)| \leq \epsilon \), then we have a Chebyshev interpolation polynomial \( P_d(x) := \frac{2}{f(1)} + \sum_{k=1}^{d} c_k T_k \) where \( c_k := \frac{2}{f(1)} \int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}} \, dx \) 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 \( \Pi \) and \( \Pi' \) span on the corresponding subset of \( \{ \{0\}, \{1\} \}^s \),\(^{16}\) and implementing the Chebyshev interpolation polynomial with 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 \( \text{Eval}_{P_d} \) 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 \( f : \mathbb{R} \to \mathbb{R} \) be a continuous function bounded on \( I \subseteq [-1, 1] \). If there exists a degree-\( d \) polynomial \( P_d \) that approximates \( h : [-1, 1] \to \mathbb{R} \), where \( h \) approximates \( f \) only on \( I \), such that \( \max_{x \in [-1, 1]} |h(x) - P_d(x)| \leq \epsilon \), then Chebyshev interpolation yields another degree-\( d \) polynomial \( P_d \) satisfying the following conditions: \( \max_{x \in I} |f(x) - P_d(x)| \leq O(\epsilon \log d) \) and \( \max_{x \in [-1, 1]} |P_d(x)| \leq 1 \). Furthermore, we have an algorithm \( A_f \) 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 \( P_d^{(SV)}(A) \) using \( O(d^2 \|c\|_1) \) queries}\(^{37}\). In addition, \( \|c\|_1 \) 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 \( I = [\beta, 1] \) for any \( \beta \geq 2^{-O(s(n))} \), whereas the techniques from [MY23] are restricted to functions that are bounded on the interval \( I = [-1, 1] \). 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 \( BQ_U \), \( coRQ_U \), and \( RQ_U \).

**Computing the coefficients.** We will implement the evaluation oracle \( \text{Eval}_{P_d} \) 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 \( poly(s(n)) \) 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 \( \Pi \) and \( \Pi' \) 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 \( \|c\|_1 \) arises from renormalizing the bitstring indexed encoding via amplitude amplification.
function $f$ that is bounded on the interval $I = [-1, 1]$, we can use standard numerical integral techniques, given that the integrand’s second derivative in $\{c_k\}_{k=0}^d$ is bounded by $\text{poly}(d)$.

However, implementing the evaluation oracle for piecewise-smooth functions $f$ on an interval $I \subseteq [-1, 1]$ 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 $I$ by multiplying with a Gaussian error function, resulting in $h$ which approximates $f$ only on $I$. 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 $A_f$ for piecewise-smooth functions.

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 $\text{Tr}(P_d(A)\rho)$, where $\rho$ is a (mixed) quantum state prepared by a quantum circuit $Q_\rho$, $A$ 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].

\[
\begin{align*}
|0\rangle & \xrightarrow{H} |0\rangle \quad |0\rangle \\
|0\rangle & \xrightarrow{H} |0\rangle \quad |0\rangle \\
|0\rangle & \xrightarrow{Q_\rho} |0\rangle \quad |0\rangle \\
|0\rangle & \xrightarrow{U_{P_d(A)}} |0\rangle \quad |0\rangle
\end{align*}
\]

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

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 $T(Q_\rho, U_A, P_d)$, where we exclude the precision for simplicity. The measurement outcome of $T(Q_\rho, U_A, P_d)$ will be 0 with a probability close to $1 + \frac{\text{Tr}(P_d(A)\rho)}{2}$. This property allows us to estimate $\text{Tr}(P_d(A)\rho)$ within an additive error $\epsilon$ using $O(1/\epsilon^2)$ sequential repetitions, resulting in a BQL containment.

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

---

18We 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.

19The 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 $\text{BPL} \subseteq \text{BQL}$ [FR21].
sign function. Similarly, $T(Q_i, U_{\rho_i}, P_{d}^{\text{gn}})$ is utilized in GAPQED, where $U_{\rho_i}$ is a block-encoding of $\rho_i$ for $i \in \{0, 1\}$, and $P_{d}^{\text{gn}}$ is a space-efficient polynomial approximation of the normalized logarithmic function. Both $P_{d}^{\text{gn}}$ and $P_{d}^{\text{hn}}$ can be obtained by employing Theorem 1.4.\textsuperscript{20}

**Making the error one-sided.** The main challenge is constructing a unitary $U$ of interest, such as $T(Q_0, U_A, P_d)$, 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 $\text{CertQHS}_{\log}$ and express $\text{HS}^2(\rho_0, \rho_1)$ as a linear combination of $\text{Tr}(\rho_0^2), \text{Tr}(\rho_1^2)$, and $\text{Tr}(\rho_0 \rho_1)$. We thus design a unitary quantum algorithm employing the LCU technique, which accepts with probability $(\frac{1}{2} + \frac{1}{4} \text{HS}^2(\rho_0, \rho_1))^2$, 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 coRQUL, the resulting algorithm is indeed in coRQUL.

Moving on to $\text{CertQSD}_{\log}$, we consider the quantum circuit $U_i = T(Q_i, U_{\rho_i}, P_{d}^{\text{gn}})$ for $i \in \{0, 1\}$. Let $p_i$ be the probability that the measurement outcome of $U_i |0\rangle$ in Figure 1 is 0. Since our space-efficient QSVT preserves parity, specifically the approximation polynomial $P_{d}^{\text{gn}}$ satisfies $P_{d}^{\text{gn}}(0) = 0$,\textsuperscript{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 $A$ to meet the condition of exact amplitude amplification for yes instances. Further analysis shows that $A$ accepts with probability polynomially away from $1$ for no instances. We thus can conclude a coRQUL containment similar to $\text{CertQHS}_{\log}$.

### 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 of space-bounded quantum computation. 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, may shed light on Question (iii) since finding SU(2) rotation angles appears easier.

\textsuperscript{20}In particular, $P_{d}^{\text{hn}}$ is given in Corollary 3.7, as well as $P_{d}^{\text{gn}}$ is given in Corollary 3.10.

\textsuperscript{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 $\epsilon$-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, WZ23].

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 $S(\rho)$ can be estimated within additive error $\epsilon$ with query complexity $\tilde{O}(N/\epsilon^{1.5})$. If we know the reciprocal $\kappa$ of the minimum non-zero eigenvalue of $\rho$, then $S(\rho)$ can be estimated with query complexity $\tilde{O}(\kappa^2/\epsilon)$ [CLW20]. We can estimate $S(\rho)$ within multiplicative error $\epsilon$ with query complexity $\tilde{O}(n^{1/2}+\frac{1+\eta}{\epsilon^2})$ [GHS21], provided that $S(\rho) = \Omega(\epsilon + 1/\eta)$. If $\rho$ is of rank $r$, then $S(\rho)$ 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 (Theorem 1.1), as well as a novel family of natural BQL-complete problems (Theorem 1.2).
- von Neumann entropy. $S(\rho) := -\text{Tr}(\rho \ln \rho)$ for any quantum state $\rho$.

- Quantum Jensen-Shannon divergence. $\text{QJS}(\rho_0, \rho_1) := S\left(\frac{\rho_0 + \rho_1}{2}\right) - \frac{S(\rho_0) + S(\rho_1)}{2}$.

The trace distance and the squared Hilbert-Schmidt distance reach the minimum of 0 when $\rho_0$ equals $\rho_1$, 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 $\rho_0$ and a mixed state $\rho_1$, $F^2(\rho_0, \rho_1) = \text{Tr}(\rho_0 \rho_1)$.
- For two pure states $\rho_0$ and $\rho_1$, $\text{Tr}(\rho_0 \rho_1) = 1 - \text{HS}^2(\rho_0, \rho_1)$.

Moreover, we have $\text{HS}^2(\rho_0, \rho_1) = \frac{1}{2}(\text{Tr}(\rho_0^2) + \text{Tr}(\rho_1^2)) - \text{Tr}(\rho_0 \rho_1)$. 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 $\rho_0$ and $\rho_1$,

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

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 $S(\sum_i p_i |i\rangle\langle i| \otimes \rho_i) = H(D) + \sum_i p_i S(\rho_i)$.

Let us now turn our attention to the quantum Jensen-Shannon divergence, which is defined in [MLP05]. For simplicity, we define $\text{QJS}_2(\rho_0, \rho_1) := \text{QJS}(\rho_0, \rho_1)/\ln 2$ 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 $\rho_0$ and $\rho_1$, we have

$$1 - H_2\left(\frac{1 - \text{td}(\rho_0, \rho_1)}{2}\right) \leq \text{QJS}_2(\rho_0, \rho_1) \leq \text{td}(\rho_0, \rho_1).$$

Here, the binary entropy $H_2(p) := -p \log(p) - (1 - p) \log(1 - p)$.

### 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 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 $\text{BQ}_s\text{SPACE}[s(n)]$. 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 \{HADAMARD, CNOT, T\}. For a promise problem \( L = (L_{\text{yes}}, L_{\text{no}}) \), we say that a family of quantum circuits \( \{C_x : x \in L\} \) is \( t(n) \)-time-bounded if there is a deterministic Turing machine that, on any input \( x \in L \), runs in time \( O(t(|x|)) \), and outputs a description of \( C_x \) such that \( C_x \) accepts (resp., rejects) if \( x \in L_{\text{yes}} \) (resp., \( x \in L_{\text{no}} \)). Similarly, we say that a family of quantum circuits \( \{C_x : x \in L\} \) is \( s(n) \)-space-bounded if there is a deterministic Turing machine that, on any input \( x \in L \), 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 \( x \in L_{\text{yes}} \) (resp., \( x \in L_{\text{no}} \)), as well as \( C_x \) is acting on \( O(s(|x|)) \) qubits and has \( 2^{O(s(|x|))} \) gates.

Definition 2.6 (BQSPACE\([s(n), a(n), b(n)]\), adapted from Definition 5 in [FR21]). Let \( s : \mathbb{N} \to \mathbb{N} \) 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 \( (L_{\text{yes}}, L_{\text{no}}) \) is in BQSPACE\([s(n), a(n), b(n)]\) if there exists a family of \( s(n) \)-space-bounded (unitary) quantum circuits \( \{C_x \} \), 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 \( x \in L_{\text{yes}} \), whereas \( \Pr[C_x \text{ accepts } x] \leq b(|x|) \) if \( x \in L_{\text{no}} \).

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 BQSPACE\([s(n), a(n), b(n)]\) as long as \( a(n) - b(n) \geq 2^{-O(s(n))} \), which follows from [FKL16] or our space-efficient QSVT-based construction in Section 3.4. We thereby define BQSPACE\([s(n)] := \text{BQSPACE}[s(n), 2/3, 1/3] \) to represent (two-sided) bounded-error unitary quantum space, and BQL := BQSPACE\([O(\log n)] \) 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 \( \Phi \) mapping from density matrices on \( k_1 \) qubits to density matrices on \( k_2 \) qubits, we can exactly simulate this quantum channel \( \Phi \) 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 BQSPACE\([s(n)]\).

Definition 2.7 (BQSPACE\([s(n), a(n), b(n)]\), adapted from Definition 7 in [FR21]). Let \( s : \mathbb{N} \to \mathbb{N} \) 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 \( (L_{\text{yes}}, L_{\text{no}}) \) is in BQSPACE\([s(n), a(n), b(n)]\) if there exists a family of \( s(n) \)-space-bounded general quantum circuits \( \{\Phi_x \} \), where \( n = |x| \), satisfying the following holds:

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

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 BQSPACE\([s(n)] \subseteq 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 BQSPACE\([s(n)] \subseteq \text{BQSPACE}[O(s(n))] \). In addition, it is evident that BQSPACE\([s(n)] \) can achieve error reduction since it admits sequential repetition simply by resetting working qubits. Therefore, we define BQSPACE\([s(n)] := \text{BQSPACE}[s(n), 2/3, 1/3] \).
to represent (two-sided) bounded-error general quantum space, and denote general quantum logspace by \( \text{BQL} := \text{BQSPACE}[O(|\log n|)] \).

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

- \( \text{RQ}_{U}\text{SPACE}[s(n), a(n)] := \text{BQ}_{U}\text{SPACE}[s(n), a(n), 0] \);
- \( \text{coRQ}_{U}\text{SPACE}[s(n), b(n)] := \text{BQ}_{U}\text{SPACE}[s(n), 1, b(n)] \).

Note that \( \text{RQ}_{U}\text{SPACE}[s(n), a(n)] \) and \( \text{coRQ}_{U}\text{SPACE}[s(n), b(n)] \) can achieve error reduction, as shown in \cite{Wat01} or our space-efficient QSOrT-based construction in Section 3.4. We define \( \text{RQ}_{U}\text{SPACE}[s(n)] := \text{BQ}_{U}\text{SPACE}[s(n), \frac{1}{2}, 0] \) and \( \text{coRQ}_{U}\text{SPACE}[s(n)] := \text{BQ}_{U}\text{SPACE}[s(n), 1, \frac{1}{2}] \) to represent one-sided bounded-error unitary quantum space, as well as logspace counterparts

\( \text{RQ}_{U}\text{L} := \text{RQ}_{U}\text{SPACE}[O(|\log n|)] \) and \( \text{coRQ}_{U}\text{L} := \text{coRQ}_{U}\text{SPACE}[O(|\log n|)] \).

Remark 2.8: \( \text{RQ}_{U}\text{L} \) and \( \text{coRQ}_{U}\text{L} \) are gateset-dependent. We observe that changing the gateset in space-efficient Solovay-Kitaev theorem \cite{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 \( \text{RQ}_{U}\text{SPACE}[s(n)] \) and \( \text{coRQ}_{U}\text{SPACE}[s(n)] \), 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:

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

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 \cite{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 \cite{Pow67}.

**Lemma 2.10 (Near-minimax approximation by Chebyshev interpolation, adapted from Theorem 6.13 in \cite{MH02}).** For any continuous function \( f: [-1, 1] \to \mathbb{R} \), if there exists an explicit degree-\( d \) polynomial \( P_d \in \mathbb{R}[x] \) such that \( \max_{x \in [-1, 1]} |f(x) - P_d(x)| \leq \epsilon \), 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 $A$ 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 $A$, 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 $E[X] = p$. Now let $\hat{X} = \frac{1}{m} \sum_{i=1}^{m} X_i$ be the additive-error estimation, where $\hat{X}_i$ is the outcome of $A$ in the $i$-th repetition. By the Chernoff-Hoeffding bound (e.g., Theorem 4.12 in [MU17]), we know that $\Pr[|\hat{X} - p| \geq \epsilon] \leq 2 \exp(-2m\epsilon^2)$. By choosing $m = 2e^{-2}$, this choice of $m$ ensures that each run of $A$ 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. \qed

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, we will use Lemma 2.11 for convenience, given that $BQL = BQUL$ [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 $\rho_0$ and $\rho_1$ are two $n$-qubit mixed quantum states. There is a $(2n + 1)$-qubit quantum circuit that outputs 0 with probability $\frac{1 + \operatorname{Tr}(\rho_0 \rho_1)}{2}$, using 1 sample of each $\rho_0$ and $\rho_1$ 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 $\ell$-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 $\tilde{B}$ from $B$ by adding an additional column and row. Let $\tilde{B}[i, j]$ denote the entry at the $i$-th column and the $j$-th row of $\tilde{B}$. Specifically,

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

Next, we view $\tilde{B}$ as a transition matrix of a Markov chain since $\tilde{B}$ 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 $(u, v) \in E$ if $B(u, v) > 0$. In particular, the probability that a $k$-step random walk starting at node $s$ and ending at node $t$ is exactly $\tilde{B}^k[s, t] = B^k[s, t]$. This is because the walker who visits the dummy node $\perp$ will not reach other nodes.

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

### 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 $(a,a,\epsilon)$-projected unitary encoding of a linear operator $A$ if \( \|A - \alpha \Pi U \Pi \| \leq \epsilon \), where $U$ and orthogonal projectors $\Pi$ and $\Pi$ act on $s + a$ qubits, and both rank($\Pi$) and 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 $\Pi$ and $\Pi$ span on $\tilde{S}, \tilde{S} \subseteq \{0, 1\}^{(a+s)}$, respectively.\footnote{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.} In particular, for any $|\tilde{s}_i\rangle \in \tilde{S}$ and $|s_j\rangle \in S$, we have a matrix representation $A_{\tilde{s}_i, s_j}(i, j) := \langle \tilde{s}_i | U | s_j \rangle$ 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 $A = (|0\rangle \otimes I_s) U (|0\rangle \otimes I_s)$ 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 $\Pi$ and $\Pi$ on a finite-dimensional Hilbert space $\mathcal{H}_U$. Namely, $A = \tilde{\Pi} U \Pi$. Then there exists orthonormal bases of $\Pi$ and $\Pi$ such that $\Pi$: \{ $|\psi_i\rangle : i \in [d]$ \}, where $d := \text{rank}(\Pi)$, of a subspace $\text{Img}(\Pi) = \text{span}\{ |\psi_i\rangle \}$; $\tilde{\Pi}$: \{ $|\tilde{\psi}_i\rangle : i \in [\tilde{d}]$ \}, where $\tilde{d} := \text{rank}(\tilde{\Pi})$, of a subspace $\text{Img}(\tilde{\Pi}) = \text{span}\{ |\tilde{\psi}_i\rangle \}$. 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 $f: \mathbb{R} \rightarrow \mathbb{C}$ be an even or odd function. We consider a linear operator $A \in \mathbb{C}^{d \times d}$ satisfying the singular value decomposition $A = \sum_{i=1}^{\min(d, \tilde{d})} \sigma_i |\tilde{\psi}_i\rangle \langle \psi_i|$. We define the singular value transformation corresponding to $f$ as follows:

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

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

It is worth noting that $f^{(SV)}(A) = f(A)$ 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 $f: \mathbb{R} \rightarrow \mathbb{R}$ be a continuous function bounded on the closed interval of interest $I \subseteq [-1, 1]$. If there exists a degree-$d$ polynomial $P_d$ that approximates $h: [-1, 1] \rightarrow \mathbb{R}$, where $h$ approximates $f$ only on $I$, such that $\max_{x \in [-1, 1]} |h(x) - P_d(x)| \leq \epsilon$, then Chebyshev interpolation yields another degree-$d$ polynomial $P_d$ satisfying the following conditions: $\max_{x \in I} |f(x) - P_d(x)| \leq O(\epsilon \log d)$ and $\max_{x \in [-1, 1]} |P_d(x)| \leq 1$.\hfill $\square$
Moreover, we have space-efficient classical algorithms for computing any entry in the coefficient vector \( c \) of the Chebyshev interpolation polynomial \( P_d \):

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

Furthermore, for any \((1, a, 0)\)-bitstring indexed encoding \( U \) of \( A = \tilde{\Pi} \Pi \), 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 \( P_d^{(SV)}(A) \) that acts on \( O(s(n)) \) qubits by using \( O(d^2\|c\|_1) \) queries to \( U \).

We remark that we can apply Theorem 3.4 to general forms of the projected unitary encoding \( U \) with orthogonal projectors \( \Pi \) and \( \tilde{\Pi} \), as long as such an encoding meets the conditions: (1) The basis of \( \Pi \) and \( \tilde{\Pi} \) admits a well-defined order; (2) Both controlled-\( \Pi \) 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 $P_d^{(f)}$ 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 $T_k(\cos \theta) = \cos(k\theta)$:

$$P_d^{(f)}(x) = \frac{c_0}{2} + \sum_{k=1}^{d} c_k T_k(x)$$

where $c_k := \frac{2}{\pi} \int_{-\pi}^{\pi} F_k(\theta) \, d\theta$ 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 $\gamma$, then any entry of the coefficient vector $c = (c_0, \ldots, c_d)$ can be computed in deterministic time $O(d^{\gamma+1}/2^{\epsilon-1/2} t(\ell))$ and space $O(\log(d^{\gamma+1}/2^{\epsilon-3/2}))$, where evaluating $F(\theta)$ in $\ell$-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 $c$ has a norm bounded by $\|c\|_1 \leq O(Bd)$.

**Proof.** To apply Chebyshev interpolation to a bounded continuous function $f(x)$, we begin with a degree-$d$ polynomial $P_d^{(f)}$ such that $\max_{x \in [-1,1]} |f(x) - P_d^{(f)}(x)| \leq \epsilon$. 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 $P_d^{(f)} = \frac{c_0}{2} + \sum_{k=1}^{d} c_k T_k$, where $c_k = \frac{2}{\pi} \int_{-\pi}^{\pi} F_k(\theta) \, d\theta$ and $F_k(\theta) := \cos(k\theta) f(\cos \theta)$, and additionally satisfies the error bound: $\max_{x \in [-1,1]} |f(x) - P_d^{(f)}(x)| \leq O(\epsilon \log d)$.

**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:

$$\int_{-\pi}^{\pi} F_k(x) \, dx \approx \frac{\pi}{m} \left( F_k(x_0) + \sum_{i=1}^{m} F_k(x_i) + F_k(x_m) \right)$$

where $x_i := -\pi + \frac{\pi}{m} l$ for $l = 0, 1, \ldots, m$.

(3.1)

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

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

(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 $\varepsilon_{d,k} = \sum_{k=1}^{d} \varepsilon_{d,k} \leq \epsilon$. 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 $\varepsilon_{d,k} \leq O(d^{\gamma+1}/m^2) \leq O(\epsilon)$. Moreover, to guarantee that the accumulated error is $O(\epsilon)$ in Equation (3.1), we need to evaluate the integrand $F(\theta)$ with $\ell$-bit precision, where $\ell = O(\log (m/\epsilon)) = O(\log(\epsilon^{-3/2} d^{(\gamma+1)/2}))$. In addition, note that $c_k = \frac{2}{\pi} \int_{-\pi}^{\pi} F_k(\theta) \, d\theta \leq 2 \cdot \max_{x \in [-1,1]} |f(x)| \leq 2B$, we know that the coefficient vector $c$ satisfies $\|c\|_1 = \sum_{k=0}^{d} |c_k| \leq O(Bd)$.

**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., $\ell$-bit precision) in $O(\ell)$ space. The space complexity required for computing the numerical integration is the number of bits required to index the interval coefficients and represent the resulting coefficients. To be specific, the space complexity is

$$\max \{O(\log m), O\left(\log \frac{m}{\epsilon}\right), \log \|c\|_\infty\} \leq O(\max \{\log(\epsilon^{-1/2} d^{\gamma+1}/2^{1/2} B), 2B\}).$$

Here, $\|c\|_\infty = \max_{0 \leq k \leq d} \left| \int_{-\pi}^{\pi} \cos(k\theta) f(\cos \theta) \, d\theta \right| \leq \max_{0 \leq k \leq d} \max_{-\pi \leq \theta \leq 0} O(|f(\cos \theta)|) \leq O(B)$, 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 $\ell$-bit precision requires only deterministic $O(\ell)$ space:
Remark 3.6 (Space-efficient evaluation of holonomic functions). Holonomic functions encompass several commonly used functions,\footnote{For a more detailed introduction, please refer to [BZ10, Section 4.9.2].} 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 $\ell$-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 $P^\text{sgn}_d := \frac{c_0}{2} + \sum_{k=1}^d c_k T_k \in \mathbb{R}[x]$ of degree $d \leq \tilde{C}_\text{sgn} \delta^{-1} \log \epsilon^{-1}$, where $\tilde{C}_\text{sgn}$ is a universal constant. Any entry of the coefficient vector $c := (c_0, \cdots, c_d)$ can be computed in deterministic time $\tilde{O}(\epsilon^{-1/2} d^2)$ and space $O(\log(\epsilon^{-1} d))$. Furthermore, the polynomial $P^\text{sgn}_d$ satisfies the following conditions:

\[
\forall x \in [-1, 1] \setminus [-\delta, \delta], |\text{sgn}(x) - P^\text{sgn}_d(x)| \leq C_\text{sgn} \epsilon \log d, \quad \text{where } C_\text{sgn} \text{ is a universal constant},
\]

\[
\forall x \in [-1, 1], |P^\text{sgn}_d(x)| \leq 1.
\]

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

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

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

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

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

\[
\max_{x \in [-1, 1]} |\hat{P}^\text{sgn}_d(x) - \text{erf}(\kappa x)| \leq \epsilon.
\]

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

\[
\max_{0 \leq k \leq d, \xi \in [-\pi, 0]} |F^\text{sgn}_k(\xi)| \leq \frac{2}{\sqrt{\pi}} \kappa + \frac{4}{\sqrt{\pi}} \kappa^2 + \frac{4}{\sqrt{\pi}} \kappa^3 + \frac{4}{\sqrt{\pi}} k \kappa.
\]

Note that $\kappa \leq O(d)$ and $k \leq d$, Fact 3.7.2 indicates that $\max_{\xi \in [-\pi, 0]} |F^\text{sgn}_k(\xi)| \leq O(d^2)$ for any $0 \leq k \leq d$. Hence, we result in an approximation polynomial $\hat{P}^\text{sgn}_d$ by Lemma 3.5 satisfies that $\max_{x \in [-1, 1]} |\text{erf}(\kappa x) - \hat{P}^\text{sgn}_d(x)| \leq O(\epsilon \log d)$, which additionally derives that

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

Here, $C_\text{sgn}$ is a universal constant. Moreover, we specify the bound of $\|c^\text{sgn}\|_1$ 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 $c^\text{sgn}$ corresponding to a degree-$d$ polynomial $\hat{P}^\text{sgn}_d$, we have $\|c^\text{sgn}\|_1 \leq \tilde{C}_\text{sgn} \log d$ where $\tilde{C}_\text{sgn}$ is a universal constant.
In addition, the coefficient vector $\tilde{c}_{\text{sgn}}$ can be computed in deterministic space $O(\log(d\epsilon^{-1}))$. As the evaluation of the integrand $F(\theta)$ requires $\ell$-bit precision where $\ell = O(\log(\epsilon^{-3/2}d^2))$, together with Remark 3.6, $\tilde{c}_{\text{sgn}}$ can be computed in deterministic time $O(\epsilon^{-1/2}d^2)$.

Finally, we obtain that $|F_d^{\text{sgn}}(x)| \leq 1 + \epsilon$ for any $x \in [-1, 1]$ since $|\text{sgn}(x)| \leq 1$ for any $x$. We finish the proof by normalizing $F_d^{\text{sgn}}$, in particular, considering $F_d^{\text{sgn}}(x) := (1 + \epsilon)^{-1} F_d^{\text{sgn}}$. It is evident to verify that $F_d^{\text{sgn}}$ 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 $f : [-x_0 - r - \delta, x_0 + r + \delta] \to \mathbb{R}$ such that $f(x_0 + x) = \sum_{i=0}^{\infty} a_i x^i$ for all $x \in [-r - \delta, \epsilon + \delta]$, where $x_0 \in [-1, 1], r \in (0, 2], \delta \in (0, r]$. Assume that $\sum_{i=0}^{\infty} (r + \delta)^i |a_i| \leq B$ where $B > 0$. Let $\epsilon \in (0, 1/2])$ 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 $c^{(P)}$ can be computed in bounded-error randomized time $\tilde{O}(\max\{(|\delta|^{-3} \epsilon^{-2} B^2, d^2 \epsilon^{-1/2} B^2\})$ and space $O(\log(d^4(\delta^{-4})^{-4}\epsilon^{-1} B^4))$ where $\delta := \frac{\delta}{2(r + \delta)}$, such that

$$||f(x) - P(x)||_{|x_0-r,x_0+r|} \leq \epsilon(\log d),$$

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

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

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

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 $f: \mathbb{R} \to \mathbb{R}$ be a real-valued function such that $|f(x) - \sum_{k=0}^{K} a_k x^k| \leq \epsilon/4$ for all $x \in I_\delta$, the interval $I_\delta := [-1 + \delta, 1 - \delta]$ and $\|a\|_1 \leq O(\max\{\epsilon^{-1}, \delta^{-1}\})$. Then there is a coefficient vector $c \in \mathbb{C}^{2M+1}$ such that

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

Here $M := \max(2[\delta^{-1} \log(4\|a\|_1 \epsilon^{-1})] + 0)$ and $\|c\|_1 \leq \|a\|_1$. Moreover, the coefficient vector $c$ 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 $\|a\|_1 \geq \epsilon/2$.\footnote{This is because if $\|a\|_1 < \epsilon/2$, then $\|f\|_\infty \leq \|f(x) - \sum_{k=0}^{K} a_k x^k\|_\infty + \|\sum_{k=0}^{K} a_k x^k\|_\infty \leq \epsilon/4 + \|a\|_1 < \epsilon$, implying that $M = 0$ and $c = 0$.}

**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.
Proposition 3.9.1 (First approximation). Let \( f(x) := \sum_{k=0}^{K} a_k x^k \) such that \( \| f - f_1 \|_\infty \leq \epsilon / 4 \). Then we know that \( f(x) = \sum_{k=0}^{K} a_k \sum_{l=0}^{\infty} b_l^{(k)} \sin(x \ell) \) where the coefficients \( b_l^{(k)} \) satisfy that

\[
 b_l^{(k+1)} = \sum_{l' = 0}^{l} b_l^{(k)} b_{l-l'}^{(1)} \quad \text{where } b_l^{(1)} = \begin{cases} 0 & \text{if } l \text{ is even}, \\ \left( \frac{l-1}{2} \right) \left( \frac{l}{2} \right) & \text{if } l \text{ is odd}. \end{cases} \tag{3.3}
\]

Furthermore, the coefficients \( \{ b_l^{(k)} \} \) satisfies the following: (1) \( \| b_l^{(k)} \|_1 = 1 \) for all \( k \geq 1 \); (2) \( b_l^{(k)} \) is entry-wise non-negative for all \( k \geq 1 \); (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 \( f_2(x) := \sum_{k=0}^{K} a_k \sum_{l=0}^{L} b_l^{(k)} \sin(x \ell) \) where \( L := \lceil \delta^{-2} \ln(4\|a\|_1 e^{-1}) \rceil \), then we have that \( \| f - f_2 \|_\infty \leq \epsilon / 4 \).

The third approximation approximates the functions \( \sin(x) \) in \( f_2(x) \) using a tail bound of the binomial distribution. Notably, this construction not only quadratically improves the dependence on \( \delta \), but also ensures that the integrand’s second derivative is bounded when combined with Lemma 3.5.

Proposition 3.9.3 (Third approximation). Let \( f_3(x) \) be polynomial approximations of \( f \) that depends on the parity of \( f \) such that \( \| f - f_3 \| \leq \epsilon / 2 \) and \( M = \lceil \delta^{-1} \ln(4\|a\|_1 e^{-1}) \rceil \), then we have

\[
 f_3^{(\text{even})}(x) := \sum_{m=-M}^{M} c_m^{(\text{even})} \cos(\pi x m) \quad \text{where } c_m^{(\text{even})} := (-1)^m \sum_{k=0}^{K} a_k \sum_{l=0}^{L/2} b_l^{(k)} \left( \frac{2 l}{m} \right)^2 2^{2l};
\]

\[
 f_3^{(\text{odd})}(x) := \sum_{m=-M}^{M} c_m^{(\text{odd})} \sin(\pi x(m + 1/2)) \quad \text{where } c_m^{(\text{odd})} := (-1)^m \sum_{k=0}^{K} a_k \sum_{l=0}^{(L-1)/2} b_l^{(k)} \left( \frac{2 l}{m + l + 1} \right)^2 2^{2l-1}. \tag{3.4}
\]

We then notice that the rearrangement of terms in Equation (3.4) can be directly applied to the definition of \( f_3(x) \) in Proposition 3.9.3. As a consequence, we obtain the following bound on the accumulative error: \( \| f - f_3 \| \leq \| f - f_1 \| + \| f_1 - f_2 \| + \| f_2 - f_3 \| \leq \epsilon \). Additionally, we remark that \( \| c \|_1 \leq \| a \|_1 \), since \( \| b_l^{(k)} \|_1 = 1 \) (see Proposition 3.9.1) and \( \sum_{m=0}^{M} \left( \frac{1}{2} \right)^m = 2^l \).

Analyzing time and space complexity. To evaluate the bounded polynomial approximation \( f_3(x) \) with \( \epsilon \) accuracy, it is necessary to approximate the summand with \( \ell \)-bit precision, where \( \ell = O(\log(KLMe^{-1})) = O(\log(\delta^{-4}e^{-1})) \). Since the summand is a product of a constant number of holonomic functions, approximating \( b_l^{(k)} \) with \( \ell \)-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.

\[\text{In particular, the summand in } f_3(x) = c_m^{(\text{even})} \cos(\pi x m) + c_m^{(\text{odd})} \sin(\pi x(m + 1/2)) \text{ if } f \text{ is neither even nor odd.}\]
We now present a bounded-error randomized algorithm for estimating \( t_i^{(k)} \). As \( b^{(1)} \) is entrywise non-negative and \( \sum_{i=1}^n b_i^{(1)} < \| b^{(1)} \|_1 = 1 \) following Proposition 3.9.1, we can express the recursive formula in Equation (3.3) as the matrix powering of a sub-stochastic matrix \( B_1 \):

\[
B_1^k := \begin{pmatrix}
\vdots & \vdots & \ddots & \vdots \\
0 & b_1^{(1)} & \cdots & b_{n-1}^{(1)} \\
0 & 0 & \ddots & 0 \\
0 & 0 & \cdots & 0 \\
\end{pmatrix}^k = \begin{pmatrix}
\vdots & \vdots & \ddots & \vdots \\
0 & b_1^{(k)} & \cdots & b_{n-1}^{(k)} \\
0 & 0 & \ddots & 0 \\
0 & 0 & \cdots & 0 \\
\end{pmatrix} := B_k.
\]

In addition, we approximate the sub-stochastic matrix \( B_1 \) by truncating it using Lemma 3.9. Specifically, we repeat the algorithm \( B_1 \) using Lemma 3.11. Next, we ensure that \( f \) is computed by truncating it using Lemma 3.9. Let us begin by defining a linear transformation \( B \) of \( f \) by applying Lemma 3.5.

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 \( \tilde{f}(x) \) of the given function \( f(x) \) by truncating it using Lemma 3.9. Next, we ensure that \( \tilde{f}(x) \) 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 the 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 \( b_i := a_i(r+\delta)^j \), then it is evident that \( g(y) := \sum_{i=0}^\infty b_i y^j \) 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 \( \delta' := \frac{\delta}{\pi(r+\delta)} \) and \( J := \lceil (\delta')^{-1} \log(12B\epsilon^{-1}) \rceil \). This ensures that the truncation error \( \varepsilon_j^{(g)} := |g(y) - \sum_{j=0}^{J} b_j y^j| \) for any \( y \in [-1 + \delta', 1 - \delta'] \) satisfies the following:

\[
\varepsilon_j^{(g)} = \sum_{j=J}^\infty |b_j y^j| \leq \sum_{j=J}^\infty |b_j (1 - \delta')^j| \leq (1 - \delta')^J \sum_{j=J}^\infty |b_j| \leq (1 - \delta')^J B \leq e^{-\delta'} J B \leq \frac{\epsilon}{12} := \varepsilon'.
\]

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

\[
\tilde{g}(y) := \begin{cases}
\sum_{m=-M}^M c_m^{(even)} \cos(\pi y m), & \text{if } f \text{ is even} \\
\sum_{m=-M}^M c_m^{(odd)} \sin(\pi y (m + \frac{1}{2})), & \text{if } f \text{ is odd}
\end{cases},
\]

\[
\sum_{m=-M}^M \left( c_m^{(even)} \cos(\pi y m) + c_m^{(odd)} \sin(\pi y (m + \frac{1}{2})) \right), & \text{otherwise}
\]

\[
(3.5)
\]
By appropriately choosing $M = O((\delta')^{-1}\log (\|\hat{b}\|_1/\epsilon')) = O(\delta^{-1}\log (B/\epsilon))$, we obtain that the vectors of coefficients $c^{(\text{even})}$ and $c^{(\text{odd})}$ satisfy $\|c^{(\text{even})}\|_1 \leq \|\hat{b}\|_1 \leq B$ and similarly $\|c^{(\text{odd})}\| \leq B$. Plugging $f(x) = \hat{g}(L(x))$ 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 $[x_0 - r - \delta/2, x_0 + r + \delta/2]$:

\[
\hat{f}(x) = \hat{g}\left(\frac{x - x_0}{r + \delta}\right) = \begin{cases} 
\sum_{m=-M}^{M} c_m^{(\text{even})} \cos\left(\pi m \left(\frac{x - x_0}{r + \delta}\right)\right), & \text{if } f \text{ is even} \\
\sum_{m=-M}^{M} c_m^{(\text{odd})} \sin\left(\pi m \left(\frac{x - x_0}{r + \delta}\right)\right), & \text{if } f \text{ is odd} . 
\end{cases}
\]

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 $\epsilon$-close to 1 on the interval $[x_0 - r, x_0 + r]$ and is $\epsilon$-close to 0 on the interval $[-1, 1] \setminus [x_0 - r - 2\delta, x_0 + r + 2\delta]$, where $\epsilon := \epsilon/(3B)$ and $\delta := \delta/4$. Moreover, $|R(x)| \leq 1$ 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:

\[
R(x) := \frac{1}{2} \left[ \text{erf}(\kappa(x-x_0+r+\delta)) - \text{erf}(\kappa(x-x_0-r-\delta)) \right] \quad \text{where} \quad \kappa := 2\sqrt{\frac{2}{\delta}} \log 2 \frac{\sqrt{\pi}}{\sqrt{\pi} \epsilon} - 2\log 2 \frac{\sqrt{\pi} B}{\sqrt{\pi} \epsilon} . \tag{3.6}
\]

Bounded polynomial approximation with Chebyshev interpolation. We hereby 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 $d = O(\delta^{-1}\log (Be^{-1}))$ 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 $f: [-x_0 - r - \delta, x_0 + r + \delta] \rightarrow \mathbb{R}$ and be such that $f(x_0 + x) := \sum_{l=0}^{\infty} a_l x^l$ for all $x \in [-r, r]$. Suppose $B > 0$ is such that $\sum_{l=0}^{\infty} (r + \delta)^l |a_l| \leq B$. Let $\epsilon \in (0, \frac{2}{2r}]$, there is a $\epsilon/3$-precise Fourier approximation $\hat{f}(x)$ of $f(x)$ on the interval $[x_0 - r + \delta/2, x_0 + r + \delta/2]$, where $\hat{f}(x) := \sum_{m=-M}^{M} \Re\left[ c_m e^{-\frac{\text{i} m}{2\pi} x x_0} e^{\frac{\text{i} m}{2\pi} x^2} \right]$ and $\|\hat{c}\| \leq B$. We have a time-efficient polynomial $P^* \in \mathbb{R}[x]$ of degree $O(\delta^{-1}\log (Be^{-1}))$ s.t.

\[
\|\hat{f}(x) R(x) - P^*(x)\|_{[x_0-r, x_0+r]} \leq \epsilon, \\
\|P^*(x)\|_{[-1, 1]} \leq \epsilon + \|\hat{f}(x) R(x)\|_{[x_0-r-\delta/2, x_0+r+\delta/2]} \leq \epsilon + B, \tag{3.7}
\]

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 $0 \leq k \leq d$. 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 $F_k(\theta) = \sum_{m=-M}^{M} c_m (H_{k,m}^{(+) - H_{k,m}^{(-)})}$ 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

\[
H_{k,m}^{(\pm)}(\theta) := \cos\left(\pi m \left(\frac{\cos \theta - x_0}{r + \delta}\right)\right) \cdot \cos(k\theta) \cdot \text{erf}\left(\kappa \left(\cos \theta - x_0 \pm r \pm \frac{\delta}{4}\right)\right). \tag{3.8}
\]

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

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

Moreover, the integrand is $F_k(\theta) = \sum_{m=-M}^{M} \left( c_m^{(even)} (H_{k,m}^{(+) - H_{k,m}^{(-)})} + c_m^{(odd)} (H_{k,m}^{(+) - H_{k,m}^{(-)})} \right)$ when $f$ is neither even nor odd, where $H_{k,m}^{(\pm)}$ and $H_{k,m}^{(\pm)}$ follow from Equation (3.8) and Equation (3.9),
respectively. Regardless of the parity of \( f \), we have that the second derivative \( F''(\theta) \leq O(Bd^3) \).

Together with Fact 3.8.2, we are ready to apply Lemma 3.5 to \( h(x) = \tilde{f}(x)/R(x) \), 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 \( I_{\text{int}} := [x_0 - r, x_0 + r] \) and \( I_{\text{ext}} := [x_0 - r - \delta/2, x_0 + r + \delta/2] \) to obtain:

\[
\|f(x) - P(x)\|_{I_{\text{int}}} \leq \|f(x) - h(x)\|_{I_{\text{int}}} + \|h(x) - P(x)\|_{I_{\text{int}}} \leq \epsilon + O(\epsilon \log d) = O(\epsilon \log d),
\]

\[
\|P(x) - 0\|_{I_{\text{ext}}} \leq \|P(x) - h(x)\|_{I_{\text{ext}}} + \|h(x) - 0\|_{I_{\text{ext}}} \leq O(\epsilon \log d) + O(\epsilon) \leq O(\epsilon \log d).
\]

(3.10)

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

Analyzing time and space complexity. The construction of \( \tilde{f}(x) \) can be implemented in bounded-error randomized time \( O((\delta')^{-5}e^{-2}B^2) \) and space \( O((\delta')^{-4}e^{-1}B) \), given that this construction uses Lemma 3.9 with \( \delta' = \frac{\epsilon}{2r(1+\delta)} \in (0, \frac{1}{2}] \) and \( \epsilon' = \frac{\epsilon}{3(1+\delta)} \). Having \( \tilde{f}(x) \), we can construct a bounded polynomial approximation \( \hat{h}(x) \) deterministically using Lemma 3.5. This construction can be implemented in deterministic time \( O(d^2e^{-1/2}B) \) and space \( O(d^3e^{-1}B) \) 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 \( O(\max\{ (\delta')^{-5}e^{-2}B^2, d^2e^{-1/2}B \}) \) and space \( O(\log(d^4(\delta')^{-4}e^{-1}B)) \).

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 logarithmic function). Let \( \beta \in (0,1] \) and \( \epsilon \in (0,1/2) \), there is an even polynomial \( P \) of degree \( d \leq \tilde{C}_n\beta^{-1} \log \epsilon^{-1} \) where \( \tilde{C}_n \) is a universal constant such that

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

\[
\forall x \in [-1, 1], \quad |P(x)| \leq 1.
\]

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

Proof. Consider the function \( f(x) := \frac{\ln(x)}{2\ln(2/\beta)} \). 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 \( B = 1/2 \). 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 \( C_{\ln}^\epsilon \), the approximation guarantees the following inequalities:

\[
\|f(x) - \tilde{P}(x)\|_{[\beta, 2-\beta]} \leq C_{\ln}^\epsilon \log d = \frac{\eta}{4},
\]

\[
\|\tilde{P}(x)\|_{[-1,1]} \leq B + C_{\ln}^\epsilon \log d \leq \frac{1}{2} + C_{\ln}^\epsilon \log d = \frac{1}{2} + \frac{\eta}{4},
\]

\[
\|\tilde{P}(x)\|_{[-1,\beta/2]} \leq C_{\ln}^\epsilon \log d - \frac{\eta}{4}.
\]

As indicated in Lemma 11 of [GL20], since the Taylor series of \( f(x) \) at \( x = 1 \) is

\[
\frac{1}{\ln(2/\beta)} \sum_{i=1}^{\infty} \frac{(-1)^i x^{i-1}}{i},
\]

we obtain that

\[
B = f\left(\frac{\beta}{2} - 1\right) = \frac{1}{\ln(2/\beta)} \sum_{i=1}^{\infty} \frac{(-1)^i x^{i-1}}{i} = -\frac{1}{2 \ln(2/\beta)} \sum_{i=1}^{\infty} \frac{(-1)^i x^{i-1}}{i} (\beta/2 - 1)^i = -\frac{1}{2 \ln(2/\beta)} \ln \frac{\beta}{2} = \frac{1}{2}.
\]
Additionally, the coefficient vector $c^{(P)}$ of $\hat{P}$ satisfies that $\|c^{(P)}\|_1 \leq O(Bd) \leq \hat{C}_n d$ where $\hat{C}_n$ is a universal constant. Notice that $\delta' = \frac{\delta}{2(1 + \beta)} = \frac{\beta}{2(1 - \beta^{1/2})} = \Theta(\beta)$, our utilization of Theorem 3.8 yields a bounded-error randomized algorithm that requires $O(\log(d^4(\delta')^{-4}e^{-1}B)) = O(\log(d^4\beta^{-4}e^{-1}))$ space and $\tilde{O}(\max\{\beta^{-5}e^{-2}, c^2\beta^{-1/2}\}B^2) = \tilde{O}(\max\{\beta^{-5}e^{-2}, d^2e^{-1/2}\})$ 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:

$$
\|f(x) - P(x)\|_{[\beta, 1]} \leq \|f(x) - \frac{1}{1 + \eta} \tilde{P}(x)\|_{[\beta, 1]} + \|\frac{1}{1 + \eta} \tilde{P}(-x)\|_{[\beta, 1]} \\
\leq \|f(x) - \tilde{P}(x)\|_{[\beta, 1]} + \|\frac{1}{1 + \eta} \tilde{P}(x) - \frac{1}{1 + \eta} \tilde{P}(-x)\|_{[\beta, 1]} \\
\leq \frac{\eta}{2} + \frac{\eta}{1 + \eta} \cdot \left(\frac{1}{2} + \frac{\eta}{2}\right) + \frac{1}{1 + \eta} \cdot \frac{\eta}{2} \\
= \frac{\eta}{2} + \frac{\eta}{1 + \eta} \cdot \frac{1 + \eta}{2} + \frac{1}{1 + \eta} \cdot \frac{\eta}{2} \\
\leq \eta.
$$

(3.12)

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

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

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 $P_d(x) = \frac{d}{2} + \sum^{d}_{k=1} c_k T_k(x)$ 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 $\text{Eval}$ that returns $\hat{c}_k$ with precision $\epsilon := O(c_2^2/d)$, then we have a $(1, a', 144d\sqrt{\pi} ||c||^2_2 + 36e_2 ||c||_1)$-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 $O(d^2 ||c||_1)$ uses of $U$, $U^\dagger$, $C_H \text{NOT}$, $C_H \text{NOT}$, and $O(d^2 ||c||_1)$ multi-controlled single-qubit gates.\(^{30}\) Moreover, we can compute the description of the resulting quantum circuit in deterministic time $\tilde{O}(d^2 ||c||_1 \log(d/e_2))$ and space $O(\max\{s(n), \log(d/e_2)\})$\(^{31}\), also $O(d^2 ||c||_1)$ oracle calls to $\text{Eval}$ with precision $\epsilon$.

Furthermore, our construction straightforwardly extends to any linear (possibly non-Hermitian) operator $A$ by simply replacing $P_d(A)$ with $P_d^{(SV)}(A)$ 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 $P_d(x) = \sum^{(d-1)/2}_{l=0} c_{2l+1} T_{2l+1}(x)$, as described in Theorem 3.11, although the actual implementation results in a slightly different polynomial, $\hat{P}_d(x) = \sum^{(d-1)/2}_{l=0} \hat{c}_{2l+1} T_{2l+1}(x)$. 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 \( k \leq 2^\Omega(s) \), there exists a new \((1,a+1,4k\sqrt{\epsilon})\)-bitstring indexed encoding \( V \) of \( T_k^{(SV)}(A) \) that acts on \( s+a+1 \) qubits. This implementation requires \( k \) uses of \( U \), \( U^\dagger \), \( C_{|\Pi \rangle \langle \Pi|} \), 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 \( \Pi \) are the corresponding orthogonal projectors of the bitstring indexed encoding \( U \). If \( A \) and \( A' \) satisfy the conditions \( \|A - A'\| + \|A + A'\| \leq 1 \) and \( \|A + A'\|^2 \leq \zeta \), then \( V \) is a \((1,a+1,\sqrt{\epsilon}/\sqrt{k})\)-bitstring indexed encoding of \( T_k^{(SV)}(A) \).

**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 \( \prod_{j=1}^{k} \left( \begin{array}{cc} \exp(i\phi_j) & 0 \\ 0 & \exp(-i\phi_j) \end{array} \right) \right) = T_k(x) \).

Then we implement the quantum singular value transformation \( T_k^{(SV)}(A) \), 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 \( P^{(SV)}(\Pi U \Pi) \) with a single ancillary qubit. Moreover, this implementation in [GSLW19, Figure 3] makes \( k \) uses of \( U \), \( U^\dagger \), \( C_{|\Pi \rangle \langle \Pi|} \), and single-qubit gates.

Owing to the robustness of QSVT (Lemma 22 in [GSLW18], full version of [GSLW19]), we have that \( \|T_k^{(SV)}(U) - T_k^{(SV)}(U')\| \leq 4k\sqrt{\|A - A'\|} = 4k\sqrt{\epsilon} \), where \( U' \) is a \((1,a,0)\)-bitstring indexed encoding of \( A \). Moreover, with a tighter bound for \( A \) and \( A' \), namely \( \|A - A'\| + \|A + A'\|^2 \leq 1 \), we can deduce that \( \|T_k^{(SV)}(U) - T_k^{(SV)}(U')\| \leq k \sqrt{2} \sqrt{1 - \|A + A'\|^2/2} \|A - A'\| \leq \sqrt{2} \sqrt{k} \epsilon \) following [GSLW18, Lemma 23], indicating an improved dependence of \( \epsilon \). 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 \( P_{\gamma} \) is an \( \epsilon \)-state preparation operator for \( \gamma \) if \( P_{\gamma}[0] := \sum_{i=1}^{m} \sqrt{y_i} |i\rangle \) for some \( \gamma \) such that \( \|\gamma\|/\|\gamma\|_1 - \gamma_1 \leq \epsilon \).

**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 \((|\gamma\|_1, a, \epsilon_i)\)-bitstring indexed encoding \( U_i \) acting on \( s+a \) qubits associated with projections \( \Pi_i \) and \( \Pi'_i \). Also each \( y_i \) \((1 \leq i \leq m)\) can be expressed in \( O(s(n)) \) bits with an evaluation oracle \( \text{Eval} \) that returns \( y_i^\prime \) with precision \( \epsilon := O(\epsilon_2^2/m) \). Then utilizing an \( \epsilon_2 \)-state preparation operator \( P_{\gamma} \) for \( \gamma \) acting on \( O(\log m) \) qubits, and \( a \) \((s+a+\)
[log m])-qubit unitary $W = \sum_{i=0}^{m-1} |i\rangle\langle i| \otimes U_i + (I - \sum_{i=0}^{m-1} |i\rangle\langle i|) \otimes I$, we can implement a $\left(\|\psi\|_1, a + [\log m], \epsilon_1, |\psi\|_1^2 \right)$-bitstring indexed encoding of $A$ acting on $s + a + [\log m]$ qubits with a single use of $W, P_y, P_y^T$. In addition, the classical pre-processing can be implemented in deterministic time $\tilde{O}(m^2 \log(m/\epsilon))$ and space $O(\log(m/\epsilon^2))$;\(^{31}\) as well as $m^2$ oracle calls to $\text{Eval}$ with precision $\epsilon$.

Proof. For the $\epsilon_2$-state preparation operator $P_y$ such that $P_y|\bar{0}\rangle = \sum_{i=1}^{m} \sqrt{y_i} |i\rangle$, 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 $|\psi\rangle := \sum_{i=1}^{l} \sqrt{y_i} |i\rangle$, where $l = [\log m]$ and $y_i$ are real amplitudes associated with an evaluation oracle $\text{Eval}(i, \epsilon)$ that returns $\hat{y}_i$ up to accuracy $\epsilon$ we can prepare $|\psi\rangle$ up to accuracy $\epsilon$ 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 $\epsilon := O(\epsilon^2/m)$.

Now consider the bitstring indexed encoding $(P_y^T \otimes I_\alpha) W (P_y \otimes I_\alpha)$ of $A$ acting on $s+a+[\log m]$ qubits. Let $y'_i := y_i/\|\psi\|_1$, then we obtain the implementation error:

$$\left\| A - \|\psi\|_1 (|\bar{0}\rangle \langle \bar{0}| \otimes \tilde{I}) (P_y^T \otimes I_\alpha) W (P_y \otimes I_\alpha) (|\bar{0}\rangle \langle \bar{0}| \otimes \tilde{I}) \right\| = \left\| A - \|\psi\|_1 \sum_{i=0}^{m-1} \hat{y}_i \tilde{U}_i \tilde{U}_i \right\| \\
\leq \left\| \|\psi\|_1 \sum_{i=0}^{m-1} \hat{y}'_i \tilde{U}_i \tilde{U}_i \right\| + \epsilon_1 \|\psi\|_1 \sum_{i=0}^{m-1} (y'_i - \hat{y}_i) \|\tilde{U}_i \tilde{U}_i \| \\
\leq \epsilon_1 \|\psi\|_1^2 + \epsilon_2 \|\psi\|_1.
$$

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 $0 \leq i < m$. \(\square\)

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 $\alpha$. 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 $(\alpha, a, \epsilon)$-bitstring indexed encoding of $A$, where $\alpha > 1$ and $0 < \epsilon < 1$, and $A$ is a linear operator acting on $(n)$ 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 $U$, $U^\dagger$, $\text{CNOT}_0$, $\text{CNOT}_1$, 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 $P_d(A) = \frac{a_0}{2} + \sum_{k=1}^{d} c_k T_k(A)$ where $T_k(A)$ corresponding to a $(1, a + 1, 4k\sqrt{c_1})$-bitstring indexed encoding $V_k$. Employing Lemma 3.14, we result in a $\left(\|c\|_1, \hat{a}, 4k\sqrt{c_1}\|c\|_1^2 + \epsilon_2\|c\|_1^2\right)$-bitstring indexed encoding $\hat{V}$ where $\hat{a} := a + [\log d] + 1$. Moreover, by utilizing Lemma 3.15, we obtain a $(1, a', 144k\sqrt{c_1}\|c\|_1^2 + 36\epsilon_2\|c\|_1^2)$-bitstring indexed encoding $V$ acts on $s+a'$ qubits where $a' := a + a + [\log d] + 3$. A direct calculation demonstrates that this implementation makes $\sum_{k=1}^{d} k \cdot O(\|c\|_1) = O(d^2\|c\|_1)$ uses of $U$, $U^\dagger$, $\text{CNOT}_0$, $\text{CNOT}_1$.

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

27
Furthermore, our construction straightforwardly extends to any non-Hermitian \( A \) and \( \hat{A} \) and \( \langle \text{Sign polynomial with space-efficient coefficients applied to bitstring indexed}

Corollary 3.16

the other hand, the logarithmic function is a piecewise-smooth function that is bounded by \( \log \) polynomial with space-efficient coefficients applied to bitstring indexed

Corollary 3.17

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:

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

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 \((1, a + \log d) + 3, 144C_{\text{sgn}}^{2d} e^{\epsilon_1/2} \log^2(d) + 36C_{\text{sgn}} \epsilon_2 \log(d))\)-bitstring indexed encoding \( V \) of \( P_d^{\text{sgn}}(A) \), where \( P_d^{\text{sgn}} \) is a space-efficient bounded polynomial approximation of the sign function specified in Corollary 3.7, and \( C_{\text{sgn}} \) is a universal constant. This implementation requires \( O(d^2 \log d) \) uses of \( U \), \( U^\dagger \), \( C_1 \text{HOT} \), \( C_1 \text{NOT} \), and \( O(d^2 \log d) \) multi-controlled single-qubit gates. Moreover, we can compute the description of \( V \) in deterministic time \( \tilde{O}(\epsilon_2^{-1}d^{3/2}) \) and space \( O(s(n)) \).

Furthermore, our construction straightforwardly extends to any non-Hermitian (but linear) matrix \( A \) by simply replacing \( P_d^{\text{sgn}}(A) \) with \( P_d^{\text{SV}}(A) \) defined in the same way as Definition 3.3.

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

Furthermore, the quantum circuit of \( V \) makes \( O(d^2 \log d) \) uses of \( U \), \( U^\dagger \), \( C_1 \text{HOT} \), and \( C_1 \text{NOT} \) 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 \( \epsilon \) can be computed in \( O(\log(\epsilon_2^{-2}d^5)) \) space. Additionally, the time complexity for computing the description of \( V \) is

\[
\max\{\tilde{O}(d^2 \log d \log(d/\epsilon_2)), d^2 \log d \cdot \tilde{O}(\epsilon_2^{-1}d^{3/2})\} = \tilde{O}(\epsilon_2^{-1}d^{3/2}).
\]

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 \( d \leq 2^{O(s(n))} \),
\[\epsilon_2 \geq 2^{-O(s(n))}, \text{ and } \beta \geq 2^{-O(s(n))}, \] we have a \((1, a + \lceil \log d \rceil, 3, 144\hat{C}_{ln}e^{1/2}d^3 + 36\hat{C}_{ln}e_2d)\)-bitstring indexed encoding \(V\) of \(P_{dn}^U(A)\), where \(P_{dn}^U\) 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 \(U, U^1, C_{l1} \text{NOT}, C_{l2} \text{NOT}, \) 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 \(\tilde{O}(\max\{\beta^{-5}e_2^{-4}d^2, e_2^{-1}d_1^{11/2}\})\) and space \(O(s(n))\).

**Proof.** In Corollary 3.10, we can express \(P_{dn}^U(x)\) as \(a_0 + \sum_{k=1}^{d} c_k T_k(x)\), where \(d = O(\delta^{-1} \log \epsilon^{-1})\). For any \(l_\beta(x)\), we have \(|l_\beta(x) - P_{dn}^U(x)| \leq O(\epsilon/\log d) := \epsilon_2\) for all \(x \in [\beta, 1]\). To implement Eval with precision \(\epsilon\), we can compute the corresponding entry \(c_i\) of the coefficient vector by a bounded-error randomized algorithm. This requires \(O(\log(\epsilon^{-1}d^4\beta^{-4})) = O(\log(\beta^{-4}e_2^{-2}d^3))\) space and \(\tilde{O}(\max\{\beta^{-5}\epsilon^{-2}, \epsilon^{-1/2}d^2\}) = \tilde{O}(\max\{\beta^{-5}e_2^{-4}d^2, e_2^{-1}d_3^{1/2}\})\) time. Using Theorem 3.11, we conclude that \(P_{dn}^U\) has a \((1, a', 144C_{ln}^{1/2}d^3 + 36\hat{C}_{ln}e_2d)\)-bitstring indexed encoding \(V\) that acts on \(s + a'\) qubits, where \(a' := a + \lceil \log d \rceil + 3\) and \(\|c\|_1 \leq \hat{C}_{ln}d\).

Furthermore, the quantum circuit of \(V\) makes \(O(d^3)\) uses of \(U, U^1, C_{l1} \text{NOT}, \) and \(C_{l2} \text{NOT}\) 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 \(\epsilon\) can be computed in \(O(\log(\beta^{-4}e_2^{-2}d^3))\) space. The time complexity for computing the description of \(V\) is given by:

\[
\max\{\tilde{O}(d^3 \log(d/\epsilon_2)), d^3 \cdot \tilde{O}(\max\{\beta^{-5}e_2^{-4}d^2, e_2^{-1}d_3^{1/2}\})\} = \tilde{O}(\max\{\beta^{-5}e_2^{-4}d^5, e_2^{-1}d_1^{11/2}\}). \tag{3.13}
\]

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. \(\square\)

### 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., RQ\(_{d1}\) and coRQ\(_{d1}\)) have been proven in [Wat01], and the two-sided error scenario (e.g., BQ\(_{d1}\)) 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

\[
\text{BQ}_{d1}\text{SPACE}[s(n), a(n), b(n)] \subseteq \text{BQ}_{d1}\text{SPACE}[s(n) + \lceil \log d \rceil + 1, 1, 2^{-l(n)}, 2^{-l(n)}].
\]

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

\[
\text{RQ}_{d1}\text{SPACE}[s(n), a(n)] \subseteq \text{RQ}_{d1}\text{SPACE}[s(n) + \lceil \log d_0 \rceil + 1, 1, 2^{-l(n)}] \quad \text{where} \quad d_0 := \frac{l(n)}{\max\{\sqrt{a} - \sqrt{1 - a}, \sqrt{1 - b} - \sqrt{1 - b}\}},
\]

\[
\text{coRQ}_{d1}\text{SPACE}[s(n), b(n)] \subseteq \text{coRQ}_{d1}\text{SPACE}[s(n) + \lceil \log d_1 \rceil + 1, 1, 2^{-l(n)}] \quad \text{where} \quad d_1 := \frac{l(n)}{\max\{\sqrt{a} - \sqrt{1 - a}, \sqrt{1 - b} - \sqrt{1 - b}\}}.
\]

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 BQ\(_{d1}\), RQ\(_{d1}\), and coRQ\(_{d1}\)). For deterministic logspace computable functions \(a(n), b(n), \) and \(l(n)\) satisfying \(a(n) - b(n) \geq 1/\log(n)\) and \(l(n) \leq 2^{O(s(n))}\), we have that for any \(l(n) \leq 2^{O(s(n))}\):

\[
\text{RQ}_{d1}\text{SPACE}[s(n), a(n)] \subseteq \text{RQ}_{d1}\text{SPACE}[s(n) + \lceil \log d_0 \rceil + 1, 1, 2^{-l(n)}] \quad \text{where} \quad d_0 := \frac{l(n)}{\max\{\sqrt{a} - \sqrt{1 - a}, \sqrt{1 - b} - \sqrt{1 - b}\}},
\]

\[
\text{coRQ}_{d1}\text{SPACE}[s(n), b(n)] \subseteq \text{coRQ}_{d1}\text{SPACE}[s(n) + \lceil \log d_1 \rceil + 1, 1, 2^{-l(n)}] \quad \text{where} \quad d_1 := \frac{l(n)}{\max\{\sqrt{a} - \sqrt{1 - a}, \sqrt{1 - b} - \sqrt{1 - b}\}}.
\]
probability at most

Consider an unknown quantum state $A$ with a singular value either above or below $\sqrt{\log \frac{1}{\epsilon}}$. We can distinguish the two cases with error probability at most $\epsilon := O(\epsilon \log d)$ 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 $O(\log \log d)$ uses of $U$, $U^\dagger$, $C_\Pi$NOT, $C_\bar{\Pi}$NOT, and $O(\log \log d)$ multi-controlled single-qubit gates. Also, we can compute the description of the implementation in deterministic time $O(\epsilon^{-1} b^{(1/2)/\alpha})$ 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 $\mathsf{BQ}_{\mathsf{L}} \mathsf{SPACE}[s(n)]$ circuit $C_x$ accepts is $\Pr[C_x \text{ accepts}] = \| \langle 1 \rangle (1)_{\text{out}} C_x (0^{k+m}) \|_2^2 \geq a$ for yes instances, whereas $\Pr[C_x \text{ accepts}] = \| \langle 1 \rangle (1)_{\text{out}} C_x (0^{k+m}) \|_2^2 \leq b$ 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 $\Pi_{\text{in}} := \langle 0 \rangle \langle 0 \rangle^{k+m}$ and $\Pi_{\text{out}} := \langle 1 \rangle (1)_{\text{out}} \otimes I_{m+k-1}$. Since $\| M_x \| = \sigma_{\text{max}}(M_x)$ where $\sigma_{\text{max}}(M_x)$ 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 $\epsilon := 2^{-(n)}$, 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 $\rho_0$ denote the mixed state obtained by running $Q_1$ on the all-zero state $\langle 0 \rangle$ 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 $\rho_0$ and $\rho_1$ are easily distinguished or almost indistinguishable. Likewise, we also define a space-bounded quantum state certification problem to decide whether $\rho_0$ and $\rho_1$ 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, namely there is a deterministic $O(s(n))$-space Turing machine with time complexity $\ldots$

\[ \text{For instance, the construction in [FL18, Remark 11], or [PY86, BLT92] in general.} \]
poly(s(n)) can uniformly generate the corresponding gate sequences, then Definition 4.1 can be extended to any s(n) satisfying Ω(log n) ≤ s(n) ≤ poly(n).

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, \(\text{GAPQSD}_{\log}\)). Consider deterministic logspace computable functions \(\alpha(n)\) and \(\beta(n)\), satisfying \(0 ≤ \beta(n) < \alpha(n) ≤ 1\) and \(\alpha(n) − \beta(n) ≥ 1/poly(n)\). Then the promise is that one of the following holds:

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

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

Likewise, we can define \(\text{GAPQJS}_{\log}\) and \(\text{GAPQHS}_{\log}\), also the certification version \(\overline{\text{CERTQHS}}_{\log}\), in a similar manner to Definition 4.3 by replacing the distance-like measure accordingly:

- **\(\text{GAPQJS}_{\log}[\alpha(n), \beta(n)]\):** Decide whether \(\text{QJS}_2(\rho_0, \rho_1) ≥ \alpha(n)\) or \(\text{QJS}_2(\rho_0, \rho_1) ≤ \beta(n)\);
- **\(\text{GAPQHS}_{\log}[\alpha(n), \beta(n)]\):** Decide whether \(\text{HS}_2(\rho_0, \rho_1) ≥ \alpha(n)\) or \(\text{HS}_2(\rho_0, \rho_1) ≤ \beta(n)\).

Furthermore, we use the notation \(\overline{\text{CERTQSD}}_{\log}\) to indicate the complement of \(\text{CERTQSD}_{\log}\) with respect to the chosen parameter \(\alpha(n)\), and so does \(\overline{\text{CERTQHS}}_{\log}\).

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

- **Yes instance:** A pair of quantum circuits \((Q_0, Q_1)\) such that \(S(\rho_0) − S(\rho_1) ≥ g(n)\);
- **No instance:** A pair of quantum circuits \((Q_0, Q_1)\) such that \(S(\rho_1) − S(\rho_0) ≥ g(n)\).

**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 \(\text{coRQUL}\)-complete problems. By relaxing the error requirement from one-sided to two-sided, Theorem 4.6 identifies a new family of natural \(\text{BQL}\)-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 \(\alpha(n) ≥ 1/poly(n)\), is as follows:

1. \(\overline{\text{CERTQSD}}_{\log}[\alpha(n)]\) is \(\text{coRQUL}\)-complete;
2. \(\overline{\text{CERTQHS}}_{\log}[\alpha(n)]\) is \(\text{coRQUL}\)-complete.

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

1. \(\text{GAPQSD}_{\log}[\alpha(n), \beta(n)]\) is \(\text{BQL}\)-complete;
2. \(\text{GAPQED}_{\log}[g(n)]\) is \(\text{BQL}\)-complete;
3. \(\text{GAPQJS}_{\log}[\alpha(n), \beta(n)]\) is \(\text{BQL}\)-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. GAPQHS\([\alpha(n), \beta(n)]\) 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 coRQUSPACE\([s(n)]\) and 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 GAPQSD\(_{log}\) in Section 4.2, as well as the BQL containment of GAPQED\(_{log}\) and GAPQJS\(_{log}\) in Section 4.3. Subsequently, in Section 4.4, we focus on making the error one-sided and establish the coRQU\(_L\) containment of CertQSD\(_{log}\) and CertQHS\(_{log}\). Additionally, we show the BQL containment of 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 \(T\). Specifically, the space-efficient tester \(T\) 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 \(\hat{x}\) of \(x\) with high probability through sequential repetition (Lemma 2.11).

To construct \(T\), 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.

\[
\begin{aligned}
|0\rangle &\xrightarrow{H} |0\rangle \\
|\bar{0}\rangle &\xrightarrow{U_{P_d}(A)} |\bar{0}\rangle \\
|0\rangle \otimes r &\xrightarrow{Q} |0\rangle \\
|0\rangle &\xrightarrow{H} x
\end{aligned}
\]

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

**Constructing a space-efficient quantum tester.** We now provide a formal definition and the detailed construction of the quantum tester \(T\). The quantum circuit shown in Figure 2 defines the quantum tester \(T(Q, U_A, P_d, \epsilon)\) 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 \(\rho\) where \(\rho\) 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 \(P_d = \frac{a_d}{2} + \sum_{k=1}^{d} c_k T_k \in \mathbb{R}[x]\) where \(T_k\) is the \(k\)-th Chebyshev polynomial, with \(d \leq 2^{O(s(n))}\), such that the coefficients \(c := (c_0, \cdots, c_d)\) can be computed in bounded-error randomized space \(O(s(n))\);
- \(\epsilon\) 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 \(\hat{T}(Q, U_A, P_d, \epsilon, \epsilon_H, \delta)\), namely a quantum algorithm that computes an additive-error estimation \(\hat{x}\) of the output \(x\) from
the tester \( T(Q, U_A, P_d, \epsilon) \). Technically speaking, \( \hat{T} \) outputs \( \hat{x} \) such that \(|x - \hat{x}| \leq \epsilon \|c\|_1 + \epsilon_H \) with probability at least \( 1 - \delta \). Now we will demonstrate that both the tester \( T \) and the corresponding estimation procedure \( \hat{T} \) are space-efficient:

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

We first provide two useful lemmas for implementing our quantum tester \( T \). 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 \( \rho \) 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 \( \rho = \text{Tr}_a(|\rho\rangle \langle \rho|) \). Then, we can construct an \((a + s)\)-qubit quantum circuit \( \hat{U} \) that is an \( O(0.0) \)-block-encoding of \( \rho \), 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 \( \rho \), outputs 0 with probability \( \frac{1 + \text{Re}(\text{Tr}(A\rho))}{2} \).

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 \( O(d^2 \|c\|_1) \) queries to \( U_A \), where \( a = s - r + \lceil \log d \rceil + 3 \) and \( A'_{P_d} \) is specified in Theorem 3.11 satisfying \( \|P_d(A) - A'_{P_d}\| \leq \epsilon \|c\|_1 \). 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 \( c \) of \( P_d \). As the quantum tester \( T(Q, U_A, P_d, \epsilon) \) is mainly based on the Hadamard test, by employing Lemma 4.9, we have that \( T \) outputs 0 with probability \[ \Pr[x = 0] = \frac{1}{2} (1 + \text{Re}(\text{Tr}(A'_{P_d} \rho))) = \frac{1}{2} (1 + \text{Re}(\text{Tr}(P_d(A)\rho))) \leq \epsilon \|c\|_1. \]

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

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

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

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

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

4.2 GapQSD\textsubscript{log} is in BQL

In this subsection, we demonstrate Theorem 4.10 by constructing a quantum algorithm that incorporates testers \( T(Q, U_{\frac{m-1}{m-2}}, P^\text{sgn}_d, \epsilon) \) 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 $\alpha(n) - \beta(n) \geq 1/\text{poly}(n)$, we have that $\text{GapQSD}_{\log}[\alpha(n), \beta(n)]$ is in $\text{BQL}$.

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

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

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

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

**Params**: $\varepsilon := \frac{2-\beta}{4}$, $\delta := \frac{\varepsilon}{2^{2+\gamma}}$, $\epsilon := \frac{\varepsilon}{2(2C_{\text{sgn}} + 2C_{\text{sgn}})C_{\text{sgn}}2^{r+3} + 1}$, $d := \tilde{C}_{\text{sgn}}\delta^{-1}\log^{-1}\varepsilon^{-1}$, $\epsilon_H := \frac{\varepsilon}{7}$.

1. Construct block-encodings of $\rho_0$ and $\rho_1$, denoted by $U_{\rho_0}$ and $U_{\rho_1}$, 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 $\rho_{0-\rho_1}$, denoted by $U_{\rho_0-\rho_1}$, using $O(1)$ queries to $U_{\rho_0}$ and $U_{\rho_1}$ and $O(s(n))$ ancillary qubits by Lemma 3.14.

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

3. Set $x_0 := \tilde{T}(Q_0, U_{\rho_0-\rho_1}, P_{d}^{\text{sgn}}, \epsilon, \epsilon_H, 1/10)$, $x_1 := \tilde{T}(Q_1, U_{\rho_0-\rho_1}, P_{d}^{\text{sgn}}, \epsilon, \epsilon_H, 1/10)$.
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 $\rho_0$ and $\rho_1$, respectively. According to Lemma 4.8, we can construct $O(s)$-qubit quantum circuits $U_{\rho_0}$ and $U_{\rho_1}$ that encode $\rho_0$ and $\rho_1$ 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 $Q_{\rho_0}$ and $Q_{\rho_1}$, as well as $O(1)$ one- and two-qubit quantum gates.

Let $\delta := \frac{\varepsilon}{2^{2+\gamma}}$, $\epsilon := \frac{\varepsilon}{2(2C_{\text{sgn}} + 2C_{\text{sgn}})C_{\text{sgn}}2^{r+3} + 1}$ and $d := \tilde{C}_{\text{sgn}}\delta^{-1}\log^{-1}\varepsilon^{-1} = 2^{O(s)}$ where $\tilde{C}_{\text{sgn}}$ comes from Corollary 3.7. Let $P_{d}^{\text{sgn}} \in \mathbb{R}[x]$ be the polynomial specified in Corollary 3.7. Let $\epsilon_H = \varepsilon/4$. By employing Corollary 3.16 and the corresponding estimation procedure $\tilde{T}(Q_1, U_{\rho_0-\rho_1}, P_{d}^{\text{sgn}}, \epsilon, \epsilon_H, 1/10)$ from Lemma 4.7, we obtain the values $x_i$ for $i \in \{0, 1\}$, ensuring the following inequalities:

$$\Pr\left[|x_i - \text{Tr}\left(P_{d}^{\text{sgn}} \left(\frac{\rho_0 - \rho_1}{2}\right)\rho_i\right)| \leq \tilde{C}_{\text{sgn}}\epsilon \log d + \epsilon_H\right] \geq \frac{9}{10} \text{ for } i \in \{0, 1\}. \tag{4.11}$$

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 $\tilde{T}(Q_1, U_{\rho_0-\rho_1}, P_{d}^{\text{sgn}}, \epsilon, \epsilon_H, 1/10)$ can be computed in deterministic time $\tilde{O}(d^{9/2}/\epsilon)$ and space $O(\log s)$. 

Now let $x := (x_0 - x_1)/2$. We will finish the correctness analysis of Algorithm 1 by showing $\Pr[|x - \text{td}(\rho_0, \rho_1)|] \leq \varepsilon > 0.8$ through Equation (4.11). By considering the approximation error of $P_{d}^{\text{sgn}}$ 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.** $\Pr[|x - \text{td}(\rho_0, \rho_1)|] \leq \tilde{C}_{\text{sgn}}\epsilon \log d + \epsilon_H + 2\tilde{C}_{\text{sgn}}\epsilon \log d + 2^{r+1}\delta > 0.8$. 

34
Consequently, it is left to show that \( \hat{C}_{\mathrm{sgn}} \epsilon \log d + \epsilon_H + 2C_{\mathrm{sgn}} \epsilon \log d + 2^r + 1 \delta \leq \epsilon \) for the aforementioned choice of \( \delta, \epsilon, \) and \( \epsilon_H \). Note that \( \epsilon_H = \epsilon/4 \) and \( 2^r + 1 \delta = \epsilon/4 \), we complete the correctness analysis by choosing \( \epsilon := \delta'/2 \log(\delta'-1) \) with \( \delta' := \delta/(2(\hat{C}_{\mathrm{sgn}} + 2C_{\mathrm{sgn}})\hat{C}_{\mathrm{sgn}}) \leq 1/2 \) and subsequently deriving the following inequality:

\[
(\hat{C}_{\mathrm{sgn}} + 2C_{\mathrm{sgn}}) \epsilon \log d \leq (\hat{C}_{\mathrm{sgn}} + 2C_{\mathrm{sgn}}) \epsilon \log(\delta'-1 \log(\epsilon^{-1})) \leq (\hat{C}_{\mathrm{sgn}} + 2C_{\mathrm{sgn}})\delta' \leq \epsilon/2.
\]

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

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

\[\square\]

4.3 \text{GAPQED}_{\log} and \text{GAPQJS}_{\log} are in \text{BQL}

In this subsection, we will demonstrate Theorem 4.11 by devising a quantum algorithm that encompasses testers \( T(Q_i, U_{\rho_i}, P_{d_i}^\text{lin}, \epsilon) \) 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 \text{GAPQJS}_{\log} \) is in \text{BQL} via a reduction from \text{GAPQJS}_{\log} to \text{GAPQED}_{\log}.

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

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

\begin{algorithm}
\textbf{Input} : Quantum circuits \( Q_i \) that prepares the purification of \( \rho_i \) for \( i \in \{0,1\} \).
\textbf{Output} : An additive-error estimation of \( S(\rho_0) - S(\rho_1) \).
\textbf{Params} : \( \epsilon := \frac{\epsilon}{2}, \beta := \min\left\{ \frac{\epsilon}{2^{r+\log(2\epsilon/\beta)}, 1/2}, d := \tilde{C}_\text{ln} \cdot \frac{4\log \epsilon}{\beta \epsilon}, \epsilon := \frac{\beta \epsilon}{4\log \tilde{C}_\text{ln}(\tilde{C}_\text{ln}+\tilde{C}_\text{ln}) \log(1/\beta)} \cdot \epsilon_H \cdot \frac{1}{8\log(1/\beta)} \right\}, \epsilon_H := \frac{\epsilon}{8\log(1/\beta)}.

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

Let \( P_{d_i}^\text{lin} \) be the degree- \( d \) polynomial specified in Corollary 3.10 with parameters \( \delta \) and \( \epsilon \) such that its Chebyshev coefficients are computable in bounded-error randomized space \( O(\log(d/\epsilon)) \);

2. Set \( x_0 := \tilde{T}(Q_0, U_{\rho_0}, P_{d_i}^\text{lin}, \epsilon, \epsilon_H, 1/10), x_1 := \tilde{T}(Q_1, U_{\rho_1}, P_{d_i}^\text{lin}, \epsilon, \epsilon_H, 1/10) \);

3. Compute \( x = (x_0 - x_1) \ln(2/\beta) \). Return “yes” if \( x > 0 \), and “no” otherwise.

\end{algorithm}

Let us now demonstrate the correctness and computational complexity of Algorithm 2. We concentrate on the scenario with \( s(n) = \Theta(\log n) \) and \( \epsilon = g/4 \geq 2^{-O(s)} \). Our strategy is to estimate the entropy of each of \( \rho_0 \) and \( \rho_1 \), respectively. We assume that \( Q_0 \) and \( Q_1 \) are \( s \)-qubit quantum circuits that prepare the purifications of \( \rho_0 \) and \( \rho_1 \), respectively. By Lemma 4.8, we can construct \( (1, O(s), 0) \)-block-encodings \( U_{\rho_0} \) and \( U_{\rho_1} \) for \( \rho_0 \) and \( \rho_1 \), respectively, using \( O(1) \) queries to \( Q_0 \) and \( Q_1 \) as well as \( O(1) \) one- and two-qubit quantum gates.

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

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

Here, the implementation uses $O(d^3)$ queries to $U\rho_0$ and $O(d^3)$ multi-controlled single-qubit gates. Moreover, the circuit descriptions of $\hat{T}(Q_i, U, P_{d}^{\rho_i}, \epsilon, \epsilon_H, 1/10)$ can be computed in bounded-error time $O(d^3/\epsilon^4)$ and space $O(s)$.

We will finish the correctness analysis of Algorithm 2 by demonstrating $\Pr[|x_i - S(\rho_i)| \leq \epsilon] \geq 0.9$ through Equation (4.2). By considering the approximation error of $P_{d}^{\rho_i}$ 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 $i \in \{0, 1\}$:

$$\Pr\left[|x_i \ln \left( \frac{2}{\hat{x}_i} \right) - S(\rho_i)| \leq 2 \ln \left( \frac{2}{\hat{x}_i} \right) \left( \hat{C}_i \epsilon + \epsilon_H + C_i \epsilon \log d + 2^{r \epsilon + 1} \beta \right) \right] \geq \frac{9}{10}. \quad (4.3)$$

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

$$2 \ln(\beta^{-1})(\hat{C}_i \epsilon + C_i \epsilon \log(d)) \leq 2 \ln(\beta^{-1})(\hat{C}_i + C_i) \epsilon \log(d) = 2 \ln(\beta^{-1})(\hat{C}_i + C_i) \hat{C}_i \beta^{-1} \epsilon \log(\epsilon^{-1}) \leq 2 \ln(\beta^{-1})(\hat{C}_i + C_i) \hat{C}_i \beta^{-1} \delta = \epsilon/2. \quad \text{Here, the first line is because of } \log(d) \leq d, \text{ the third line owes to the fact that } \epsilon \log(\epsilon^{-1}) \leq \delta, \text{ and the last line is due to choice of } \delta.$$

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 $O(d^3/\epsilon_H^2) = O(2^{5r}/\epsilon^4)$ queries to $Q_0$ and $Q_1$. Furthermore, its circuit description can be computed in bounded-error randomized time $\hat{O}(d^{11}/\epsilon^4) = \hat{O}(2^{11r}/\epsilon^{15})$. \hfill $\square$

**GAPQJS$_{\log}$ is in BQL.** It is noteworthy that we can achieve GAPQJS$_{\log} \in$ BQL by employing the estimation procedure $\hat{T}$ in Algorithm 2 for three according states, given that the quantum Jensen-Shannon divergence $QJS(\rho_0, \rho_1)$ is a linear combination of $S(\rho_0), S(\rho_1)$, and $S\left(\frac{\rho_0 + \rho_1}{2}\right)$. Nevertheless, the log-space Karp reduction from GAPQJS$_{\log}$ to GAPQED$_{\log}$ (Corollary 4.12) allows us to utilize $\hat{T}$ 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 $\alpha(n) - \beta(n) \geq 1/\text{poly}(n)$, we have that 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 $\rho'_i := \frac{1}{2} |0\rangle \langle 0| \otimes \rho_0 + \frac{1}{2} |1\rangle \langle 1| \otimes \rho_1$, where $\rho_0$ and $\rho_1$ 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 $\rho'_0$ and $\rho'_1$, which are the output of $(s(n) + 2)$-qubit quantum circuits $Q'_0$ and $Q'_1$. 

36
and $Q'_1$, 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 $B' := (p_0, p_1)$ is an independent random bit with entropy $H(B') = 1 - \frac{1}{2}[\alpha(n) + \beta(n)]$. Let $S_2(\rho) := S(\rho)/\ln 2$ for any quantum state $\rho$, then we have derived that:

$$S_2(\rho'_0) - S_2(\rho'_1) = S_2(B', Y)\rho'_0 - S_2(B, Y)\rho'_1$$

$$= [H(B') + S_2(Y|B')\rho'_0] - [H(B) + S_2(Y|B)\rho'_1]$$

$$= S_2(Y)\rho'_0 - S_2(Y|B)\rho'_1 + H(B') - H(B)$$

$$= S_2(Y)\rho'_0 - S_2(Y|B)\rho'_1 - \frac{1}{2}[\alpha(n) + \beta(n)]$$

$$= S_2(\rho_0(1 + \frac{1}{2}(p_0 + p_1)) - \frac{1}{2}(S_2(\rho_0) + S_2(\rho_1)) - \frac{1}{2}[\alpha(n) + \beta(n)]$$

$$= QJS_2(\rho_0, p_1) - \frac{1}{2}[\alpha(n) + \beta(n)].$$

(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 $\text{GapQJS}_{\text{log}}[\alpha(n), \beta(n)]$, we can define $g(n) := \ln 2 (\alpha(n - 1) - \beta(n - 1))$ and conclude that:

- If $\text{QJS}_2(\rho_0, \rho_1) \geq \alpha(n)$, then $S(\rho'_0) - S(\rho'_1) \geq \ln 2 (\alpha(n) - \beta(n)) = g(n + 1)$;
- If $\text{QJS}_2(\rho_0, \rho_1) \leq \beta(n)$, then $S(\rho'_0) - S(\rho'_1) \leq -\ln 2 (\alpha(n) - \beta(n)) = -g(n + 1)$.

As $\rho'_1$ and $\rho'_0$ 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, $\text{GapQJS}_{s(n)}[\alpha(n), \beta(n)]$ is logspace Karp reducible to $\text{GapQED}_{s+1}[g(n)]$ by mapping $(Q_0, Q_1)$ to $(Q'_0, Q'_1)$. □

4.4 $\text{CertQSD}_{\text{log}}$ and $\text{CertQHS}_{\text{log}}$ are in $\text{coROU}_L$

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 $U|0\rangle = \sin(\theta)|\psi_0\rangle + \cos(\theta)|\psi_1\rangle$, where $|\psi_0\rangle$ and $|\psi_1\rangle$ are normalized pure states and $\langle \psi_0|\psi_1\rangle = 0$. Let $G = -U(I - 2|0\rangle\langle 0|)U^\dagger(I - 2|\psi_0\rangle\langle \psi_0|)$ be the Grover operator. Then, for every integer $j \geq 0$, we have $G^jU|0\rangle = \sin((2j + 1)\theta)|\psi_0\rangle + \cos((2j + 1)\theta)|\psi_1\rangle$. In particular, with a single application of $G$, we obtain $GU|0\rangle = \sin(3\theta)|\psi_0\rangle + \cos(3\theta)|\psi_1\rangle$, signifying that $GU|0\rangle = |\psi_0\rangle$ 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 $T$ from Lemma 4.7), Lemma 4.13 guarantees that the resulting algorithm $A$ accepts with probability exactly 1 for yes instances ($\rho_0 = \rho_1$). However, achieving $A$ to accept with probability polynomially deviating from 1 for no instances requires additional efforts, leading to the $\text{coROU}_L$ containment established through error reduction for $\text{coROU}_L$ (Corollary 3.19). In a nutshell, demonstrating $\text{coROU}_L$ containment entails satisfying the desired property, which is achieved differently for $\text{CertQSD}_{\text{log}}$ and $\text{CertQHS}_{\text{log}}$.

---

34To construct $Q'_1$, we follow these steps: We start by applying a $\text{HADAMARD}$ gate on $B$ followed by a $\text{CNOT}_{B \rightarrow R}$ 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 $S = (Y, Z)$ is an s(n)-qubit register initialized on $|0\rangle$. 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 $\rho'_1$ by tracing out $R$ and the qubits in $Z$. In addition, we can construct $Q'_0$ similarly.
4.4.1 CertQSD\textsubscript{log} is in coRQU\textsubscript{L}

Our algorithm in Theorem 4.14 relies on the quantum tester \( T(Q_i, U_{2\alpha-\rho}, P_d^{\text{sgn}}, \epsilon) \) 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, \( T(Q_i, U_{2\alpha-\rho}, P_d^{\text{sgn}}, \epsilon) \) outputs 0 with probability exactly 1/2 when \( \rho_0 = \rho_1 \), enabling us to derive the coRQU\textsubscript{L} containment through a relatively involved analysis for cases when \( td(\rho_0, \rho_1) \geq \alpha \).

**Theorem 4.14.** For any deterministic logspace computable function \( \alpha(n) \geq 1/\text{poly}(n) \), we have that CertQSD\textsubscript{log}[\alpha(n)] is in coRQU\textsubscript{L}.

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

**Algorithm 3:** Space-efficient algorithm for CertQSD\textsubscript{log}.

**Input:** Quantum circuits \( Q_i \) that prepares the purification of \( \rho_i \) for \( i \in \{0, 1\} \).

**Output:** Return “yes” if \( \rho_0 = \rho_1 \), and “no” otherwise.

**Params:** \( \epsilon := \frac{\alpha}{4}, \delta := \frac{\epsilon}{2d+13}, \epsilon := \frac{\epsilon}{2(C_{\text{sgn}}+2C_{\text{sgn}})C_{\text{sgn}}2^{r+3}/\epsilon} \) and \( d := \tilde{C}_{\text{sgn}}\delta^{-1}\log \epsilon^{-1} \).

1. Construct block-encodings of \( \rho_0 \) and \( \rho_1 \), denoted by \( U_{\rho_0} \) and \( U_{\rho_1} \), 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 \( \rho_{2\alpha-\rho} \), denoted by \( U_{2\alpha-\rho} \), using \( O(1) \) queries to \( U_{\rho_0} \) and \( U_{\rho_1} \) and \( O(s(n)) \) ancillary qubits by Lemma 3.14.
3. Let \( P_d^{\text{sgn}} \) be the degree-\( d \) odd polynomial specified in Corollary 3.7 with parameters \( \delta \) and \( \epsilon \) such that its Chebyshev coefficients are computable in deterministic space \( O(\log(d/\epsilon)) \).
4. Let \( G_i := (H \otimes U_i)(I - 2|0\rangle\langle 0|)(I - 2I\Pi_0) \) for \( i \in \{0, 1\} \), where \( \Pi_0 \) is the projector onto the subspace spanned by \( \{|0\rangle\langle 0|\} \) over all \( |\varphi\rangle \).
5. Measure the first two qubits of \( G_i(I \otimes U_i)|0\rangle\langle 0|0\rangle \), and let \( x_{d0} \) and \( x_{d1} \) be the outcomes, respectively. Return “yes” if \( x_{d0} = x_{d1} = x_{10} = x_{11} = 0 \), and “no” otherwise.

**Constructing the unitary of interest via the space-efficient QSVT.** We consider the setting with \( s(n) = \Theta(\log n) \) and \( \epsilon = \alpha/2 \). Suppose \( Q_0 \) and \( Q_1 \) are \( s(n) \)-qubit quantum circuits that prepare the purifications of \( \rho_0 \) and \( \rho_1 \), respectively. Similar to Algorithm 1, we first construct an \( O(s) \)-qubit quantum circuit \( U_{2\alpha-\rho} \) that is a \( (1, O(s), 0) \)-block-encoding of \( \rho_{2\alpha-\rho} \), using \( O(1) \) queries to \( Q_0, Q_1, P_{\rho_0}, P_{\rho_1} \) and \( O(1) \) one- and two-qubit quantum gates. Let \( \delta := \frac{\epsilon}{2d+13}, \epsilon := \frac{\epsilon}{2(C_{\text{sgn}}+2C_{\text{sgn}})C_{\text{sgn}}2^{r+3}/\epsilon} \) and \( d := \tilde{C}_{\text{sgn}}\delta^{-1}\log \epsilon^{-1} = 2^{O(s)} \) where \( \tilde{C}_{\text{sgn}} \) comes from Corollary 3.7. Let \( P_d^{\text{sgn}} \in \mathbb{R}[x] \) be the odd polynomial specified in Corollary 3.7. Let \( U_i := T(Q_i, U_{2\alpha-\rho}, P_d^{\text{sgn}}, \epsilon) \) for \( i \in \{0, 1\} \), then we have the following equalities with \( 0 \leq p_0, p_1 \leq 1 \):

\[
U_{\rho_0}|0\rangle\langle 0| = \sqrt{p_0}|0\rangle\langle 0| + \sqrt{1 - p_0}|1\rangle\langle 1|,
\]

\[
U_{\rho_1}|0\rangle\langle 0| = \sqrt{p_1}|0\rangle\langle 0| + \sqrt{1 - p_1}|1\rangle\langle 1|.
\]

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

\[
(H \otimes U_i)|0\rangle\langle 0| = \sqrt{p_i/2}|0\rangle\langle 0| + \sqrt{p_i/2}|1\rangle\langle 1| + \sqrt{1 - p_i/2}|1\rangle\langle 0| + \sqrt{1 - p_i/2}|0\rangle\langle 1| \cdot \sqrt{1 - p_i/2} |\psi_i\rangle.
\]
Making the error one-sided by exact amplitude amplification. Consider the Grover operator \(G_i := -(H \otimes U_i)(I - 2|0\rangle\langle 0|)(H \otimes U_i^\dagger)(I - 2\Pi_0), \) where \(\Pi_0\) is the projector onto the subspace spanned by \(|0\rangle\langle 0|\) over all \(|\varphi\rangle\). By employing the exact amplitude amplification (Lemma 4.13), we can obtain that:

\[
G_i(H \otimes U_i)|0\rangle\langle 0| = \sin(3\theta_i)|0\rangle\langle 0|\psi_0) + \cos(3\theta_i)|\perp_i\rangle \text{ where } \sin^2\theta_i = \frac{1}{2} \text{ when } \theta_i \in [0, \frac{\pi}{4}]. \tag{4.4}
\]

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

- For yes instances \((\rho_0 = \rho_1), U_{p_{\text{yes}}}^\rho(\varphi_{0} - \varphi_1)\) is a \((1, O(s), 0)\)-block-encoding of the zero operator, following from Remark 3.12. Consequently, \(T(Q_i, U_{\alpha - \rho_1}^\rho, P_d, \epsilon)\) 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 \(x_{i0} = x_{i1} = x_{10} = x_{11} = 0\) with certainty, which completes the analysis.

- For no instances \((\text{td}(\rho_0, \rho_1) \geq \alpha)\), \(U_{p_{\text{yes}}}^\rho(\varphi_{0} - \varphi_1)\) is a \((1, O(s), 0)\)-block-encoding of \(A\) satisfying \(\|A - P_d^{\rho_0}(\varphi_{0} - \varphi_1)\| \leq \hat{C}_{\text{sgn}}\epsilon \log d\). Let \(p_1\) be the probability that \(T(Q_i, U_{\alpha - \rho_1}^\rho, P_d, \epsilon)\) outputs 0 for \(i \in \{0, 1\}\), then \(p_1 = \frac{1}{2}(1 + \text{Re}(\text{Tr}(\rho_i A)))\) following from Lemma 4.7. A direct calculation similar to Proposition 4.10.1 indicates that:

\[
|(p_0 - p_1) - \text{td}(\rho_0, \rho_1)| \leq \hat{C}_{\text{sgn}}\epsilon \log d + 2C_{\text{sgn}}\epsilon \log d + 2^{-r+1}\delta.
\]

Under the choice of \(\delta, \epsilon, \) and \(d\) in the proof of Theorem 4.11, we obtain that \(|(p_0 - p_1) - \text{td}(\rho_0, \rho_1)| \leq \epsilon\) which yields that \(\text{max}\{|p_0 - 1/2|, |p_1 - 1/2|\} \geq \epsilon/2\).\(^{35}\)

Note that \(\text{Pr}[x_{i0} = x_{i1} = 0] = \sin^2(3\theta_i)\) for \(i \in \{0, 1\}\), Algorithm 3 will return “yes” with probability \(p_{\text{yes}} = \sin^2(3\theta_0)\sin^2(3\theta_1)\). We provide an upper bound for \(p_{\text{yes}}\) 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 \(\text{max}\{|p_0 - 1/2|, |p_1 - 1/2|\} \geq \epsilon/2\), then \(f(\theta_0, \theta_1) \leq 1 - \epsilon^2/4 = 1 - \alpha^2/16.\)

Consequently, we finish the analysis by noticing \(p_{\text{yes}} = f(\theta_0, \theta_1) \leq 1 - \epsilon^2/4 = 1 - \alpha^2/16.\)

Now we analyze the complexity of Algorithm 3. Following Lemma 4.7, we can compute \(x_{i0}, x_{i1}, x_{10}, x_{11}\) in BQL. The quantum circuit that computes \(x_{i0}, x_{i1}, x_{10}, x_{11}\) takes \(O(d^2 \log d) = \hat{O}(2^{d/2}/\alpha^2)\) queries to \(Q_0\) and \(Q_1\), and its circuit description can be computed in deterministic time \(\hat{O}(d^{9/2}/\alpha) = \hat{O}(2^{d/5}/\alpha^{5.5})\). Finally, we conclude the \(\text{coRQUL}\) containment of \(\text{CERTQSD}_{\text{log}}\) by applying error reduction for \(\text{coRQUL}(\text{Corollary 3.19})\) to Algorithm 3.

4.4.2 \(\text{CERTQHS}_{\text{log}}\) is in \(\text{coRQUL}\)

Our algorithm in Theorem 4.15 is based on the observation that by expressing \(\text{HS}^2(\rho_0, \rho_1)\) as a summation of \(\frac{1}{2}\text{Tr}(\rho_0^2), \frac{1}{2}\text{Tr}(\rho_1^2),\) and \(-\text{Tr}(\rho_0 \rho_1)\), 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 \(\alpha(n) \geq 1/\text{poly}(n)\), we have that \(\text{CERTQHS}_{\text{log}}[\alpha(n)]\) is in \(\text{coRQUL}\).

**Proof.** We first provide a formal algorithm in Algorithm 4.\(^{35}\) This inequality is because \(|p_0 - p_1| \geq \text{td}(\rho_0, \rho_1) - \epsilon \geq 2\epsilon - \epsilon = \epsilon.\)
Algorithm 4: Space-efficient algorithm for \text{CertQHS}_{\log}.

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

\textbf{Output}: Return “yes” if $\rho_0 = \rho_1$, and “no” otherwise.

1. Construct subroutines $T_{ij} := \text{SWAP}(\rho_i, \rho_j)$ for $(i,j) \in \{(0,0), (1,1), (0,1)\}$, which output 0 with probability $p_{ij}$. The subroutine $\text{SWAP}(\rho_i, \rho_j)$ involves applying $Q_i$ and $Q_j$ to prepare quantum states $\rho_i$ and $\rho_j$, respectively, and then employing the SWAP test (Lemma 2.12) on these states $\rho_i$ and $\rho_j$.

2. Construct a block-encoding of $\rho(\frac{1}{2} + \frac{\text{HS}^2(\rho_0, \rho_1)}{4})$ where $\rho(p) := p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|$, 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 $GU|0\rangle$ in the computational basis. Return “yes” if the measurement outcome is an all-zero string, and “no” otherwise.

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

We define $T_{ij} := \text{SWAP}(\rho_i, \rho_j)$ for $(i,j) \in \mathcal{I} := \{(0,0), (1,1), (0,1)\}$, with the control qubit in $\text{SWAP}(\rho_i, \rho_j)$ serving as the output qubit of $T_{ij}$. By introducing another ancillary qubit, we construct $T_{ij}' := \text{CNOT}(I \otimes T_{ij})$ for $(i,j) \in \mathcal{I}$, where $\text{CNOT}$ 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 $\rho(p_{ij})$ with $\rho(p_{ij}) := p_{ij}|0\rangle\langle 0| + (1-p_{ij})|1\rangle\langle 1|$ 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 $(1,0,0)$-block-encoding of $\rho(p_{ij})$, using $O(1)$ queries to $T_{ij}'$ and $O(1)$ one- and two-qubit quantum gates. Notably, $(X \otimes I)T_{00}'$, with $X$ acting on the qubit of $\rho(p_2)$, prepares the purification of $X\rho(p_0)X^\dagger = p_0|1\rangle\langle 1| + (1-p_0)|0\rangle\langle 0| = \rho(1-p_0)$, leading to the equality:

$$\rho(p_0, p_1) := \frac{1}{4}\rho(p_0) + \frac{1}{4}\rho(p_1) + \frac{1}{2}\rho(1-p_0) = \rho\left(\frac{1}{2} + \frac{\text{HS}^2(\rho_0, \rho_1)}{4}\right).$$

Consequently, we employ Lemma 3.14 to construct a unitary quantum circuit $U$ that is a $(1,m,0)$-block-encoding of $\rho(\frac{1}{2} + \frac{\text{HS}^2(\rho_0, \rho_1)}{4})$ 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:

$$U|0\rangle|0\rangle^\otimes m = \left(\frac{1}{2} + \frac{\text{HS}^2(\rho_0, \rho_1)}{4}\right)|0\rangle|0\rangle^\otimes m + \cos(\theta)|\perp\rangle,$$

where $|0\rangle|0\rangle^\otimes m|\perp\rangle = 0$. \hfill (4.5)

\text{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 $\text{HS}^2(\rho_0, \rho_1) = 0$. 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”.

40
• For no instances, we have \( HS^2(\rho_0, \rho_1) \geq \alpha \). According to Equation (4.5), we derive that:

\[
\sin(\theta) = \frac{1}{2} + \frac{HS^2(\rho_0, \rho_1)}{4} \geq \frac{1}{2} + \frac{\alpha}{4} \quad \text{and} \quad \frac{1}{4} \leq \sin^2(\theta) = \left( \frac{1}{2} + \frac{HS^2(\rho_0, \rho_1)}{4} \right)^2 \leq \left( \frac{1}{2} + \frac{\alpha}{4} \right)^2 = \frac{9}{16}. \quad (4.6)
\]

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 \( \alpha \leq 1 \) and then we know that the acceptance probability \( \Pr[\text{acceptance probability}] \) 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 \( 0 \leq \alpha \leq 1 \).

Combining Equation (4.6) and Proposition 4.15.1, we have that \( \Pr[\text{acceptance probability}] \) is \( O(1) \) queries to \( Q_0 \) and \( Q_1 \). Finally, we finish the proof by applying error reduction from \( \text{coRQ}_U \) (Corollary 3.19) to Algorithm 3.

\[\square\]

### 4.5 BQL- and \( \text{coRQ}_U \)-hardness for space-bounded state testing problems

We will prove that space-bounded state testing problems mentioned in Theorem 4.6 are \( \text{BQL} \)-hard, which implies their \( \text{BQL} \)-hardness since \( \text{BQL} = \text{BQL} \) [FR21]. Similarly, all space-bounded state certification problems mentioned in Theorem 4.5 are \( \text{coRQ}_U \)-hard.

#### 4.5.1 Hardness results for \( \text{GapQSD}_{\log} \), \( \text{GapQHS}_{\log} \), and their certification version

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

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

**Proof.** Consider a promise problem \( (L_{\text{yes}}, L_{\text{no}}) \in \text{BQL}[a(n), b(n)] \), then we know that the acceptance probability \( \Pr[\text{acceptance probability}] \) for \( x \in L_{\text{yes}} \), whereas \( \Pr[\text{acceptance probability}] \) for \( x \in L_{\text{no}} \). Now we notice that the acceptance probability is the fidelity between a single-qubit pure state \( \rho_0 \) and a single-qubit mixed state \( \rho_1 \) that generates by two logarithmic-qubit quantum circuits \( Q_0 \) and \( Q_1 \), respectively:

\[
\Pr[\text{acceptance probability}] = ||1\rangle \langle 1\text{out}|C_x|0\rangle||^2 = \text{Tr}\left(1\rangle \langle 1\text{out}|\text{Tr}_{\text{out}}(C_x|0\rangle \langle 0|C_x^\dagger)\right) = F^2\left(1\rangle \langle 1\text{out}|\text{Tr}_{\text{out}}(C_x|0\rangle \langle 0|C_x^\dagger)\right)
\]

(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 \( \rho_0 \) and \( \rho_1 \) by tracing out all non-output qubits. By utilizing Lemma 2.2, we have derived that:

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

Therefore, we demonstrate that \( \text{GapQSD}_{\log}[1 - \sqrt{a(n)}, \sqrt{1 - b(n)}] \) is \( \text{BQL}[a(n), b(n)] \)-hard. \( \square \)

To construct pure states, adapted from the construction in Lemma 4.16, we replace the final measurement in the \( \text{BQL} \) 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.

41
Lemma 4.17 ($\text{GapQHS}_{\log}$ is BQUL-hard). For any deterministic logspace computable functions $a(n)$ and $b(n)$ such that $a(n) - b(n) \geq 1/\text{poly}(n)$, we have that $\text{GapQHS}_{\log}[1 - a^2(n), 1 - b^2(n)]$ is BQUL-$[a(n), b(n)]$-hard.

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

$$C_x^\prime := C_x^O \text{CNOT}_O \to F X_O C_x.$$  

And we say that $C_x^\prime$ 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:

$$\Pr[C_x^\prime \text{ accepts}] = (\langle \langle \bar{0} | 0 \rangle \otimes | 0 \rangle_F) C_x^O \text{CNOT}_O \to F X_O C_x(\bar{0} \otimes | 0 \rangle_F) \|^2 = (\langle \langle \bar{0} \otimes | 0 \rangle_F | 1 \rangle_O \otimes I_F + | 0 \rangle_O \otimes X_F) C_x(\bar{0} \otimes | 0 \rangle_F) \|^2 = (\langle \langle \bar{0} \otimes | 0 \rangle_F | 1 \rangle_O C_x | 0 \rangle_F \|^2 = \Pr^2[C_x \text{ accepts}].$$

Here, the second line owes to CNOT$_O \to F = | 0 \rangle_O \otimes I_F + | 1 \rangle_O \otimes X_F$, and the last line is because of Equation (4.7). Interestingly, by defining two pure states $\rho_0 := | 0 \rangle \otimes | 0 \rangle_F$ and $\rho_1 := C_x^O | 0 \rangle \otimes | 0 \rangle_F$ corresponding to $Q_0 = I$ and $Q_1 = C_x^O$, respectively, we deduce the following from Equation (4.8):

$$\Pr[C_x^\prime \text{ accepts}] = \text{Tr}(\rho_0 \rho_1) = 1 - \text{HS}^2(\rho_0, \rho_1).$$

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

- For yes instances, $\Pr[C_x \text{ accepts}] \geq a(n)$ implies that $\text{HS}^2(\rho_0, \rho_1) \leq 1 - a^2(n)$;

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

We thus complete the proof of $\text{GapQHS}_{\log}[1 - a^2(n), 1 - b^2(n)]$ is BQUL-$[a(n), b(n)]$-hard. \qed

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

1. $\text{GapQSD}_{\log}[\alpha(n), \beta(n)]$ is BQUL-hard for $\alpha \leq 1 - 1/p(n)$ and $\beta \geq 1/p(n)$;

2. $\text{CertQSD}_{\log}[^\gamma(n)]$ is coRQUL-hard for $\gamma \leq 1 - 1/p(n)$;

3. $\text{GapQHS}_{\log}[\alpha(n), \beta(n)]$ is BQUL-hard for $\alpha \leq 1 - 1/p(n)$ and $\beta \geq 1/p(n)$;

4. $\text{CertQHS}_{\log}[\gamma(n)]$ is coRQUL-hard for $\gamma \leq 1 - 1/p(n)$.

Proof. Firstly, it is important to note that BQUL is closed under complement, as demonstrated in [Wat99, Corollary 4.8]. By combining error reduction for BQUL (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 coRQUL (Corollary 3.19) and set $a = 1$ in Lemma 4.16 (resp., Lemma 4.17). \qed
4.5.2 Hardness results for GAPQJS_{log} and GAPQED_{log}

We demonstrate the BQUL-hardness of GAPQJS_{log} by reducing GAPQSD_{log} to GAPQJS_{log}, following a similar approach as shown in [Liu23, Lemma 4.11].

**Lemma 4.19 (GAPQJS_{log} is BQUL-hard).** For any functions \( \alpha(n) \) and \( \beta(n) \) are computable in deterministic logspace, we have \( \text{GAPQJS}_{\text{log}}[\alpha(n), \beta(n)] \) is BQUL-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 \( \text{GAPQSD}_{\text{log}}[1 - 1/p(n), 1/p(n)] \) to \( \text{GAPQJS}_{\text{log}}[\alpha(n), \beta(n)] \). Consider logarithmic-qubit quantum circuits \( Q_0 \) and \( Q_1 \), which is an instance of GAPQSD_{log}. We will obtain \( \rho_0 \) 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 \( \text{td}(\rho_0, \rho_1) \geq 1 - 1/p(n) \), then Lemma 2.4 yields that
  \[
  \text{QJS}_2(\rho_0, \rho_1) \geq 1 - H_2 \left( \frac{1 - \text{td}(\rho_0, \rho_1)}{2} \right) \geq 1 - H_2 \left( \frac{1}{2p(n)} \right) \geq 1 - \frac{\sqrt{2}}{\sqrt{p(n)}} \geq \alpha(n),
  \]
  where the third inequality owing to \( H_2(x) \leq 2\sqrt{x} \) for all \( x \in [0, 1] \).
- If \( \text{td}(\rho_0, \rho_1) \leq 1/p(n) \), then Lemma 2.4 indicates that
  \[
  \text{QJS}_2(\rho_0, \rho_1) \leq \text{td}(\rho_0, \rho_1) \leq \frac{1}{p(n)} \leq \beta(n).
  \]

Therefore, we can utilize the same quantum circuits \( Q_0 \) and \( Q_1 \), along with their corresponding quantum states \( \rho_0 \) and \( \rho_1 \), respectively, to establish a logspace Karp reduction from \( \text{GAPQSD}_{\text{log}}[1 - 1/p(n), 1/p(n)] \) to \( \text{GAPQJS}_{\text{log}}[\alpha(n), \beta(n)] \), as required.

By combining the reduction from \( \text{GAPQSD}_{\text{log}} \) to \( \text{GAPQJS}_{\text{log}} \) (Lemma 4.19) and the reduction from \( \text{GAPQJS}_{\text{log}} \) to \( \text{GAPQED}_{\text{log}} \) (Corollary 4.12), we will demonstrate that the BQUL-hardness for \( \text{GAPQED}_{\text{log}} \) through reducing \( \text{GAPQSD}_{\text{log}} \) to \( \text{GAPQED}_{\text{log}} \).

**Corollary 4.20 (GAPQED_{log} is BQUL-hard).** For any function \( g(n) \) are computable in deterministic logspace, we have \( \text{GAPQED}_{\text{log}}[g(n)] \) is BQUL-hard for \( g(n) \leq \frac{\ln 2}{2} \left( 1 - \frac{\sqrt{2}}{\sqrt{p(n-1)}} - \frac{1}{p(n-1)} \right) \), 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 \( \text{GAPQJS}_{\text{log}}[\alpha(n), \beta(n)] \) is BQUL-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 \( \rho_0 \) and \( \rho_1 \), 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 \( \rho'_0 = (p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|) \otimes \left( \frac{1}{2} \rho_0 + \frac{1}{2} \rho_1 \right) \) satisfying \( H_2(p) = 1 - \frac{1}{2} (\alpha(n) + \beta(n)) \) 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, \( \text{GAPQED}_{\text{log}}[g(n)] \) is BQUL-hard as long as

\[
  g(n) = \frac{\ln 2}{2} \left( \alpha(n-1) - \beta(n-1) \right) \leq \frac{\ln 2}{2} \left( 1 - \frac{\sqrt{2}}{\sqrt{p(n-1)}} - \frac{1}{p(n-1)} \right).
\]

Therefore, \( \text{GAPQSD}_{s(n)}[\alpha(n), \beta(n)] \) is logspace Karp reducible to \( \text{GAPQED}_{s+1}[g(n)] \) 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


47


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

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 \( F_k(\theta) := \text{erf}(\kappa \cos \theta) \cos(k\theta) \), max max \( |F''_k(\xi)| \leq \frac{2}{\sqrt{\pi}} \kappa + k^2 + \frac{4}{\sqrt{\pi}} \kappa^3 + \frac{4}{\sqrt{\pi}} k \kappa \).

Proof. Through a straightforward calculation, we have derived that

\[
|F''_k(\theta)| = \frac{2}{\sqrt{\pi}} |\kappa \exp(-\kappa^2 \cos^2 \theta) \cos(\kappa \theta)| + |k^2 \cos(k \theta) \text{erf}(\kappa \cos(\theta))| \\
+ \frac{4}{\sqrt{\pi}} |\kappa \exp(-\kappa^2 \cos^2 \theta) \cos(\kappa \theta) \sin^2 \theta| \\
+ \frac{4}{\sqrt{\pi}} |k \kappa \exp(-\kappa^2 \cos^2 \theta) \sin \theta \cos(k \theta)| \\
\leq \frac{2}{\sqrt{\pi}} \kappa + k^2 + \frac{4}{\sqrt{\pi}} \kappa^3 + \frac{4}{\sqrt{\pi}} k \kappa.
\]

(A.1)

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

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

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

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 \( \|f - \hat{f}_1\|_\infty \leq \epsilon/4 \). Then we know that \( \hat{f}_1(x) = \sum_{k=0}^{K} a_k \sum_{l=0}^{\infty} b_l^{(k)} \sin\left(\frac{l \pi x}{2}\right) \) where the coefficients \( b_l^{(k)} \) satisfy that

\[
b_l^{(k+1)} = \sum_{l'=0}^{l} b_l^{(k)} b_{l-l'}^{(1)} \quad \text{where} \quad b_l^{(1)} = \begin{cases} 0 & \text{if } l \text{ is even} \\ \left(\frac{l+1}{l}\right)^{2l+1} \cdot \frac{2}{\pi} & \text{if } l \text{ is odd} \end{cases}, \quad (3.3)
\]

Furthermore, the coefficients \( \{b_l^{(k)}\} \) satisfies the following: (1) \( \|b^{(k)}\|_1 = 1 \) for all \( k \geq 1 \); (2) \( b^{(k)} \) is entry-wise non-negative for all \( k \geq 1 \); (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 \( x = \frac{2}{\pi} \cdot \text{arcsin}(\sin(\frac{\pi x}{2})) \) 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 \( \|f - \hat{f}_1\| \leq \epsilon/4 \) by the assumption. Let \( b^{(k)} \) be the coefficients of \( \left(\frac{\text{arcsin} y}{\pi/2}\right)^k = \sum_{l=0}^{\infty} b_l^{(k)} y^l \) for all \( y \in [-1, 1] \), then we result in our first approximation. Moreover, we observe that \( \frac{\pi}{2} b^{(1)} = \left(\frac{\text{arcsin} y}{\pi/2}\right) \sum_{l=0}^{\infty} b_l^{(1)} y^l \) for \( k > 1 \), which derives Equation (3.3) by comparing the coefficients. In addition, notice that \( \|b^{(k)}\|_1 = \sum_{l=0}^{\infty} b_l^{(k)} 1^l = \left(\frac{\text{arcsin} 1}{\pi/2}\right)^k = 1 \), 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 \( \hat{f}_2(x) := \sum_{k=0}^{K} a_k \sum_{l=0}^{L} b_l^{(k)} \sin\left(\frac{l \pi x}{2}\right) \) where \( L := \lceil \delta^{-2} \ln(4\|a\|_1 \epsilon^{-1}) \rceil \), then we have that \( \|\hat{f}_1 - \hat{f}_2\|_\infty \leq \epsilon/4 \).
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:

\[
\left| \sum_{l=L}^{\infty} b_l^{(k)} \sin \left( \frac{\pi x}{l} \right) \right| \leq \sum_{l=L}^{\infty} b_l^{(k)} |\sin \left( \frac{\pi x}{l} \right)| \leq \sum_{l=L}^{\infty} b_l^{(k)} |1 - \delta^2| \leq (1 - \delta^2)L \sum_{l=L}^{\infty} b_l^{(k)} \leq (1 - \delta^2)L.
\]

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

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 \( \|\hat{f}_2 - \hat{f}_3\| \leq \epsilon/2 \) and \( M = \lfloor \delta^{-1} \ln(4\|a\|_1 \epsilon^{-1}) \rfloor \), then we have

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

\[
\hat{f}_3^{(\text{odd})}(x) := \sum_{k=0}^{K} a_k \sum_{l=0}^{(L-1)/2} (-1)^l 2^{-l} b_l^{(k)} \sum_{m' = 0}^{l - M} (-1)^{m'} 2^{l+1} \sin(\pi x (m' - l - \frac{l}{2})).
\]

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 \( \hat{f}_3(x) := \hat{f}_3^{(\text{even})}(x) + \hat{f}_3^{(\text{odd})}(x) \).

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 \( \sin^l(z) = (\cos \frac{iz}{2})^l = \left( \frac{e^{-iz}}{2} \right)^l \sum_{m=0}^{l} \exp(iz(2m - l)) \) by a direct calculation, which implies the counterpart for real-valued functions:

\[
\sin^l(z) = \begin{cases} 
2^{-l}(-1)^{(l+1)/2} \sum_{m=0}^l (-1)^{m'} \binom{l}{m'} \sin(z(2m' - l)), & \text{if } l \text{ is odd}; \\
2^{-l}(-1)^{l/2} \sum_{m=0}^l (-1)^{m'} \binom{l}{m'} \cos(z(2m' - l)), & \text{if } l \text{ is even}.
\end{cases}
\] (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 \( l \leq L \), we have derived that:

\[
\left( \sum_{m' = 0}^{(l/2) - M} 2^{-l} \binom{l}{m'} \right) = \sum_{m' = (l/2) + M}^{l} 2^{-l} \binom{l}{m'} \leq e^{-\frac{2lM}{2}} \leq \frac{e}{4\|a\|_1} \leq \frac{e}{4\|a\|_1}.
\] (A.3)

Here, we choose \( M = \lfloor \delta^{-1} \ln(4\|a\|_1 \epsilon^{-1}) \rfloor \), and the last inequality is because of the assumption \( \epsilon \leq 2\|a\|_1 \). 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) \( l \) for odd (resp., even) functions. If the function \( f \) is neither even nor odd, we must consider all \( l \leq L \). Plugging Equation (A.3) into Equation (A.2), we can derive that:

\[
\begin{align*}
\text{If } l \text{ is odd, } \left\| \sin^l(z) - 2^{-l}(-1)^{(l+1)/2} \sum_{m' = (l+1)/2 - M}^{(l+1)/2 + M} (-1)^{m'} \binom{l}{m'} \sin(z(2m' - l)) \right\|_{\infty} & \leq \frac{e}{2\|a\|_1}; \\
\text{If } l \text{ is even, } \left\| \sin^l(z) - 2^{-l}(-1)^{l/2} \sum_{m' = l/2 - M}^{l/2 + M} (-1)^{m'} \binom{l}{m'} \cos(z(2m' - l)) \right\|_{\infty} & \leq \frac{e}{2\|a\|_1};
\end{align*}
\] (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 \( \sum_{k=0}^{K} a_k \sum_{l=0}^{L} |b_l^{(k)}| \leq \sum_{k=0}^{K} |a_k| = \|a\|_1 \) with Equation (A.4), we achieve that \( \|\hat{f}_2 - \hat{f}_3\|_{\infty} \leq \epsilon/2 \). \( \square \)
A.1.3 Omitted proof in Theorem 3.8

Fact 3.8.2. Consider the integrand $F_k(\theta) = \sum_{m=-M}^{M} c_m e^{\frac{k}{2}(H^{(+)}_{k,m} - H^{(-)}_{k,m})}$ 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

$$H^{(\pm)}_{k,m}(\theta) := \cos\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right) \cdot \text{erf}\left(\kappa \left(\cos \theta - x_0 \mp r \mp \frac{\delta}{4}\right)\right).$$

(3.8)

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

$$H^{(\pm)}_{k,m}(\theta) := \sin\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right) \cdot \cos(\kappa \theta) \cdot \text{erf}\left(\kappa \left(\cos \theta - x_0 \mp r \mp \frac{\delta}{4}\right)\right).$$

(3.9)

Moreover, the integrand is $F_k(\theta) = \sum_{m=-M}^{M} \left(\frac{c_m}{2}(H^{(+)\pm}_{k,m} - \tilde{H}^{(-)}_{k,m}) + \frac{c_m^{(\text{odd})}}{2}(\tilde{H}^{(+)\pm}_{k,m} - \tilde{H}^{(-)}_{k,m})\right)$ when $f$ is neither even nor odd, where $\tilde{H}_{k,m}$ 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 $F_k(\theta)$:

$$|F_k''(\theta)| \leq \sum_{m=-M}^{M} \left| \frac{d^2}{d\theta^2} H^{(+)\pm}_{k,m}(\theta) - \frac{d^2}{d\theta^2} H^{(-)\pm}_{k,m}(\theta) \right| \leq \left\| \frac{d^2}{d\theta^2} H^{(+)\pm}_{k,m}(\theta) \right\|_{\theta = \theta_0} \left(\left| \frac{d^2}{d\theta^2} H^{(-)\pm}_{k,m}(\theta) \right|_{\theta = \theta_0} \right).$$

(A.5)

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

$$\frac{d^2}{d\theta^2} H^{(+)\pm}_{k,m}(\theta) = -k^2 \cos(\theta) \cos\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right) \cdot \text{erf}(\kappa \left(\cos \theta - x_0 \mp r \mp \frac{\delta}{4}\right))$$

$$- \frac{\sin^2(\theta) \cos(\theta) \cos\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right)}{k^2 \cos(\theta) \cos\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right)} \cdot \text{erf}(\kappa \left(\cos \theta - x_0 \mp r \mp \frac{\delta}{4}\right))$$

$$+ \frac{\sin(\theta) \cos(\theta) \cos\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right)}{\kappa \cos(\theta) \cos\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right)} \cdot \text{erf}(\kappa \left(\cos \theta - x_0 \mp r \mp \frac{\delta}{4}\right))$$

$$- \frac{2k}{\sqrt{\pi}} \cos(\theta) \cos\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right) e^{-\kappa^2 \left(\cos \theta - x_0 \mp r \mp \frac{\delta}{4}\right)^2}$$

$$- \frac{4\sqrt{\pi} m}{\kappa \cos(\theta) \cos\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right)} \cdot \text{erf}(\kappa \left(\cos \theta - x_0 \mp r \mp \frac{\delta}{4}\right) e^{-\kappa^2 \left(\cos \theta - x_0 \mp r \mp \frac{\delta}{4}\right)^2}$$

$$+ \frac{4\sqrt{\pi} m \sin(\theta) \cos(\theta) \cos\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right)}{\kappa \cos(\theta) \cos\left(\theta \frac{\cos \theta - x_0}{r + \delta}\right)} e^{-\kappa^2 \left(\cos \theta - x_0 \mp r \mp \frac{\delta}{4}\right)^2}$$

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

$$\left| \frac{d^2}{d\theta^2} H^{(+)\pm}_{k,m}(\theta) \right| \leq k^2 + \frac{\pi}{\sqrt{\pi}} + \frac{\pi}{\delta r} + 1 \frac{\kappa^2}{\delta^2} + m \cdot \left( \frac{\pi}{\delta r} + \frac{2\pi k}{\delta r} + \frac{4\sqrt{\pi} \kappa}{\delta r} + \frac{m^2}{\delta^2} \right)$$

$$\leq d^2 + O(d) + O(d^2) + O(d^3) + M \cdot \left( \frac{O(1)}{\delta r} + O(d) + O(d) + M^2 \cdot \frac{O(1)}{\delta r} \right)$$

(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 \( H^{(±)}_{k,m}(θ) \) when \( f \) is odd:

\[
\frac{d^2}{dθ^2} H^{(±)}_{k,m}(θ) = -k^2 \cos(kx) \sin\left(\frac{π(m+\frac{1}{2})}{δ+r}(\cos(x)−x_0)\right) \text{erf}\left(\kappa\left(\cos(x)−x_0 \mp r \mp \frac{δ}{4}\right)\right) \\
- \frac{π(m+\frac{1}{2})}{δ+r} \cos(\cos(kx)) \cos\left(\frac{π(m+\frac{1}{2})}{δ+r}(\cos(x)−x_0)\right) \text{erf}\left(\kappa\left(\cos(x)−x_0 \mp r \mp \frac{δ}{4}\right)\right) \\
- \frac{π^2(m+\frac{1}{2})^2}{(δ+r)^2} \sin^2(x) \cos(kx) \sin\left(\frac{π(m+\frac{1}{2})}{δ+r}(\cos(x)−x_0)\right) \text{erf}\left(\kappa\left(\cos(x)−x_0 \mp r \mp \frac{δ}{4}\right)\right) \\
+ \frac{2πk(m+\frac{1}{2})}{δ+r} \sin(x) \sin(kx) \cos\left(\frac{π(m+\frac{1}{2})}{δ+r}(\cos(x)−x_0)\right) \text{erf}\left(\kappa\left(\cos(x)−x_0 \mp r \mp \frac{δ}{4}\right)\right) \\
+ \frac{4πk(m+\frac{1}{2})}{δ+r} \sin^2(x) \cos(kx) \cos\left(\frac{π(m+\frac{1}{2})}{δ+r}(\cos(x)−x_0)\right) e^{-κ^2(\cos(x)−x_0)\mp r\mp δ^2} \\
- \frac{2k}{\sqrt{π}} \cos(x) \cos(kx) \sin\left(\frac{π(m+\frac{1}{2})}{δ+r}(\cos(x)−x_0)\right) e^{-κ^2(\cos(x)−x_0)\mp r\mp δ^2} \\
+ \frac{4k}{\sqrt{π}} \sin(x) \sin(kx) \sin\left(\frac{π(m+\frac{1}{2})}{δ+r}(\cos(x)−x_0)\right) e^{-κ^2(\cos(x)−x_0)\mp r\mp δ^2} \\
- \frac{4k}{\sqrt{π}} \sin^2(x) \cos(kx) \cos(x)−x_0 \mp r \mp \frac{δ}{4}) \sin\left(\frac{π(m+\frac{1}{2})}{δ+r}(\cos(x)−x_0)\right) e^{-κ^2(\cos(x)−x_0)\mp r\mp δ^2}.
\]

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

### 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 \( |ψ⟩ := \sum_{i=1}^{m} y_i|i⟩ \), where \( l = \lceil \log m \rceil \) and \( y_i \) are real amplitudes associated with an evaluation oracle \( \text{Eval}(i, ε) \) that returns \( y_i \) up to accuracy \( ε \) we can prepare \( |ψ⟩ \) up to accuracy \( ε \) in deterministic time \( O(m^2 \log(m/ε)) \) and space \( O(\log(m/ε^2)) \), together with \( m^2 \) evaluation oracle calls with precision \( ε := O(ε^2/m) \).

**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 \( |ψ⟩ \) expresses the weight \( W_x \) as a telescoping product, given by

\[
\forall x \in \{0,1\}^l, \quad W_x = W_{x_1} \cdot \frac{W_{x_1} \cdot W_{x_1 x_2}}{W_{x_1}} \cdot \frac{W_{x_1 x_2 x_3}}{W_{x_1 x_2}} \cdots \frac{W_x}{W_{x_{n-1}}} \quad \text{where} \quad W_x := \sum_{y \in \{0,1\}^{|x|−|x|} } \langle x y | ψ ⟩^2.
\]

(A.7)

To estimate \( |ψ⟩ \) up to accuracy \( ε \) in the \( l_2 \) norm, it suffices to approximate each weight \( W_x \) up to additive error \( ε := O(ε^2/m) \), as indicated in \[MP16, Section III.A\]. To compute \( W_x \), we need \( 2^{l−|x|} \) oracle calls to \( \text{Eval}(i, ε) \). Evaluating all terms in Equation (A.7) requires computing \( W_{x_1}, W_{x_1 x_2}, \ldots, W_x \) 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 \( \text{Eval}(i, ε) \). As we need to compute Equation (A.7) for all \( x \in \{0,1\}^l \), the overall number of oracle calls to \( \text{Eval}(i, ε) \) is \( 2^{2^l} = m^2 \). The remaining computation can be achieved in deterministic time \( \tilde{O}(m^2 \log(m/ε)) \) and space \( O(\log(m/ε)) \) where the time complexity is because of the iterated integer multiplication. \( \square \)

**Lemma 3.15** (Renormalizing bitstring indexed encoding). Let \( U \) be an \( (α, a, ε) \)-bitstring indexed encoding of \( A \), where \( α > 1 \) and \( 0 < ε < 1 \), 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 \( (1, a + 2, 36ε) \)-bitstring indexed encoding of \( A \). This implementation requires \( O(α) \) uses of \( U, U^\dagger, C_{U} \text{NOT}, \text{C}_{U} \text{NOT} \), and \( O(α) \) single-qubit gates. Moreover, the description of the resulting quantum circuit can be computed in deterministic time \( O(α) \) and space \( O(s) \).

**Proof.** Following Definition 3.1, we have \( ||A − αUH ||_1 ≤ ε \), where \( H \) and \( Π \) are the corresponding orthogonal projectors. Because \( U \) is a \( (1, a, ε/α) \)-bitstring indexed encoding \( A/α \), we obtain that \( ||A/α|| ≤ ||U|| + ε/α = 1 + ε/α \), equivalently \( ||A|| ≤ α + ε \).
Adjusting the encoding through a single-qubit rotation. Consider an odd integer \( k := 2\lfloor \pi (\alpha + 1)/2 \rfloor + 1 \leq 9\alpha = O(\alpha) \) and \( \gamma := (\alpha + \varepsilon) \sin(\pi/k) \leq 1 \). We define new orthogonal projectors \( \Pi' := \tilde{\Pi} \otimes |0\rangle \langle 0| \) and \( \Pi'' := \Pi \otimes |0\rangle \langle 0| \), and combine them with \( U' = U \otimes R_\gamma \), where \( R_\gamma = \left( \frac{\gamma}{\sqrt{1 - \gamma^2}} \right) \). By noting that \( \tilde{\Pi}'U'\Pi'' = \gamma \tilde{\Pi}U \Pi \otimes |0\rangle \langle 0| \), we deduce that \( U' \) is a \((1, a + 1, \gamma \varepsilon/\alpha)\)-bitstring indexed encoding of \( \gamma A/\alpha \otimes |0\rangle \langle 0| \), which is consequently a \((1, a + 1, 2\gamma \varepsilon/\alpha)\)-bitstring indexed encoding of \( \sin(2\pi/k) \cdot (A \otimes |0\rangle \langle 0|) \). An error bound follows:

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

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 \((1, a + 1, 2\gamma \varepsilon/\alpha)\)-bitstring indexed encoding of \( A \otimes |0\rangle \langle 0| \) when \( k = 1 \). Let \( \varepsilon := 2\gamma \varepsilon/\alpha \), 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:

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

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

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 \( \tilde{A} := \gamma A \otimes |0\rangle \langle 0| \), then we have \( A' = \tilde{A} \), indicating that \( \| A' - \tilde{A} \| \leq \varepsilon \), \( \| A' + \tilde{A} \| \leq \varepsilon \), \( \| A' - \tilde{A} \| \| A' + \tilde{A} \| \leq 1 \), and \( \| A' - \tilde{A} \| + \| A' + \tilde{A} \| \leq 1/3 + 4/9 < 1 \). By employing Lemma 3.13, as well as the facts that \( \sqrt{\frac{\varepsilon}{1 - \varepsilon}} < 2 \) and \( 2k\varepsilon = 4k\gamma \varepsilon/\alpha \leq 36\varepsilon \), we can construct a \((1, a + 1, 36\varepsilon)\)-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 \( U, U^\dagger, C_\Pi \text{NOT}, C_{\Pi^\dagger} \text{NOT}, \) and \( O(\alpha) \) single-qubit gates. Furthermore, the description of the resulting quantum circuit can be computed in deterministic time \( O(\alpha) \) and space \( O(s) \).

\[\square\]

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 \( \alpha \) or below \( \beta \). We can distinguish the two cases with error probability at most \( \varepsilon := O(\varepsilon \log d) \) using a degree-\( d \) quantum singular value transformation where 

\[ d = \frac{\log 1/\varepsilon}{\max\{\beta - \alpha, k \sqrt{1 - \alpha^2} \sqrt{1 - \beta^2}\}}. \]

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 \( U, U^\dagger, C_\Pi \text{NOT}, C_{\Pi^\dagger} \text{NOT}, \) and \( O(d^3 \log d) \) multi-controlled single-qubit gates. Also, we can compute the description of the implementation in deterministic time \( O(\varepsilon^{-1} d^3/2) \) 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 \( V \).

\[^36\text{This is because } \| A' + \tilde{A} \| \leq \| A' \| + \| A' \| + \| A' - \tilde{A} \| \leq 2 \sin(\pi/2k) + \varepsilon \leq 2 \sin(\pi/6) + 1/3 = 4/3.\]
associated a \((1, \epsilon)\)-bitstring indexed encoding \(U_P\) such that
\[
\begin{align*}
\left\| \Pi'_{\geq t+\delta} U_P \Pi'_{\geq t+\delta} - I \otimes \sum_{i:a \geq t+\delta} |\tilde{\psi_i}\rangle\langle\tilde{\psi_i}| \right\| &\leq \epsilon, \\
\left\| (|+\rangle \otimes \Pi'_{\leq t-\delta}) U_P (|+\rangle \otimes \Pi_{\leq t-\delta}) - 0 \right\| &\leq \epsilon.
\end{align*}
\]
where \(\Pi' := \begin{cases} \Pi, & \text{if } \beta - \alpha \geq \sqrt{1 - \alpha^2} - \sqrt{1 - \beta^2} \\
I - \Pi, & \text{otherwise.} \end{cases}\)

(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 \(\Pi'\) in Equation (A.8) in accordance with the proof presented in [GSLW19, Theorem 20].

With the construction of the resulting bitstring indexed encoding \(\Pi' U_P \Pi\) for an odd polynomial \(P\), we then apply an \(\epsilon\)-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 \(\text{Im}g(|+\rangle \langle+| \otimes \Pi')\), there exists a singular value \(\sigma_i\) above \(\alpha\) (resp., \(\sqrt{1 - \alpha^2}\)); Otherwise, all singular value \(\sigma_i\) must be below \(\beta\) (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 \(m = O(\delta^{-1} \log \frac{1}{\epsilon})\), which approximates an odd function \(Q(x) := \frac{1}{2}[(1 - \epsilon) \cdot \text{sgn}(x + t) + (1 - \epsilon) \cdot \text{sgn}(x - t) + 2\epsilon \cdot \text{sgn}(x)]\) on the interval \([-1, 1] \cup (-t - \delta, -t + \delta) \cup (t - \delta, t + \delta)\) with \(\epsilon^2/4\) precision. By leveraging the space-efficient odd degree-\(d\) polynomial approximation \(P_d^{\text{sgn}}(x)\) of the sign function, as specified in Corollary 3.7, we then obtain:
\[
P(x) = \frac{1}{2}[(1 - \epsilon) \cdot P_d^{\text{sgn}}(x + t) + (1 - \epsilon) \cdot P_d^{\text{sgn}}(x - t) + 2\epsilon \cdot P_d^{\text{sgn}}(x)]. \quad \text{(A.9)}
\]
Hence, we ensure that \(|P(x)| \leq 1\) for any \(-1 \leq x \leq 1\), and \((1 - \epsilon)^2 P(x) \in [0, \epsilon]\) if \((1 - \epsilon)^2 x \in [0, t - \delta]\), as well as \((1 - \epsilon)^2 P(x) \in [1 - \epsilon, 1]\) if \((1 - \epsilon)^2 x \in [t + \delta, 1]\) for \(z \in \{0, 1\}\). To achieve the resulting bitstring indexed encoding \(U_P\) of \(P(x)\) 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 \(\epsilon = O(\epsilon \log d)\).

\[\square\]

### 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 \(\alpha(n) - \beta(n) \geq 1/\text{poly}(n)\), we have that \(\text{GapQHS}_{\log}[\alpha(n), \beta(n)]\) is in BQL.

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

### B.1 Omitted proofs in BQL contains

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

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

\[\text{By applying [GSLW18, Definition 12] (the full version of [GSLW19]) to } \Pi' := I - \Pi, \text{ we know that } |\psi\rangle \text{ is a singular vector of } \Pi' \Pi \text{ with a singular value of at least } \sqrt{1 - \alpha^2} \text{ in the first case, or with a singular value of at most } \sqrt{1 - \beta^2} \text{ in the second case. Additionally, in one-sided error scenarios, if } \alpha = 0, \text{ then } b - a = b \geq 1 - \sqrt{1 - \alpha^2} = \sqrt{1 - \beta^2} \text{; while if } b = 1, \text{ then } b - a = 1 - a \leq \sqrt{1 - \alpha^2} = \sqrt{1 - \beta^2}.\]
inequality from Equation (4.1), which holds with probability at least 0.92 > 0.8:
\[
\left| \frac{x_0 - x_1}{2} - \text{td}(\rho_0, \rho_1) \right| = \left| \frac{x_0 - x_1}{2} - \text{Tr} \left( \frac{\rho_0 - \rho_1}{2} \text{sgn} \left( \frac{\rho_0 - \rho_1}{2} \right) \right) \right| \\
\leq \left| \frac{x_0 - x_1}{2} - \text{Tr} \left( \frac{\rho_0 - \rho_1}{2} P_{d}^{\text{sgn}} \left( \frac{\rho_0 - \rho_1}{2} \right) \right) \right| + \left| \text{Tr} \left( \frac{\rho_0 - \rho_1}{2} P_{d}^{\text{sgn}} \left( \frac{\rho_0 - \rho_1}{2} \right) \right) - \text{Tr} \left( \frac{\rho_0 - \rho_1}{2} \text{sgn} \left( \frac{\rho_0 - \rho_1}{2} \right) \right) \right|.
\]

For the first term, by noting the QSVT implementation error in Corollary 3.17, we have
\[
\left| \frac{x_0 - x_1}{2} - \text{Tr} \left( \frac{\rho_0 - \rho_1}{2} P_{d}^{\text{sgn}} \left( \frac{\rho_0 - \rho_1}{2} \right) \right) \right| \leq \tilde{C}_{\text{sgn}} \epsilon \log d + \epsilon_H. \quad (B.1)
\]

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,
\[
\left| \text{Tr} \left( \frac{\rho_0 - \rho_1}{2} P_{d}^{\text{sgn}} \left( \frac{\rho_0 - \rho_1}{2} \right) \right) - \text{Tr} \left( \frac{\rho_0 - \rho_1}{2} \text{sgn} \left( \frac{\rho_0 - \rho_1}{2} \right) \right) \right| \leq \sum_j |\lambda_j P_{d}^{\text{sgn}}(\lambda_j) - \lambda_j \text{sgn}(\lambda_j) |. \quad (B.2)
\]

We split the summation over \( j \) into three separate summations: \( \sum_j = \sum_{\lambda > \delta} + \sum_{\lambda > \delta} + \sum_{-\delta \leq \lambda \leq \delta} \). By noticing the approximation error of \( P_{d}^{\text{sgn}} \) in Corollary 3.7, we can then obtain the following results for each of the three summations:
\[
\sum_{\lambda > \delta} |\lambda_j P_{d}^{\text{sgn}}(\lambda_j) - \lambda_j \text{sgn}(\lambda_j) | = \sum_{\lambda > \delta} |\lambda_j | P_{d}^{\text{sgn}}(\lambda_j) - 1 | \leq \sum_{\lambda > \delta} |\lambda_j | C_{\text{sgn}} \epsilon \log d \leq C_{\text{sgn}} \epsilon \log d,
\]
\[
\sum_{\lambda < -\delta} |\lambda_j P_{d}^{\text{sgn}}(\lambda_j) - \lambda_j \text{sgn}(\lambda_j) | = \sum_{\lambda < -\delta} |\lambda_j | P_{d}^{\text{sgn}}(\lambda_j) + 1 | \leq \sum_{\lambda < -\delta} |\lambda_j | C_{\text{sgn}} \epsilon \log d \leq C_{\text{sgn}} \epsilon \log d,
\]
\[
\sum_{-\delta \leq \lambda_j \leq \delta} |\lambda_j P_{d}^{\text{sgn}}(\lambda_j) - \lambda_j \text{sgn}(\lambda_j) | \leq \sum_{-\delta \leq \lambda_j \leq \delta} 2 |\lambda_j | \leq 2^{r+1} \delta.
\]

Hence, we derive the following inequality by summing over the aforementioned three inequalities:
\[
\sum_j |\lambda_j P_{d}^{\text{sgn}}(\lambda_j) - \lambda_j \text{sgn}(\lambda_j) | \leq 2^{r+1} \delta + 2C_{\text{sgn}} \epsilon \log d. \quad (B.3)
\]

By combining Equation (B.1), Equation (B.2), and Equation (B.3), we conclude that
\[
\left| \frac{x_0 - x_1}{2} - \text{td}(\rho_0, \rho_1) \right| \leq \tilde{C}_{\text{sgn}} \epsilon \log d + \epsilon_H + 2C_{\text{sgn}} \epsilon \log d + 2^{r+1} \delta. \quad \square
\]

**Proposition 4.11.1.** The following inequality holds for \( i \in \{0, 1\} \):
\[
\text{Pr} \left[ \left| x_i \ln \left( \frac{\delta}{2} \right) - S(\rho_i) \right| \leq 2 \ln \left( \frac{\delta}{2} \right)
\left( \tilde{C}_{\text{ln}d \epsilon d} + \epsilon_H + C_{\text{ln} \epsilon \log d + 2^{r+1} \beta} \right) \right] \geq \frac{9}{10}.
\]

**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:
\[
|x_0 \ln(2/\beta) - S(\rho_0)| = \left| x_0 \ln(2/\beta) - \ln(2/\beta) \text{Tr} \left( P_{d}^{\text{ln}}(\rho_0) \rho_0 \right) \right| + \left| \ln(2/\beta) \text{Tr} \left( P_{d}^{\text{ln}}(\rho_0) \rho_0 \right) - S(\rho_0) \right|.
\]

For the first term, by noting the QSVT implementation error in Corollary 3.17, we have
\[
\left| x_0 \ln(2/\beta) - \ln(2/\beta) \text{Tr} \left( P_{d}^{\text{ln}}(\rho_0) \rho_0 \right) \right| \leq 2 \ln(1/\beta) \left( \tilde{C}_{\text{ln}d \epsilon d} + \epsilon_H \right). \quad (B.4)
\]

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,
\[
\left| \ln(2/\beta) \text{Tr} \left( P_{d}^{\text{ln}}(\rho_0) \rho_0 \right) - S(\rho_0) \right| \leq \sum_j |\lambda_j P_{d}^{\text{ln}}(\lambda_j) \ln(2/\beta) - \lambda_j \ln(1/\lambda_j) |. \quad (B.5)
\]

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

\[
\sum_{\lambda_j > \delta} |\lambda_j P_d^{\ln}(\lambda_j) \ln(2/\beta) - \lambda_j \ln(1/\lambda_j)| = \sum_{\lambda_j > \delta} |\lambda_j| |P_d^{\ln}(\lambda_j) \ln(2/\beta) - \ln(1/\lambda_j)| \\
\leq \sum_{\lambda_j > \delta} |\lambda_j| C_{\ln} \epsilon \log d \ln(2/\beta) \\
\leq 2C_{\ln} \epsilon \log d \ln(1/\beta),
\]

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

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

\[
\sum_j |\lambda_j P_d^{\ln}(\lambda_j) \ln(2/\beta) - \lambda_j \ln(1/\lambda_j)| \leq 2C_{\ln} \epsilon \log d \ln(1/\beta) + 2^{r+2}\delta \ln(1/\beta).
\]

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

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

\section*{B.2 Omitted proofs in coRQUL containments}

\textbf{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 \( \max\{|p_0 - 1/2|, |p_1 - 1/2|\} \geq \epsilon/2 \), then \( f(\theta_0, \theta_1) \leq 1 - \epsilon^2/4 \).

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

\[
f(\theta_0, \theta_1) = (2p_0^3 - 6p_0^2 + \frac{9}{2}p_0) \left(2p_1^3 - 6p_1^2 + \frac{9}{2}p_1\right) \\
\leq \left(1 - \left(p_0 - \frac{1}{2}\right)^2\right) \left(1 - \left(p_1 - \frac{1}{2}\right)^2\right) \\
\leq 1 - \max\{|p_0 - \frac{1}{2}|, |p_1 - \frac{1}{2}|\}^2 \\
\leq 1 - \frac{\epsilon^2}{4}.
\]

\textbf{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 \( f\left(\frac{1}{2} + \frac{q}{4}\right)^2 \leq 1 - \frac{q^2}{2} \) for any \( 0 \leq q \leq 1 \).

\textbf{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:

\[
f\left(\frac{1}{2} + \frac{q}{4}\right)^2 = \frac{q^6}{256} + \frac{3q^5}{64} + \frac{9q^4}{64} - \frac{q^3}{8} - \frac{3q^2}{4} + 1 \leq 1 - \frac{q^2}{2}.
\]

Equivalently, it suffices to show that \( g(x) := -\frac{x^4}{256} - \frac{3x^3}{64} - \frac{9x^2}{64} + x + \frac{1}{2} \geq 0 \) 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 \( \min\{g(0), g(1)\} = \{\frac{1}{4}, \frac{47}{256}\} > 0 \), as desired.