Succinct Arguments for QMA in the Quantum Random Oracle Model

Alessandro Chiesa
alessandro.chiesa@epfl.ch
EPFL

Zihan Hu
zihan.hu@epfl.ch
EPFL

November 20, 2026

Abstract

Succinct arguments are a fundamental cryptographic primitive for verifying computational claims with small communication. In the classical setting, succinct arguments for NP can be constructed from unstructured hardness alone (e.g., hash functions) by compiling probabilistically checkable proofs (PCPs) or interactive oracle proofs (IOPs) for NP via the commit-and-open paradigm. In contrast, known succinct arguments for QMA rely on "structured" cryptographic primitives, or on the quantum PCP conjecture.

We construct the first succinct argument for QMA in the quantum random oracle model (QROM) without relying on additional cryptographic assumptions or unproven conjectures. This yields succinct arguments for QMA from unstructured hardness alone, showing that ideal hash functions not only suffice for succinct arguments for NP but also for QMA.

Underlying our result is an efficiency-preserving transformation that compiles quantum interactive oracle proofs (QIOPs), a recently introduced interactive generalization of quantum PCPs, into quantum arguments for the same language, via a natural quantum commit-and-open paradigm. Our transformation applies to every QIOP with public-query soundness, a notion that we formalize to capture a natural requirement of the commit-and-open paradigm and is satisfied by a known QIOP for QMA. As a key ingredient in our transformation, we formalize and construct extractable vector commitments for quantum states with local openings in the QROM, which may be of independent interest.

Keywords: succinct arguments for QMA; quantum random oracle model; quantum interactive oracle proofs

Contents

1 Introduction **1**
- 1.1 Our results . . . 1
- 1.2 Related work . . . 4

2 Preliminaries **6**
- 2.1 Quantum states, operators, and circuits . . . 6
- 2.2 Oracle circuits and oracle algorithms . . . 7
- 2.3 Compressed oracles . . . 8
- 2.4 Relations and languages . . . 11
- 2.5 Quantum interactive arguments . . . 11
- 2.6 Quantum interactive oracle proofs . . . 12

3 Public-query quantum interactive oracle proofs **15**

4 Defining extractable quantum state commitments **17**
- 4.1 A canonical quantum state commitment . . . 17
- 4.2 The extractability definition . . . 18

5 Construction of extractable commitments to quantum states **20**
- 5.1 An extractable basic succinct commitment scheme . . . 20
- 5.2 Extractor for the commitment scheme . . . 20
- 5.3 Analysis of the commitment scheme . . . 22
- 5.4 Proof of Lemma 5.6 . . . 24
- 5.5 Proof of Claim 5.7 . . . 32

6 Defining extractable quantum state vector commitments **36**
- 6.1 Syntax for quantum state vector commitments . . . 36
- 6.2 The extractability definition . . . 37

7 Construction of extractable quantum state vector commitments **39**
- 7.1 Labeling the Merkle tree . . . 39
- 7.2 An extractable quantum state vector commitment scheme . . . 39
- 7.3 Extractor for the vector commitment scheme . . . 41
- 7.4 Security analysis of the vector commitment scheme . . . 42
- 7.5 Proof of Lemma 7.4 . . . 45

8 Quantum interactive arguments based on QIOPs **52**
- 8.1 Our transformation . . . 52
- 8.2 The malicious QIOP prover . . . 54
- 8.3 Proof of Theorem 8.1 . . . 54

A The QIOP that we use **59**

Acknowledgments **60**

AI disclosure **60**

References **60**

1 Introduction

Succinct arguments enable a verifier to check a computational claim using communication substantially smaller than the cost of checking the claim directly. A remarkable feature of the classical theory is that such arguments do not require public-key cryptography or other highly structured assumptions: suitable hash functions suffice. Indeed, the *commit-and-open paradigm* compiles probabilistically checkable proofs and interactive oracle proofs for NP into succinct arguments using hash-based vector commitments [Kil92; Mic00; BG08; BCS16; CDGS23]. This paradigm is also known to remain secure against quantum adversaries under unstructured post-quantum assumptions [CMS19; CMSZ21; CDDGS25].

These results naturally raise the question of whether succinct arguments based solely on unstructured hardness can be extended from NP to its quantum analogue, QMA, which involves a quantum witness and can be verified in quantum polynomial time. In particular, [BLM26] poses the following question:

Are there succinct arguments for QMA *based on hash functions, or any unstructured hardness?*

A line of work [Bar+22; MNZ24; GKNV25; BLM26] constructs succinct arguments for QMA, even with a classical verifier. However, these constructions rely on structured hardness assumptions (see Section 1.2).

A different approach, more closely aligned with our goal of succinct arguments for QMA from unstructured hardness, is to extend the classical commit-and-open paradigm to the fully quantum setting [CM24; GJMZ23]. In particular, [GJMZ23] proposes a quantum analogue of Kilian's protocol [Kil92] that compiles a quantum probabilistically checkable proof (QPCP) into a quantum interactive argument while preserving the efficiency of the QPCP. Crucially, [GJMZ23] requires only unstructured hardness: any collapsing hash function, or more generally any succinct quantum state commitment (see Section 1.2 for more details).

However, obtaining a succinct argument for QMA via their transformation requires a QPCP for QMA with polynomial length and small query complexity, whose existence is a weak form of the QPCP conjecture [AALV09; AAV13] and remains a major open problem in quantum complexity theory.

In sum, succinct arguments for QMA have not matched their classical counterparts in terms of assumptions. Known constructions rely on structured cryptographic hardness, or on an unresolved complexity conjecture.

1.1 Our results

We construct the first succinct argument for QMA in the quantum random oracle model (QROM) [BDFLSZ11], without relying on additional cryptographic assumptions or unsolved conjectures.

Theorem 1. *In the QROM, there exists a succinct quantum interactive argument for* QMA *with communication complexity* $\mathsf{poly}(\lambda, \log \nu)$ *for instance size* ν *and security parameter* λ.

Here all parties, including malicious ones, have quantum query access to a uniformly sampled random oracle with output length λ. Thus, our result establishes the feasibility of succinct arguments for QMA from an idealized form of unstructured hardness, without additional cryptographic assumptions; this answers the aforementioned open question of [BLM26] in the oracle model. Establishing an analogous result in the plain model (from collapsing hash functions) is done in independent and concurrent work.[1]

[1] [BM26] constructs succinct arguments for QMA from (a non-black-box use of) collapsing hash functions in the standard model, via quantum-succinct blind delegation and the communication-compression compiler of [BLM26]. Our approach is different: we give a quantum commit-and-open compiler for QIOPs and construct the extractable quantum-state vector commitments to instantiate it in the QROM. The two works independently resolve the feasibility question through different techniques and in different models.

We obtain this result from a general transformation, which is our main technical contribution: a quantum analogue of the classical hash-based commit-and-open transformation. We compile any public-query quantum interactive oracle proof (QIOP) into a quantum interactive argument in the QROM while essentially preserving its efficiency. Instantiating the compiler with the recent QIOP for QMA in [SV26] yields Theorem 1. Public-query soundness is inherent to this commit-and-open approach because the prover must learn the queried locations in order to open them; we formulate the appropriate quantum analogue further below.

Theorem 2 (Theorem 8.1, informal)**.** *There exists a transformation $\mathbb{T}$ satisfying the following. Let* QIOP *be a QIOP for a relation R, with round complexity* k*, proof length* l*, query complexity* q*, and verifier-to-prover communication complexity* vc*. Then* $\mathsf{QARG} \coloneqq \mathbb{T}[\mathsf{QIOP}]$ *is a quantum interactive argument for R in the QROM with communication complexity $O(\lambda\mathsf{k} + \lambda\mathsf{q}\log\mathsf{l} + \mathsf{vc})$. If* QIOP *has public-query soundness error s_{pq}, then* QARG *has soundness error $O(s_{\mathsf{pq}} + \mathsf{k}\cdot\mathsf{poly}(t,\mathsf{l},\mathsf{q})\cdot 2^{-\lambda})$ against t-query quantum adversaries.*

Extending the commit-and-open paradigm to an appropriate notion of QIOP requires overcoming delicate definitional and technical challenges. In particular, we introduce a suitable notion of *quantum-state vector commitment* (QSVC) with a strong extraction property, and prove that a natural quantum-state tree commitment (QSTC) in the QROM satisfies it. Below we elaborate on these challenges and results.

Flavor of QIOP. Quantum interactive oracle proofs (QIOPs) [SV25] extend quantum probabilistically checkable proofs (QPCPs) to the interactive setting, just as classical IOPs extend PCPs [BCS16; RRR16]. Several recent works consider different, sometimes incomparable, models of QIOPs [SV25; SV26; CGMV26]. We formulate our compiler for a QIOP model that simultaneously supports the relevant features appearing across these constructions: superposition queries, the return of previously submitted quantum proof states, and adaptive queries made before those states are returned. This formulation also identifies expressive QIOP features supported by our QROM compiler, thereby providing a flexible target model for future QIOP constructions. See Section 2.6 for details.

- **Superposition queries.** We equip the QIOP verifier with *superposition query access* to the quantum proof oracles [CGMV26]. Informally, the verifier can specify query sets in superposition in a quantum register and coherently swap the corresponding locations of the quantum proof oracle to designated registers held by the verifier. Superposition query access generalizes the "point access" used in the definition of QPCPs and some QIOP constructions [SV25; SV26].
- **Returning proof oracles.** The verifier may return quantum proof oracles from earlier rounds to the prover after querying them, and the size of the returned registers is not charged to its query complexity or verifier-to-prover communication. This feature plays a key role in known QIOPs [SV25; SV26; CGMV26]. In contrast to a classical prover, a quantum prover cannot generally retain a copy of a transmitted proof state. The verifier may therefore need to return a previously submitted proof oracle for use by the prover in later rounds.
- **Adaptive queries and delayed returns.** We allow the verifier to retain quantum proof oracles across multiple rounds and make *adaptive* superposition queries to them in later rounds until they are returned to the prover. Delayed returns are used in the QIOP in [SV26] (but not, e.g., the QIOP in [CGMV26]).

Each feature creates an obstacle for commit-and-open compilation, addressed via the QSVC interface below.

Public-query soundness. The commit-and-open paradigm in the classical setting does not apply to every IOP: the argument prover learns the verifier's query locations to open the corresponding proof locations, so the IOP must satisfy *public-query soundness*. That is, soundness must hold even when the verifier's queries are leaked to the malicious prover whenever they occur [CDGS23]. We need an appropriate quantum

formulation because the query locations may be in superposition and therefore cannot, in general, be revealed by measurement without disturbing the verifier's computation.

We introduce a definition of *public-query soundness* for QIOPs, where soundness holds even when the malicious prover has access to the query location register immediately before and after each superposition query. That is, immediately before the query, the location registers are transferred to the malicious prover, who may act jointly on them and its private state before returning them. The same interaction occurs immediately after the query. Public-query soundness requires the original soundness guarantee to continue to hold even with this additional interface. This notion extends classical public-query soundness [CDGS23], and is satisfied by the efficient QIOP in [SV26] (see more in Appendix A). See Definition 3.2 for the formal definition.

Quantum-state vector commitment schemes. At a minimum, achieving succinctness in the compiled protocol requires a commitment scheme that produces a short commitment to a long quantum state and provides efficient openings of a few locations. In the classical setting, succinct commitment schemes with such local openings are known as vector commitments, whose standard commitment, opening, and verification procedures suffice for the commit-and-open paradigm [BCS16; CDGS23; CDDGS25].

In the quantum setting, this standard vector-commitment interface does not suffice when query locations may be in superposition, even when formulated for quantum messages.

- In a Merkle opening, the prover selects authentication-path registers according to query locations. If those locations are in superposition, the registers retained by the prover become entangled with the verifier's location register. Unless this which-path information is coherently erased, the location register is dephased, and even an honest execution of the compiled protocol may not reproduce the QIOP verifier's computation.
- A distinct problem arises when the QIOP verifier returns a previously submitted proof oracle. The compiled verifier holds only a succinct commitment to that state, so the commitment scheme must allow the prover to recover the updated underlying state after the verifier has queried it.

Prior work [GJMZ23] defines quantum-state commitments for compiling QPCPs with classical (non-superposition) queries. They do not provide a standalone quantum-state vector-commitment interface, nor provide the additional procedures that we require for coherent openings and returned proof states.

In light of this, we provide a new definition of *quantum-state vector commitment* (QSVC) in the QROM. Our definition incorporates key additional procedures: (a) an Update procedure to allow the honest prover to coherently erase its record of the query locations; (b) a Recover procedure to allow the honest prover to recover the underlying quantum message when the verifier returns a commitment, thereby supporting the return of proof oracles in a QIOP. These procedures (see Section 6.1 for details) ensure *completeness* of the resulting succinct argument for the aforementioned broad class of QIOPs. We also need a suitable *security* notion, discussed next.

Online quantum extraction of quantum messages. We formulate a notion of *online extractability* for QSVCs in the QROM. Informally, a QSVC scheme is extractable if there is an efficient extractor, with access to the compressed-oracle database, such that the following two ways of implementing superposition query access to the committed quantum message are indistinguishable. (Both implementations abort if any validity check fails.)

- *Real access.* The receiver verifies the supplied local openings and performs the desired superposition query through the commitment.
- *Extracted access.* Before seeing any openings, the extractor recovers a complete quantum state associated with the commitment. It later validates each opening, performs the same query directly on the extracted state, and reconstructs the updated commitment.

Crucially, extraction occurs online and before the sender chooses its openings. Formalizing the above intuition is subtle. The extractor must check the validity of the openings after the commitment has already been used

to extract the message; see Definition 6.3 for the formal definition.

Our online extractability notion enables a straightline reduction from the soundness of the compiled quantum argument to the public-query soundness of the underlying QIOP. Given a malicious argument prover, the reduction constructs a malicious QIOP prover by using the QSVC extractor to recover the committed proof states. When the QIOP verifier returns a proof state after querying it, the reduction reconstructs a corresponding commitment and resumes the simulation. The swap-binding notion for quantum-state commitments in [GJMZ23] does not provide the online guarantee required for such a straightline reduction.

Extractable quantum-state tree commitment. We construct an extractable QSVC in the QROM (unconditionally) where a commitment has size $\mathsf{poly}(\lambda)$ and an opening proof has size $\mathsf{poly}(\lambda, q, \log n)$, for message length n and a superposition query where each branch is for at most q locations. (This efficiency is comparable to that of the Merkle commitment scheme in the ROM.) The extraction error is as follows.

Theorem 3 (Theorem 7.3, informal)**.** *There exists a QSVC in the QROM (with the above efficiency and) with extraction error $\xi_{\mathsf{Ext}} = O(t^3 \cdot \mathsf{poly}(n, \lambda) \cdot 2^{-\lambda})$ against t-query quantum adversaries that send a single opening. Here n is the length of the message and λ is the security parameter (output length of the oracle).*

For simplicity, the theorem states the extraction error for a single opening. More generally, our formal definition of extraction supports multiple openings (see Definition 6.3) to handle QIOPs with adaptive queries across rounds, and our analysis directly establishes an extraction error for this adaptive query setting.

The underlying construction is a *quantum-state tree commitment* (QSTC). We combine a Merkle tree structure with a basic extractable succinct quantum-state commitment (see Sections 4 and 5 for details). This follows the structure in [GJMZ23] with two differences: (i) a random oracle replaces the cryptographic hash function; (ii) our construction additionally includes the extra procedures required by our QSVC definition.

Informally, at each tree node, the basic commitment coherently evaluates the random oracle on the computational-basis labels of the message register and, separately, on its Hadamard-basis labels. The two resulting image registers form the commitment, while the message register is retained as the opening. Using the compressed-oracle database, the extractor coherently recovers preimages associated with both image registers. Opening verification can then be related to an EPR-type consistency test between the two recovered descriptions. Excluding branches containing a random-oracle collision (and other small probability events), this test ensures that extracted access behaves like honest recovery of the committed state. We compose this basic commitment in a Merkle tree and prove that extraction remains secure under local, adaptive openings.

Our analysis of this construction addresses online extraction, coherent openings, and adaptive queries; all of these are not considered in [GJMZ23]. See Construction 7.1 for the full construction.

1.2 Related work

Hash-based succinct arguments for NP. The commit-and-open paradigm [Kil92; Mic00; BCS16; CDGS23] originated with Kilian's protocol [Kil92] and was later generalized from PCPs to IOPs [BCS16]; it has become a standard approach to constructing hash-based succinct arguments for NP. Using a hash-based vector commitment scheme (typically a Merkle commitment scheme), these transformations map a public-coin IOP into a succinct public-coin interactive argument. The prover replaces each long proof string with a succinct commitment and subsequently opens only the locations queried by the verifier. (To obtain a non-interactive argument, one further applies the Fiat–Shamir transformation.) Security of the hash-based commit-and-open paradigm is studied in two settings: (i) *in the plain model*, classical security is established in [CDGS23] based on collision-resistant hash functions and post-quantum security is established in [CDDGS25] based on collapsing hash functions; and (ii) *in the random oracle model*, classical security is established in [BCS16; CY24] and post-quantum security is established in [CMS19; CDHZ26].

Succinct arguments for QMA. Succinct arguments for QMA can be constructed under various assumptions. A line of work constructs such arguments with classical verifiers based on either Mahadev's measurement protocol [Bar+22; Fai24; GKNV25][2] or the compiled non-local game paradigm [MNZ24; BLM26]. Most recently, [BLM26] shows the feasibility of succinct arguments for QMA from oblivious state preparation (OSP) and collapsing hash functions. Since OSP can be based on plain trapdoor claw-free functions, their result is the first such construction that does not inherently rely on the hardness of learning with errors. No construction of OSP from purely unstructured assumptions is currently known.

A different approach, more closely related to ours, applies the commit-and-open paradigm directly to quantum proofs. [CM24] proposes a candidate transformation that compiles a QPCP into a succinct interactive quantum argument by committing to the quantum proof succinctly and later opening only the locations queried by the verifier, and [GJMZ23] constructs another transformation following the same paradigm and proves the security of their transformation. Neither transformation relies on structured cryptographic assumptions; we discuss the assumptions underlying their commitment schemes below. However, obtaining a succinct argument for QMA through their approaches requires an efficient QPCP for QMA, whose existence remains a conjecture.

Commitments to quantum states. Two prior works study quantum-state vector commitments by adapting the Merkle tree paradigm to quantum states. [CM24] proposes a candidate construction called the quantum Merkle tree in the quantum Haar random oracle model, using a Haar random unitary to compress the quantum state at each level of the tree until we reach the root, without proving the security of the candidate construction against a malicious sender. [GJMZ23] studies how to obtain a quantum-state vector commitment in the plain model by first constructing a succinct quantum-state commitment (without local opening) and composing them using a Merkle tree. Their construction can be based on collapsing hash functions and even on potentially weaker quantum cryptographic assumptions. Neither work formalizes quantum-state vector commitments as a standalone cryptographic primitive with a dedicated security definition.

A related notion is the classical commitment to quantum states in [GKNV25]. Their scheme allows a quantum sender to produce a classical commitment to a quantum state and later provide classical openings corresponding to measurements of selected qubits in either the computational or Hadamard basis. Openings for measurement results in these two bases do not suffice for our application.

Extractable commitments. Online extractability requires that an extractor, given some trapdoor information, can recover the underlying message from the commitment without rewinding the sender. [Pas03] shows that the basic hash-based commitment ("hash the message") is extractable in the ROM. [DFMS22] prove the same scheme is post-quantum extractable in the QROM; given a classical commitment y from a quantum adversary, it is possible to recover the message x given the quantum database obtained via the compressed oracle technique [Zha19]. [CDHZ26] further strengthens this notion by showing that such an extraction can be applied coherently to commitments y appearing in superposition within the adversary's random oracle queries without being detected by the adversary. These works consider only commitments to classical messages and therefore do not directly provide extractability for commitments to quantum states.

[2] [Fai24] additionally assumes a variant of the QPCP conjecture and constructs a classical-verifier succinct argument for QMA with non-succinct setup in the QROM.

2 Preliminaries

For every non-negative integer n, we use $[n]$ to denote the set $\{1, 2, \cdots, n\}$. In particular, $[0]$ denotes the empty set. We denote the empty string by ε. Let $\{0,1\}^{<\ell}, \{0,1\}^{\leq \ell}$ be the sets of strings of length less than ℓ and no more than ℓ, respectively.

For every game G, we use $\Pr[\mathsf{G}]$ to denote the probability that the output of game G is 1.

We use $\mathsf{hm}(z)$ to denote the Hamming weight of a vector z (the number of non-zero elements in z).

2.1 Quantum states, operators, and circuits

We use the standard bra-ket notation. We abbreviate the tensor product $|0\rangle^{\otimes n}$ as $|0^n\rangle$, or simply $|0\rangle$ when the number of qubits is clear from context.

A *register* is a named finite-dimensional complex Hilbert space. We use serif font, for example, A, to represent registers. For registers $\mathrm{A}, \mathrm{B}, \mathrm{C}$, the concatenation ABC is the tensor product of the associated Hilbert spaces. Sometimes we need to group several registers into one register, in which case, we use notations like $\mathrm{A} \coloneqq (\mathrm{B}, \mathrm{C})$. We also often divide a register into several registers, in which case, we use notations like $(\mathrm{A}, \mathrm{B}) \coloneqq \mathrm{C}$ when the size of each register is clear from the context. For a linear transformation L and a quantum state $\boldsymbol{\rho}$, we sometimes add a subscript R to them to emphasize that L_{R} is acting on the register R, and $\boldsymbol{\rho}_{\mathrm{R}}$ is a quantum state in the register R. We denote the identity transformation over a register R as I_{R}.

For a vector $|\phi\rangle$, we write $\| |\phi\rangle \|$ to denote its ℓ_2 norm $\| |\phi\rangle \| \coloneqq \sqrt{\langle\phi|\phi\rangle}$. For a linear transformation L, the operator norm of L is denoted by $\|L\| \coloneqq \max_{|\phi\rangle} \|L|\phi\rangle\|$ where the max is among all the vectors of norm 1. Then for two operators A and B, $\|A+B\| \leq \|A\| + \|B\|$ and $\|AB\| \leq \|A\|\|B\|$. If A and B satisfy $A^\dagger B = 0$ and $AB^\dagger = 0$ (i.e. they have orthogonal images and orthogonal supports), then

$$\|A+B\| \leq \max\{\|A\|, \|B\|\} \ . \tag{1}$$

In particular, the operator norm of a controlled operator $A = \sum |x\rangle\langle x| \otimes A^x$ can be upper bounded by the maximum of the norms of the operators A^x as shown in [DFMS22]:

$$\|A\| \leq \max_x \|A^x\| \ . \tag{2}$$

For two linear transformations A and B, their commutator is $[A, B] \coloneqq AB - BA$, whose norm measures how nearly A and B commute. For three operators A, B and C such that $\|A\|, \|B\|, \|C\| \leq 1$, if both A and B are almost commutative with C, AB is almost commutative with C, formally, as shown in [CMS19]:

$$\| [AB, C] \| \leq \| [A, C] \| + \| [B, C] \| \ . \tag{3}$$

We fix the universal gate set $\{H, \mathsf{CNOT}, T\}$ [NC10]. In this work, we consider *unitary quantum circuits* consisting of unitary gates from this gate set. The size of a unitary quantum circuit C is the number of gates in C. For a unitary quantum circuit C, the inverse of C, denoted as $C^\dagger$, has size linear in the size of C.

A quantum algorithm can, without loss of generality, be written in the form of introducing ancilla qubits initialized as $|0\rangle$, applying a unitary quantum circuit, and making measurements, by the deferred measurement principle. Thus we sometimes specify a quantum algorithm with a unitary quantum circuit together with the ancilla register and the output register.

For two registers A and B of the same size, we use $\mathsf{SWAP}_{\mathrm{AB}}$ to denote the unitary that maps $|\phi\rangle_{\mathrm{A}} |\psi\rangle_{\mathrm{B}}$ to $|\psi\rangle_{\mathrm{A}} |\phi\rangle_{\mathrm{B}}$ for each $|\phi\rangle$ and $|\psi\rangle$. X is the Pauli matrix which flips a qubit,

$$\mathsf{X} \coloneqq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \ .$$

2.2 Oracle circuits and oracle algorithms

For a function $f\colon \{0,1\}^* \to \{0,1\}^m$, an *oracle-aided unitary quantum circuit* C^f is a unitary quantum circuit with the additional query gate $U_f\colon |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$. Notice that the inverse of U_f is U_f. For an oracle-aided quantum circuit C^f, the inverse of C^f can also be implemented with oracle access to f.

More generally, we consider stateful oracles, which can be written as $U(\mathrm{S})$ for a unitary U and a state register S. An oracle-aided unitary quantum circuit $C^{U(\mathrm{S})}$ is a unitary quantum circuit where all the gates do not act on the state register S, with the additional query gate U_{XYS} acting on the registers XYS where X is the query register and Y is the answer register.

We use the sans-serif font with a stateful oracle in the superscript, e.g., $\mathsf{A}^{U(\mathrm{S})}$, to denote an oracle-aided quantum algorithm. Similarly, an oracle-aided quantum algorithm can be written in the form of introducing some ancilla qubits in $|0\rangle$, applying an oracle-aided unitary quantum circuit, and making measurements, by the deferred measurement principle.

For an oracle-aided quantum algorithm $\mathsf{A}^{U(\mathrm{S})}$ which has input register I and output register O, we use $\mathrm{O} \leftarrow \mathsf{A}^{U(\mathrm{S})}(\mathrm{I})$ to denote the following process:

1. A is given an input on register I, which might entangle with the state register S of the oracle U.
2. After interacting with the oracle U, A outputs a state on register O, which might also be entangled with the state register S.

An algorithm A with oracle access to multiple oracles $(U_i(\mathrm{S}_i))_{i\in[t]}$ may query them in superposition, the registers $(\mathrm{S}_i)_{i\in[t]}$ may not be distinct. Specifically, A has an oracle-selection register N, and a query is implemented by the controlled gate

$$\sum_{i\in[t]} |i\rangle\langle i|_{\mathrm{N}} \otimes (U_i)_{\mathrm{XYS}_i} \ .$$

We use *query probability mass* to quantify how heavily an oracle is queried.

Definition 2.1. *Let* A *be an algorithm with access to multiple oracles* $(U_i(\mathrm{S}_i))_{i\in[t]}$. *We define the* **query probability mass** *of* A *on the* i*-th unitary* U_i *at its* j*-th query as*

$$\mathsf{mass}(\mathsf{A}, i, j) \coloneqq \| \, |i\rangle\langle i|_{\mathrm{N}} |\psi_j\rangle \, \|^2 \ ,$$

where $|\psi_j\rangle$ *is the joint quantum state of the algorithm* A *and the state registers of the unitaries* $(U_i)_{i\in[t]}$ *just before the algorithm makes its* j*-th oracle query.*

A quantum query can simultaneously access each oracle, but the sum of query probability mass across different oracles cannot exceed 1.

Remark 2.2. Since $\sum_{i\in[t]} |i\rangle\langle i|_{\mathrm{N}} = I_{\mathrm{N}}$, for every algorithm A with access to oracles $(U_i(\mathrm{S}_i))_{i\in[t]}$, and integer j,

$$\sum_{i\in[t]} \mathsf{mass}(\mathsf{A}, i, j) = 1 \ .$$

We define the total (query) probability mass as the sum of the query probability masses over all oracle queries.

Definition 2.3. *For an algorithm* A *that makes* q *queries to oracles* $(U_i(\mathrm{S}_i))_{i\in[t]}$, *we define the* **total query probability mass** *of* A *on the* i*-th unitary* U_i *as follows:*

$$\mathsf{TotalMass}(\mathsf{A}, i) \coloneqq \sum_{j=1}^{q} \mathsf{mass}(\mathsf{A}, i, j) \ .$$

In the paper, we slightly overload the notation to use $\mathsf{TotalMass}(\mathsf{A}, U_i(\mathrm{S}_i))$ to also denote the total query (probability) mass of A to U_i.

2.3 Compressed oracles

In this work, we consider quantum random oracle models (QROMs) where every party has quantum access to an oracle f, sampled uniformly at random from the set of all the functions from $\{0,1\}^*$ to $\{0,1\}^m$. We use RO to emphasize that it is a random oracle and write the process of sampling a random oracle RO as $\mathsf{RO} \leftarrow \mathcal{U}(m)$.

For $y \in \{0,1\}^m$, we define $|\hat{y}\rangle := 2^{-m/2} \sum_{y' \in \{0,1\}^m} (-1)^{\langle y,y' \rangle} |y'\rangle$ where $\langle y, y' \rangle$ is the inner product of y and y'. To simplify notation, we often use $|\hat{0}^m\rangle$ to denote the state $|\hat{0}\rangle^{\otimes m} = 2^{-m/2} (|0\rangle + |1\rangle)^{\otimes m} = |\widehat{0^m}\rangle$.

Since an efficient oracle-aided quantum algorithm with running time T cannot query RO on input length greater than T, we often set the domain X to be a set of bit strings of bounded length instead of $\{0,1\}^*$ in order to represent it in a finite-dimensional complex Hilbert space.

Applying the query gate $U_{\mathsf{RO}} \colon |x\rangle |y\rangle \to |x\rangle |y \oplus \mathsf{RO}(x)\rangle$ is equivalent to applying the unitary

$$U \colon |x\rangle |y\rangle |f\rangle \to |x\rangle |y \oplus f(x)\rangle |f\rangle$$

to a state whose last register is initialized as the truth table of RO on X. Since U commutes with computational basis measurement on the last register, we can initialize a database register $\mathrm{D} := \bigotimes_{x \in X} \mathrm{D[x]}$ to be a uniform superposition of the truth tables of all the possible functions from X to $\{0,1\}^m$ and use U as the query gate, instead of sampling RO and using U_{RO}.

More interestingly, Zhandry [Zha19] introduced a way to compress the database register D, taking advantage of the fact that most of the registers $\mathrm{D[x]}$ are not queried and thus remain in $|\hat{0}^m\rangle$ during the process. Formally, the state of each $\mathrm{D[x]}$ lies in the span of $\{|y\rangle\}_{y \in \{0,1\}^m} \cup \{|\perp\rangle\}$ where $\perp$ is a special symbol. We can apply

$$F_{\mathrm{D}} := \bigotimes_{x \in X} F_{\mathrm{D[x]}}$$

with

$$F_{\mathrm{D[x]}} := |\hat{0}^m\rangle\langle\perp|_{\mathrm{D[x]}} + |\perp\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} + \sum_{y \in \{0,1\}^m / \{0^m\}} |\hat{y}\rangle\langle\hat{y}|_{\mathrm{D[x]}}$$

to the state $\bigotimes_{x \in X} \left(\frac{1}{\sqrt{2^m}} \sum_{y \in \{0,1\}^m} |y\rangle \right)_{\mathrm{D[x]}} = \bigotimes_{x \in X} |\hat{0}^m\rangle_{\mathrm{D[x]}}$ to get an initial state $\bigotimes_{x \in X} |\perp\rangle_{\mathrm{D[x]}}$ before any query is made.

Upon receiving a query with query register X and answer register Y, we can apply the oracle unitary

$$O_{\mathrm{XYD}} := \sum_{x \in X} |x\rangle\langle x|_{\mathrm{X}} \otimes F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]Y}} F_{\mathrm{D[x]}}$$

where $\mathsf{CNOT}_{\mathrm{D[x]Y}} |y_x\rangle_{\mathrm{D[x]}} |y\rangle_{\mathrm{Y}} = |y_x\rangle_{\mathrm{D[x]}} |y \oplus y_x\rangle_{\mathrm{Y}}$ for $y, y_x \in \{0,1\}^m$ and $\mathsf{CNOT}_{\mathrm{D[x]Y}}$ acts as an identity on $|\perp\rangle_{\mathrm{D[x]}} |y\rangle_{\mathrm{Y}}$ for $y \in \{0,1\}^m$.

For every set $S \subseteq X$, the register $\mathrm{D}[S] := \bigotimes_{x \in S} \mathrm{D[x]}$ records the images of the set S. We view a database D as an array of length $|X|$ with alphabet $\{0,1\}^m \cup \{\perp\}$. We use $D[x]$ to denote the image of x, as stored in the database D, and we use $D[S]$ to denote the array of images of $x \in S$, as stored in the database D. The size of D is denoted as $|D|$, which is the number of non-$\perp$ elements in the array D.

Let Π_{NoCol} be the projector that projects to all the databases without any collision. Formally, we denote the set of all possible databases as $S_D := (\{0,1\}^m \cup \{\perp\})^X$ and the set of all databases without collisions as

$S_{\mathsf{NoCollision}} \coloneqq \{D \in S_D : \forall x, x' \in X \text{ s.t. } x \neq x', D[x] \neq \bot, \text{ and } D[x'] \neq \bot, \text{ we have that} D[x] \neq D[x']\}$, and we have that

$$\Pi_{\mathsf{NoCol}} \coloneqq \sum_{D \in S_{\mathsf{NoCollision}}} |D\rangle\langle D| \ .$$

Π_t is the projector that projects to all the databases with size at most t. Formally,

$$\Pi_t \coloneqq \sum_{D \in S_t} |D\rangle\langle D| \ ,$$

where

$$S_t \coloneqq \{D \in S_D : |D| \leq t\} \ .$$

We will use the following lemma from [CMS19].

Lemma 2.4 (Lemmas 5.10 and 6.11 in [CMS19])**.** *For every random oracle output length m and bound t for the size of the database,*

$$\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}})(\Pi_t O \Pi_t)\Pi_{\mathsf{NoCol}}\| \leq \sqrt{6t} \cdot 2^{-m/2} \ .$$

Readers may wonder how to implement the above compressed oracle efficiently. As in [Zha19], we can make the compressed oracle efficient by mapping the extremely long table into a short database. There exists a unitary U_{ToDB} that takes a compressed table and returns a compressed database. Namely, given input $\bigotimes_{x \in X} |z_x\rangle_{\mathrm{D}[x]}$ where $z_x \in \{0,1\}^m \cup \{\bot\}$, U_{ToDB} returns $|x_1, z_{x_1}, x_2, z_{x_2}, \ldots, x_k, z_{x_k}\rangle$ where $x_1, \ldots, x_k$ enumerate all $x \in X$ such that $z_x \neq \bot$, in increasing order. All the computations can also be done with this short database efficiently.

We show that restricting a procedure to run on databases without collisions won't change the final result too much.

Lemma 2.5. *Let A be a t-query algorithm with query access to the oracles $O(\mathrm{D})$ and $U(\mathrm{D})$ for a unitary U such that $[\Pi_{\mathsf{NoCol}}, U] = 0$ and $[\Pi_q, U] = 0$ for every q. Let O denote the output register of the algorithm A. Denote A's query mass to $O(\mathrm{D})$ as $w \coloneqq \mathsf{TotalMass}(\mathsf{A}, O(\mathrm{D}))$. Let D be a computationally unbounded distinguisher.*

Let $\mathsf{A}_{\mathit{NoCollision}}$ be a variant of A that runs as A except that before and after each query, the measurement $\{\Pi_{\mathsf{NoCol}}, I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}\}$ is performed over D until A outputs register O. If at least one of the measurement outcomes is $I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}$, then $\mathsf{A}_{\mathit{NoCollision}}$ outputs 0 and O; otherwise, $\mathsf{A}_{\mathit{NoCollision}}$ outputs 1 and O.

Then for every random oracle output length m,

$$\left| \sqrt{\Pr\left[b = 1 \;\middle|\; \begin{array}{l} \mathrm{D} \leftarrow |\bot\rangle \\ \mathrm{O} \leftarrow \mathsf{A}^{O(\mathrm{D}),U(\mathrm{D})} \\ b \leftarrow D(\mathrm{O},\mathrm{D}) \end{array} \right]} - \sqrt{\Pr\left[b \wedge b' = 1 \;\middle|\; \begin{array}{l} \mathrm{D} \leftarrow |\bot\rangle \\ (b', \mathrm{O}) \leftarrow \mathsf{A}_{\mathit{NoCollision}}^{O(\mathrm{D}),U(\mathrm{D})} \\ b \leftarrow D(\mathrm{O},\mathrm{D}) \end{array} \right]} \right|^2$$
$$\leq 24 \cdot t^2 w \cdot 2^{-m} \ .$$

In general, the same bound holds for A', which runs as A except that the measurement $\{\Pi_{\mathsf{NoCol}}, I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}\}$ is performed over D before the i-th query for i in a prescribed set $I \subseteq [t+1]$:

$$\left| \sqrt{\Pr\left[b = 1 \;\middle|\; \begin{array}{l} \mathrm{D} \leftarrow |\bot\rangle \\ \mathrm{O} \leftarrow \mathsf{A}^{O(\mathrm{D}),U(\mathrm{D})} \\ b \leftarrow D(\mathrm{O},\mathrm{D}) \end{array} \right]} - \sqrt{\Pr\left[b \wedge b' = 1 \;\middle|\; \begin{array}{l} \mathrm{D} \leftarrow |\bot\rangle \\ (b', \mathrm{O}) \leftarrow \mathsf{A}'^{O(\mathrm{D}),U(\mathrm{D})} \\ b \leftarrow D(\mathrm{O},\mathrm{D}) \end{array} \right]} \right|^2$$
$$\leq 24 \cdot t^2 w \cdot 2^{-m} \ .$$

Proof. We prove the first part of the lemma. The second part of the lemma can be proved with the same idea.

By the deferred measurement principle, we can assume A always performs a unitary U_i to prepare the i-th query for $i \in [t]$ and performs the unitary U_{t+1} to prepare the output register. Without loss of generality, we can assume the distinguisher D performs a projective measurement $\{M_0, M_1\}$.

Define $|\psi_i\rangle \coloneqq U_{i+1}O' \cdots O'U_1 |\bar{0}\rangle$ where the unitary

$$O' \coloneqq |0\rangle\langle 0|_{\mathrm{N}} \otimes O + |1\rangle\langle 1|_{\mathrm{N}} \otimes U$$

is the controlled unitary that implements the superposition queries to the two oracles O and U. Then the state $|\psi_t\rangle = U_{t+1}O' \cdots O'U_1 |\bar{0}\rangle$ is the state on registers (O, D) immediately after A is finished.

Define $|\phi_i\rangle \coloneqq U_{i+1}O'_{\text{NoCollision}} \cdots O'_{\text{NoCollision}}U_1 |\bar{0}\rangle$, where

$$O'_{\text{NoCollision}} \coloneqq \Pi_{\mathsf{NoCol}}O'\Pi_{\mathsf{NoCol}} \ .$$

Then the subnormalized state $|\phi_t\rangle = U_{t+1}O'_{\text{NoCollision}} \cdots O'_{\text{NoCollision}}U_1 |\bar{0}\rangle$ is the state on registers (O, D) immediately after A is finished conditioned on all of the measurement outcomes being Π_{NoCol}.

Then we have that

$$\begin{aligned}
&\left| \sqrt{\Pr\left[b = 1 \;\middle|\; \begin{array}{l} \mathrm{D} \leftarrow |\perp\rangle \\ \mathrm{O} \leftarrow \mathsf{A}^{O(\mathrm{D}),U(\mathrm{D})} \\ b \leftarrow D(\mathrm{O},\mathrm{D}) \end{array} \right]} - \sqrt{\Pr\left[b \wedge b' = 1 \;\middle|\; \begin{array}{l} \mathrm{D} \leftarrow |\perp\rangle \\ (b', \mathrm{O}) \leftarrow \mathsf{A}^{O(\mathrm{D}),U(\mathrm{D})}_{\text{NoCollision}} \\ b \leftarrow D(\mathrm{O},\mathrm{D}) \end{array} \right]} \right|^2 \\
&\leq \left| \| M_1 |\psi_t\rangle \| - \| M_1 |\phi_t\rangle \| \right|^2 \\
&\leq \| M_1(|\psi_t\rangle - |\phi_t\rangle) \|^2 \qquad \text{(By the triangle inequality)} \\
&\leq \| \, |\psi_t\rangle - |\phi_t\rangle \, \|^2 \ .
\end{aligned}$$

Since one query to the oracle O increases the size of the database by at most 1, and one query to the oracle U does not change the size of the database (as $[\Pi_q, U] = 0$ for every q), the states satisfy that for every $i \in [t]$,

$$|\psi_i\rangle = U_{i+1}\Pi_t O'\Pi_t \cdots \Pi_t O'\Pi_t U_1 |\bar{0}\rangle$$

and

$$|\phi_i\rangle = U_{i+1}\Pi_t O'_{\text{NoCollision}}\Pi_t \cdots \Pi_t O'_{\text{NoCollision}}\Pi_t U_1 |\bar{0}\rangle \ .$$

As a result,

$$\begin{aligned}
&\| \, |\psi_t\rangle - |\phi_t\rangle \, \| \\
&\leq \sum_{i=0}^{t-1} \| U_{t+1}\Pi_t O'_{\text{NoCollision}} \cdots O'_{\text{NoCollision}}\Pi_t U_{i+2}\Pi_t (O'_{\text{NoCollision}} - O')\Pi_t |\psi_i\rangle \, \| \qquad \text{(By the triangle inequality)} \\
&\leq \sum_{i=0}^{t-1} \| U_{t+1}\Pi_t O'_{\text{NoCollision}} \cdots O'_{\text{NoCollision}}\Pi_t U_{i+2}(\Pi_{\mathsf{NoCol}} + I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}})\Pi_t (O'_{\text{NoCollision}} - O')\Pi_t |\psi_i\rangle \, \| \\
&\leq \sum_{i=0}^{t-1} \| \Pi_{\mathsf{NoCol}}\Pi_t (O'_{\text{NoCollision}} - O')\Pi_t |\psi_i\rangle \, \| + \| (I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}})\Pi_t O'\Pi_t |\psi_{t-1}\rangle \, \| \\
&\leq \sum_{i=0}^{t-1} \| \Pi_{\mathsf{NoCol}}\Pi_t (O'_{\text{NoCollision}} - O')\Pi_t |\psi_i\rangle \, \| + \sum_{i=0}^{t-1} \| (I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}})\Pi_t O'\Pi_t\Pi_{\mathsf{NoCol}} |\psi_i\rangle \, \| \\
&\qquad\qquad \text{(By the triangle inequality)}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=0}^{t-1} \|\Pi_{\mathsf{NoCol}}\Pi_t(O'_{\text{NoCollision}} - O')\Pi_t \left|0\right\rangle\!\left\langle 0\right|_{\mathrm{N}} \left|\psi_i\right\rangle \| + \sum_{i=0}^{t-1} \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}})\Pi_t O' \Pi_t \Pi_{\mathsf{NoCol}} \left|0\right\rangle\!\left\langle 0\right|_{\mathrm{N}} \left|\psi_i\right\rangle \| \\
&\leq \sum_{i=0}^{t-1} \left(\|\Pi_{\mathsf{NoCol}}\Pi_t(O'_{\text{NoCollision}} - O')\Pi_t\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}})\Pi_t O' \Pi_t \Pi_{\mathsf{NoCol}}\| \right) \| \left|0\right\rangle\!\left\langle 0\right|_{\mathrm{N}} \left|\psi_i\right\rangle \| \\
&= 2 \sum_{i\in[t]} \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}})\left(\Pi_t O \Pi_t\right)\Pi_{\mathsf{NoCol}}\| \sqrt{\mathsf{mass}(\mathsf{A}, O(\mathrm{D}), i)} \\
&\leq 2\sqrt{6t} \cdot 2^{-m/2} \sqrt{t \sum_{i\in[t]} \mathsf{mass}(\mathsf{A}, O(\mathrm{D}), i)} \qquad \text{(By Lemma 2.4 and Cauchy–Schwarz inequality)} \\
&\leq 2\sqrt{6t} \cdot 2^{-m/2} \sqrt{tw} \ ,
\end{aligned}
$$

where in the sixth line, we use that $[\Pi_{\mathsf{NoCol}}, U] = 0$ and $[\Pi_t, U] = 0$, and thus $\Pi_{\mathsf{NoCol}}\Pi_t(O'_{\text{NoCollision}} - O')\Pi_t \left|1\right\rangle\!\left\langle 1\right|_{\mathrm{N}} = 0$ and $(I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}})\Pi_t O' \Pi_t \Pi_{\mathsf{NoCol}} \left|1\right\rangle\!\left\langle 1\right|_{\mathrm{N}} = 0$.

The lemma follows from the above two inequalities. □

2.4 Relations and languages

We consider quantum arguments for relations, where the honest prover additionally gets a quantum witness as input. Thus we need to define a relation between the classical instance and the quantum witness state.

Definition 2.6 (Relations and languages)**.** *A relation R is a set of tuples of a classical string and a quantum state. Namely, $R \subseteq \{(\mathbb{x}, \boldsymbol{\rho}) : \mathbb{x} \in \{0,1\}^*, \boldsymbol{\rho}$ is a quantum state$\}$.*

A language for a relation R is defined as $L(R) = \{\mathbb{x} : \exists \boldsymbol{\rho}, (\mathbb{x}, \boldsymbol{\rho}) \in R\}$.

We say a relation has a decider $\mathsf{Decider}$*, if* $\mathsf{Decider}$ *is a quantum algorithm such that for every $(\mathbb{x}, \boldsymbol{\rho}) \in R$,*

$$\Pr\left[\mathsf{Decider}(\mathbb{x}, \boldsymbol{\rho}) = 1\right] \geq \frac{2}{3} \ ,$$

and for every $\mathbb{x} \notin L(R)$ and quantum state $\boldsymbol{\rho}$,

$$\Pr\left[\mathsf{Decider}(\mathbb{x}, \boldsymbol{\rho}) = 1\right] \leq \frac{1}{3} \ .$$

A decider is efficient, if it takes a quantum state of $\mathsf{poly}(\nu)$ *qubits, and runs in time* $\mathsf{poly}(\nu)$ *where ν is the length of the instance $\mathbb{x}$.*

Then QMA is the set of languages $L(R)$ such that R has an efficient decider.

2.5 Quantum interactive arguments

A quantum interactive argument $(\mathcal{P}, \mathcal{V})$ for a relation R consists of a quantum polynomial-time prover $\mathcal{P}$ and a quantum polynomial-time verifier $\mathcal{V}$ that exchange quantum messages. Both parties receive an instance $\mathbb{x}$ as input, while $\mathcal{P}$ receives a quantum state $\boldsymbol{\rho}$ as the witness. After the interaction, the verifier $\mathcal{V}$ outputs a bit, indicating whether $\mathcal{V}$ accepts. We write $\Pr[\langle \mathcal{P}(\mathbb{x}, \boldsymbol{\rho}), \mathcal{V}(\mathbb{x})\rangle = 1]$ for the probability that the verifier accepts after the interaction.

In this paper, we consider quantum interactive arguments in the quantum random oracle model where both prover $\mathcal{P}$ and $\mathcal{V}$ have quantum query access to a random oracle RO sampled from $\mathcal{U}(m)$. We write $\Pr[\langle \mathcal{P}^{\mathsf{RO}}(\mathbb{x}, \boldsymbol{\rho}), \mathcal{V}^{\mathsf{RO}}(\mathbb{x})\rangle = 1]$ for the probability that the verifier accepts after the interaction when the prover and the verifier have access to RO.

Definition 2.7 (Completeness)**.** *A quantum interactive argument $(\mathcal{P}, \mathcal{V})$ for a relation R in the quantum random oracle model has completeness c_{QARG} if for every integer λ, ν, function $\mathsf{RO} : \{0,1\}^* \to \{0,1\}^m$ where the output length m of the random oracle is a function of the security parameter λ specified by the scheme, and $(\mathbb{x}, \boldsymbol{\rho}) \in R$ such that $|\mathbb{x}| \leq \nu$,*

$$\Pr[\langle \mathcal{P}^{\mathsf{RO}}(\mathbb{x}, \boldsymbol{\rho}), \mathcal{V}^{\mathsf{RO}}(\mathbb{x})\rangle = 1] \geq c_{\mathsf{QARG}}(\nu, \lambda) \ .$$

Definition 2.8 (Soundness)**.** *A quantum interactive argument $(\mathcal{P}, \mathcal{V})$ for a relation R in the quantum random oracle model has soundness s_{QARG} if for every integer λ, ν, t, and t-query quantum adversary $\widetilde{\mathcal{P}}$,*

$$\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq \nu & \mathsf{RO} \leftarrow \mathcal{U}(m) \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathcal{P}}^{\mathsf{RO}} \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathcal{P}}^{\mathsf{RO}}, \mathcal{V}^{\mathsf{RO}}(\mathbb{x})\rangle \end{array}\right] \leq s_{\mathsf{QARG}}(\nu, \lambda, t) \ ,$$

where the output length m of the random oracle is a function of the security parameter λ specified by the scheme.

2.6 Quantum interactive oracle proofs

Quantum interactive oracle proofs (QIOPs), introduced in [SV25], are proof systems that combine the QPCPs and the quantum interactive proof (QIP) systems. Basically, QIOPs are QIPs such that the verifier does not necessarily read all qubits of the prover's message. We adopt the formulation of [CGMV26] and extend it to allow the verifier to keep a proof register across rounds and query it in subsequent rounds before returning it to the prover.

Superposition access to quantum registers. We adopt the notion of superposition access to quantum registers from [CGMV26], which allows an algorithm to access a list of quantum registers of the same size coherently. Generally, for a list of registers $(\mathrm{B}_i)_{i\in L}$ with the same size, a quantum *superposition query* of width w to the oracle $(\mathrm{B}_i)_{i\in L}$ consists of w pairs of registers $(\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}]}$. The oracle applies the unitary

$$U^{\mathsf{qry}} := \sum_{i\in L} |i\rangle\langle i|_{\mathrm{Loc}_\iota} \otimes \mathsf{SWAP}\,[\mathrm{B}_i, \mathrm{Y}_\iota]$$

on the registers $(\mathrm{Loc}_\iota, \mathrm{Y}_\iota, (\mathrm{B}_i)_{i\in L})$ for each $\iota \in [\mathsf{w}]$ before sending the registers $(\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}]}$ back. We say that an algorithm A has quantum *superposition access* to a list of registers $(\mathrm{B}_i)_{i\in L}$ with query depth d and query width w if its computation consists of unitaries $(\mathsf{A}^{(i)})_{i\in\{0,1,\ldots,\mathsf{d}\}}$, none of which acts directly on $(\mathrm{B}_i)_{i\in L}$, interleaved with d quantum superposition queries with width at most w to the oracle $(\mathrm{B}_i)_{i\in L}$, where d and w are polynomially bounded. We denote an algorithm with such access by $\mathsf{A}^{(\mathrm{B}_i)_{i\in L}}$. For simplicity, we only consider algorithms whose superposition queries all have the same width. We define the query complexity of A to be $\mathsf{q} := \mathsf{d}\cdot\mathsf{w}$.

QIOP protocol. We denote a k-round QIOP protocol by $(\mathbf{P}, \mathbf{V})$ where $\mathbf{P} = (\mathbf{P}_i)_{i\in[\mathsf{k}]}$ is a tuple of polynomial-time quantum algorithms and $\mathbf{V} = (\mathbf{V}_i)_{i\in[\mathsf{k}]}$ is a tuple of polynomial-time quantum algorithms with superposition access to registers. We specify different registers for messages and the internal states for different rounds. In more detail, we name the register that holds the i-th prover's message by Mp_i, and the register that holds the i-th prover's internal state by P_i, and similarly, name the register that holds the i-th verifier's message by Mv_i, and the register that holds the i-th verifier's internal state by V_i. We view the prover's message registers that have not been returned yet as held by a trusted third party $\mathbf{O}$, who answers the

verifier's superposition queries to these registers. When the prover needs a previously submitted register, the trusted third party removes that register from the verifier's query access and sends it back to the prover.

Before the interaction, both $\mathbf{P}$ and $\mathbf{V}$ receive the same instance $\mathbb{x}$ for the relation R. The honest prover $\mathbf{P}$ also receives a quantum witness $\boldsymbol{\rho}$ in register P_1. We initialize V_1 to the all-zero state, set Mv_0 to be an empty register, and set $J_1 \coloneqq \emptyset$.

The prover $\mathbf{P}$ and verifier $\mathbf{V}$ then interact as follows. In round i, the prover runs $\mathbf{P}_i$ on the instance $\mathbb{x}$, the verifier's message register Mv_{i-1}, the returned proof registers $(\mathrm{Mp}_j)_{j\in J_i}$, and its private working register P_i. It produces a message in register $\mathrm{Mp}_i \coloneqq (\mathrm{Mp}_i[j])_{j\in[\mathsf{l}_i]}$ and an updated private state in register P_{i+1}. The prover sends Mp_i to the trusted third party $\mathbf{O}$. At this point, $\mathbf{O}$ holds the registers $(\mathrm{Mp}_{i'}[j'])_{(i',j')\in L}$ where $L \coloneqq \{(i',j') : i' \in I_i, j' \in [\mathsf{l}_{i'}]\}$ and I_i denotes the set of indices of the prover's messages that have yet to be returned. For $i \in [\mathsf{k}-1]$, the verifier $\mathbf{V}$, with *superposition access* to the registers $(\mathrm{Mp}_{i'}[j'])_{(i',j')\in L}$, computes $(J_{i+1}, \mathrm{Mv}_i, \mathrm{V}_{i+1}) \leftarrow \mathbf{V}_i^{(\mathrm{Mp}_{i'}[j'])_{(i',j')\in L}}(\mathbb{x}, \mathrm{V}_i)$. The verifier sends Mv_i to the prover and asks $\mathbf{O}$ to return the registers $(\mathrm{Mp}_j)_{j\in J_{i+1}}$ to the prover. In the final round $i = \mathsf{k}$, the verifier outputs $b \leftarrow \mathbf{V}_{\mathsf{k}}^{(\mathrm{Mp}_{i'}[j'])_{(i',j')\in L}}(\mathbb{x}, \mathrm{V}_{\mathsf{k}})$.

We summarize the above interaction as follows:

$$\left[\begin{array}{l}
\mathrm{P}_1 \leftarrow \boldsymbol{\rho} \\
(\mathrm{Mp}_1, \mathrm{P}_2) \leftarrow \mathbf{P}_1(\mathbb{x}, \mathrm{P}_1) \\
(\mathrm{Mp}_1[j])_{j\in[\mathsf{l}_1]} \coloneqq \mathrm{Mp}_1 \\
I_1 \leftarrow \{1\}, L \leftarrow \{(1,j) : j \in [\mathsf{l}_1]\} \\
\text{For } i = 2, \ldots, \mathsf{k}: \\
\quad (J_i, \mathrm{Mv}_{i-1}, \mathrm{V}_i) \leftarrow \mathbf{V}_{i-1}^{(\mathrm{Mp}_{i'}[j'])_{(i',j')\in L}}(\mathbb{x}, \mathrm{V}_{i-1}) \\
\quad (\mathrm{Mp}_i, \mathrm{P}_{i+1}) \leftarrow \mathbf{P}_i(\mathbb{x}, \mathrm{Mv}_{i-1}, \mathrm{P}_i, (\mathrm{Mp}_j)_{j\in J_i}) \\
\quad (\mathrm{Mp}_i[j])_{j\in[\mathsf{l}_i]} \coloneqq \mathrm{Mp}_i \\
\quad I_i \leftarrow (I_{i-1} \cup \{i\}) \setminus J_i, L \leftarrow \{(i',j') : i' \in I_i, j' \in [\mathsf{l}_{i'}]\} \\
b \leftarrow \mathbf{V}_{\mathsf{k}}^{(\mathrm{Mp}_{i'}[j'])_{(i',j')\in L}}(\mathbb{x}, \mathrm{V}_{\mathsf{k}})
\end{array}\right],$$

which we abbreviate by $b \leftarrow \langle \mathbf{P}(\mathbb{x}, \boldsymbol{\rho}), \mathbf{V}(\mathbb{x})\rangle$.

For a computationally unbounded quantum adversary $\widetilde{\mathbf{P}}$, we similarly write $\mathbb{x} \leftarrow \widetilde{\mathbf{P}}$ and $b \leftarrow \langle \widetilde{\mathbf{P}}, \mathbf{V}(\mathbb{x})\rangle$ for the soundness experiment in which $\widetilde{\mathbf{P}}$ first outputs an instance $\mathbb{x}$ and then interacts with $\mathbf{V}(\mathbb{x})$ in place of the honest QIOP prover, where b denotes the verifier's decision bit.

We say that a k-round quantum interactive oracle proof $\mathsf{QIOP} = (\mathbf{P}, \mathbf{V})$ for a relation R has completeness c and soundness s if the following holds.

Definition 2.9 (Completeness)**.** *For every integer ν and instance-witness pair $(\mathbb{x}, \boldsymbol{\rho}) \in R$ such that $|\mathbb{x}| \leq \nu$,*

$$\Pr\left[\langle \mathbf{P}(\mathbb{x}, \boldsymbol{\rho}), \mathbf{V}(\mathbb{x})\rangle = 1\right] \geq c(\nu) \ .$$

Definition 2.10 (Soundness)**.** *For every integer ν and computationally unbounded quantum adversary $\widetilde{\mathbf{P}}$,*

$$\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq \nu & \mathbb{x} \leftarrow \widetilde{\mathbf{P}} \\ \wedge\, \mathbb{x} \notin L(R) & b \leftarrow \langle \widetilde{\mathbf{P}}, \mathbf{V}(\mathbb{x})\rangle \\ \wedge\, b = 1 & \end{array}\right] \leq s(\nu) \ .$$

We consider the following efficiency measures of a QIOP.

- The *round complexity* k is the number of rounds of interactions between the prover and the verifier.

- The *query depth* is the total query depth of the verifier algorithms $(\mathbf{V}_i)_{i\in[\mathsf{k}]}$. Namely, let d_i denote the query depth of $\mathbf{V}_i$, and we define the *query depth* of the QIOP

$$\mathsf{d} := \sum_{i\in[\mathsf{k}]} \mathsf{d}_i \ .$$

- The *query width* is an upper bound for the query width of verifier algorithms $(\mathbf{V}_i)_{i\in[\mathsf{k}]}$. Namely, let w_i denote the query width of $\mathbf{V}_i$, and we define the *query width* of the QIOP

$$\mathsf{w} := \max_{i\in[\mathsf{k}]} \mathsf{w}_i \ .$$

- The *query complexity* is the total query complexity of $(\mathbf{V}_i)_{i\in[\mathsf{k}]}$. Namely, let $\mathsf{q}_i := \mathsf{w}_i \cdot \mathsf{d}_i$ denote the query complexity of $\mathbf{V}_i$, and we define the *query complexity* of the QIOP

$$\mathsf{q} := \sum_{i\in[\mathsf{k}]} \mathsf{q}_i \ .$$

- The *total proof length* is the sum of the lengths of the prover messages over all rounds, while the *maximum proof length* is the maximum length of a prover message in any round. Namely,

$$\mathsf{l} := \sum_{i\in[\mathsf{k}]} \mathsf{l}_i, \ \mathsf{l}_{\max} := \max_{i\in[\mathsf{k}]} \mathsf{l}_i \ .$$

- For each $i \in [\mathsf{k}-1]$, let vc_i denote the sum of the size of the verifier's message register Mv_i in qubits and the number of bits used to describe J_{i+1}. We define the *verifier-to-prover communication* of the QIOP

$$\mathsf{vc} := \sum_{i\in[\mathsf{k}-1]} \mathsf{vc}_i \ .$$

Notice that we do not count the returned prover messages toward the verifier-to-prover communication, following [SV26; CGMV26], since they are returned by the trusted third party $\mathbf{O}$, rather than by the verifier.

3 Public-query quantum interactive oracle proofs

As in the classical IBCS transformation [CDGS23], our compiler from quantum interactive oracle proofs to succinct quantum interactive arguments applies only to public-query QIOPs. This restriction arises because the query locations must be revealed to the argument prover in order to open the corresponding locations, and this additional information may be exploited by a malicious argument prover.

In the classical setting, the public-query property roughly requires soundness to hold even when the query locations are revealed to the prover at the time the queries are made [CDGS23]. In the quantum setting, however, the query locations are stored in superposition in the registers $(\mathrm{Loc}_i)_{i\in[\mathsf{w}]}$. We therefore adopt a natural quantum analogue of the classical notion. Namely, we say that a QIOP is *public-query* if its soundness continues to hold even when the verifier's superposition queries are replaced by *leaked superposition queries*.

Informally, a leaked superposition query with respect to an adversary B is performed as follows. Upon receiving a query $(\mathrm{Loc}_i, \mathrm{Y}_i)_{i\in[\mathsf{w}]}$, the trusted third party $\mathbf{O}$ sends the query location registers $(\mathrm{Loc}_i)_{i\in[\mathsf{w}]}$ to B. The adversary B may then apply an arbitrary quantum operation jointly to the registers $(\mathrm{Loc}_i)_{i\in[\mathsf{w}]}$ and the private register before returning $(\mathrm{Loc}_i)_{i\in[\mathsf{w}]}$ to $\mathbf{O}$, who answers the query by applying the same query unitary U^{qry} as in the ordinary superposition query setting. After $\mathbf{O}$ performs the query, the query location registers $(\mathrm{Loc}_i)_{i\in[\mathsf{w}]}$ are leaked to B again, who may then apply another arbitrary quantum operation jointly to the registers $(\mathrm{Loc}_i)_{i\in[\mathsf{w}]}$ and the private register before returning $(\mathrm{Loc}_i)_{i\in[\mathsf{w}]}$. We formalize this interaction below.

Definition 3.1 (Leaked superposition queries)**.** *Let $(\mathrm{M}_i)_{i\in L}$ be a list of registers of equal size, and let B be an adversary with internal register* J*.*

Let A *be an algorithm with leaked superposition access to $(\mathrm{M}_i)_{i\in L}$ of query depth* d *and query width* w*. Let x denote the classical input of* A*, let $\mathrm{I}^{(q)}$ denote its private register after the q-th invocation of its internal unitary, and let* O *denote its output register.*

The computation of A *with superposition queries leaked to the adversary B is given by the following experiment:*

$$\left[\begin{array}{l}
((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{I}^{(1)}) \leftarrow \mathsf{A}^{(0)}(x, \mathrm{I}^{(0)}) \\
((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{J}) \leftarrow B((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{J}) \\
\textit{For } q = 1, \ldots, \mathsf{d}-1: \\
\quad ((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}]}, (\mathrm{M}_i)_{i\in L}) \leftarrow \mathbf{O}((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}]}, (\mathrm{M}_i)_{i\in L}) \\
\quad ((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{J}) \leftarrow B((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{J}) \\
\quad ((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{I}^{(q+1)}) \leftarrow \mathsf{A}^{(q)}(x, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{I}^{(q)}) \\
\quad ((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{J}) \leftarrow B((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{J}) \\
((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}]}, (\mathrm{M}_i)_{i\in L}) \leftarrow \mathbf{O}((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}]}, (\mathrm{M}_i)_{i\in L}) \\
((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{J}) \leftarrow B((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{J}) \\
(\mathrm{O}, \mathrm{I}^{(\mathsf{d}+1)}) \leftarrow \mathsf{A}^{(\mathsf{d})}(x, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}]}, \mathrm{I}^{(\mathsf{d})}),
\end{array}\right]$$

where $\mathbf{O}$ *is the trusted party that answers the superposition queries by applying the unitary*

$$U^{\mathsf{qry}} := \sum_{i\in L} |i\rangle\langle i|_{\mathrm{Loc}_\iota} \otimes \mathsf{SWAP}\,[\mathrm{M}_i, \mathrm{Y}_\iota]$$

on the registers $(\mathrm{Loc}_\iota, \mathrm{Y}_\iota, (\mathrm{M}_i)_{i\in L})$ for each $\iota \in [\mathsf{w}]$.

We abbreviate this computation as

$$(\mathrm{O}, \mathrm{I}^{(\mathsf{d}+1)}) \leftarrow \mathsf{A}^{\mathsf{Leaked}(\mathbf{O}, B, (\mathrm{M}_i)_{i\in L})}(x, \mathrm{I}^{(0)}).$$

Notice that the adversary's private register I *is also updated as in the experiment above and is left implicit in this notation. For algorithms with multiple inputs or outputs, we use the analogous notation.*

The public-query soundness experiment follows the soundness experiment in Definition 2.10, with each of the verifier's superposition queries replaced by a leaked superposition query. Formally, we write $\mathbb{x} \leftarrow \widetilde{\mathbf{P}}$ and $b \leftarrow \langle \widetilde{\mathbf{P}}, \mathbf{V}(\mathbb{x})\rangle_{\mathsf{pq}}$ to denote the following experiment:

$$\left[\begin{array}{l}
(\mathbb{x}, \mathrm{P}_1) \leftarrow \widetilde{\mathbf{P}} \\
(\mathrm{Mp}_1, \mathrm{P}_2) \leftarrow \widetilde{\mathbf{P}}(\mathbb{x}, \mathrm{P}_1) \\
(\mathrm{Mp}_1[j])_{j\in[\mathsf{l}_1]} \coloneqq \mathrm{Mp}_1 \\
I_1 \leftarrow \{1\}, L \leftarrow \{(1,j) : j \in [\mathsf{l}_1]\} \\
\text{For } i = 2, \ldots, \mathsf{k}: \\
\quad (J_i, \mathrm{Mv}_{i-1}, \mathrm{V}_i) \leftarrow \mathbf{V}_{i-1}^{\mathsf{Leaked}(\mathbf{O}, \widetilde{\mathbf{P}}, (\mathrm{Mp}_{i'}[j'])_{(i',j')\in L})}(\mathbb{x}, \mathrm{V}_{i-1}) \\
\quad (\mathrm{Mp}_i, \mathrm{P}_{i+1}) \leftarrow \widetilde{\mathbf{P}}(\mathbb{x}, \mathrm{Mv}_{i-1}, \mathrm{P}_i, (\mathrm{Mp}_j)_{j\in J_i}) \\
\quad (\mathrm{Mp}_i[j])_{j\in[\mathsf{l}_i]} \coloneqq \mathrm{Mp}_i \\
\quad I_i \leftarrow I_{i-1} \cup \{i\} \setminus J_i, L \leftarrow \{(i',j') : i' \in I_i, j' \in [\mathsf{l}_{i'}]\} \\
b \leftarrow \mathbf{V}_{\mathsf{k}}^{\mathsf{Leaked}(\mathbf{O}, \widetilde{\mathbf{P}}, (\mathrm{Mp}_{i'}[j'])_{(i',j')\in L})}(\mathbb{x}, \mathrm{V}_{\mathsf{k}})
\end{array}\right] .$$

Public-query soundness bounds the probability that the verifier accepts a no instance in this experiment.

Definition 3.2 (Public-query soundness)**.** *A quantum interactive oracle proof* $\mathsf{QIOP} = (\mathbf{P}, \mathbf{V})$ *for a relation* R *has public-query soundness* s_{pq} *if for every integer* ν *and computationally unbounded adversary* $\widetilde{\mathbf{P}}$*,*

$$\Pr\left[\begin{array}{l|l} |\mathbb{x}| \le \nu & \mathbb{x} \leftarrow \widetilde{\mathbf{P}} \\ \wedge\, \mathbb{x} \notin L(R) & b \leftarrow \langle \widetilde{\mathbf{P}}, \mathbf{V}(\mathbb{x})\rangle_{\mathsf{pq}} \\ \wedge\, b = 1 & \end{array}\right] \le s_{\mathsf{pq}}(\nu) \ .$$

Public-coin QIOPs. A QIOP is *public-coin* if each verifier's message register Mv_i consists of fresh classical random coins. Moreover, for each round $i \in [\mathsf{k}]$, both the returned index set J_i and the classical query set used by the verifier for point queries in that round are determined by publicly known deterministic functions of the instance $\mathbb{x}$ and the random coins contained in the preceding verifier's message registers $(\mathrm{Mv}_j)_{j\in[i-1]}$.

For public-coin QIOPs, we assume without loss of generality that, immediately after each leakage in round i, the verifier derives the query set from the instance $\mathbb{x}$ and the random coins in $(\mathrm{Mv}_j)_{j\in[i-1]}$, and the verifier rejects if the query set given by the prover differs from this derived set. These checks do not affect the ordinary execution of the public-coin QIOP and ensure that the malicious prover does not modify the query set. Moreover, the leaked query locations are determined by the instance $\mathbb{x}$ and the random coins in the preceding verifier's message registers $(\mathrm{Mv}_j)_{j\in[i-1]}$. Thus, they reveal no additional information to the malicious prover. Consequently, we obtain the following.

Lemma 3.3. *Every public-coin* QIOP *with soundness* s *has public-query soundness* $s_{\mathsf{pq}}(\nu) = s(\nu)$.

4 Defining extractable quantum state commitments

We describe the syntax of (non-interactive) quantum state commitments we will consider, and then provide a formal definition of extractability.

4.1 A canonical quantum state commitment

Following [Yan22; GJMZ23], we consider quantum state commitments in the following canonical form, where to commit to a quantum message, the sender always applies a unitary quantum circuit to the quantum message along with enough ancilla qubits initialized as $|0\rangle$, and the receiver always does the inverse of what the sender does and checks whether the ancilla qubits are $|0\rangle$.

Definition 4.1 (Quantum state commitment scheme in the QROM)**.** *Let $m \in \mathbb{N}$ be the output length of the random oracle and $n \in \mathbb{N}$ be the length of the quantum state which we want to commit to. Let RO be sampled uniformly at random from all the functions with range $\{0,1\}^m$.*

In the QROM, a quantum state commitment (QSC) *is a pair of oracle-aided quantum algorithms $(\mathsf{Com}^{\mathsf{RO}}, \mathsf{Check}^{\mathsf{RO}})$ with oracle access to RO where $\mathsf{Com}^{\mathsf{RO}}$ and $\mathsf{Check}^{\mathsf{RO}}$ are specified by a $\mathsf{poly}(m,n)$-qubit ancilla register A and an oracle-aided $\mathsf{poly}(m,n)$-size unitary quantum circuit $C^{\mathsf{RO}}_{m,n}$, as follows.*

$\mathsf{Com}^{\mathsf{RO}}(\boldsymbol{\rho}_{\mathrm{M}})$*:*
1. *Initialize A as all $|0\rangle$ states.*
2. *To commit to an n-qubit quantum message $\boldsymbol{\rho}_{\mathrm{M}}$, apply $C^{\mathsf{RO}}_{m,n}$ to the registers M and A to get a state $\boldsymbol{\sigma}_{\mathrm{CO}}$, which is divided into two registers C and O.*
3. *Output the state on C as the commitment, and the state on O as the decommitment.*

$\mathsf{Check}^{\mathsf{RO}}(\boldsymbol{\sigma}_{\mathrm{CO}})$*:*
1. *Apply the inverse of $C^{\mathsf{RO}}_{m,n}$ to the registers C and O to get a state $\boldsymbol{\varrho}_{\mathrm{MA}}$.*
2. *Make a computational measurement on the register A.*
3. *If the outcome is not all 0, output $b = 0$ and a dummy quantum message $|0^n\rangle$ on register M.*
4. *Otherwise, output $b = 1$ and the state $\boldsymbol{\rho}_{\mathrm{M}}$ in register M.*

A commitment is succinct *if the length of C is less than the length of the quantum message.*
A commitment is n-to-t if the quantum message has n qubits while C has t qubits.

A canonical commitment scheme always satisfies the perfect correctness as defined below.

Definition 4.2 (Perfect correctness)**.** *Let $m \in \mathbb{N}$ be the output length of the random oracle. Let RO be sampled uniformly at random from all the functions with range $\{0,1\}^m$. A quantum state commitment scheme $(\mathsf{Com}^{\mathsf{RO}}, \mathsf{Check}^{\mathsf{RO}})$ has perfect correctness if for every function RO with output length m, unbounded quantum adversary A, and unbounded quantum distinguisher D, the following holds:*

$$\Pr\left[1 \leftarrow D^{\mathsf{RO}}(b,\mathrm{M},\mathrm{E}) \;\middle|\; \begin{array}{l}(\mathrm{M},\mathrm{E}) \leftarrow A^{\mathsf{RO}} \\ (\mathrm{C},\mathrm{O}) \leftarrow \mathsf{Com}^{\mathsf{RO}}(\mathrm{M}) \\ (b,\mathrm{M}) \leftarrow \mathsf{Check}^{\mathsf{RO}}(\mathrm{C},\mathrm{O})\end{array}\right] = \Pr\left[1 \leftarrow D^{\mathsf{RO}}(1,\mathrm{M},\mathrm{E}) \mid (\mathrm{M},\mathrm{E}) \leftarrow A^{\mathsf{RO}}\right] \ .$$

Lemma 4.3. *Every canonical quantum state commitment scheme $(\mathsf{Com}^{\mathsf{RO}}, \mathsf{Check}^{\mathsf{RO}})$ defined in Definition 4.1 has perfect correctness (Definition 4.2).*

Proof. By definition, $\mathsf{Check}^{\mathsf{RO}}$ does exactly the inverse of $\mathsf{Com}^{\mathsf{RO}}$. Since in $\mathsf{Com}^{\mathsf{RO}}$, A is properly initialized, before the computational measurement in $\mathsf{Check}^{\mathsf{RO}}$, qubits of A are all in the $|0\rangle$ state, so $\mathsf{Check}^{\mathsf{RO}}$ always outputs $b = 1$, and the original quantum message (while preserving the entanglement between M and E). □

4.2 The extractability definition

We first provide the syntax of an extractor in this case. In addition to the state in the commitment register C, an extractor should have some side information of the oracle, provided by a quantum simulator. Ideally, the simulator maintains the side information in its internal state register S without disturbing any adversary's execution.

Definition 4.4 (Quantum simulator)**.** *A stateful oracle U_{Sim} with the state register S is a $\xi_{\mathsf{Sim}}(t,m)$-*quantum simulator *for the random oracle if for any t-query quantum adversary A,*

$$\left|\Pr\left[1 \leftarrow A^{U_{\mathsf{Sim}}(\mathrm{S})}(|\bot\rangle_{\mathrm{S}})\right] - \Pr\left[1 \leftarrow A^{\mathsf{RO}} | \mathsf{RO} \leftarrow \mathcal{U}(m)\right]\right| \leq \xi_{\mathsf{Sim}}(t,m) \ ,$$

where m is the output length of the random oracle.

A ξ_{Sim}-quantum simulator is a perfect *quantum simulator for the random oracle if $\xi_{\mathsf{Sim}} = 0$.*

Theorem 4.5 ([Zha19])**.** *O with the state register D is a perfect quantum simulator for the random oracle, where*

$$O_{\mathrm{XYD}} = \sum_{x \in X} |x\rangle\langle x|_{\mathrm{X}} \otimes F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]Y}} F_{\mathrm{D[x]}}$$

is the oracle unitary defined in Section 2.3.

Now we are ready to define the syntax of an extractor.

Definition 4.6 (Quantum message extractor)**.** *A* quantum message extractor *$\mathcal{E}$ with query access to U_{Sim} and U_{Ext} (both have the state register S) has the following syntax:*

1. *$\mathcal{E}.\mathsf{ExtMsg}$ takes as input a commitment in register C, makes queries to U_{Sim} and U_{Ext}, and outputs two registers M and Aux (which will be used for $\mathsf{AltCheck}$).*
2. *$\mathcal{E}.\mathsf{AltCheck}$ takes as input the opening in register O and the auxiliary information in register Aux, and outputs $b = 0$ or 1, indicating whether the sender gives a valid opening.*

We will write the above two procedures as $(\mathrm{M}, \mathrm{Aux}) \leftarrow \mathcal{E}.\mathsf{ExtMsg}^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}(\mathrm{C})$ and $b \leftarrow \mathcal{E}.\mathsf{AltCheck}(\mathrm{O}, \mathrm{Aux})$.

Definition 4.7 (Extractability)**.** *A quantum state commitment scheme in the quantum random oracle model $(\mathsf{Com}^{\mathsf{RO}}, \mathsf{Check}^{\mathsf{RO}})$ is $(\xi^{(1)}_{\mathsf{Ext}}, \xi^{(2)}_{\mathsf{Ext}}, \xi^{(3)}_{\mathsf{Ext}})$-*extractable *if there exist a $\xi^{(1)}_{\mathsf{Ext}}$-quantum simulator U_{Sim} with state register S for the random oracle, a stateful oracle U_{Ext} with the same state register S, and a polynomial-time quantum message extractor $\mathcal{E}$ with query access to U_{Sim} and U_{Ext} as in Definition 4.6 such that the following holds:*

1. *(U_{Ext} does not disturb the simulation.) For every integer m, $\| [U_{\mathsf{Ext}}, U_{\mathsf{Sim}}] \|^2 \leq \xi^{(2)}_{\mathsf{Ext}}(m)$, where m is the output length of the random oracle.*
2. *Two parallel U_{Sim} oracles commute, and two parallel U_{Ext} oracles commute.*
3. *($\mathcal{E}$ gives the only state that the adversary A can open to.) For every integer m, t, real numbers $w_1, w_2 \in [0,t]$ such that $w_1 + w_2 \leq t$, t-query quantum adversary A with query access to U_{Sim} and U_{Ext} such that $\mathsf{TotalMass}(A, U_{\mathsf{Sim}}) \leq w_1$ and $\mathsf{TotalMass}(A, U_{\mathsf{Ext}}) \leq w_2$, and unbounded quantum distinguisher D,*

$$\left|\sqrt{\Pr\left[\mathsf{SimWorld}(A,D)\right]} - \sqrt{\Pr\left[\mathsf{ExtWorld}(A,D)\right]}\right|^2 \leq \xi^{(3)}_{\mathsf{Ext}}(t, w_1, w_2, m) \ ,$$

where m is the output length of the random oracle, and the games $\mathsf{SimWorld}(A,D)$ and $\mathsf{ExtWorld}(A,D)$ are defined below:

- $\mathsf{SimWorld}(A, D)$*:*
  (a) The game initializes the state register: $\mathrm{S} \leftarrow |\perp\rangle$.
  (b) The adversary generates a commitment: $(\mathrm{C}, \mathrm{E}) \leftarrow A^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}$.
  (c) The adversary generates an opening: $(\mathrm{O}, \mathrm{E}) \leftarrow A^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}(\mathrm{E})$.
  (d) The game checks if the opening is valid: $(b, \mathrm{M}) \leftarrow \mathsf{Check}^{U_{\mathsf{Sim}}(\mathrm{S})}(\mathrm{C}, \mathrm{O})$.
  (e) The game generates the output: If $b = 0$*, output 0; otherwise, compute* $b' \leftarrow D(\mathrm{M}, \mathrm{E}, \mathrm{S})$ *and output* b'.
- $\mathsf{ExtWorld}(A, D)$*:*
  (a) The game initializes the state register: $\mathrm{S} \leftarrow |\perp\rangle$.
  (b) The adversary generates a commitment: $(\mathrm{C}, \mathrm{E}) \leftarrow A^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}$.
  (c) The game uses the extractor to extract the underlying message: $(\mathrm{M}, \mathrm{Aux}) \leftarrow \mathcal{E}.\mathsf{ExtMsg}^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}(\mathrm{C})$.
  (d) The adversary generates an opening: $(\mathrm{O}, \mathrm{E}) \leftarrow A^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}(\mathrm{E})$.
  (e) The game uses the alternative check to check if the opening is valid: $b \leftarrow \mathcal{E}.\mathsf{AltCheck}(\mathrm{O}, \mathrm{Aux})$.
  (f) The game generates the output: If $b = 0$*, output 0; otherwise, compute* $b' \leftarrow D(\mathrm{M}, \mathrm{E}, \mathrm{S})$ *and output* b'.

5 Construction of extractable commitments to quantum states

5.1 An extractable basic succinct commitment scheme

In this work, we will use the following basic succinct commitment in the quantum random oracle model. This is essentially an idealization of the commitment scheme in [GJMZ23], with a cryptographic hash function replaced by its idealization, a random oracle RO.

Construction 5.1. Let RO be sampled uniformly from all the functions with range $\{0,1\}^m$. We construct the *oracle-aided circuit* $C^{\mathsf{RO}}_{m,n}$ as follows, where n is the length of the quantum message $\boldsymbol{\rho}_{\mathrm{M}}$.

$C^{\mathsf{RO}}_{m,n}(\boldsymbol{\rho}_{\mathrm{MA}})$:

1. Rename the first half of the ancilla qubits as Z_1, and the second half of the ancilla qubits as Z_2.
2. Compute the hash value in the standard basis: make a query $U_{\mathsf{RO}}\colon |x\rangle|y\rangle \to |x\rangle|y\oplus \mathsf{RO}(x)\rangle$ with query register M and answer register Z_1.
3. Compute the hash value in the Hadamard basis.
   (a) Apply the Hadamard gate on each qubit of the register M.
   (b) Make a query $U_{\mathsf{RO}}\colon |x\rangle|y\rangle \to |x\rangle|y\oplus \mathsf{RO}(x)\rangle$ with query register M and answer register Z_2.
   (c) Apply the Hadamard gate on each qubit of the register M.
4. Set $\mathrm{C} \coloneqq \mathrm{Z}_1\mathrm{Z}_2$ and $\mathrm{O} \coloneqq \mathrm{M}$.

Construction 5.2 (An extractable quantum state commitment)**.** Let RO be sampled uniformly from all the functions with range $\{0,1\}^m$. $(\mathsf{Com}^{\mathsf{RO}}, \mathsf{Check}^{\mathsf{RO}})$ is a canonical quantum state commitment as defined in Definition 4.1, where the ancilla register A is a $2m$-qubit register, and the oracle-aided circuit $C^{\mathsf{RO}}_{m,n}$ is instantiated as Construction 5.1.

Perfect correctness. This candidate construction in Construction 5.2 is in the canonical form as Definition 4.1. By Lemma 4.3, this candidate construction has perfect correctness.

5.2 Extractor for the commitment scheme

Let $(\mathsf{Com}^{\mathsf{RO}}, \mathsf{Check}^{\mathsf{RO}})$ be the quantum state commitment scheme in Construction 5.2. We consider the following perfect quantum simulator for the random oracle

$$O_{\mathrm{XYD}} = \sum_{x\in X} |x\rangle\langle x|_{\mathrm{X}} \otimes F_{\mathrm{D[x]}}\mathsf{CNOT}_{\mathrm{D[x]Y}}F_{\mathrm{D[x]}} \ ,$$

which maintains some side information in register D during the simulation without disturbing any adversary's execution.

Let's construct an extractor for $(\mathsf{Com}^{\mathsf{RO}}, \mathsf{Check}^{\mathsf{RO}})$ and the quantum simulator $O(\mathrm{D})$. Our extractor will use the side information from the database register D. Let's start with several setups.

Recall that m is the output length of the random oracle and X is the set of possible query positions made by the adversary. Without loss of generality, we can assume $X = \{0,1\}^n$ where n is the length of the quantum message.

Let $\mathrm{W} \coloneqq \mathrm{BM}'$ for a single-qubit register B and an n-qubit register M'. For each $y \in \{0,1\}^m$, the purified measurement $M^{(y)}_{\mathrm{DW}}$ writes the first $x \in X$ (if any) such that the image of x in D is y into the register M'. Otherwise, it flips the flag register B. Namely,

$$M^{(y)}_{\mathrm{DW}} \coloneqq \sum_{x\in X} \Sigma^{(y)}_x \otimes \mathsf{X}^x_{\mathrm{M}'} + \Sigma^{(y)}_\perp \otimes \mathsf{X}_{\mathrm{B}} \ , \tag{4}$$

where $\{\Sigma_x^{(y)}\}_{x\in X\cup\{\perp\}}$ is the measurement on the database register D that outputs the first preimage $x \in X$ (if any) of y and outputs $\perp$ otherwise; in other words, for every $x \in X$,

$$\Sigma_x^{(y)} \coloneqq \bigotimes_{x'<x} \left(I_{\mathrm{D[x']}} - |y\rangle\langle y|_{\mathrm{D[x']}}\right) \otimes |y\rangle\langle y|_{\mathrm{D[x]}} \quad ,$$

and

$$\Sigma_\perp^{(y)} \coloneqq I_{\mathrm{D}} - \sum_{x\in X} \Sigma_x^{(y)} \quad .$$

The following lemma states that the oracle unitary O_{XYD} almost commutes with the above purified measurement $M_{\mathrm{DW}}^{(y)}$.

Lemma 5.3 (Theorem 3.1 and Lemma 3.4 in [DFMS22])**.** *For every $y \in \{0,1\}^m$ and every operator V such that it acts only on* $\mathrm{D[x]}$ *within the database register* D *and it does not act on* W*, the purified measurement $M_{\mathrm{DW}}^{(y)}$ defined in Equation* (4) *almost commutes with V, as long as $|y\rangle\langle y|_{\mathrm{D[x]}}$ almost commutes with V:*

$$\| \left[V, M_{\mathrm{DW}}^{(y)}\right] \| \leq 3\| \left[V, |y\rangle\langle y|_{\mathrm{D[x]}}\right] \| + \| \left[V, \Sigma_\perp^{(y)}\right] \| \leq 4\| \left[V, |y\rangle\langle y|_{\mathrm{D[x]}}\right] \| \quad .$$

Besides, the purified measurement $M_{\mathrm{DW}}^{(y)}$ almost commutes with the oracle unitary O_{XYD}:

$$\| \left[O_{\mathrm{XYD}}, M_{\mathrm{DW}}^{(y)}\right] \| \leq 8\sqrt{2}\cdot 2^{-m/2} \quad .$$

We can also use a value in a quantum register T as y, and do $M_{\mathrm{DW}}^{(y)}$ coherently, as the unitary Inv:

$$\mathsf{Inv}_{\mathrm{TDW}} \coloneqq \sum_{y\in\{0,1\}^m} |y\rangle\langle y|_{\mathrm{T}} \otimes M_{\mathrm{DW}}^{(y)} \quad . \tag{5}$$

Now we are ready to present the quantum message extractor $\mathcal{E}$.

Construction 5.4 (The extractor $\mathcal{E}$)**.** We consider the following $\mathcal{E}$ for $(\mathsf{Com}^{\mathsf{RO}}, \mathsf{Check}^{\mathsf{RO}})$ with query access to $O(\mathrm{D})$ and $\mathsf{Inv}(\mathrm{D})$.

$\mathcal{E}$:

1. $\mathcal{E}.\mathsf{ExtMsg}$: Upon receiving a commitment in register C, $\mathcal{E}.\mathsf{ExtMsg}$ does the following.
   (a) Parse C as $\mathrm{Z_1}$ and $\mathrm{Z_2}$.
   (b) Initialize two n-qubit registers $\mathrm{M_1}$ and $\mathrm{M_2}$ in the zero state, and three single-qubit registers B, $\mathrm{B_1}$ and $\mathrm{B_2}$ in the zero state, and set working registers $\mathrm{W_1} \coloneqq \mathrm{B_1M_1}$, $\mathrm{W_2} \coloneqq \mathrm{B_2M_2}$.
   (c) Call $\mathsf{Inv}_{\mathrm{Z_2DW_2}}$ and $O_{\mathrm{M_2Z_2D}}$.
   (d) Call $\mathsf{Inv}_{\mathrm{Z_1DW_1}}$ and $O_{\mathrm{M_1Z_1D}}$.
   (e) Apply the Hadamard gate on each qubit of the register $\mathrm{M_2}$.
   (f) Run CNOT on each qubit of the register $\mathrm{M_1}$ (as the control bit) and register $\mathrm{M_2}$ (as the target bit).
   (g) Set $\mathrm{M} \coloneqq \mathrm{M_1}$ and $\mathrm{Aux} \coloneqq \mathrm{B_1B_2M_2Z_1Z_2}$. Output the state in registers M and Aux.
2. $\mathcal{E}.\mathsf{AltCheck}$: Upon receiving an opening in register O, $\mathcal{E}.\mathsf{AltCheck}$ checks whether the state in registers O and Aux is

   $$|\psi_{\mathsf{AltCheck}}\rangle \coloneqq \frac{1}{\sqrt{|X|}} \sum_{x\in X} |0,0,0^m,0^m,x,x\rangle_{\mathrm{B_1B_2Z_1Z_2OM_2}}$$

   and outputs $b = 1$ if so. Otherwise, output $b = 0$.

5.3 Analysis of the commitment scheme

Theorem 5.5 (Extractability)**.** *Construction 5.2 is* $(\xi^{(1)}_{\mathsf{Ext}}, \xi^{(2)}_{\mathsf{Ext}}, \xi^{(3)}_{\mathsf{Ext}})$*-extractable with the quantum simulator* $O(\mathrm{D})$*, the extractor oracle* $\mathsf{Inv}(\mathrm{D})$*, and the extractor* $\mathcal{E}$ *in Construction 5.4 where*

$$\begin{aligned}\xi^{(1)}_{\mathsf{Ext}} &:= 0 \ , \\ \xi^{(2)}_{\mathsf{Ext}}(m) &:= 2^{-m+7} \ , \\ \xi^{(3)}_{\mathsf{Ext}}(t, w_1, w_2, m) &:= 2^{-m+14}(t+2)^2(w_1+w_2+3) \ .\end{aligned}$$

Define $\mathsf{CNOT}_{\mathrm{XX'}} |\perp\rangle |y\rangle = |y\rangle |\perp\rangle$ (in case that at least one of X and X′ is $|\perp\rangle$, do the swap unitary.)

We first present a lemma that is used to prove Theorem 5.5. The following lemma basically says that the extractor can extract offline the message from the commitment (after the adversary produces the opening) as long as there is no collision during the process of $\mathcal{E}.\mathsf{ExtMsg}$ and Check.

To this end, we define a *no-collision variant for an algorithm with access to* $O(\mathrm{D})$ *and* $\mathsf{Inv}(\mathrm{D})$. For every algorithm A with oracle access to $O(\mathrm{D})$ and $\mathsf{Inv}(\mathrm{D})$, let $\mathsf{A}_{\text{NoCollision}}$ denote the algorithm which has an additional query access that tells whether the database has a collision by making a projective measurement $\{\Pi_{\mathsf{NoCol}}, I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}\}$. $\mathsf{A}_{\text{NoCollision}}$ does the same thing as A except that $\mathsf{A}_{\text{NoCollision}}$ always makes a projective measurement $\{\Pi_{\mathsf{NoCol}}, I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}\}$ (by issuing a query) before and after applying each unitary, and outputs a bit $b_{\text{NoCollision}}$, indicating whether all of the measurement results say there is no collision (i.e. the result is Π_{NoCol}), in addition to the original outputs. We omit the additional query access and write $\mathsf{A}^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}_{\text{NoCollision}}$ as the additional query access is clear from the context.

Lemma 5.6. *Let* $(\mathsf{Com}^{\mathsf{RO}}, \mathsf{Check}^{\mathsf{RO}})$ *be the quantum state commitment in Construction 5.2, and* $\mathcal{E}$ *be the extractor in Construction 5.4 with oracle access to* $O(\mathrm{D})$ *and* $\mathsf{Inv}(\mathrm{D})$ *that works on an internal database register* D*.*

For every integer m*,* t*, real numbers* $w_1, w_2 \in [0, t]$ *such that* $w_1 + w_2 \leq t$*,* t*-query quantum adversary* A *such that* $\mathsf{TotalMass}(A, O) \leq w_1$ *and* $\mathsf{TotalMass}(A, \mathsf{Inv}) \leq w_2$*, and unbounded quantum distinguisher* D*,*

$$\left|\sqrt{\Pr\left[\mathsf{SimWorld}^*(A, D)\right]} - \sqrt{\Pr\left[\mathsf{OffExtWorld}^*(A, D)\right]}\right|^2 \leq (12t + 16 + 32t\sqrt{w_2})^2 \cdot 2^{-m} \ ,$$

where

- m *is the output length of the random oracle;*
- *The game* $\mathsf{SimWorld}^*(A, D)$ *is obtained by replacing the algorithm* Check *in the game* SimWorld *with the algorithm* $\mathsf{Check}_{\textit{NoCollision}}$*;*
- *The game* $\mathsf{OffExtWorld}^*(A, D)$ *is obtained by doing extraction after the adversary outputs the openings, and replacing the algorithm* $\mathcal{E}.\mathsf{ExtMsg}$ *by* $\mathcal{E}.\mathsf{ExtMsg}_{\textit{NoCollision}}$ *in the game* $\mathsf{ExtWorld}(A, D)$*.*

Specifically, the followings are the formal definitions of the games $\mathsf{SimWorld}^*(A, D)$ *and* $\mathsf{OffExtWorld}^*(A, D)$*:*

- $\mathsf{SimWorld}^*(A, D)$*:*
  1. *The game initializes the database register:* $\mathrm{D} \leftarrow |\perp\rangle$*.*
  2. *The adversary generates a commitment:* $(\mathrm{C}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}$*.*
  3. *The adversary generates an opening:* $(\mathrm{O}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}(\mathrm{E})$*.*

4. *The game checks if the opening is valid:* $(b_{\mathit{NoCollision}}, b, \mathrm{M}) \leftarrow \mathsf{Check}^{O(\mathrm{D})}_{\mathit{NoCollision}}(\mathrm{C}, \mathrm{O})$.
5. *The game generates the output: If* $b_{\mathit{NoCollision}} \wedge b = 0$*, output 0; otherwise, compute* $b' \leftarrow D(\mathrm{M}, \mathrm{E}, \mathrm{D})$ *and output* b'.

- $\mathsf{OffExtWorld}^*(A, D)$:
  1. *The game initializes the database register:* $\mathrm{D} \leftarrow |\perp\rangle$.
  2. *The adversary generates a commitment:* $(\mathrm{C}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}), \mathsf{Inv}(\mathrm{D})}$.
  3. *The adversary generates an opening:* $(\mathrm{O}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}), \mathsf{Inv}(\mathrm{D})}(\mathrm{E})$.
  4. *The game uses the extractor to extract the underlying message:* $(b_{\mathit{NoCollision}}, \mathrm{M}, \mathrm{Aux}) \leftarrow \mathcal{E}.\mathsf{ExtMsg}^{O(\mathrm{D}), \mathsf{Inv}(\mathrm{D})}_{\mathit{NoCollision}}(\mathrm{C})$.
  5. *The game uses the alternative check to check if the opening is valid:* $b \leftarrow \mathcal{E}.\mathsf{AltCheck}(\mathrm{O}, \mathrm{Aux})$.
  6. *The game generates the output: If* $b_{\mathit{NoCollision}} \wedge b = 0$*, output 0; otherwise, compute* $b' \leftarrow D(\mathrm{M}, \mathrm{E}, \mathrm{D})$ *and output* b'.

The proof of Lemma 5.6 is deferred to Section 5.4. We first show how Lemma 5.6 implies Theorem 5.5.

Proof of Theorem 5.5. We use Zhandry's compressed oracle O as the unitary U_{Sim} to do the simulation and use D as the state register S. By Theorem 4.5, O simulates the random oracle perfectly and thus $\xi^{(1)}_{\mathsf{Ext}} = 0$.

We use the preimage-finding unitary Inv as the unitary U_{Ext}. By Lemma 5.3 and Equations (2) and (5),

$$\| [U_{\mathsf{Ext}}, U_{\mathsf{Sim}}] \|^2 \leq \max_y \| \left[M^{(y)}_{\mathrm{DW}}, O_{\mathrm{XYD}} \right] \|^2 \leq (8\sqrt{2} \cdot 2^{-m/2})^2 = 2^{-m+7} \ ,$$

and thus $\xi^{(2)}_{\mathsf{Ext}}(m) = 2^{-m+7}$.

By the definition of O and Inv, two O oracles commute, and two Inv oracles commute no matter which registers they act on.

To bound $\xi^{(3)}_{\mathsf{Ext}}$, we first introduce a new game $\mathsf{OffExtWorld}$ that does the extraction offline, where the difference between $\mathsf{OffExtWorld}$ and $\mathsf{ExtWorld}$ is highlighted in blue.

$\mathsf{OffExtWorld}(A, D)$:
1. The game initializes the database register: $\mathrm{D} \leftarrow |\perp\rangle$.
2. The adversary generates a commitment: $(\mathrm{C}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}), \mathsf{Inv}(\mathrm{D})}$.
3. The adversary generates an opening: $(\mathrm{O}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}), \mathsf{Inv}(\mathrm{D})}(\mathrm{E})$.
4. The game uses the extractor to extract the underlying message: $(\mathrm{M}, \mathrm{Aux}) \leftarrow \mathcal{E}.\mathsf{ExtMsg}^{O(\mathrm{D}), \mathsf{Inv}(\mathrm{D})}(\mathrm{C})$.
5. The game uses the alternative check to check if the opening is valid: $b \leftarrow \mathcal{E}.\mathsf{AltCheck}(\mathrm{O}, \mathrm{Aux})$.
6. The game generates the output: If $b = 0$, output 0; otherwise, compute $b' \leftarrow D(\mathrm{M}, \mathrm{E}, \mathrm{D})$ and output b'.

The only difference between $\mathsf{OffExtWorld}$ and $\mathsf{ExtWorld}$ is whether the adversary first generates the opening or the game first extracts the underlying message. As a result, for every t-query quantum adversary A and unbounded quantum distinguisher D,

$$\left| \sqrt{\Pr\left[\mathsf{ExtWorld}(A, D)\right]} - \sqrt{\Pr\left[\mathsf{OffExtWorld}(A, D)\right]} \right| \leq 4t \cdot \| [U_{\mathsf{Ext}}, U_{\mathsf{Sim}}] \| \leq 32\sqrt{2} \cdot t \cdot 2^{-m/2} \ . \quad (6)$$

It remains to bound $\left| \sqrt{\Pr\left[\mathsf{SimWorld}(A, D)\right]} - \sqrt{\Pr\left[\mathsf{OffExtWorld}(A, D)\right]} \right|$.

Then the only difference between $\mathsf{SimWorld}(A, D)$ and $\mathsf{SimWorld}^*(A, D)$ is whether we check if there are collisions before and after each $O(\mathrm{D})$ query in the algorithm Check. By Lemma 2.5,

$$\left| \sqrt{\Pr\left[\mathsf{SimWorld}(A, D)\right]} - \sqrt{\Pr\left[\mathsf{SimWorld}^*(A, D)\right]} \right|$$

$$\leq 2\sqrt{6(t+2)^2(w_1+2)} \cdot 2^{-m/2} \ , \tag{7}$$

since Check makes two queries to the random oracle.

By the same reasoning,

$$\left|\sqrt{\Pr\left[\mathsf{OffExtWorld}(A,D)\right]} - \sqrt{\Pr\left[\mathsf{OffExtWorld}^*(A,D)\right]}\right| \leq 2\sqrt{6(t+2)^2(w_1+2)} \cdot 2^{-m/2} \ , \tag{8}$$

since $\mathcal{E}.\mathsf{ExtMsg}$ makes two queries to the random oracle.

Combining Equations (7) and (8) and Lemma 5.6, we can obtain

$$\begin{aligned}&\left|\sqrt{\Pr\left[\mathsf{SimWorld}(A,D)\right]} - \sqrt{\Pr\left[\mathsf{OffExtWorld}(A,D)\right]}\right| \\ &\leq 4\sqrt{6}\cdot(t+2)\sqrt{w_1+2}\cdot 2^{-m/2} + (12t+16+32t\sqrt{w_2})\cdot 2^{-m/2} \ ,\end{aligned}$$

which can be further combined with Equation (6) to get that

$$\begin{aligned}&\left|\sqrt{\Pr\left[\mathsf{SimWorld}(A,D)\right]} - \sqrt{\Pr\left[\mathsf{ExtWorld}(A,D)\right]}\right| \\ &\leq 4\sqrt{6}\cdot(t+2)\sqrt{w_1+2}\cdot 2^{-m/2} + (60t+16+32t\sqrt{w_2})\cdot 2^{-m/2} \ ,\end{aligned}$$

and therefore,

$$\begin{aligned}&\xi^{(3)}_{\mathsf{Ext}}(t,w_1,w_2,m) \\ &\leq \left(4\sqrt{6}\cdot(t+2)\sqrt{w_1+2}\cdot 2^{-m/2} + (60t+16+32t\sqrt{w_2})\cdot 2^{-m/2}\right)^2 \\ &\leq 2^{-m+2}\left(96(t+2)^2(w_1+2) + 3600t^2 + 256 + 1024t^2w_2\right) \qquad \text{(By Cauchy–Schwarz inequality)} \\ &\leq 2^{-m+14}(t+2)^2(w_1+w_2+3) \ .\end{aligned}$$

□

5.4 Proof of Lemma 5.6

We need the following claims to prove Lemma 5.6.

Let $\Pi_{\mathsf{NoHatZero}} \coloneqq \prod_x (I_{\mathrm{D[x]}} - |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}})$ be the projector that checks whether each point of the database register is set to be non-$\hat{0}^m$.

Claim 5.7. *For every query bound t,*

$$\left\|\Pi_t\left[\Pi_{\mathsf{NoCol}}, \Pi_{\mathsf{NoHatZero}}\right]\Pi_t\right\| \leq 2^{-(m-3)/2}t \ ,$$

and

$$\left\|\Pi_t\left[\mathsf{Inv}_{\mathrm{TDW}}, \Pi_{\mathsf{NoHatZero}}\right]\Pi_t\right\| \leq 2^{-m/2+4}\sqrt{t} \ .$$

We defer the proof to Section 5.5.

Claim 5.8.

$$H_{\mathrm{M}}^{\otimes n}\mathsf{CNOT}_{\mathrm{MM_2}}H_{\mathrm{M}}^{\otimes n}\mathsf{CNOT}_{\mathrm{MM_1}}|0,0\rangle_{\mathrm{M_1M_2}} = H_{\mathrm{M_2}}^{\otimes n}\mathsf{CNOT}_{\mathrm{M_1M_2}}\mathsf{SWAP}_{\mathrm{M_1M}}|\phi_{\mathrm{EPR}}\rangle_{\mathrm{M_1M_2}} \ ,$$

where $|\phi_{\mathrm{EPR}}\rangle \coloneqq \frac{1}{\sqrt{|X|}}\sum_{z\in X}|z,z\rangle$.

Proof. This equation can be shown by a direct computation. We omit the proof. □

Claim 5.9. *Let* $\mathrm{W} \coloneqq \mathrm{BM}'$. *For every subnormalized state* $|\phi\rangle$ *on registers* M, T, E *and* D,

$$\begin{aligned}&\| \langle 0|_{\mathrm{WT}} \mathsf{CNOT}_{\mathrm{MM}'} O_{\mathrm{M}'\mathrm{TD}} \mathsf{Inv}_{\mathrm{TDW}} |\phi\rangle_{\mathrm{MTED}} |0\rangle_{\mathrm{W}} - \langle 0|_{\mathrm{T}} O_{\mathrm{MTD}} |\phi\rangle_{\mathrm{MTED}} \| \\ &\leq \frac{4}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}) |\phi\rangle_{\mathrm{MTED}} \| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{MTED}} \| \ .\end{aligned}$$

Furthermore, for every subnormalized state $|\psi\rangle$ *on registers* M, E *and* D,

$$\begin{aligned}&\|\Pi_{\mathsf{NoCol}} \mathsf{Inv}_{\mathrm{TDW}} O_{\mathrm{M}'\mathrm{TD}} \mathsf{CNOT}_{\mathrm{MM}'} |\psi\rangle_{\mathrm{MED}} |0\rangle_{\mathrm{WT}} - \Pi_{\mathsf{NoCol}} O_{\mathrm{MTD}} |\psi\rangle_{\mathrm{MED}} |0\rangle_{\mathrm{WT}} \| \\ &\leq \frac{2}{\sqrt{2^m}} + \sqrt{2} \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\psi\rangle_{\mathrm{MED}} \| \ .\end{aligned}$$

Proof. Let $|\theta\rangle \coloneqq \frac{1}{\sqrt{2}}(|\perp\rangle - |\hat{0}^m\rangle)$ be a normalized quantum state. By the definition of F, we can get that for each $y \in \{0,1\}^m$, $F|y\rangle = |y\rangle + \frac{1}{\sqrt{2^{m-1}}} |\theta\rangle$.

We start with the first inequality. We write $|\phi\rangle_{\mathrm{MTED}} \coloneqq \sum_{x \in X, y \in \{0,1\}^m, e, D} \alpha_{x,y,e,D} |x,y,e,D\rangle_{\mathrm{MTED}}$. Then by the definition of Inv, we can compute $\mathsf{Inv}_{\mathrm{TDW}} |\phi\rangle_{\mathrm{MTED}} |0\rangle_{\mathrm{W}}$ according to whether $y \in \mathsf{Im}(D)$. Specifically,

$$\begin{aligned}&\mathsf{Inv}_{\mathrm{TDW}} |\phi\rangle_{\mathrm{MTED}} |0\rangle_{\mathrm{W}} \\ &= \sum_{\substack{x,e,x',y,D \\ \text{s.t. } D[x']=y \text{ and } \forall x''<x', D[x'] \neq y}} \alpha_{x,y,e,D} |x,y,e,D,0,x'\rangle_{\mathrm{MTEDBM}'} + \sum_{x,e,D,y \notin \mathsf{Im}(D)} \alpha_{x,y,e,D} |x,y,e,D,1,0\rangle_{\mathrm{MTEDBM}'} \ ,\end{aligned}$$

which implies

$$\langle 0|_{\mathrm{B}} \mathsf{Inv}_{\mathrm{TDW}} |\phi\rangle_{\mathrm{MTED}} |0\rangle_{\mathrm{W}} = \sum_{\substack{x,e,x',y,D \\ \text{s.t. } D[x']=y \text{ and } \forall x''<x', D[x''] \neq y}} \alpha_{x,y,e,D} |x,y,e,D,x'\rangle_{\mathrm{MTEDM}'} \ .$$

Moreover,

$$\begin{aligned}&\langle 0|_{\mathrm{T}} O_{\mathrm{M}'\mathrm{TD}} \sum_{\substack{x,e,x',y,D \\ \text{s.t. } D[x']=y \text{ and } \forall x''<x', D[x''] \neq y}} \alpha_{x,y,e,D} |x,y,e,D,x'\rangle_{\mathrm{MTEDM}'} \\ &= \sum_{x',x,e} |x',x,e\rangle_{\mathrm{M}'\mathrm{ME}} \sum_{\substack{y,D \text{ s.t. } D[x']=y \\ \text{and } \forall x''<x', D[x''] \neq y}} \alpha_{x,y,e,D} \langle 0|_{\mathrm{T}} F_{\mathrm{D}[\mathrm{x}']} \mathsf{CNOT}_{\mathrm{D}[\mathrm{x}']\mathrm{T}} F_{\mathrm{D}[\mathrm{x}']} |y,D\rangle_{\mathrm{TD}} \\ &= \sum_{x',x,e} |x',x,e\rangle_{\mathrm{M}'\mathrm{ME}} \sum_{\substack{y,D \text{ s.t. } D[x']=y \\ \text{and } \forall x''<x', D[x''] \neq y}} \alpha_{x,y,e,D} \langle 0|_{\mathrm{T}} F_{\mathrm{D}[\mathrm{x}']} \mathsf{CNOT}_{\mathrm{D}[\mathrm{x}']\mathrm{T}} (|y\rangle + \frac{1}{\sqrt{2^{m-1}}} |\theta\rangle)_{\mathrm{D}[\mathrm{x}']} |y, D - x'\rangle_{\mathrm{TD}[\mathrm{X}/\{\mathrm{x}'\}]} \\ &= \sum_{x',x,e} |x',x,e\rangle_{\mathrm{M}'\mathrm{ME}} \sum_{\substack{y,D \text{ s.t. } D[x']=y \\ \text{and } \forall x''<x', D[x''] \neq y}} \alpha_{x,y,e,D} F_{\mathrm{D}[\mathrm{x}']} |D\rangle_{\mathrm{D}} \\ &+ \frac{1}{\sqrt{2^{m-1}}} \langle 0|_{\mathrm{T}} \sum_{\substack{x,e,x',y,D \\ \text{s.t. } D[x']=y \text{ and } \forall x''<x', D[x''] \neq y}} F_{\mathrm{D}[\mathrm{x}']} \mathsf{CNOT}_{\mathrm{D}[\mathrm{x}']\mathrm{T}} \alpha_{x,y,e,D} |x',x,e\rangle_{\mathrm{M}'\mathrm{ME}} |\theta\rangle_{\mathrm{D}[\mathrm{x}']} |y, D - x'\rangle_{\mathrm{TD}[\mathrm{X}/\{\mathrm{x}'\}]} \ .\end{aligned}$$

Therefore,

$$\begin{aligned}
&\langle 0|_{\mathrm{M'T}} \mathsf{CNOT}_{\mathrm{MM'}} O_{\mathrm{M'TD}} \sum_{\substack{x,e,x',y,D \\ \text{s.t. } D[x']=y \text{ and } \forall x''<x', D[x'']\neq y}} \alpha_{x,y,e,D} |x,y,e,D,x'\rangle_{\mathrm{MTEDM'}} \\
&= \sum_{x,e} |x,e\rangle_{\mathrm{ME}} \sum_{\substack{y,D \text{ s.t. } D[x]=y \\ \text{and } \forall x''<x, D[x'']\neq y}} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} |D\rangle_{\mathrm{D}} \\
&+ \frac{1}{\sqrt{2^{m-1}}} \langle 0|_{\mathrm{T}} \sum_{\substack{x,e,y,D \\ \text{s.t. } D[x]=y \text{ and } \forall x''<x, D[x'']\neq y}} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} |x,e\rangle_{\mathrm{ME}} |\theta\rangle_{\mathrm{D[x]}} |y, D-x\rangle_{\mathrm{TD[X/\{x\}]}} \quad ,
\end{aligned}$$

which implies

$$\begin{aligned}
&\| \langle 0|_{\mathrm{WT}} \mathsf{CNOT}_{\mathrm{MM'}} O_{\mathrm{M'TD}} \mathsf{Inv}_{\mathrm{TDW}} |\phi\rangle_{\mathrm{MTED}} |0\rangle_{\mathrm{W}} - \sum_{\substack{x,e,y,D \text{ s.t. } D[x]=y \\ \text{and } \forall x''<x, D[x'']\neq y}} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} |x,e,D\rangle_{\mathrm{MED}} \| \\
&= \frac{1}{\sqrt{2^{m-1}}} \| \langle 0|_{\mathrm{T}} \sum_{\substack{x,e,y,D \\ \text{s.t. } D[x]=y \text{ and } \forall x''<x, D[x'']\neq y}} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} |x,e\rangle_{\mathrm{ME}} |\theta\rangle_{\mathrm{D[x]}} |y, D-x\rangle_{\mathrm{TD[X/\{x\}]}} \| \\
&\leq \frac{1}{\sqrt{2^{m-1}}} \| \sum_{\substack{x,e,y,D \\ \text{s.t. } D[x]=y \text{ and } \forall x''<x, D[x'']\neq y}} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} |x,e\rangle_{\mathrm{ME}} |\theta\rangle_{\mathrm{D[x]}} |y, D-x\rangle_{\mathrm{TD[X/\{x\}]}} \| \\
&= \frac{1}{\sqrt{2^{m-1}}} \| \left(\sum_{x} |x\rangle\langle x|_{\mathrm{M}} \otimes F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} \right) \sum_{\substack{x,e,y,D \\ \text{s.t. } D[x]=y \\ \wedge \forall x''<x, D[x'']\neq y}} \alpha_{x,y,e,D} |x,e\rangle_{\mathrm{ME}} |\theta\rangle_{\mathrm{D[x]}} |y, D-x\rangle_{\mathrm{TD[X/\{x\}]}} \| \\
&= \frac{1}{\sqrt{2^{m-1}}} \| \sum_{\substack{x,e,y,D \\ \text{s.t. } D[x]=y \\ \wedge \forall x''<x, D[x'']\neq y}} \alpha_{x,y,e,D} |x,e,y,D-x\rangle_{\mathrm{METD[X/\{x\}]}} \| \\
&= \frac{1}{\sqrt{2^{m-1}}} \sqrt{\sum_{\substack{x,e,y,D \\ \text{s.t. } D[x]=y \\ \wedge \forall x''<x, D[x'']\neq y}} |\alpha_{x,y,e,D}|^2} \\
&\leq \frac{1}{\sqrt{2^{m-1}}} \quad . \quad \text{(By normalization, we have that } \textstyle\sum_{x,y,e,D} |\alpha_{x,y,e,D}|^2 \leq 1 \text{ for the subnormalized state } |\phi\rangle.)
\end{aligned}$$

On the other hand,

$$\begin{aligned}
&\langle 0|_{\mathrm{T}} O_{\mathrm{MTD}} |\phi\rangle_{\mathrm{MTED}} \\
&= \sum_{x,e} |x,e\rangle_{\mathrm{ME}} \sum_{y,D} \alpha_{x,y,e,D} \langle 0|_{\mathrm{T}} F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} F_{\mathrm{D[x]}} |y,D\rangle_{\mathrm{TD}} \\
&= \sum_{x,e} |x,e\rangle_{\mathrm{ME}} \sum_{y,D \text{ s.t. } D(x)=\perp} \alpha_{x,y,e,D} \langle 0|_{\mathrm{T}} F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} F_{\mathrm{D[x]}} |y,D\rangle_{\mathrm{TD}} \\
&+ \sum_{x,e} |x,e\rangle_{\mathrm{ME}} \sum_{y,D \text{ s.t. } D(x)=y} \alpha_{x,y,e,D} \langle 0|_{\mathrm{T}} F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} F_{\mathrm{D[x]}} |y,D\rangle_{\mathrm{TD}}
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{x,e} |x,e\rangle_{\mathrm{ME}} \sum_{y,D \text{ s.t. } D(x)\neq\perp \text{ and } D(x)\neq y} \alpha_{x,y,e,D} \langle 0|_{\mathrm{T}} F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} F_{\mathrm{D[x]}} |y,D\rangle_{\mathrm{TD}} \\
&= \frac{1}{\sqrt{2^m}} \sum_{x,e} |x,e\rangle_{\mathrm{ME}} \sum_{y,D \text{ s.t. } D(x)=\perp} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} |y\rangle_{\mathrm{D[x]}} |D\rangle_{\mathrm{D[X/\{x\}]}} \\
&+ \sum_{x,e} |x,e\rangle_{\mathrm{ME}} \sum_{y,D \text{ s.t. } D(x)=y} \alpha_{x,y,e,D} \left(F_{\mathrm{D[x]}} |D\rangle_{\mathrm{D}} + \frac{1}{\sqrt{2^{m-1}}} \langle 0|_{\mathrm{T}} F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} |y\rangle_{\mathrm{T}} |\theta\rangle_{\mathrm{D[x]}} |D-x\rangle_{\mathrm{D[X/\{x\}]}} \right) \\
&+ \frac{1}{\sqrt{2^{m-1}}} \sum_{x,e} |x,e\rangle_{\mathrm{ME}} \sum_{y,D \text{ s.t. } D(x)\neq\perp \text{ and } D(x)\neq y} \alpha_{x,y,e,D} \langle 0|_{\mathrm{T}} F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} |y\rangle_{\mathrm{T}} |\theta\rangle_{\mathrm{D[x]}} |D-x\rangle_{\mathrm{D[X/\{x\}]}} \quad ,
\end{aligned}
$$

which implies

$$
\begin{aligned}
&\| \langle 0|_{\mathrm{T}} O_{\mathrm{MTD}} |\phi\rangle_{\mathrm{MTED}} - \sum_{x,e,y,D \text{ s.t. } D[x]=y} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} |x,e,D\rangle_{\mathrm{MED}} \| \\
&\leq \frac{1}{\sqrt{2^m}} \| \sum_{x,e} |x,e\rangle_{\mathrm{ME}} \sum_{y,D \text{ s.t. } D(x)=\perp} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} |y\rangle_{\mathrm{D[x]}} |D\rangle_{\mathrm{D[X/\{x\}]}} \| \\
&+ \frac{1}{\sqrt{2^{m-1}}} \| \sum_{x,e} |x,e\rangle_{\mathrm{ME}} \sum_{y,D \text{ s.t. } D(x)\neq\perp} \alpha_{x,y,e,D} \langle 0|_{\mathrm{T}} F_{\mathrm{D[x]}} \mathsf{CNOT}_{\mathrm{D[x]T}} |y\rangle_{\mathrm{T}} |\theta\rangle_{\mathrm{D[x]}} |D-x\rangle_{\mathrm{D[X/\{x\}]}} \| \\
&\leq \frac{1}{\sqrt{2^m}} + \frac{1}{\sqrt{2^{m-1}}} \| \frac{1}{\sqrt{2^{m+1}}} \sum_{x,e,y,D \text{ s.t. } D(x)\neq\perp} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} |x,e\rangle_{\mathrm{ME}} |y\rangle_{\mathrm{D[x]}} |D-x\rangle_{\mathrm{D[X/\{x\}]}} \| \\
&+ \frac{1}{\sqrt{2^m}} \| \sum_{x,e,D \text{ s.t. } D(x)\neq\perp} \alpha_{x,0,e,D} F_{\mathrm{D[x]}} |x,e\rangle_{\mathrm{ME}} |\perp\rangle_{\mathrm{D[x]}} |D\rangle_{\mathrm{D[X/\{x\}]}} \| \\
&= \frac{1}{\sqrt{2^m}} + \frac{1}{2^m} \| \sum_{x,e,y,D \text{ s.t. } D(x)\neq\perp} \alpha_{x,y,e,D} |x,e\rangle_{\mathrm{ME}} |y\rangle_{\mathrm{D[x]}} |D-x\rangle_{\mathrm{D[X/\{x\}]}} \| \\
&+ \frac{1}{\sqrt{2^m}} \| \sum_{x,e,D \text{ s.t. } D(x)\neq\perp} \alpha_{x,0,e,D} |x,e\rangle_{\mathrm{ME}} |D-x\rangle_{\mathrm{D}} \| \\
&= \frac{1}{\sqrt{2^m}} + \frac{1}{2^m} \sqrt{\sum_{x,e,y,D \text{ s.t. } D(x)=\perp} \left| \sum_z \alpha_{x,y,e,D+[x\mapsto z]} \right|^2} \\
&+ \| \left(\sum_x |x\rangle\langle x|_{\mathrm{M}} \otimes |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \right) \sum_{x,e,D} \alpha_{x,0,e,D} |x,0,e,D\rangle_{\mathrm{MTED}} \| \\
&\leq \frac{1}{\sqrt{2^m}} + \frac{1}{2^m} \sqrt{\sum_{x,e,y,D \text{ s.t. } D(x)=\perp} 2^m \cdot \sum_z \left| \alpha_{x,y,e,D+[x\mapsto z]} \right|^2} + \| (I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{MTED}} \| \\
&\leq \frac{2}{\sqrt{2^m}} + \| (I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{MTED}} \| \quad .
\end{aligned}
$$

Therefore, we have that

$$
\begin{aligned}
&\| \langle 0|_{\mathrm{WT}} \mathsf{CNOT}_{\mathrm{MM'}} O_{\mathrm{M'TD}} \mathsf{Inv}_{\mathrm{TDW}} |\phi\rangle_{\mathrm{MTED}} |0\rangle_{\mathrm{W}} - \langle 0|_{\mathrm{T}} O_{\mathrm{MTD}} |\phi\rangle_{\mathrm{MTED}} \| \\
&\leq \frac{4}{\sqrt{2^m}} + \| (I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{MTED}} \|
\end{aligned}
$$

$$
\begin{aligned}
&+ \| \sum_{x,e,y,D \text{ s.t. } D[x]=y} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} \left|x,e,D\right\rangle_{\mathrm{MED}} - \sum_{\substack{x,e,y,D \text{ s.t. } D[x]=y \\ \text{and } \forall x''<x, D[x'']\neq y}} \alpha_{x,y,e,D} F_{\mathrm{D[x]}} \left|x,e,D\right\rangle_{\mathrm{MED}} \| \\
&= \frac{4}{\sqrt{2^m}} + \| (I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) \left|\phi\right\rangle_{\mathrm{MTED}} \| + \| \sum_{\substack{x,e,y,D \text{ s.t. } D[x]=y \\ \text{and } \exists x''<x, D[x'']=y}} \alpha_{x,y,e,D} \left|x,e,D\right\rangle_{\mathrm{MED}} \| \\
&= \frac{4}{\sqrt{2^m}} + \| (I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) \left|\phi\right\rangle_{\mathrm{MTED}} \| + \sqrt{\sum_{\substack{x,e,y,D \text{ s.t. } D[x]=y \\ \text{and } \exists x''<x, D[x'']=y}} \left|\alpha_{x,y,e,D}\right|^2} \ .
\end{aligned}
$$

Recall that $\left|\phi\right\rangle_{\mathrm{MTED}} \coloneqq \sum_{x\in X, y\in\{0,1\}^m, e, D} \alpha_{x,y,e,D} \left|x,y,e,D\right\rangle_{\mathrm{MTED}}$. We can get that

$$
\| (I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}) \left|\phi\right\rangle_{\mathrm{MTED}} \|^2 = \sum_{x,y,e,D\notin S_{\mathsf{NoCollision}}} \left|\alpha_{x,y,e,D}\right|^2 \ .
$$

Thus by the non-negativity of each $|\alpha_{x,y,e,D}|$,

$$
\begin{aligned}
&\| \left\langle 0\right|_{\mathrm{WT}} \mathsf{CNOT}_{\mathrm{MM'}} O_{\mathrm{M'TD}} \mathsf{Inv}_{\mathrm{TDW}} \left|\phi\right\rangle_{\mathrm{MTED}} \left|0\right\rangle_{\mathrm{W}} - \left\langle 0\right|_{\mathrm{T}} O_{\mathrm{MTD}} \left|\phi\right\rangle_{\mathrm{MTED}} \| \\
&\leq \frac{4}{\sqrt{2^m}} + \| (I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}) \left|\phi\right\rangle_{\mathrm{MTED}} \| + \| (I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) \left|\phi\right\rangle_{\mathrm{MTED}} \| \ .
\end{aligned}
$$

We now prove the second inequality. Define

$$
\begin{aligned}
\left|\psi_0\right\rangle &\coloneqq \Pi_{\mathsf{NoCol}} O_{\mathrm{MTD}} \left|\psi\right\rangle_{\mathrm{MED}} \left|0\right\rangle_{\mathrm{WT}} \ , \\
\left|\psi_1\right\rangle &\coloneqq \Pi_{\mathsf{NoCol}} \mathsf{Inv}_{\mathrm{TDW}} O_{\mathrm{M'TD}} \mathsf{CNOT}_{\mathrm{MM'}} \left|\psi\right\rangle_{\mathrm{MED}} \left|0\right\rangle_{\mathrm{WT}} \\
&= \Pi_{\mathsf{NoCol}} \mathsf{Inv}_{\mathrm{TDW}} \mathsf{CNOT}_{\mathrm{MM'}} O_{\mathrm{MTD}} \left|\psi\right\rangle_{\mathrm{MED}} \left|0\right\rangle_{\mathrm{WT}} \\
&= \mathsf{Inv}_{\mathrm{TDW}} \mathsf{CNOT}_{\mathrm{MM'}} \left|\psi_0\right\rangle \ ,
\end{aligned}
$$

where we use that copying the value in M and querying M' is equivalent to querying M and then performing the copy because M' is initialized as all-zero states.

We show that to bound the left hand side of the second inequality, which equals $\| \left|\psi_0\right\rangle - \left|\psi_1\right\rangle \|$, it suffices to bound the error $\|\Pi_{\neq} \left|\psi_0\right\rangle \|$, where $\Pi_{\neq} \coloneqq \sum_{x,y,D \text{ s.t. } D[x]\neq y} \left|x,y,D\right\rangle\!\left\langle x,y,D\right|_{\mathrm{MTD}}$ projects to the bad branches such that $D[x] \neq y$ after making the query O_{MTD} where T is initialized as all-zero states.

This is because, for $\left|\psi_0\right\rangle \coloneqq \sum_{x,y,e,D\in S_{\mathsf{NoCollision}}} \alpha_{x,y,e,D} \left|x,y,e,D\right\rangle_{\mathrm{MTED}} \left|0\right\rangle_{\mathrm{W}}$,

$$
\left|\psi_1\right\rangle = \mathsf{Inv}_{\mathrm{TDW}} \mathsf{CNOT}_{\mathrm{MM'}} \left|\psi_0\right\rangle = \sum_{x,y,e,D\in S_{\mathsf{NoCollision}}} \alpha_{x,y,e,D} \left|x,y,e,D\right\rangle_{\mathrm{MTED}} \left|g(x,y,D)\right\rangle_{\mathrm{W}} \ ,
$$

for a classical function g such that $g(x,y,D) = 0$ if and only if $D[x] = y$. Thus

$$
\begin{aligned}
\| \left|\psi_1\right\rangle - \left|\psi_0\right\rangle \| &= \| \sum_{x,y,e,D\in S_{\mathsf{NoCollision}}} \alpha_{x,y,e,D} \left|x,y,e,D\right\rangle_{\mathrm{MTED}} (\left|g(x,y,D)\right\rangle_{\mathrm{W}} - \left|0\right\rangle_{\mathrm{W}}) \| \\
&= \sqrt{2} \| \sum_{x,y,e,D\in S_{\mathsf{NoCollision}} \text{ s.t. } g(x,y,D)\neq 0} \alpha_{x,y,e,D} \left|x,y,e,D\right\rangle_{\mathrm{MTED}} \| \\
&= \sqrt{2} \|\Pi_{\neq} \left|\psi_0\right\rangle \| \ .
\end{aligned}
$$

Intuitively, the error $\|\Pi_{\neq}\left|\psi_0\right\rangle\|$ should be small because T is initialized as 0, and after making a query, it should store $D[x]$. This can be shown by direct calculation.

We decompose the input as $\left|\psi\right\rangle = \Pi_{\mathsf{NoHatZero}}\left|\psi\right\rangle + (I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\left|\psi\right\rangle$. Then

$$
\begin{aligned}
&\|\Pi_{\neq}\left|\psi_0\right\rangle\| \\
&= \|\Pi_{\neq}\Pi_{\mathsf{NoCol}}O_{\mathrm{MTD}}\left|\psi\right\rangle_{\mathrm{MED}}\left|0\right\rangle_{\mathrm{WT}}\| \\
&\leq \|\Pi_{\neq}O_{\mathrm{MTD}}\left|\psi\right\rangle_{\mathrm{MED}}\left|0\right\rangle_{\mathrm{T}}\| \\
&\leq \|\Pi_{\neq}O_{\mathrm{MTD}}\Pi_{\mathsf{NoHatZero}}\left|\psi\right\rangle_{\mathrm{MED}}\left|0\right\rangle_{\mathrm{T}}\| + \|\Pi_{\neq}O_{\mathrm{MTD}}(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\left|\psi\right\rangle_{\mathrm{MED}}\left|0\right\rangle_{\mathrm{T}}\| \\
&\leq \|\Pi_{\neq}F_{\mathrm{D}}\sum_x \left|x\right\rangle\!\left\langle x\right|_{\mathrm{M}} \otimes \mathsf{CNOT}_{\mathrm{D[x]T}}F_{\mathrm{D}}\Pi_{\mathsf{NoHatZero}}\left|\psi\right\rangle_{\mathrm{MED}}\left|0\right\rangle_{\mathrm{T}}\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\left|\psi\right\rangle_{\mathrm{MED}}\| \ .
\end{aligned}
$$

We write the state $F_{\mathrm{D}}\Pi_{\mathsf{NoHatZero}}\left|\psi\right\rangle_{\mathrm{MED}} = \sum_{x,e,f}\alpha_{x,e,f}\left|x,e,f\right\rangle_{\mathrm{MED}}$. Here f does not contain $\perp$ on any location since the state before applying F_{D} is supported over $\Pi_{\mathsf{NoHatZero}}$. We bound the first term in the above inequality:

$$
\begin{aligned}
&\|\Pi_{\neq}F_{\mathrm{D}}\sum_x \left|x\right\rangle\!\left\langle x\right|_{\mathrm{M}} \otimes \mathsf{CNOT}_{\mathrm{D[x]T}}F_{\mathrm{D}}\Pi_{\mathsf{NoHatZero}}\left|\psi\right\rangle_{\mathrm{MED}}\left|0\right\rangle_{\mathrm{T}}\| \\
&= \|\Pi_{\neq}F_{\mathrm{D}}\sum_{x,e,f}\alpha_{x,e,f}\left|x,e,f\right\rangle_{\mathrm{MED}}\left|f[x]\right\rangle_{\mathrm{T}}\| \\
&= \|\Pi_{\neq}\sum_{x,e,f}\alpha_{x,e,f}F_{\mathrm{D[x]}}\left|x,e,f\right\rangle_{\mathrm{MED}}\left|f[x]\right\rangle_{\mathrm{T}}\| \\
&= \|\Pi_{\neq}\sum_{x,e,f}\alpha_{x,e,f}\left|x,e,f-x\right\rangle_{\mathrm{MED[X/\{x\}]}}(\left|f(x)\right\rangle + \frac{1}{\sqrt{2^{m-1}}}\left|\theta\right\rangle)_{\mathrm{D[x]}}\left|f[x]\right\rangle_{\mathrm{T}}\| \\
&= \frac{1}{\sqrt{2^{m-1}}}\|\Pi_{\neq}\sum_{x,e,f}\alpha_{x,e,f}\left|x,e,f-x\right\rangle_{\mathrm{MED[X/\{x\}]}}\left|\theta\right\rangle_{\mathrm{D[x]}}\left|f[x]\right\rangle_{\mathrm{T}}\| \\
&\leq \frac{1}{\sqrt{2^{m-1}}} \ .
\end{aligned}
$$

The second inequality of this claim follows from the above equations. □

Claim 5.10. *Define* $V_{\mathsf{Check}} \coloneqq \Pi_{\mathsf{NoCol}}O_{\mathrm{MZ_1D}}\Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}O_{\mathrm{MZ_2D}}\Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}$ *to be the operation that* $\mathsf{Check}_{\textit{NoCollision}}$ *does, and define* $\left|\psi_{\mathsf{Check}}\right\rangle \coloneqq \left|0^{2m}\right\rangle_{\mathrm{Z_1Z_2}}$ *to be the state that* $\mathsf{Check}_{\textit{NoCollision}}$ *projects onto to see if the opening is valid.*

Define $V_{\mathsf{ExtMsg}} \coloneqq \mathsf{SWAP}_{\mathrm{M_1M}}\mathsf{CNOT}_{\mathrm{M_1M_2}}H_{\mathrm{M_2}}^{\otimes n}\Pi_{\mathsf{NoCol}}O_{\mathrm{M_1Z_1D}}\Pi_{\mathsf{NoCol}}\mathsf{Inv}_{\mathrm{Z_1DW_1}}O_{\mathrm{M_2Z_2D}}\Pi_{\mathsf{NoCol}}\mathsf{Inv}_{\mathrm{Z_2DW_2}}$ *to be the operation that* $\mathcal{E}.\mathsf{ExtMsg}_{\textit{NoCollision}}$ *does (*$\mathsf{SWAP}_{\mathrm{M_1M}}$ *is just for renaming the registers), and define* $\left|\psi_{\mathsf{AltCheck}}\right\rangle \coloneqq \frac{1}{\sqrt{|X|}}\sum_{x\in X}\left|0,0,0^m,0^m,x,x\right\rangle_{\mathrm{B_1B_2Z_1Z_2M_1M_2}}$ *to be the state that* $\mathcal{E}.\mathsf{ExtMsg}_{\textit{NoCollision}}$ *projects onto to see if the opening is valid.*

Then for every subnormalized state $\left|\phi\right\rangle_{\mathrm{CODE}}$,

$$
\begin{aligned}
&\|(\left\langle\psi_{\mathsf{Check}}\right| V_{\mathsf{Check}} - \left\langle\psi_{\mathsf{AltCheck}}\right| V_{\mathsf{ExtMsg}}\left|0\right\rangle_{\mathrm{W_1W_2}})\left|\phi\right\rangle_{\mathrm{CODE}}\| \\
&\leq \frac{8}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\left|\phi_1\right\rangle\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\left|\phi_2\right\rangle\| \ ,
\end{aligned}
$$

where $\left|\phi_1\right\rangle \coloneqq \Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}O_{\mathrm{MZ_2D}}\Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}\left|\phi\right\rangle_{\mathrm{CODE}}$ *and* $\left|\phi_2\right\rangle \coloneqq \Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}\left|\phi\right\rangle_{\mathrm{CODE}}$.

Furthermore, for every subnormalized state $|\psi\rangle_{\mathrm{MED}}$,

$$\begin{aligned}&\|(V_{\mathsf{Check}}^{\dagger}|\psi_{\mathsf{Check}}\rangle|0\rangle_{\mathrm{W_1W_2}} - V_{\mathsf{ExtMsg}}^{\dagger}|\psi_{\mathsf{AltCheck}}\rangle)|\psi\rangle_{\mathrm{MED}}\|\\&\leq \frac{4}{\sqrt{2^m}} + \sqrt{2}\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})|\psi_1\rangle\| + \sqrt{2}\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})|\psi_2\rangle\| \ ,\end{aligned}$$

where $|\psi_1\rangle \coloneqq H_{\mathrm{M}}^{\otimes n}\Pi_{\mathsf{NoCol}}O_{\mathrm{MZ_1D}}\Pi_{\mathsf{NoCol}}|\psi\rangle_{\mathrm{MED}}|\psi_{\mathsf{Check}}\rangle$ *and* $|\psi_2\rangle \coloneqq \Pi_{\mathsf{NoCol}}|\psi\rangle_{\mathrm{MED}}$.

Proof. For the first inequality, invoking Claim 5.9 on the subnormalized state $|\phi_1\rangle$, we obtain that

$$\begin{aligned}&\|\langle 0|_{\mathrm{W_1Z_1}}\mathsf{CNOT}_{\mathrm{MM_1}}O_{\mathrm{M_1Z_1D}}\mathsf{Inv}_{\mathrm{Z_1DW_1}}|\phi_1\rangle|0\rangle_{\mathrm{W_1}} - \langle 0|_{\mathrm{Z_1}}O_{\mathrm{MZ_1D}}|\phi_1\rangle\|\\&\leq \frac{4}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}})|\phi_1\rangle\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})|\phi_1\rangle\|\\&= \frac{4}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})|\phi_1\rangle\| \ ,\end{aligned} \tag{9}$$

where $\mathrm{W_1} \coloneqq (\mathrm{M_1}, \mathrm{B_1})$, $\mathrm{M} \coloneqq \mathrm{O}$, and $(\mathrm{Z_1}, \mathrm{Z_2}) \coloneqq \mathrm{C}$.

Notice that by the definition of V_{Check} and $|\psi_{\mathsf{Check}}\rangle$,

$$\langle\psi_{\mathsf{Check}}|V_{\mathsf{Check}}|\phi\rangle_{\mathrm{CODE}} = \langle 0|_{\mathrm{Z_2}}\langle 0|_{\mathrm{Z_1}}\Pi_{\mathsf{NoCol}}O_{\mathrm{MZ_1D}}|\phi_1\rangle \ .$$

By Equation (9) and the triangle inequality,

$$\begin{aligned}&\|\langle 0|_{\mathrm{W_1Z_1Z_2}}\Pi_{\mathsf{NoCol}}\mathsf{CNOT}_{\mathrm{MM_1}}O_{\mathrm{M_1Z_1D}}\mathsf{Inv}_{\mathrm{Z_1DW_1}}|\phi_1\rangle|0\rangle_{\mathrm{W_1}} - \langle\psi_{\mathsf{Check}}|V_{\mathsf{Check}}|\phi\rangle_{\mathrm{CODE}}\|\\&\leq \frac{4}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})|\phi_1\rangle\| \ .\end{aligned} \tag{10}$$

Again invoking Claim 5.9 on the subnormalized state $|\phi_2\rangle$, we obtain that

$$\begin{aligned}&\|\langle 0|_{\mathrm{W_2Z_2}}\mathsf{CNOT}_{\mathrm{MM_2}}O_{\mathrm{M_2Z_2D}}\mathsf{Inv}_{\mathrm{Z_2DW_2}}|\phi_2\rangle|0\rangle_{\mathrm{W_2}} - \langle 0|_{\mathrm{Z_2}}O_{\mathrm{MZ_2D}}|\phi_2\rangle\|\\&\leq \frac{4}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}})|\phi_2\rangle\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})|\phi_2\rangle\|\\&\leq \frac{4}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})|\phi_2\rangle\| \ ,\end{aligned} \tag{11}$$

where $\mathrm{W_2} \coloneqq (\mathrm{M_2}, \mathrm{B_2})$.

Notice that $|\phi_1\rangle = \Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}O_{\mathrm{MZ_2D}}|\phi_2\rangle$. By Equation (11) and the triangle inequality,

$$\begin{aligned}&\|\langle 0|_{\mathrm{W_2Z_2}}\Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}\mathsf{CNOT}_{\mathrm{MM_2}}O_{\mathrm{M_2Z_2D}}\mathsf{Inv}_{\mathrm{Z_2DW_2}}|\phi_2\rangle|0\rangle_{\mathrm{W_2}} - \langle 0|_{\mathrm{Z_2}}|\phi_1\rangle\|\\&\leq \frac{4}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})|\phi_2\rangle\| \ .\end{aligned} \tag{12}$$

Equations (10) and (12) imply that

$$\begin{aligned}&\|\langle\psi_{\mathsf{Check}}|V_{\mathsf{Check}}|\phi\rangle_{\mathrm{CODE}} - \langle 0|_{\mathrm{W_1W_2Z_1Z_2}}V'_{\mathsf{Check}}|\phi\rangle_{\mathrm{CODE}}|0\rangle_{\mathrm{W_1W_2}}\|\\&\leq \frac{8}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})|\phi_1\rangle\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})|\phi_2\rangle\| \ ,\end{aligned}$$

where $V'_{\mathsf{Check}} \coloneqq \Pi_{\mathsf{NoCol}}\mathsf{CNOT}_{\mathrm{MM}_1}O_{\mathrm{M}_1\mathrm{Z}_1\mathrm{D}}\mathsf{Inv}_{\mathrm{Z}_1\mathrm{D}\mathrm{W}_1}\Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}\mathsf{CNOT}_{\mathrm{MM}_2}O_{\mathrm{M}_2\mathrm{Z}_2\mathrm{D}}\mathsf{Inv}_{\mathrm{Z}_2\mathrm{D}\mathrm{W}_2}\Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}$.

It remains to show that $\langle 0|_{\mathrm{W}_1\mathrm{W}_2\mathrm{Z}_1\mathrm{Z}_2} V'_{\mathsf{Check}} |0\rangle_{\mathrm{W}_1\mathrm{W}_2} = \langle\psi_{\mathsf{AltCheck}}| V_{\mathsf{ExtMsg}} |0\rangle_{\mathrm{W}_1\mathrm{W}_2}$.

We use Claim 5.8 to prove the above equation. Let $\mathrm{W} \coloneqq (\mathrm{W}_1, \mathrm{W}_2)$.

$$
\begin{aligned}
&\langle 0|_{\mathrm{W}_1\mathrm{W}_2\mathrm{Z}_1\mathrm{Z}_2} V'_{\mathsf{Check}} |0\rangle_{\mathrm{W}} \\
&= \langle 0|_{\mathrm{W}_1\mathrm{W}_2\mathrm{Z}_1\mathrm{Z}_2} \Pi_{\mathsf{NoCol}}\mathsf{CNOT}_{\mathrm{MM}_1}O_{\mathrm{M}_1\mathrm{Z}_1\mathrm{D}}\mathsf{Inv}_{\mathrm{Z}_1\mathrm{D}\mathrm{W}_1}\Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}\mathsf{CNOT}_{\mathrm{MM}_2}O_{\mathrm{M}_2\mathrm{Z}_2\mathrm{D}}\mathsf{Inv}_{\mathrm{Z}_2\mathrm{D}\mathrm{W}_2}\Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n} |0\rangle_{\mathrm{W}} \\
&= \langle 0|_{\mathrm{W}_1\mathrm{W}_2\mathrm{Z}_1\mathrm{Z}_2} \Pi_{\mathsf{NoCol}}\mathsf{CNOT}_{\mathrm{MM}_1}H_{\mathrm{M}}^{\otimes n}\mathsf{CNOT}_{\mathrm{MM}_2}H_{\mathrm{M}}^{\otimes n}O_{\mathrm{M}_1\mathrm{Z}_1\mathrm{D}}\mathsf{Inv}_{\mathrm{Z}_1\mathrm{D}\mathrm{W}_1}\Pi_{\mathsf{NoCol}}O_{\mathrm{M}_2\mathrm{Z}_2\mathrm{D}}\mathsf{Inv}_{\mathrm{Z}_2\mathrm{D}\mathrm{W}_2}\Pi_{\mathsf{NoCol}} |0\rangle_{\mathrm{W}} \\
&= \langle\phi_{\mathrm{EPR}}|_{\mathrm{M}_1\mathrm{M}_2} \langle 0|_{\mathrm{B}_1\mathrm{B}_2\mathrm{Z}_1\mathrm{Z}_2} \mathsf{SWAP}_{\mathrm{M}_1\mathrm{M}}\mathsf{CNOT}_{\mathrm{M}_1\mathrm{M}_2}H_{\mathrm{M}_2}^{\otimes n}\Pi_{\mathsf{NoCol}}O_{\mathrm{M}_1\mathrm{Z}_1\mathrm{D}}\Pi_{\mathsf{NoCol}}\mathsf{Inv}_{\mathrm{Z}_1\mathrm{D}\mathrm{W}_1}O_{\mathrm{M}_2\mathrm{Z}_2\mathrm{D}}\Pi_{\mathsf{NoCol}}\mathsf{Inv}_{\mathrm{Z}_2\mathrm{D}\mathrm{W}_2} |0\rangle_{\mathrm{W}} \\
&= \langle\psi_{\mathsf{AltCheck}}| V_{\mathsf{ExtMsg}} |0\rangle_{\mathrm{W}} \quad , \qquad \text{(Definitions of } |\psi_{\mathsf{AltCheck}}\rangle \text{ and } V_{\mathsf{ExtMsg}}\text{)}
\end{aligned}
$$

where we use the commutativity of unitaries on different registers, and the commutativity of Π_{NoCol} and Inv.

The second inequality follows similarly by invoking Claim 5.9 on the states $|\psi_1\rangle$ and $|\psi_2\rangle$. □

Now we are ready to prove Lemma 5.6.

Proof of Lemma 5.6. By the triangle inequality,

$$
\begin{aligned}
&\left|\sqrt{\Pr\left[\mathsf{SimWorld}^*(A,D)\right]} - \sqrt{\Pr\left[\mathsf{OffExtWorld}^*(A,D)\right]}\right|^2 \\
&\leq \|(\langle\psi_{\mathsf{Check}}| V_{\mathsf{Check}} - \langle\psi_{\mathsf{AltCheck}}| V_{\mathsf{ExtMsg}} |0\rangle_{\mathrm{W}_1\mathrm{W}_2}) |\phi\rangle_{\mathrm{CODE}}\|^2 \quad , \qquad (13)
\end{aligned}
$$

where $|\phi\rangle_{\mathrm{CODE}}$ is the joint state after the adversary provides the opening.

By Claim 5.10,

$$
\begin{aligned}
&\|(\langle\psi_{\mathsf{Check}}| V_{\mathsf{Check}} - \langle\psi_{\mathsf{AltCheck}}| V_{\mathsf{ExtMsg}} |0\rangle_{\mathrm{W}_1\mathrm{W}_2}) |\phi\rangle_{\mathrm{CODE}}\| \\
&\leq \frac{8}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi_1\rangle\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi_2\rangle\| \quad ,
\end{aligned}
$$

where $|\phi_1\rangle \coloneqq \Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n}O_{\mathrm{MZ}_2\mathrm{D}}\Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n} |\phi\rangle_{\mathrm{CODE}}$ and $|\phi_2\rangle \coloneqq \Pi_{\mathsf{NoCol}}H_{\mathrm{M}}^{\otimes n} |\phi\rangle_{\mathrm{CODE}}$.

We bound the term $\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi_1\rangle\|$. Notice that $|\phi\rangle_{\mathrm{CODE}}$ has database size at most t,

$$
\begin{aligned}
&\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi_1\rangle\| \\
&= \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\Pi_{t+1} |\phi_1\rangle\| \\
&\leq 2\|\Pi_{t+1}\left[\Pi_{\mathsf{NoCol}}, \Pi_{\mathsf{NoHatZero}}\right]\Pi_{t+1}\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{CODE}}\| \\
&\leq 2(t+1)\cdot 2^{-(m-3)/2} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{CODE}}\| \quad .
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
&\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi_2\rangle\| \\
&= \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\Pi_{t+1} |\phi_2\rangle\| \\
&\leq \|\Pi_t\left[\Pi_{\mathsf{NoCol}}, \Pi_{\mathsf{NoHatZero}}\right]\Pi_t\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{CODE}}\| \\
&\leq t\cdot 2^{-(m-3)/2} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{CODE}}\| \quad .
\end{aligned}
$$

Furthermore, as the error for $(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})$ only accumulates when the adversary queries Inv,

$$\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{CODE}}\|$$

$$
\begin{aligned}
&\leq \sum_{i\in[t]} \|\Pi_t \left[\mathsf{Inv}_{\mathrm{TDW}}, \Pi_{\mathsf{NoHatZero}}\right] \Pi_t\| \sqrt{\mathsf{mass}(\mathsf{A}, \mathsf{Inv}, i)} \\
&\leq 2^{-m/2+4}\sqrt{t}\sqrt{t\sum_{i\in[t]} \mathsf{mass}(\mathsf{A}, \mathsf{Inv}, i)} \\
&\leq 2^{-m/2+4} \cdot t \cdot \sqrt{w_2} \ .
\end{aligned}
$$

Combining the above inequalities, we obtain

$$
\begin{aligned}
&\|(\langle\psi_{\mathsf{Check}}| V_{\mathsf{Check}} - \langle\psi_{\mathsf{AltCheck}}| V_{\mathsf{ExtMsg}} |0\rangle_{\mathrm{W_1W_2}}) |\phi\rangle_{\mathrm{CODE}} \| \\
&\leq (12t + 16 + 32t\sqrt{w_2}) \cdot 2^{-m/2} \ . \qquad (14)
\end{aligned}
$$

Then the claim follows from Equation (13). □

5.5 Proof of Claim 5.7

We first prove a bound for a more general operator A.

Claim 5.11. *For an operator A, if $A = A^\dagger$, and for every $x \in X$, A commutes with $|\perp\rangle\langle\perp|_{\mathrm{D[x]}}$, then the following inequality holds:*

$$
\|\Pi_t \left[\Pi_{\mathsf{NoHatZero}}, A\right] \Pi_t\| \leq 2\sqrt{t} \max_{x\in X} \|\Pi_t \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} A\, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \Pi_t\| \ .
$$

Proof. To bound $\|\Pi_t \left[\Pi_{\mathsf{NoHatZero}}, A\right] \Pi_t\|$, it suffices to bound $\|\Pi_t\Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t\|$ because by the triangle inequality,

$$
\begin{aligned}
&\|\Pi_t \left[\Pi_{\mathsf{NoHatZero}}, A\right] \Pi_t\| \\
\leq &\|\Pi_t\Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t\| + \|\Pi_t \left(-I + \Pi_{\mathsf{NoHatZero}}\right) A\Pi_{\mathsf{NoHatZero}}\Pi_t\| \\
= &\|\Pi_t\Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t\| + \|\Pi_t\Pi_{\mathsf{NoHatZero}} A^\dagger \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t\| \\
= &2\|\Pi_t\Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t\| \ .
\end{aligned}
$$

We define a family of projectors $\Delta_z := \bigotimes_{x\in X} \left(\frac{I}{2} + (-1)^{z[x]+1} \left(\frac{I}{2} - |\perp\rangle\langle\perp|\right)\right)_{\mathrm{D[x]}}$ for each vector $z \in \{0,1\}^X$. That is, Δ_z projects onto $|\perp\rangle\langle\perp|$ on the registers D[x] when $z[x] = 0$, and projects onto $I - |\perp\rangle\langle\perp|$ otherwise.

Then $\{\Delta_z\}_z$ is in fact a measurement, where the outcome z indicates the locations where the database is non-empty. We observe that $\Pi_{\mathsf{NoHatZero}}$ commutes with Δ_z, and in addition, since A commutes with $|\perp\rangle\langle\perp|_{\mathrm{D[x]}}$ for every $x \in X$, A commutes with Δ_z, for each $z \in \{0,1\}^X$. As a result,

$$
\begin{aligned}
&\|\Pi_t\Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t\| \\
= &\|\sum_{z,z'\in\{0,1\}^X} \Delta_z\Pi_t\Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t\Delta_{z'}\| \\
= &\|\sum_{z,z'\in\{0,1\}^X} \Delta_z\Delta_{z'}\Pi_t\Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t\| \\
= &\|\sum_{z\in\{0,1\}^X} \Delta_z\Delta_z\Pi_t\Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t\|
\end{aligned}
$$

$$
\begin{aligned}
&= \| \sum_{z \in \{0,1\}^X} \Delta_z \Pi_t \Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t \Delta_z \| \\
&\leq \max_{z \in \{0,1\}^X} \| \Delta_z \Pi_t \Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t \Delta_z \| \\
&= \max_{z \in \{0,1\}^X \text{ s.t. } \mathsf{hm}(z) \leq t} \| \Delta_z \Pi_t \Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t \Delta_z \| \qquad (15)
\end{aligned}
$$

where we use Equation (1), and the fact that for z with Hamming weight more than t, $\Delta_z \Pi_t = 0$.

Now let's bound $\| \Delta_z \Pi_t \Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t \Delta_z \|$.

For z with Hamming weight $\mathsf{hm}\,(z) \leq t$, and every normalized state $|\phi\rangle$, we can write $\left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t \Delta_z |\phi\rangle$ in the following form

$$
\left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t \Delta_z |\phi\rangle = \sum_{i=1}^{\mathsf{hm}(z)} \alpha_i \, |\hat{0}^m\rangle_{\mathrm{D[x_i]}} \, |\phi_i\rangle_{\overline{\mathrm{D[x_i]}}}
$$

where x_i is the ith element in X such that $z[x] = 1$, $\overline{\mathrm{D[x_i]}}$ is all the registers excluding $\mathrm{D[x_i]}$, and $|\hat{0}^m\rangle_{\mathrm{D[x_i]}} |\phi_i\rangle_{\overline{\mathrm{D[x_i]}}}$ are orthonormal states.

Plug $\left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t \Delta_z |\phi\rangle$ into the following, and we can get that

$$
\begin{aligned}
&\| \Delta_z \Pi_t \Pi_{\mathsf{NoHatZero}} A \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t \Delta_z |\phi\rangle \| \\
=&\| \Delta_z \Pi_t \Pi_{\mathsf{NoHatZero}} A \Pi_t \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t \Delta_z |\phi\rangle \| \\
\leq& \sum_{i=1}^{\mathsf{hm}(z)} |\alpha_i| \, \| \Pi_t \Pi_{\mathsf{NoHatZero}} A \Pi_t \, |\hat{0}^m\rangle_{\mathrm{D[x_i]}} \, |\phi_i\rangle_{\overline{\mathrm{D[x_i]}}} \| \\
=& \sum_{i=1}^{\mathsf{hm}(z)} |\alpha_i| \, \| \Pi_{\mathsf{NoHatZero}} \Pi_t A \Pi_t \, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x_i]}} \, |\phi_i\rangle_{\overline{\mathrm{D[x_i]}}} \, |\hat{0}^m\rangle_{\mathrm{D[x_i]}} \| \\
\leq& \sum_{i=1}^{\mathsf{hm}(z)} |\alpha_i| \, \| \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x_i]}} \Pi_t A \Pi_t \, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x_i]}} \, |\phi_i\rangle_{\overline{\mathrm{D[x_i]}}} \, |\hat{0}^m\rangle_{\mathrm{D[x_i]}} \| \\
\leq& \left(\sum_{i=1}^{\mathsf{hm}(z)} |\alpha_i| \right) \max_{x \in X} \| \Pi_t \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} A \, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \, \Pi_t \| \\
\leq& \sqrt{t} \max_{x \in X} \| \Pi_t \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} A \, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \, \Pi_t \| \ ,
\end{aligned}
$$

where the last inequality is due to Cauchy-Schwarz, $\mathsf{hm}\,(z) \leq t$ and $\| \left(I - \Pi_{\mathsf{NoHatZero}}\right) \Pi_t \Delta_z |\phi\rangle \| \leq 1$.

Combining Equation (15) and the above inequality, we get Claim 5.11. □

Both Π_{NoCol} and Inv commute with $|\perp\rangle\langle\perp|_{\mathrm{D[x]}}$ for every $x \in X$, and thus Claim 5.7 follows from Claim 5.11 and the following two claims.

Claim 5.12. *For every $x \in X$ and every query bound t,*

$$
\| \Pi_t \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} \Pi_{\mathsf{NoCol}} \, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \, \Pi_t \| \leq 2^{-(m-1)/2} \sqrt{t} \ .
$$

Proof. For every normalized state $|\phi\rangle$, we can write $|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle$ in the following form

$$|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle = \frac{1}{\sqrt{2^m}}\sum_{y\in\{0,1\}^m}|y\rangle_{\mathrm{D[x]}}\sum_{D[X/\{x\}]\text{ has size at most }t-1}\alpha_{D[X/\{x\}]}\,|D[X/\{x\}]\rangle_{\mathrm{D[X/\{x\}]}}\quad.$$

Notice that

$$\begin{aligned}&\|\Pi_t\left(I-|\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}}\Pi_{\mathsf{NoCol}}\,|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle\,\|\\ \le&\|\left(I-|\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}}\Pi_{\mathsf{NoCol}}\,|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle\,\|\\ =&\sqrt{\|\Pi_{\mathsf{NoCol}}\,|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle\,\|^2-\|\,\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_{\mathsf{NoCol}}\,|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle\,\|^2}\end{aligned}$$

To bound $\|\Pi_{\mathsf{NoCol}}\,|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle\,\|^2$, we plug in $|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle$ to get that

$$\begin{aligned}&\|\Pi_{\mathsf{NoCol}}\,|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle\,\|^2\\ =&\|\Pi_{\mathsf{NoCol}}\frac{1}{\sqrt{2^m}}\sum_{y\in\{0,1\}^m}|y\rangle_{\mathrm{D[x]}}\sum_{D[X/\{x\}]\text{ has size at most }t-1}\alpha_{D[X/\{x\}]}\,|D[X/\{x\}]\rangle_{\mathrm{D[X/\{x\}]}}\,\|^2\\ =&\|\frac{1}{\sqrt{2^m}}\sum_{\substack{D[X/\{x\}]\text{ has size at most }t-1\\ \text{and it doesn't have collisions}}}\alpha_{D[X/\{x\}]}\sum_{y\in\{0,1\}^m\text{ and }y\notin D[X/\{x\}]}|y\rangle_{\mathrm{D[x]}}\,|D[X/\{x\}]\rangle_{\mathrm{D[X/\{x\}]}}\,\|^2\\ =&\sum_{\substack{D[X/\{x\}]\text{ has size at most }t-1\\ \text{and it doesn't have collisions}\\ y\in\{0,1\}^m\text{ and }y\notin D[X/\{x\}]}}\frac{1}{2^m}\left|\alpha_{D[X/\{x\}]}\right|^2\\ \le&\sum_{\substack{D[X/\{x\}]\text{ has size at most }t-1\\ \text{and it doesn't have collisions}}}\left|\alpha_{D[X/\{x\}]}\right|^2\end{aligned}$$

Similarly,

$$\begin{aligned}&\|\,\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_{\mathsf{NoCol}}\,|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle\,\|^2\\ =&\|\,\langle\hat{0}^m|_{\mathrm{D[x]}}\,\frac{1}{\sqrt{2^m}}\sum_{\substack{D[X/\{x\}]\text{ has size at most }t-1\\ \text{and it doesn't have collisions}}}\alpha_{D[X/\{x\}]}\sum_{y\in\{0,1\}^m\text{ and }y\notin D[X/\{x\}]}|y\rangle_{\mathrm{D[x]}}\,|D[X/\{x\}]\rangle_{\mathrm{D[X/\{x\}]}}\,\|^2\\ =&\sum_{\substack{D[X/\{x\}]\text{ has size at most }t-1\\ \text{and it doesn't have collisions}}}\left|\alpha_{D[X/\{x\}]}\sum_{y\in\{0,1\}^m\text{ and }y\notin D[X/\{x\}]}\frac{1}{2^m}\right|^2\\ \ge&\sum_{\substack{D[X/\{x\}]\text{ has size at most }t-1\\ \text{and it doesn't have collisions}}}\left|\alpha_{D[X/\{x\}]}\right|^2\left(1-\frac{t}{2^m}\right)^2\quad.\end{aligned}$$

As a result, for any normalized state $|\phi\rangle$,

$$\begin{aligned}&\|\Pi_t\left(I-|\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}}\Pi_{\mathsf{NoCol}}\,|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle\,\|\\ \le&\sqrt{\|\Pi_{\mathsf{NoCol}}\,|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle\,\|^2-\|\,\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_{\mathsf{NoCol}}\,|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\,\Pi_t\,|\phi\rangle\,\|^2}\end{aligned}$$

$$\leq \sqrt{\sum_{\substack{D[X/\{x\}] \text{ has size at most } t-1 \\ \text{and it doesn't have collisions}} \left|\alpha_{D[X/\{x\}]}\right|^2} \cdot \sqrt{1-\left(1-\frac{t}{2^m}\right)^2}$$
$$\leq 2^{-(m-1)/2}\sqrt{t} \ ,$$

which by definition, implies

$$\|\Pi_t \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} \Pi_{\mathsf{NoCol}} |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \Pi_t\| \leq 2^{-(m-1)/2}\sqrt{t} \ .$$

□

Claim 5.13. *For every $x \in X$ and every query bound t,*

$$\|\Pi_t \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} \mathsf{Inv}_{\mathrm{TDW}} |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \Pi_t\| \leq 2^{-m/2+3} \ .$$

Proof. Recall that

$$\mathsf{Inv}_{\mathrm{TDW}} = \sum_{y\in\{0,1\}^m} |y\rangle\langle y|_{\mathrm{T}} \otimes M^{(y)}_{\mathrm{DW}}$$

is a controlled unitary. By Equation (2),

$$\| \left[\mathsf{Inv}_{\mathrm{TDW}}, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\right] \| \leq \max_{y\in\{0,1\}^m} \| \left[M^{(y)}_{\mathrm{DW}}, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\right] \| \ .$$

For every $y \in \{0,1\}^m$, since $|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}$ only acts on D[x] within the database register D, by Lemma 5.3,

$$\| \left[M^{(y)}_{\mathrm{DW}}, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\right] \| \leq 4\| \left[|y\rangle\langle y|_{\mathrm{D[x]}}, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\right] \| \leq 2^{-m/2+3} \ .$$

Therefore, $\mathsf{Inv}_{\mathrm{TDW}}$ and $|\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}$ almost commute with each other,

$$\| \left[\mathsf{Inv}_{\mathrm{TDW}}, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\right] \| \leq 2^{-m/2+3} \ .$$

Now we are ready to bound $\|\Pi_t \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} \mathsf{Inv}_{\mathrm{TDW}} |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \Pi_t\|$.

$$\begin{aligned}
&\|\Pi_t \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} \mathsf{Inv}_{\mathrm{TDW}} |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \Pi_t\| \\
\leq &\| \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} \mathsf{Inv}_{\mathrm{TDW}} |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \| \\
\leq &\| \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}} \mathsf{Inv}_{\mathrm{TDW}}\| + \| \left(I - |\hat{0}^m\rangle\langle\hat{0}^m|\right)_{\mathrm{D[x]}} \left[\mathsf{Inv}_{\mathrm{TDW}}, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\right] \| \\
\leq &\| \left[\mathsf{Inv}_{\mathrm{TDW}}, |\hat{0}^m\rangle\langle\hat{0}^m|_{\mathrm{D[x]}}\right] \| \\
\leq &2^{-m/2+3} \ .
\end{aligned}$$

□

6 Defining extractable quantum state vector commitments

We define the syntax of quantum state vector commitments, and then provide a formal definition for extractable quantum state vector commitments.

6.1 Syntax for quantum state vector commitments

We consider non-interactive quantum state vector commitments in the quantum random oracle setting.

Definition 6.1 (Quantum state vector commitments in the QROM)**.** *In the quantum random oracle model, a non-interactive* quantum state vector commitment *(QVC) for block size s and message length n is a tuple of polynomial-time oracle-aided quantum algorithms* $\mathsf{QSVC} = (\mathsf{Com}, \mathsf{Open}, \mathsf{Query}, \mathsf{Update}, \mathsf{Recover})$ *where*

- $\mathsf{QSVC.Com}^{\mathsf{RO}}(1^\lambda, \mathrm{M}) \to (\mathrm{C}, \mathrm{T})$*:* $\mathsf{QSVC.Com}^{\mathsf{RO}}$ *has oracle access to* RO *and is given as inputs the security parameter 1^λ and a $n \cdot s$-qubit quantum state on the register M, which might entangle with the register E. It outputs a state on two registers C and T. The register C will be sent to the receiver in the commitment phase and the register T contains necessary information for the local opening in the open phase.*
- $\mathsf{QSVC.Open}^{\mathsf{RO}}(1^\lambda, \mathcal{Q}, \mathrm{T}) \to (\mathrm{O}, \mathrm{R})$*: To open the commitment on positions $\mathcal{Q} \subseteq \{1, 2, \ldots, n\}$,* $\mathsf{QSVC.Open}^{\mathsf{RO}}$*, with oracle access to* RO*, takes the security parameter 1^λ, the query set $\mathcal{Q}$, and a state on T as inputs, and outputs a state on registers (O, R). The state in register O is interpreted as the opening and will be sent to the receiver for local decommitment, and R contains the unused part in the register T.*
- $\mathsf{QSVC.Query}^{\mathsf{RO}}(1^\lambda, \mathcal{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}) \to (b, \mathrm{C}, \mathrm{O}, \mathrm{Y})$*: To make a query to the message,* $\mathsf{QSVC.Query}^{\mathsf{RO}}$*, with oracle access to* RO*, takes as inputs the security parameter 1^λ, the query set $\mathcal{Q} \subseteq \{1, 2, \cdots, n\}$, and a state on registers $(\mathrm{C}, \mathrm{O}, \mathrm{Y})$. It first checks the validity of the opening and outputs a bit b to indicate whether the decommitment is valid. If it is valid, it recovers the quantum message on positions $\mathcal{Q}$, and swaps it with the register Y, then it does the reverse to get a new state on registers (C, O), and outputs a state on registers $(\mathrm{C}, \mathrm{O}, \mathrm{Y})$ together with the validity bit b.*
- $\mathsf{QSVC.Update}^{\mathsf{RO}}(1^\lambda, \mathcal{Q}, \mathrm{O}, \mathrm{R}) \to \mathrm{T}$*:* $\mathsf{QSVC.Update}^{\mathsf{RO}}$*, with oracle access to* RO*, takes as inputs the security parameter 1^λ, a query set $\mathcal{Q}$, and a state on registers (O, R), and reorganizes the states according to $\mathcal{Q}$ to erase the information about $\mathcal{Q}$. It outputs a state on the register T.*
- $\mathsf{QSVC.Recover}^{\mathsf{RO}}(1^\lambda, \mathrm{C}, \mathrm{T}) \to (b, \mathrm{M})$*: To recover the messages,* $\mathsf{QSVC.Recover}^{\mathsf{RO}}$*, with oracle access to* RO*, takes as inputs the security parameter 1^λ and a state on two registers (C, T). It outputs a validity bit b and a state on the register M, which is supposed to be the updated message after the queries.*

We slightly overload the notations $\mathsf{QSVC.Open}$*,* $\mathsf{QSVC.Query}$*, and* $\mathsf{QSVC.Update}$ *to also denote the corresponding coherent implementations when the query set is provided in the register Q. In particular, we write $(\mathrm{Q}, \mathrm{O}, \mathrm{R}) \leftarrow \mathsf{QSVC.Open}^{\mathsf{RO}}(1^\lambda, \mathrm{Q}, \mathrm{T})$, $(b, \mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}) \leftarrow \mathsf{QSVC.Query}^{\mathsf{RO}}(1^\lambda, \mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y})$, and $(\mathrm{Q}, \mathrm{T}) \leftarrow \mathsf{QSVC.Update}^{\mathsf{RO}}(1^\lambda, \mathrm{Q}, \mathrm{O}, \mathrm{R})$ for the coherent implementations. When it is clear from the context, we sometimes omit the security parameter λ.*

The vector commitment QSVC *should satisfy the following correctness and efficiency requirements.*

Perfect correctness. *An honest sender should be able to produce a valid decommitment for every part of the quantum message and recover the updated message after the queries. Namely, for every integer λ, s, function*

RO *with output length* m*, unbounded quantum adversary* A*, and unbounded quantum distinguisher* D*,*

$$\Pr\left[D^{\mathsf{RO}}(b,\mathrm{M},\mathrm{E},\mathrm{Q},\mathrm{Y}) \;\middle|\; \begin{array}{l}(\mathrm{M},\mathrm{E},\mathrm{Q},\mathrm{Y}) \leftarrow A^{\mathsf{RO}}\\ (\mathrm{C},\mathrm{T}) \leftarrow \mathsf{QSVC.Com}^{\mathsf{RO}}(1^\lambda,\mathrm{M})\\ (\mathrm{Q},\mathrm{O},\mathrm{R}) \leftarrow \mathsf{QSVC.Open}^{\mathsf{RO}}(1^\lambda,\mathrm{Q},\mathrm{T})\\ (b_1,\mathrm{Q},\mathrm{C},\mathrm{O},\mathrm{Y}) \leftarrow \mathsf{QSVC.Query}^{\mathsf{RO}}(1^\lambda,\mathrm{Q},\mathrm{C},\mathrm{O},\mathrm{Y})\\ (\mathrm{Q},\mathrm{T}) \leftarrow \mathsf{QSVC.Update}^{\mathsf{RO}}(1^\lambda,\mathrm{Q},\mathrm{O},\mathrm{R})\\ (b_2,\mathrm{M}) \leftarrow \mathsf{QSVC.Recover}^{\mathsf{RO}}(1^\lambda,\mathrm{C},\mathrm{T})\\ b \coloneqq b_1 \wedge b_2\end{array}\right]$$
$$= \Pr\left[D^{\mathsf{RO}}(1,\mathrm{M},\mathrm{E},\mathrm{Q},\mathrm{Y}) \;\middle|\; \begin{array}{l}(\mathrm{M},\mathrm{E},\mathrm{Q},\mathrm{Y}) \leftarrow A^{\mathsf{RO}}\\ (\mathrm{Q},\mathrm{M},\mathrm{Y}) \leftarrow U^{\mathsf{qry}}(\mathrm{Q},\mathrm{M},\mathrm{Y})\end{array}\right],$$

where the random oracle output length m *is a function of the security parameter* λ *specified by the scheme, and we slightly abuse notation by overloading* U^{qry} *to also denote the unitary that, controlled on* Q*, swaps* Y *with the corresponding subsets of the message register* M *(i.e., a parallel application of the unitary* U^{qry} *from Section 2.6).*

Local opening. *For a vector commitment scheme to be non-trivial, we require that the sizes of* C *and* O *grow much slower than the message length. To be more specific, the size of* C *is* $\mathsf{poly}(\lambda)$*, and the size of* O *is* $\mathsf{poly}(|\mathcal{Q}|, \log n, \lambda)$.

6.2 The extractability definition

Now we're ready to define the extractability for quantum state vector commitments. The syntax of a quantum state vector commitment is different from the syntax for the basic quantum state commitment.

Definition 6.2 (Quantum message extractor)**.** *A* quantum message extractor $\mathcal{E}$ *for a quantum state vector commitment scheme with query access to* U_{Sim} *and* U_{Ext} *(both have the state register* S*) has the following syntax:*

1. $\mathcal{E}.\mathsf{ExtMsg}^{U_{\mathsf{Sim}}(\mathrm{S}),U_{\mathsf{Ext}}(\mathrm{S})}(\mathrm{C}) \to (\mathrm{M},\mathrm{Aux})$*:* $\mathcal{E}.\mathsf{ExtMsg}$ *takes as inputs a commitment in register* C*, makes queries to* U_{Sim} *and* U_{Ext}*, and outputs a quantum state on two registers* M *and* Aux *(which will be used for* $\mathsf{AltCheck}$*).*
2. $\mathcal{E}.\mathsf{AltCheck}(\mathcal{Q},\mathrm{O},\mathrm{Aux}) \to (b,\mathrm{O},\mathrm{Aux})$*:* $\mathcal{E}.\mathsf{AltCheck}$ *takes as inputs the query set* $\mathcal{Q}$*, the opening in register* O*, and the auxiliary information in register* Aux*, and outputs* $b = 0$ *or* 1*, indicating whether the sender gives a valid opening, along with a quantum state on registers* $(\mathrm{O},\mathrm{Aux})$.
3. $\mathcal{E}.\mathsf{AltCommit}^{U_{\mathsf{Sim}}(\mathrm{S}),U_{\mathsf{Ext}}(\mathrm{S})}(\mathrm{Aux},\mathrm{M}) \to \mathrm{C}$*:* $\mathcal{E}.\mathsf{AltCommit}$ *takes as inputs a quantum state on the register* Aux*, and the updated extracted quantum message on the register* M*, makes queries to* U_{Sim} *and* U_{Ext}*, and outputs a quantum state on the register* C.

We slightly overload the notation $\mathcal{E}.\mathsf{AltCheck}$ *to also denote the corresponding coherent implementations when the query set is provided in the register* Q*. In particular, we write* $(b,\mathrm{Q},\mathrm{O},\mathrm{Aux}) \leftarrow \mathcal{E}.\mathsf{AltCheck}(\mathrm{Q},\mathrm{O},\mathrm{Aux})$ *for the coherent implementation.*

Definition 6.3 (Extractability)**.** *A quantum state vector commitment scheme* $\mathsf{QSVC} = (\mathsf{Com},\mathsf{Open},\mathsf{Query},\mathsf{Update},\mathsf{Recover})$ *is* $(\xi_{\mathsf{Ext}}^{(1)},\xi_{\mathsf{Ext}}^{(2)},\xi_{\mathsf{Ext}}^{(3)})$-extractable *if there exist a* $\xi_{\mathsf{Ext}}^{(1)}$*-quantum simulator* U_{Sim} *with state register* S *for the random oracle, a stateful oracle* U_{Ext} *with the same state register* S*, and a polynomial-time quantum message extractor* $\mathcal{E}$ *with query access to* U_{Sim} *and* U_{Ext} *as in Definition 6.2 such that the following holds:*

1. *(U_{Ext} does not disturb the simulation.) For every integer m, $\| [U_{\mathsf{Ext}}, U_{\mathsf{Sim}}] \|^2 \leq \xi^{(2)}_{\mathsf{Ext}}(m)$, where m is the output length of the random oracle.*

2. *($\mathcal{E}$ gives the only state that the adversary A can open to.) For every integer m, t, real numbers $w_1, w_2 \in [0,t]$ such that $w_1 + w_2 \leq t$, t-query p-phase quantum adversary A such that $\mathsf{TotalMass}(A, U_{\mathsf{Sim}}) \leq w_1$ and $\mathsf{TotalMass}(A, U_{\mathsf{Ext}}) \leq w_2$, and unbounded quantum distinguisher D,*

$$\left|\sqrt{\Pr\left[\mathsf{QSVCSimWorld}(A,D)\right]} - \sqrt{\Pr\left[\mathsf{QSVCExtWorld}(A,D)\right]}\right|^2 \leq \xi^{(3)}_{\mathsf{Ext}}(t,p,w_1,w_2,m,n) \ ,$$

where m is the output length of the random oracle, n is the message length of the scheme, and the games $\mathsf{QSVCSimWorld}(A,D)$ and $\mathsf{QSVCExtWorld}(A,D)$ are defined below:

- $\mathsf{QSVCSimWorld}(A,D)$*:*
  - *(a) The game initializes the registers:* $\mathrm{S} \leftarrow |\bot\rangle$, $\mathrm{Q} \leftarrow |\bar{0}\rangle$, $\mathrm{Y} \leftarrow |\bar{0}\rangle$, $\mathrm{O} \leftarrow |\bar{0}\rangle$.
  - *(b) The adversary generates a commitment:* $(\mathrm{C}, \mathrm{E}) \leftarrow A^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}$.
  - *(c) The adversary does the following:*
    For $i \in [p]$:
    - *i. The adversary generates a query location and the corresponding opening, together with an answer register for the superposition query:* $(\mathrm{Q}, \mathrm{O}, \mathrm{Y}, \mathrm{E}) \leftarrow A^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}(\mathrm{Q}, \mathrm{O}, \mathrm{Y}, \mathrm{E})$.
    - *ii. The game implements the query:* $(b_i, \mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}) \leftarrow \mathsf{QSVC.Query}^{U_{\mathsf{Sim}}(\mathrm{S})}(1^\lambda, \mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y})$.
  - *(d) The game generates the output: If $\wedge_{i\in[p]} b_i = 0$, output 0; otherwise, compute $b' \leftarrow D(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}, \mathrm{S})$ and output b'.*
- $\mathsf{QSVCExtWorld}(A,D)$*:*
  - *(a) The game initializes the registers:* $\mathrm{S} \leftarrow |\bot\rangle$, $\mathrm{Q} \leftarrow |\bar{0}\rangle$, $\mathrm{Y} \leftarrow |\bar{0}\rangle$, $\mathrm{O} \leftarrow |\bar{0}\rangle$.
  - *(b) The adversary generates a commitment:* $(\mathrm{C}, \mathrm{E}) \leftarrow A^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}$.
  - *(c) The game uses the extractor to extract the underlying message:* $(\mathrm{M}, \mathrm{Aux}) \leftarrow \mathcal{E}.\mathsf{ExtMsg}^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}(\mathrm{C})$.
  - *(d) The adversary does the following:*
    For $i \in [p]$:
    - *i. The adversary generates a query location and the corresponding opening, together with an answer register for the superposition query:* $(\mathrm{Q}, \mathrm{O}, \mathrm{Y}, \mathrm{E}) \leftarrow A^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}(\mathrm{Q}, \mathrm{O}, \mathrm{Y}, \mathrm{E})$.
    - *ii. The game uses the alternative check to check if the opening is valid:* $(b_i, \mathrm{Q}, \mathrm{O}, \mathrm{Aux}) \leftarrow \mathcal{E}.\mathsf{AltCheck}(\mathrm{Q}, \mathrm{O}, \mathrm{Aux})$.
    - *iii. The game implements the query:* $(\mathrm{Q}, \mathrm{M}, \mathrm{Y}) \leftarrow U^{\mathsf{qry}}(\mathrm{Q}, \mathrm{M}, \mathrm{Y})$.
  - *(e) The game uses alternative commit to get the commitment:* $\mathrm{C} \leftarrow \mathcal{E}.\mathsf{AltCommit}^{U_{\mathsf{Sim}}(\mathrm{S}), U_{\mathsf{Ext}}(\mathrm{S})}(\mathrm{Aux}, \mathrm{M})$.
  - *(f) The game generates the output: If $\wedge_{i\in[p]} b_i = 0$, output 0; otherwise, compute $b' \leftarrow D(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}, \mathrm{S})$ and output b'.*

7 Construction of extractable quantum state vector commitments

7.1 Labeling the Merkle tree

We make some conventions on how to label each vertex in the Merkle tree before giving the construction for the quantum state vector commitment.

A vertex v in a Merkle tree of depth d is labeled as a string $\ell \in \{0,1\}^{\leq d}$. The length of ℓ indicates the depth of the vertex v from the root, and the i-th bit of ℓ indicates whether we need to take the left edge or the right edge on the path from the root to v (0 for left and 1 for right). With the above convention, we have the following facts:

- The root of the Merkle tree is labeled as the empty string ε;
- For a string $\ell \in \{0,1\}^{<d}$, the left child of a vertex v with label ℓ is labeled as $\ell \parallel 0$;
- For a string $\ell \in \{0,1\}^{<d}$, the right child of a vertex v with label ℓ is labeled as $\ell \parallel 1$;
- For a non-empty string $\ell \in \{0,1\}^{\leq d}$, let $\mathsf{Sib}(\ell)$ be the label of the sibling of the vertex with label ℓ. Then $\mathsf{Sib}(\cdot)$ satisfies that $\mathsf{Sib}(\ell)$ is the string of length $|\ell|$ and $\mathsf{Sib}(\ell)$ equals ℓ on every bit except the last one.
- For a non-empty string $\ell \in \{0,1\}^{\leq d}$, let $\mathsf{Par}(\ell)$ be the label of the parent of the vertex with label ℓ. Then $\mathsf{Par}(\ell)$ is the prefix of ℓ of length $|\ell| - 1$.
- For a string $\ell \in \{0,1\}^{\leq d}$, let $\mathsf{Path}(\ell)$ be the set of labels of the vertices on the path from the root to the vertex v with label ℓ (including the root and the vertex v). Then $\mathsf{Path}(\ell)$ contains the $|\ell| + 1$ prefixes of ℓ (including the empty string ε and the string ℓ itself).

We overload the notation to use the string $\ell \in \{0,1\}^{\leq d}$ to mean the vertex with label ℓ.

Furthermore, we overload the functions $\mathsf{Sib}(\cdot)$ and $\mathsf{Path}(\cdot)$ to take sets as input. For every set $S \subseteq \{0,1\}^{\leq d}$, we denote the set of siblings of the vertices inside S as $\mathsf{Sib}(S) := \{\mathsf{Sib}(\ell) : \ell \in S \text{ and } \ell \neq \varepsilon\}$, and the set of vertices from the root to vertices inside S as $\mathsf{Path}(S) := \bigcup_{\ell \in S} \mathsf{Path}(\ell)$.

For each set $S \subseteq \{0,1\}^d$, we define $\mathsf{AuthPath}(S) := \mathsf{Sib}(\mathsf{Path}(S)) \setminus \mathsf{Path}(S)$.

7.2 An extractable quantum state vector commitment scheme

We apply the Merkle tree to the succinct quantum state commitment construction in Construction 5.2 to obtain a quantum state vector commitment scheme.

Construction 7.1. We assume the number of blocks n is a power of 2 (otherwise, we just add padding of 0 at the end). Denote $n = 2^d$ for an integer d.

Let RO be sampled uniformly from all the functions with range $\{0,1\}^m$, where we set m to be the security parameter λ, and let $\mathsf{CM} = (\mathsf{Com}^{\mathsf{RO}}, \mathsf{Check}^{\mathsf{RO}})$ be the succinct quantum state commitment in Construction 5.2. Let $C_{m,n}$ be the circuit for $\mathsf{Com}^{\mathsf{RO}}$ where the quantum message has $n := 4m$ qubits (in other words, it is the circuit in Construction 5.1 for $n = 4m$).

We construct $\mathsf{QSTC} = (\mathsf{Com}, \mathsf{Open}, \mathsf{Query}, \mathsf{Update}, \mathsf{Recover})$ for block size $s := 2m$ as follows.

$\mathsf{QSTC.Com}^{\mathsf{RO}}(\mathrm{M})$:

1. Divide the $(n \cdot s)$-qubit register M into n registers $(\mathrm{M}[\ell])_{\ell \in \{0,1\}^d}$ each of size s.
2. For each $\ell \in \{0,1\}^{d-1}$, define $\mathrm{M}[\ell] := (\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.
3. Initialize $n-1$ registers $(\mathrm{A}[\ell])_{\ell \in \{0,1\}^{<d}}$, each of size $2m$, as all-zero states.
4. For $j = d-1, d-2, \ldots, 0$:
   (a) For each $\ell \in \{0,1\}^j$, apply $C_{m,n}$, where the quantum message is on the registers $\mathrm{M}[\ell]$ (the quantum message has $2s = n$ qubits), the ancilla is on the register $\mathrm{A}[\ell]$, and the oracle access is implemented by RO, to obtain $(\mathrm{C}[\ell], \mathrm{O}[\ell])$.

(b) If $j \geq 1$, for each $\ell \in \{0,1\}^{j-1}$, set $\mathrm{M}[\ell] \coloneqq (\mathrm{C}[\ell,0], \mathrm{C}[\ell,1])$.

5. Set $\mathrm{C} \coloneqq \mathrm{C}[\varepsilon]$ and $\mathrm{T} \coloneqq \bigotimes_{\ell \in \{0,1\}^{<d}} \mathrm{O}[\ell]$.
6. Output C and T.

$\mathsf{QSTC.Open}^{\mathsf{RO}}(\mathcal{Q}, \mathrm{T})$:

1. Parse T as $\bigotimes_{\ell \in \{0,1\}^{<d}} \mathrm{O}[\ell]$.
2. Set $\mathrm{O} \coloneqq \bigotimes_{\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}} \mathrm{O}[\ell]$ and $\mathrm{R} \coloneqq \bigotimes_{\ell \in \{0,1\}^{<d} \setminus \mathsf{Path}(\mathcal{Q})} \mathrm{O}[\ell]$.
3. Output O and R.

$\mathsf{QSTC.Query}^{\mathsf{RO}}(\mathcal{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y})$:

1. Set $\mathrm{C}[\varepsilon] \coloneqq \mathrm{C}$, parse O as $\bigotimes_{\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}} \mathrm{O}[\ell]$, and parse Y as $\bigotimes_{\ell \in \mathcal{Q}} \mathrm{Y}[\ell]$.
2. For $j = 0, 1, \ldots, d-1$:
   (a) For each $\ell \in \mathsf{Path}(\mathcal{Q}) \cap \{0,1\}^j$, apply the inverse of $C_{m,n}$ to $(\mathrm{C}[\ell], \mathrm{O}[\ell])$, where the oracle access is implemented by RO, to obtain the committed quantum message in $\mathrm{M}[\ell]$ and the ancilla in $\mathrm{A}[\ell]$.
   (b) If $j \neq d-1$, for each $\ell \in \mathsf{Path}(\mathcal{Q}) \cap \{0,1\}^j$, parse $\mathrm{M}[\ell]$ as $(\mathrm{C}[\ell,0], \mathrm{C}[\ell,1])$.
   (c) If $j = d-1$, for each $\ell \in \mathsf{Path}(\mathcal{Q}) \cap \{0,1\}^j$, parse $\mathrm{M}[\ell]$ as $(\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.
3. For $\ell \in \mathcal{Q}$, apply SWAP operator: $(\mathrm{M}[\ell], \mathrm{Y}[\ell]) \leftarrow \mathsf{SWAP}(\mathrm{M}[\ell], \mathrm{Y}[\ell])$.
4. Measure $\mathrm{A}[\ell]$ for each $\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$ in the computational basis, and set $b = 1$ if all the measurement outcomes are 0, and otherwise set $b = 0$.
5. For each $\ell \in \mathsf{Path}(\mathcal{Q}) \cap \{0,1\}^{d-1}$, define $\mathrm{M}[\ell] \coloneqq (\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.
6. For $j = d-1, \ldots, 1, 0$:
   (a) For each $\ell \in \mathsf{Path}(\mathcal{Q}) \cap \{0,1\}^j$, apply $C_{m,n}$, where the quantum message is on the registers $\mathrm{M}[\ell]$, the ancilla is on the register $\mathrm{A}[\ell]$, and the oracle access is implemented by RO, to obtain $(\mathrm{C}[\ell], \mathrm{O}[\ell])$.
   (b) If $j \neq 0$, for each $\ell \in \mathsf{Path}(\mathcal{Q}) \cap \{0,1\}^{j-1}$, set $\mathrm{M}[\ell] \coloneqq (\mathrm{C}[\ell,0], \mathrm{C}[\ell,1])$.
7. Set $\mathrm{C} \coloneqq \mathrm{C}[\varepsilon]$, $\mathrm{O} \coloneqq \bigotimes_{\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}} \mathrm{O}[\ell]$, and $\mathrm{Y} \coloneqq \bigotimes_{\ell \in \mathcal{Q}} \mathrm{Y}[\ell]$.
8. Output $(b, \mathrm{C}, \mathrm{O}, \mathrm{Y})$.

$\mathsf{QSTC.Update}^{\mathsf{RO}}(\mathcal{Q}, \mathrm{O}, \mathrm{R})$:

1. Parse O as $\bigotimes_{\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}} \mathrm{O}[\ell]$ and parse R as $\bigotimes_{\ell \in \{0,1\}^{<d} \setminus \mathsf{Path}(\mathcal{Q})} \mathrm{O}[\ell]$.
2. Set $\mathrm{T} \coloneqq \bigotimes_{\ell \in \{0,1\}^{<d}} \mathrm{O}[\ell]$.
3. Output T.

$\mathsf{QSTC.Recover}^{\mathsf{RO}}(\mathrm{C}, \mathrm{T})$:

1. Set $\mathrm{C}[\varepsilon] \coloneqq \mathrm{C}$, and parse T as $\bigotimes_{\ell \in \{0,1\}^{<d}} \mathrm{O}[\ell]$.
2. For $j = 0, 1, \ldots, d-1$:
   (a) For each $\ell \in \{0,1\}^j$, apply the inverse of $C_{m,n}$ to $(\mathrm{C}[\ell], \mathrm{O}[\ell])$, where the oracle access is implemented by RO, to obtain the committed quantum message in $\mathrm{M}[\ell]$ and the ancilla in $\mathrm{A}[\ell]$.
   (b) If $j \neq d-1$, for each $\ell \in \{0,1\}^j$, parse $\mathrm{M}[\ell]$ as $(\mathrm{C}[\ell,0], \mathrm{C}[\ell,1])$.
   (c) If $j = d-1$, for each $\ell \in \{0,1\}^j$, parse $\mathrm{M}[\ell]$ as $(\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.
3. Measure $\mathrm{A}[\ell]$ for each $\ell \in \{0,1\}^{<d}$ in the computational basis, and set $b = 1$ if all the measurement outcomes are 0, and otherwise set $b = 0$.
4. Output (b, M).

Note that in our construction, QSTC.Recover just does the inverse of QSTC.Com.

By construction, QSTC has the following efficiency:

- The size of C is always $O(m) = O(\lambda)$ regardless of the message length.
- The size of O for a set $\mathcal{Q}$ is $O(m\,|\mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}|) = O(m\,|\mathcal{Q}|\,d) = O(\lambda\,|\mathcal{Q}|\log n)$.

Therefore, QSTC satisfies the local opening requirement of QSVC.

Moreover, the perfect completeness of QSTC follows from the perfect completeness of Com.

7.3 Extractor for the vector commitment scheme

We present the extractor for the quantum state vector commitment scheme QSTC in Construction 7.1 before showing QSTC is extractable.

We begin with the oracle access of the extractor for QSTC, which is exactly the same as the oracle access of the extractor for the basic commitment in Construction 5.4. Specifically, we use the unitary

$$O_{\mathrm{XYD}} = \sum_{x\in X} |x\rangle\langle x|_{\mathrm{X}} \otimes F_{\mathrm{D[x]}}\mathsf{CNOT}_{\mathrm{D[x]Y}}F_{\mathrm{D[x]}}$$

to simulate the random oracle, and the unitary $\mathsf{Inv}_{\mathrm{TDW}}$ as defined in Equation (5) to help the extractor, where W can be parsed as BM'.

Construction 7.2 (The extractor $\mathcal{E}_{\mathsf{QSTC}}$)**.** Let $\mathcal{E}_{\mathsf{CM}}$ be the extractor for the scheme CM in Construction 5.4. We consider the following $\mathcal{E}_{\mathsf{QSTC}}$ for $\mathsf{QSTC} = (\mathsf{Com}, \mathsf{Open}, \mathsf{Query}, \mathsf{Update}, \mathsf{Recover})$ in Construction 7.1. $\mathcal{E}_{\mathsf{QSTC}}$ has query access to $O(\mathrm{D})$ and $\mathsf{Inv}(\mathrm{D})$.

$\mathcal{E}_{\mathsf{QSTC}}$ has the following interfaces:

– $\mathcal{E}_{\mathsf{QSTC}}.\mathsf{ExtMsg}(\mathrm{C})$:
  1. Set $\mathrm{C}[\varepsilon] \coloneqq \mathrm{C}$.
  2. For $j = 0, 1, \cdots, d-1$:
     (a) For each $\ell \in \{0,1\}^j$, run $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}(\mathrm{C}[\ell])$ with the working register $(\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])$ initialized as all zero states by forwarding its queries to $O(\mathrm{D})$ and $\mathsf{Inv}(\mathrm{D})$ to obtain $\mathrm{M}[\ell]$ and $\mathrm{Aux}[\ell]$.
     (b) If $j \neq d-1$, for each $\ell \in \{0,1\}^j$, parse $\mathrm{M}[\ell]$ as $(\mathrm{C}[\ell,0], \mathrm{C}[\ell,1])$.
     (c) If $j = d-1$, for each $\ell \in \{0,1\}^j$, parse $\mathrm{M}[\ell]$ as $(\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.
  3. Set $\mathrm{M} \coloneqq \bigotimes_{\ell\in\{0,1\}^d} \mathrm{M}[\ell]$ and $\mathrm{Aux} \coloneqq \bigotimes_{\ell\in\{0,1\}^{<d}} \mathrm{Aux}[\ell]$.
  4. Output $(\mathrm{M}, \mathrm{Aux})$.

– $\mathcal{E}_{\mathsf{QSTC}}.\mathsf{AltCheck}(\mathcal{Q}, \mathrm{O}, \mathrm{Aux})$:
  1. Parse O as $\bigotimes_{\ell\in\mathsf{Path}(\mathcal{Q})\setminus\mathcal{Q}} \mathrm{O}[\ell]$ and Aux as $\bigotimes_{\ell\in\{0,1\}^{<d}} \mathrm{Aux}[\ell]$.
  2. For each $\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$, make the projective measurement as the algorithm $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCheck}$ on registers $(\mathrm{O}[\ell], \mathrm{Aux}[\ell])$ to get a result b_ℓ (while the registers $(\mathrm{O}[\ell], \mathrm{Aux}[\ell])$ still hold the post-measurement state).
  3. Set $b \coloneqq \bigwedge_{\ell\in\mathsf{Path}(\mathcal{Q})\setminus\mathcal{Q}} b_\ell$, $\mathrm{O} \coloneqq \bigotimes_{\ell\in\mathsf{Path}(\mathcal{Q})\setminus\mathcal{Q}} \mathrm{O}[\ell]$, and $\mathrm{Aux} \coloneqq \bigotimes_{\ell\in\{0,1\}^{<d}} \mathrm{Aux}[\ell]$.
  4. Output $(b, \mathrm{O}, \mathrm{Aux})$.

– $\mathcal{E}_{\mathsf{QSTC}}.\mathsf{AltCommit}(\mathrm{Aux}, \mathrm{M})$:
  1. Parse Aux as $\bigotimes_{\ell\in\{0,1\}^{<d}} \mathrm{Aux}[\ell]$.
  2. For each $\ell \in \{0,1\}^{d-1}$, define $\mathrm{M}[\ell] \coloneqq (\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.
  3. For $j = d-1, \cdots, 1, 0$:
     (a) For each $\ell \in \{0,1\}^j$ in the reverse of lexicographic order, apply the reverse of $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}$ by forwarding its queries to $O(\mathrm{D})$ and $\mathsf{Inv}(\mathrm{D})$ on $(\mathrm{M}[\ell], \mathrm{Aux}[\ell])$ to obtain a state on the register $\mathrm{C}[\ell]$ and the ancilla registers $(\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])$.

(b) If $j > 0$, for each $\ell \in \{0,1\}^{j-1}$, set $\mathrm{M}[\ell] \coloneqq (\mathrm{C}[\ell,0], \mathrm{C}[\ell,1])$.

4. Set $\mathrm{C} \coloneqq \mathrm{C}[\varepsilon]$.
5. Set $\mathrm{W} \coloneqq (\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])_{\ell \in \{0,1\}^{<d}}$.
6. Output C.

In the security analysis, we sometimes also let $\mathcal{E}_{\mathsf{QSTC}}.\mathsf{AltCommit}$ output the register W, and we sometimes also let $\mathcal{E}_{\mathsf{QSTC}}.\mathsf{ExtMsg}$ take an additional register W as an input register instead of initializing an all-zero register.

Note that in our construction, $\mathcal{E}_{\mathsf{QSTC}}.\mathsf{AltCommit}$ just does the inverse of $\mathcal{E}_{\mathsf{QSTC}}.\mathsf{ExtMsg}$ except that $\mathcal{E}_{\mathsf{QSTC}}.\mathsf{AltCommit}$ does not check whether the ancilla qubits return to the all-zero states.

7.4 Security analysis of the vector commitment scheme

We show the scheme QSTC is an extractable quantum state vector commitment scheme.

Theorem 7.3 (Extractability)**.** *The scheme* QSTC *in Construction 7.1 is* $(\xi^{(1)}_{\mathsf{Ext}}, \xi^{(2)}_{\mathsf{Ext}}, \xi^{(3)}_{\mathsf{Ext}})$*-extractable with the quantum simulator* $O(\mathrm{D})$*, the extractor oracle* $\mathsf{Inv}(\mathrm{D})$*, and the extractor* $\mathcal{E}_{\mathsf{QSTC}}$ *in Construction 7.2 where*

$$\begin{aligned}
\xi^{(1)}_{\mathsf{Ext}} &\coloneqq 0 \ , \\
\xi^{(2)}_{\mathsf{Ext}}(m) &\coloneqq 2^{-m+7} \ , \\
\xi^{(3)}_{\mathsf{Ext}}(t,p,w_1,w_2,m,n) &\coloneqq 2^{-m+22} p^2 n^2 (t+8np)^2 \left(n^2 + w_1 + w_2 + p + 1\right) \ .
\end{aligned}$$

We first present a lemma that shows for the single opening case, the world QSVCSimWorld1Open that does the query using QSTC, and the world QSVCOffExtWorld1Open that first extracts the underlying message, implements the query, and then recover the commitment are indistinguishable, as long as there is no collision during the execution of the games. Then Theorem 7.3 follows from the standard hybrid arguments and the fact that a quantum algorithm cannot obtain a collision in the database with high probability.

Lemma 7.4. *Let* QSTC *be the quantum vector state commitment in Construction 7.1, and* $\mathcal{E}_{\mathsf{QSTC}}$ *be the extractor in Construction 7.2 with oracle access to* $O(\mathrm{D})$ *and* $\mathsf{Inv}(\mathrm{D})$ *that works on an internal database register* D*.*

For every integer m*,* t*, real numbers* $w_1, w_2 \in [0,t]$ *such that* $w_1 + w_2 \leq t$*,* t*-query quantum adversary* A *such that* $\mathsf{TotalMass}(A, O) \leq w_1$ *and* $\mathsf{TotalMass}(A, \mathsf{Inv}) \leq w_2$*, and unbounded quantum distinguisher* D*,*

$$\begin{aligned}
&\left| \sqrt{\Pr\left[\mathsf{QSVCSimWorld1Open}^*(A,D)\right]} - \sqrt{\Pr\left[\mathsf{QSVCOffExtWorld1Open}^*(A,D)\right]} \right|^2 \\
&\leq \xi_{\mathsf{1Open}}(m,n,t,w_1,w_2) \ ,
\end{aligned}$$

where m *is the output length of the random oracle and* n *is the message length of the scheme, the error*

$$\xi_{\mathsf{1Open}}(m,n,t,w_1,w_2) \coloneqq 2^{-m+20} n^2 (t+8n)^2 \left(n^2 + w_2\right) \ ,$$

and the games $\mathsf{QSVCSimWorld1Open}^*(A,D)$ *and* $\mathsf{QSVCOffExtWorld1Open}^*(A,D)$ *are defined below:*

- $\mathsf{QSVCSimWorld1Open}^*(A,D)$*:*
  1. The game initializes the database register: $\mathrm{D} \leftarrow |\perp\rangle$*.*

2. *The adversary generates the query set, the commitment, the opening, and the answer register that might be entangled with the environment register:* $(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}$.
3. *The game implements the query:* $(b_{\mathit{NoCollision}}, b, \mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}) \leftarrow \mathsf{QSTC.Query}^{O(\mathrm{D})}_{\mathit{NoCollision}}(1^\lambda, \mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y})$.
4. *The game initializes the working register: set* $\mathrm{W} := (\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])_{\ell\in\{0,1\}^{<d}}$ *and initialize it as all-zero states.*
5. *The game generates the output: If* $b_{\mathit{NoCollision}} \wedge b = 0$*, output 0; otherwise, compute* $b' \leftarrow D(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}, \mathrm{D}, \mathrm{W})$ *and output* b'.

- $\mathsf{QSVCOffExtWorld1Open}^*(A, D)$*:*
1. *The game initializes the database register:* $\mathrm{D} \leftarrow |\perp\rangle$.
2. *The adversary generates the query set, the commitment, the opening, and the answer register that might be entangled with the environment register:* $(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}$.
3. *The game uses the extractor to extract the underlying message:* $(b_{\mathit{NoCollision}}, \mathrm{M}, \mathrm{Aux}) \leftarrow \mathcal{E}_{\mathsf{QSTC}}.\mathsf{ExtMsg}^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}_{\mathit{NoCollision}}(\mathrm{C})$.
4. *The game uses the alternative check to check if the opening is valid:* $(b, \mathrm{Q}, \mathrm{O}, \mathrm{Aux}) \leftarrow \mathcal{E}_{\mathsf{QSTC}}.\mathsf{AltCheck}(\mathrm{Q}, \mathrm{O}, \mathrm{Aux})$.
5. *The game implements the query:* $(\mathrm{Q}, \mathrm{M}, \mathrm{Y}) \leftarrow U^{\mathsf{qry}}(\mathrm{Q}, \mathrm{M}, \mathrm{Y})$.
6. *The game uses the extractor to recover the commitment:* $(b'_{\mathit{NoCollision}}, \mathrm{C}, \mathrm{W}) \leftarrow \mathcal{E}_{\mathsf{QSTC}}.\mathsf{AltCommit}^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}_{\mathit{NoCollision}}(\mathrm{Aux}, \mathrm{M})$.
7. *The game generates the output: If* $b_{\mathit{NoCollision}} \wedge b'_{\mathit{NoCollision}} \wedge b = 0$*, output 0; otherwise, compute* $b' \leftarrow D(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}, \mathrm{D}, \mathrm{W})$ *and output* b'.

The proof of Lemma 7.4 is deferred to Section 7.5. We first apply Lemma 7.4 to show Theorem 7.3.

Proof of Theorem 7.3. The extractor $\mathcal{E}_{\mathsf{QSTC}}$ uses the same oracles $\mathsf{Inv}(\mathrm{D})$ and $O(\mathrm{D})$ as the extractor $\mathcal{E}_{\mathsf{CM}}$ for the scheme CM in Construction 5.4. By Theorem 5.5, we have that

$$\xi^{(1)}_{\mathsf{Ext}} = 0 \ ,$$
$$\xi^{(2)}_{\mathsf{Ext}}(m) = 2^{-m+7} \ ,$$

and two O oracles commute, and two Inv oracles commute no matter which registers they act on.

To bound the term $\left|\sqrt{\Pr\left[\mathsf{QSVCSimWorld}(A, D)\right]} - \sqrt{\Pr\left[\mathsf{QSVCExtWorld}(A, D)\right]}\right|^2$, we first introduce a new game $\mathsf{QSVCOffExtWorld}$, which does the extraction after the adversary produces the openings. The differences between $\mathsf{QSVCOffExtWorld}$ and $\mathsf{QSVCExtWorld}$ are highlighted in blue.

$\mathsf{QSVCOffExtWorld}(A, D)$:
1. The game initializes the registers: $\mathrm{D} \leftarrow |\perp\rangle$, $\mathrm{Q} \leftarrow |\bar{0}\rangle$, $\mathrm{Y} \leftarrow |\bar{0}\rangle$, $\mathrm{O} \leftarrow |\bar{0}\rangle$.
2. The adversary generates a commitment: $(\mathrm{C}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}$.
3. Set $\mathrm{W} := (\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])_{\ell\in\{0,1\}^{<d}}$ and initialize it as all-zero states.
4. For $i \in [p]$:
   (a) The adversary generates a query location and the corresponding opening, together with an answer register for the superposition query: $(\mathrm{Q}, \mathrm{O}, \mathrm{Y}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}(\mathrm{Q}, \mathrm{O}, \mathrm{Y}, \mathrm{E})$.
   (b) The game uses the extractor to extract the underlying message: $(\mathrm{M}, \mathrm{Aux}) \leftarrow \mathcal{E}.\mathsf{ExtMsg}^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}(\mathrm{C}, \mathrm{W})$.
   (c) The game uses the alternative check to check if the opening is valid: $(b_i, \mathrm{Q}, \mathrm{O}, \mathrm{Aux}) \leftarrow \mathcal{E}.\mathsf{AltCheck}(\mathrm{Q}, \mathrm{O}, \mathrm{Aux})$.
   (d) The game implements the query: $(\mathrm{Q}, \mathrm{M}, \mathrm{Y}) \leftarrow U^{\mathsf{qry}}(\mathrm{Q}, \mathrm{M}, \mathrm{Y})$.
   (e) The game uses alternative commit to get the commitment: $(\mathrm{C}, \mathrm{W}) \leftarrow \mathcal{E}.\mathsf{AltCommit}^{O(\mathrm{D}),\mathsf{Inv}(\mathrm{D})}(\mathrm{Aux}, \mathrm{M})$.
5. The game generates the output: If $\left(\wedge_{i\in[p]} b_i\right) = 0$, output 0; otherwise, compute $b' \leftarrow D(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}, \mathrm{D})$ and output b'.

As the unitary part of $\mathcal{E}.\mathsf{AltCommit}$ is just the reverse of the unitary part of $\mathcal{E}.\mathsf{ExtMsg}$, the only difference between $\mathsf{QSVCOffExtWorld}$ and $\mathsf{QSVCExtWorld}$ is whether we do Item 4a first or Item 4b first. As a result, for every t-query quantum adversary A and unbounded quantum distinguisher D,

$$\begin{aligned}&\left|\sqrt{\Pr\left[\mathsf{QSVCExtWorld}(A,D)\right]}-\sqrt{\Pr\left[\mathsf{QSVCOffExtWorld}(A,D)\right]}\right|\\&\leq 4tn\cdot\|\left[O,\mathsf{Inv}\right]\|\leq 32\sqrt{2}tn\cdot 2^{-m/2}\ .\end{aligned}\tag{16}$$

It remains to bound $\left|\sqrt{\Pr\left[\mathsf{QSVCOffExtWorld}(A,D)\right]}-\sqrt{\Pr\left[\mathsf{QSVCSimWorld}(A,D)\right]}\right|$. We compare $\mathsf{QSVCSimWorld}$ and $\mathsf{QSVCOffExtWorld}$ using the following hybrids. For $j\in\{0,\ldots,p\}$, let H_j use $\mathsf{QSTC.Query}$ in the first j phases and the offline extraction procedure in the remaining phases. Thus, H_p is $\mathsf{QSVCSimWorld}$ and H_0 is $\mathsf{QSVCOffExtWorld}$.

Fix $j\in[p]$. The games H_j and H_{j-1} differ only in phase j. Their common prefix, up to the submission of the opening for the j-th phase, is an ordinary $\mathsf{QSVCSimWorld}$ prefix. We regard this prefix as the adversary of Lemma 7.4. Since the preceding $j-1$ phases make at most $4n(j-1)$ queries to O and none to Inv, this adversary has at most $t+4n(j-1)$ queries, with masses at most $w_1+4n(j-1)$ and w_2, respectively, where w_1 and w_2 refer to the adversary's query masses in $\mathsf{QSVCSimWorld}$.

Let $\mathsf{H}_j^{(0)}$ and $\mathsf{H}_j^{(1)}$ be obtained from H_j and H_{j-1}, respectively, by inserting no-collision checks only during the j-th phase. By Lemma 2.5,

$$\begin{aligned}&\left|\sqrt{\Pr\left[\mathsf{H}_j(A,D)\right]}-\sqrt{\Pr\left[\mathsf{H}_j^{(0)}(A,D)\right]}\right|\\&\leq 2\sqrt{6}\left(t+4n(j-1)+8n\right)\sqrt{w_1+4nj}\cdot 2^{-m/2}\ ,\end{aligned}$$

and

$$\begin{aligned}&\left|\sqrt{\Pr\left[\mathsf{H}_j^{(1)}(A,D)\right]}-\sqrt{\Pr\left[\mathsf{H}_{j-1}(A,D)\right]}\right|\\&\leq 2\sqrt{6}\left(t+4n(j-1)+8n\right)\sqrt{w_1+4nj}\cdot 2^{-m/2}\ .\end{aligned}$$

Moreover, by Lemma 7.4,

$$\begin{aligned}&\left|\sqrt{\Pr\left[\mathsf{H}_j^{(0)}(A,D)\right]}-\sqrt{\Pr\left[\mathsf{H}_j^{(1)}(A,D)\right]}\right|^2\\&\leq\xi_{\mathsf{1Open}}\left(m,n,t+4n(j-1),w_1+4n(j-1),w_2\right)\ .\end{aligned}$$

Therefore, by the triangle inequality, we obtain

$$\begin{aligned}&\left|\sqrt{\Pr\left[\mathsf{H}_j(A,D)\right]}-\sqrt{\Pr\left[\mathsf{H}_{j-1}(A,D)\right]}\right|^2\\&\leq\Big(4\sqrt{6}\left(t+4n(j-1)+8n\right)\sqrt{w_1+4nj}\cdot 2^{-m/2}\\&\quad+\left(\xi_{\mathsf{1Open}}\left(m,n,t+4n(j-1),w_1+4n(j-1),w_2\right)\right)^{1/2}\Big)^2\\&\leq\Big(4\sqrt{6}\left(t+4n(p-1)+8n\right)\sqrt{w_1+4np}\cdot 2^{-m/2}\end{aligned}$$

$$
\begin{aligned}
&\quad + \left(\xi_{\mathsf{1Open}}\left(m, n, t + 4n(p-1), w_1 + 4n(p-1), w_2\right)\right)^{1/2}\Big)^2 \\
&\leq 2\left(96\left(t + 4n(p-1) + 8n\right)^2\left(w_1 + 4np\right) \cdot 2^{-m} + 2^{-m+20} n^2 (t + 8np)^2\left(n^2 + w_2\right)\right) \\
&\leq 2^{-m+21} n^2 (t + 8np)^2\left(n^2 + w_1 + w_2 + p\right) \quad ,
\end{aligned}
$$

where the third inequality follows from the Cauchy–Schwarz inequality.

Summing over $j \in [p]$, we obtain

$$
\begin{aligned}
&\left|\sqrt{\Pr\left[\mathsf{QSVCSimWorld}(A, D)\right]} - \sqrt{\Pr\left[\mathsf{QSVCOffExtWorld}(A, D)\right]}\right|^2 \\
&\leq 2^{-m+21} p^2 n^2 (t + 8np)^2\left(n^2 + w_1 + w_2 + p\right) \quad .
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\left|\sqrt{\Pr\left[\mathsf{QSVCSimWorld}(A, D)\right]} - \sqrt{\Pr\left[\mathsf{QSVCExtWorld}(A, D)\right]}\right|^2 \\
&\leq 2\left|\sqrt{\Pr\left[\mathsf{QSVCExtWorld}(A, D)\right]} - \sqrt{\Pr\left[\mathsf{QSVCOffExtWorld}(A, D)\right]}\right|^2 \\
&\quad + 2\left|\sqrt{\Pr\left[\mathsf{QSVCSimWorld}(A, D)\right]} - \sqrt{\Pr\left[\mathsf{QSVCOffExtWorld}(A, D)\right]}\right|^2 \\
&\leq 2^{-m+22} p^2 n^2 (t + 8np)^2\left(n^2 + w_1 + w_2 + p + 1\right) \\
&\leq \xi^{(3)}_{\mathsf{Ext}}(t, p, w_1, w_2, m, n) \quad .
\end{aligned}
$$

□

7.5 Proof of Lemma 7.4

Lemma 7.4 mainly follows from a standard hybrid argument over Lemma 5.6.

Proof of Lemma 7.4. As in the proof of Lemma 5.6, by the triangle inequality, to prove Lemma 7.4, it is sufficient to upper bound the ℓ_2 norm of the difference between the corresponding subnormalized states at the point immediately before applying D, denoted by $|\phi_{\mathsf{QSVCSimWorld1Open}^*}\rangle$ and $|\phi_{\mathsf{QSVCOffExtWorld1Open}^*}\rangle$, respectively. Specifically, it suffices to show that

$$
\| \, |\phi_{\mathsf{QSVCSimWorld1Open}^*}\rangle - |\phi_{\mathsf{QSVCOffExtWorld1Open}^*}\rangle \, \|^2 \leq \xi_{\mathsf{1Open}}(m, n, t, w_1, w_2) \quad .
$$

We observe that, in both games $\mathsf{QSVCSimWorld1Open}^*(A, D)$ and $\mathsf{QSVCOffExtWorld1Open}^*(A, D)$, all operations performed prior to invoking the distinguisher D are controlled by the query set $\mathcal{Q}$ stored in the register Q. Thus it suffices to establish the above inequality for every fixed query set $\mathcal{Q} \subseteq \{1, 2, \ldots, n\}$, rather than for a query set chosen by the adversary in superposition, as

$$
\begin{aligned}
&\| \, |\phi_{\mathsf{QSVCSimWorld1Open}^*}\rangle - |\phi_{\mathsf{QSVCOffExtWorld1Open}^*}\rangle \, \|^2 \\
&= \sum_{\mathcal{Q}} \| \, |\phi_{\mathsf{QSVCSimWorld1Open}^* \text{ when } \mathcal{Q} \text{ is chosen}}\rangle - |\phi_{\mathsf{QSVCOffExtWorld1Open}^* \text{ when } \mathcal{Q} \text{ is chosen}}\rangle \, \|^2 \Pr\left[\mathcal{Q} \text{ is chosen}\right] \quad .
\end{aligned}
$$

To this end, we fix a set $\mathcal{Q} \subseteq \{1, 2, \cdots, n\}$, and use the standard hybrid argument together with Lemma 5.6 to replace each call to the algorithm $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}$ with $\mathsf{CM.Check}$, and each call to the inverse of $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}$ with $\mathsf{CM.Com}$. We order the elements of $\mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$ first by increasing length and then

lexicographically, and write $\mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q} = \{\ell_1, \ell_2, \ldots, \ell_T\}$. Since $\mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q} \subseteq \{0,1\}^{<d}$, the number of elements $T < 2^d = n$.

We define the following intermediate hybrid games:

$\mathsf{H}_0(A, D)$: it does the same as $\mathsf{QSVCOffExtWorld1Open}^*$ except that instead of letting the adversary produce Q, we fix the query set to be $\mathcal{Q}$.

$\mathsf{H}_1^{(i)}(A, D)$:

1. The game initializes the database register: $\mathrm{D} \leftarrow |\perp\rangle$.
2. The adversary generates the query set, the commitment, the opening, and the answer register that might be entangled with the environment register: $(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}), \mathsf{Inv}(\mathrm{D})}$.
3. Parse O as $\bigotimes_{\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}} \mathrm{O}[\ell]$ and Y as $\bigotimes_{\ell \in \mathcal{Q}} \mathrm{Y}[\ell]$.
4. The game uses the extractor to extract the underlying message:
   (a) Set $\mathrm{C}[\varepsilon] := \mathrm{C}$.
   (b) For $\ell \in \{0,1\}^{<d}$:
      i. If $\ell \in \{\ell_1, \ell_2, \ldots, \ell_i\}$,
         A. Apply $\mathsf{CM.Check}_{\text{NoCollision}}$ to $(\mathrm{C}[\ell], \mathrm{O}[\ell])$, where the oracle access is implemented by $O(\mathrm{D})$, to obtain the committed quantum message in $\mathrm{M}[\ell]$ and two validity bits $b[\ell]$ and $b_{\text{NoCollision}}[\ell]$.
         B. Output 0 if $b[\ell] \wedge b_{\text{NoCollision}}[\ell] = 0$.
         C. Prepare a state $|\psi_{\mathsf{AltCheck}}\rangle$ on the registers $(\mathrm{O}[\ell], \mathrm{Aux}[\ell])$.
      ii. If $\ell \notin \{\ell_1, \ell_2, \ldots, \ell_i\}$,
         A. Run $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}_{\text{NoCollision}}(\mathrm{C}[\ell])$ by forwarding its queries to $O(\mathrm{D})$ and $\mathsf{Inv}(\mathrm{D})$ to obtain $(\mathrm{M}[\ell], \mathrm{Aux}[\ell])$, and a validity bit $b_{\text{NoCollision}}[\ell]$, and output 0 if the bit $b_{\text{NoCollision}}[\ell] = 0$.
         B. If $\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$, run $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCheck}(\mathrm{O}[\ell], \mathrm{Aux}[\ell])$ to get a validity bit $b[\ell]$ while maintaining the post-measurement state on registers $(\mathrm{O}[\ell], \mathrm{Aux}[\ell])$, and output 0 if the bit $b[\ell] = 0$.
      iii. If $|\ell| \neq d-1$, parse $\mathrm{M}[\ell]$ as $(\mathrm{C}[\ell, 0], \mathrm{C}[\ell, 1])$.
      iv. If $|\ell| = d-1$, parse $\mathrm{M}[\ell]$ as $(\mathrm{M}[\ell, 0], \mathrm{M}[\ell, 1])$.
   (c) Set $\mathrm{M} := \bigotimes_{\ell \in \{0,1\}^d} \mathrm{M}[\ell]$.
5. The game implements the query: for $\ell \in \mathcal{Q}$, $(\mathrm{M}[\ell], \mathrm{Y}[\ell]) \leftarrow \mathsf{SWAP}(\mathrm{M}[\ell], \mathrm{Y}[\ell])$.
6. The game uses the extractor to recover the commitment: $(b'_{\text{NoCollision}}, \mathrm{C}, \mathrm{W}) \leftarrow \mathcal{E}_{\mathsf{QSTC}}.\mathsf{AltCommit}^{O(\mathrm{D}), \mathsf{Inv}(\mathrm{D})}_{\text{NoCollision}}(\mathrm{Aux}, \mathrm{M})$.
7. Output 0 if $b'_{\text{NoCollision}} = 0$.
8. Set $\mathrm{O} := \bigotimes_{\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}} \mathrm{O}[\ell]$.
9. The game generates the output: compute $b' \leftarrow D(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}, \mathrm{D}, \mathrm{W})$ and output b'.

$\mathsf{H}_2^{(i)}(A, D)$:

1. The game initializes the database register: $\mathrm{D} \leftarrow |\perp\rangle$.
2. The adversary generates the query set, the commitment, the opening, and the answer register that might be entangled with the environment register: $(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}), \mathsf{Inv}(\mathrm{D})}$.
3. Parse O as $\bigotimes_{\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}} \mathrm{O}[\ell]$ and Y as $\bigotimes_{\ell \in \mathcal{Q}} \mathrm{Y}[\ell]$.
4. The game uses the extractor to extract the underlying message:
   (a) Set $\mathrm{C}[\varepsilon] := \mathrm{C}$.
   (b) For $\ell \in \{0,1\}^{<d}$:
      i. If $\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$,

A. Apply $\mathsf{CM.Check}_{\text{NoCollision}}$ to $(\mathrm{C}[\ell], \mathrm{O}[\ell])$, where the oracle access is implemented by $O(\mathrm{D})$, to obtain the committed quantum message in $\mathrm{M}[\ell]$ and two validity bits $b[\ell]$ and $b_{\text{NoCollision}}[\ell]$.
B. Output 0 if $b[\ell] \wedge b_{\text{NoCollision}}[\ell] = 0$.

ii. If $\ell \notin \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$,
A. Run $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}_{\text{NoCollision}}(\mathrm{C}[\ell])$ by forwarding its queries to $O(\mathrm{D})$ and $\mathsf{Inv}(\mathrm{D})$ to obtain $(\mathrm{M}[\ell], \mathrm{Aux}[\ell])$, and a validity bit $b_{\text{NoCollision}}[\ell]$.
B. Output 0 if $b_{\text{NoCollision}}[\ell] = 0$.

iii. If $|\ell| \neq d-1$, parse $\mathrm{M}[\ell]$ as $(\mathrm{C}[\ell,0], \mathrm{C}[\ell,1])$.

iv. If $|\ell| = d-1$, parse $\mathrm{M}[\ell]$ as $(\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.

5. The game implements the query: for $\ell \in \mathcal{Q}$, $(\mathrm{M}[\ell], \mathrm{Y}[\ell]) \leftarrow \mathsf{SWAP}(\mathrm{M}[\ell], \mathrm{Y}[\ell])$.
6. The game uses the extractor and CM.Com to obtain the commitment:
(a) For each $\ell \in \{0,1\}^{d-1}$, define $\mathrm{M}[\ell] := (\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.
(b) For each $\ell \in \{0,1\}^{<d}$ in the reverse order:

i. If $\ell \in \{\ell_1, \ell_2, \ldots, \ell_i\}$,
A. Apply $\mathsf{CM.Com}_{\text{NoCollision}}$ to $\mathrm{M}[\ell]$, where the oracle access is implemented by $O(\mathrm{D})$, to obtain the commitment and the opening $(\mathrm{C}[\ell], \mathrm{O}[\ell])$ and a validity bit $b_{\text{NoCollision}}[\ell]$.
B. Output 0 if $b_{\text{NoCollision}}[\ell] = 0$.
C. Initialize the register $(\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])$ as all-zero states.

ii. If $\ell \notin \{\ell_1, \ell_2, \ldots, \ell_i\}$,
A. If $\ell \in \{\ell_{i+1}, \ell_{i+2}, \ldots, \ell_T\}$, prepare a state $|\psi_{\mathsf{AltCheck}}\rangle$ on the registers $(\mathrm{O}[\ell], \mathrm{Aux}[\ell])$.
B. Denote the reverse of $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}$ as an algorithm $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}$.
C. Run $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}_{\text{NoCollision}}(\mathrm{Aux}[\ell], \mathrm{M}[\ell])$ by forwarding its queries to $O(\mathrm{D})$ and $\mathsf{Inv}(\mathrm{D})$ to obtain $\mathrm{C}[\ell]$, the ancilla registers $(\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])$, and a validity bit $b_{\text{NoCollision}}[\ell]$.
D. Output 0 if the bit $b_{\text{NoCollision}}[\ell] = 0$.

iii. If $\ell \neq \varepsilon$ and the last bit of ℓ is 0, set $\mathrm{M}[\mathsf{Par}(\ell)]$ as $(\mathrm{C}[\ell], \mathrm{C}[\mathsf{Sib}(\ell)])$.
(c) Set $\mathrm{C} := \mathrm{C}[\varepsilon]$.
7. Set $\mathrm{O} := \bigotimes_{\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}} \mathrm{O}[\ell]$ and $\mathrm{W} := (\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])_{\ell \in \{0,1\}^{<d}}$.
8. The game generates the output: compute $b' \leftarrow D(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}, \mathrm{D}, \mathrm{W})$ and output b'.

$\mathsf{H}_3^{(i)}(A, D)$:

1. The game initializes the database register: $\mathrm{D} \leftarrow |\perp\rangle$.
2. The adversary generates the query set, the commitment, the opening, and the answer register that might be entangled with the environment register: $(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}) \leftarrow A^{O(\mathrm{D}), \mathsf{Inv}(\mathrm{D})}$.
3. Parse O as $\bigotimes_{\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}} \mathrm{O}[\ell]$ and Y as $\bigotimes_{\ell \in \mathcal{Q}} \mathrm{Y}[\ell]$.
4. Initialize a counter $N \leftarrow 2^d - 1 - i$.
5. Check whether the database register D has a collision by measuring it with $\{\Pi_{\mathsf{NoCol}}, I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}\}$. Output 0 if there is a collision.
6. The game uses the extractor to extract the underlying message:
(a) Set $\mathrm{C}[\varepsilon] := \mathrm{C}$.
(b) For $\ell \in \{0,1\}^{<d}$:

i. If $\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$,
A. Apply $\mathsf{CM.Check}_{\text{NoCollision}}$ to $(\mathrm{C}[\ell], \mathrm{O}[\ell])$, where the oracle access is implemented by $O(\mathrm{D})$, to obtain the committed quantum message in $\mathrm{M}[\ell]$ and two validity bits $b[\ell]$ and $b_{\text{NoCollision}}[\ell]$.

B. Output 0 if $b[\ell] \wedge b_{\text{NoCollision}}[\ell] = 0$.

ii. If $\ell \notin \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$ and $N > 0$,

A. Run $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}_{\text{NoCollision}}(\mathrm{C}[\ell])$ by forwarding its queries to $O(\mathrm{D})$ and $\mathsf{Inv}(\mathrm{D})$ to obtain $(\mathrm{M}[\ell], \mathrm{Aux}[\ell])$, and a validity bit $b_{\text{NoCollision}}[\ell]$.

B. Output 0 if $b_{\text{NoCollision}}[\ell] = 0$.

iii. If $\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$ or $N > 0$,

A. If $|\ell| < d-1$, parse $\mathrm{M}[\ell]$ as $(\mathrm{C}[\ell,0], \mathrm{C}[\ell,1])$.

B. If $|\ell| = d-1$, parse $\mathrm{M}[\ell]$ as $(\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.

iv. Decrease the counter by 1: $N \leftarrow N-1$.

7. The game implements the query: for $\ell \in \mathcal{Q}$, $(\mathrm{M}[\ell], \mathrm{Y}[\ell]) \leftarrow \mathsf{SWAP}(\mathrm{M}[\ell], \mathrm{Y}[\ell])$.
8. The game uses the extractor and $\mathsf{CM.Com}$ to obtain the commitment:
   (a) For each $\ell \in \{0,1\}^{<d}$ in the reverse order:
      i. Increase the counter by 1: $N \leftarrow N+1$.
      ii. If $\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$,
         A. If $|\ell| = d-1$, define $\mathrm{M}[\ell] := (\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.
         B. If $|\ell| < d-1$, define $\mathrm{M}[\ell] := (\mathrm{C}[\ell,0], \mathrm{C}[\ell,1])$.
         C. Apply $\mathsf{CM.Com}_{\text{NoCollision}}$ to $\mathrm{M}[\ell]$, where the oracle access is implemented by $O(\mathrm{D})$, to obtain the commitment and the opening $(\mathrm{C}[\ell], \mathrm{O}[\ell])$ and a validity bit $b_{\text{NoCollision}}[\ell]$.
         D. Output 0 if $b_{\text{NoCollision}}[\ell] = 0$.
         E. Initialize the register $(\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])$ as all-zero states.
      iii. If $\ell \notin \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$ and $N > 0$,
         A. If $|\ell| = d-1$, define $\mathrm{M}[\ell] := (\mathrm{M}[\ell,0], \mathrm{M}[\ell,1])$.
         B. If $|\ell| < d-1$, define $\mathrm{M}[\ell] := (\mathrm{C}[\ell,0], \mathrm{C}[\ell,1])$.
         C. Denote the reverse of $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}$ as an algorithm $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}$.
         D. Run $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}_{\text{NoCollision}}(\mathrm{Aux}[\ell], \mathrm{M}[\ell])$ by forwarding its queries to $O(\mathrm{D})$ and $\mathsf{Inv}(\mathrm{D})$ to obtain $\mathrm{C}[\ell]$, the ancilla registers $(\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])$, and a validity bit $b_{\text{NoCollision}}[\ell]$.
         E. Output 0 if the bit $b_{\text{NoCollision}}[\ell] = 0$.
      iv. If $\ell \notin \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$ and $N \leq 0$,
         A. Initialize the register $(\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])$ as all-zero states.
   (b) Set $\mathrm{C} := \mathrm{C}[\varepsilon]$.
9. Check whether the database register D has a collision by measuring it with $\{\Pi_{\mathsf{NoCol}}, I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}\}$. Output 0 if there is a collision.
10. Set $\mathrm{O} := \bigotimes_{\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}} \mathrm{O}[\ell]$ and $\mathrm{W} := (\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])_{\ell \in \{0,1\}^{<d}}$.
11. The game generates the output: compute $b' \leftarrow D(\mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y}, \mathrm{E}, \mathrm{D}, \mathrm{W})$ and output b'.

$\mathsf{H}_4(A, D)$: it does the same as $\mathsf{QSVCSimWorld1Open}^*$ except that instead of letting the adversary produce Q, we fix the query set to be $\mathcal{Q}$.

Let $|\phi_{\mathsf{H}_i}\rangle$ and $|\phi_{\mathsf{H}_i^{(j)}}\rangle$ denote the corresponding subnormalized states at the point immediately before applying D in games H_i and $\mathsf{H}_i^{(j)}$, respectively, for each i, j. We show through several lemmas that the states $|\phi_{\mathsf{QSVCSimWorld1Open}^*}\rangle$ and $|\phi_{\mathsf{QSVCOffExtWorld1Open}^*}\rangle$ are close in ℓ_2 norm.

Claim 7.5. $|\phi_{\mathsf{QSVCOffExtWorld1Open}^*}\rangle = |\phi_{\mathsf{H}_1^{(0)}}\rangle$, $|\phi_{\mathsf{H}_1^{(T)}}\rangle = |\phi_{\mathsf{H}_2^{(0)}}\rangle$, $|\phi_{\mathsf{H}_2^{(T)}}\rangle = |\phi_{\mathsf{H}_3^{(0)}}\rangle$, *and* $|\phi_{\mathsf{H}_3^{(2^d-1)}}\rangle = |\phi_{\mathsf{QSVCSimWorld1Open}^*}\rangle$.

Proof. Since $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCheck}$ does not act on the database register D, by the construction of $\mathcal{E}_{\mathsf{QSTC}}.\mathsf{AltCheck}$,

the game $\mathsf{H}_1^{(0)}(A, D)$ is equivalent to the game $\mathsf{H}_0(A, D)$, which is equivalent to $\mathsf{QSVCOffExtWorld1Open}^*$ when the query set is fixed to $\mathcal{Q}$, and thus

$$|\phi_{\mathsf{QSVCOffExtWorld1Open}^*}\rangle = |\phi_{\mathsf{H}_0}\rangle = |\phi_{\mathsf{H}_1^{(0)}}\rangle \quad .$$

The game $\mathsf{H}_2^{(0)}$ is equivalent to the game $\mathsf{H}_1^{(T)}$ by construction, since in the game $\mathsf{H}_1^{(T)}$, checking whether $\ell \in \{\ell_1, \ell_2, \ldots, \ell_i\}$ is just the same as checking whether $\ell \in \mathsf{Path}(\mathcal{Q}) \setminus \mathcal{Q}$, and in the game $\mathsf{H}_2^{(0)}$, we always take the branch $\ell \notin \{\ell_1, \ell_2, \ldots, \ell_i\}$. Similarly, the game $\mathsf{H}_3^{(0)}$ is equivalent to the game $\mathsf{H}_2^{(T)}$ by construction, and thus

$$|\phi_{\mathsf{H}_1^{(T)}}\rangle = |\phi_{\mathsf{H}_2^{(0)}}\rangle \quad ,$$

and

$$|\phi_{\mathsf{H}_2^{(T)}}\rangle = |\phi_{\mathsf{H}_3^{(0)}}\rangle \quad .$$

By the construction of $\mathsf{H}_3^{(2^d-1)}$, in the procedure, the counter N is always non-positive since the initialization. Moreover, Items 5 and 9 can be absorbed in $\mathsf{QSTC.Query}_{\text{NoCollision}}^{O(\mathrm{D})}(1^\lambda, \mathrm{Q}, \mathrm{C}, \mathrm{O}, \mathrm{Y})$ as the no collision variant of an algorithm would always check whether the database contains a collision before and after applying a unitary. Therefore, the game $\mathsf{H}_3^{(2^d-1)}(A, D)$ is equivalent to the game $\mathsf{H}_4(A, D)$, which is equivalent to the game $\mathsf{QSVCSimWorld1Open}^*$ when the query set is fixed to $\mathcal{Q}$, and thus

$$|\phi_{\mathsf{H}_3^{(2^d-1)}}\rangle = |\phi_{\mathsf{H}_4}\rangle = |\phi_{\mathsf{QSVCSimWorld1Open}^*}\rangle \quad .$$

□

Claim 7.6. *For $i \in [T]$,*

$$\begin{aligned}&\| \, |\phi_{\mathsf{H}_1^{(i-1)}}\rangle - |\phi_{\mathsf{H}_1^{(i)}}\rangle \, \| \\ &\leq (16n+5)(t+8n) \cdot 2^{-(m-3)/2} + 2^{-m/2+5}(t+8n)\sqrt{w_2+4n} + 8 \cdot 2^{-m/2} \quad .\end{aligned}$$

Proof. The only difference between the games $\mathsf{H}_1^{(i-1)}$ and $\mathsf{H}_1^{(i)}$ is in how the location ℓ_i is handled. In $\mathsf{H}_1^{(i)}$, the location ℓ_i is opened using $\mathsf{CM.Check}_{\text{NoCollision}}$, after which the registers $(\mathrm{O}[\ell_i], \mathrm{Aux}[\ell_i])$ are initialized to the state $|\psi_{\mathsf{AltCheck}}\rangle$. In $\mathsf{H}_1^{(i-1)}$, the location ℓ_i is instead extracted using $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}_{\text{NoCollision}}$ and is later checked using $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCheck}$.

By Claim 5.10, replacing $\mathsf{CM.Check}_{\text{NoCollision}}$ with $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}_{\text{NoCollision}}$ followed by $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCheck}$ incurs a loss of at most

$$\frac{8}{\sqrt{2^m}} + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\,|\phi_1\rangle\,\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\,|\phi_2\rangle\,\|$$

where $|\phi_1\rangle = \Pi_{\mathsf{NoCol}} H_{\mathrm{M}}^{\otimes n} O_{\mathrm{MZ_2D}} \Pi_{\mathsf{NoCol}} H_{\mathrm{M}}^{\otimes n} \, |\phi\rangle_{\mathrm{CODE}}$, $|\phi_2\rangle = \Pi_{\mathsf{NoCol}} H_{\mathrm{M}}^{\otimes n} \, |\phi\rangle_{\mathrm{CODE}}$, and the state $|\phi\rangle_{\mathrm{CODE}}$ is the joint state when the game handles the location ℓ_i.

Since A makes at most t queries and the game makes at most $8n$ queries to the two oracles, the states $|\phi_1\rangle$ and $|\phi_2\rangle$ are invariant under Π_{t+8n}.

As a result, by a similar reasoning as Lemma 5.6, since to get $|\phi\rangle_{\mathrm{CODE}}$, we make at most $8n+1$ calls to Π_{NoCol}, and make at most $t+8n$ queries to Inv with total query mass at most w_2+4n,

$$\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\,|\phi_1\rangle\,\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\,|\phi_2\rangle\,\|$$

$$
\begin{aligned}
&= \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\Pi_{t+8n} |\phi_1\rangle\| + \|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}})\Pi_{t+8n} |\phi_2\rangle\| \\
&\leq 3\|\Pi_{t+8n} [\Pi_{\mathsf{NoCol}}, \Pi_{\mathsf{NoHatZero}}] \Pi_{t+8n}\| + 2\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{CODE}}\| \\
&\leq 3(t+8n) \cdot 2^{-(m-3)/2} + 2\|\Pi_{t+8n} [\mathsf{Inv}_{\mathrm{TDW}}, \Pi_{\mathsf{NoHatZero}}] \Pi_{t+8n}\| \sqrt{(t+8n)(w_2+4n)} \\
&+ 2(8n+1)\|\Pi_{t+8n} [\Pi_{\mathsf{NoCol}}, \Pi_{\mathsf{NoHatZero}}] \Pi_{t+8n}\| \\
&\leq (16n+5)(t+8n) \cdot 2^{-(m-3)/2} + 2^{-m/2+5}(t+8n)\sqrt{w_2+4n} \ ,
\end{aligned}
$$

where we notice that only queries to Π_{NoCol} or Inv would contribute error to $\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\phi\rangle_{\mathrm{CODE}}\|$.

Initializing the registers $(\mathrm{O}[\ell_i], \mathrm{Aux}[\ell_i])$ to $|\psi_{\mathsf{AltCheck}}\rangle$ in the game $\mathsf{H}_1^{(i)}$ incurs no additional loss, since conditioned on the validity bit being 1, these registers are also in the state $|\psi_{\mathsf{AltCheck}}\rangle$ in the game $\mathsf{H}_1^{(i-1)}$.

Therefore,

$$
\begin{aligned}
&\| |\phi_{\mathsf{H}_1^{(i-1)}}\rangle - |\phi_{\mathsf{H}_1^{(i)}}\rangle \| \\
&\leq (16n+5)(t+8n) \cdot 2^{-(m-3)/2} + 2^{-m/2+5}(t+8n)\sqrt{w_2+4n} + 8 \cdot 2^{-m/2} \ .
\end{aligned}
$$

□

Claim 7.7. *For $i \in [T]$,*

$$
\begin{aligned}
&\| |\phi_{\mathsf{H}_2^{(i-1)}}\rangle - |\phi_{\mathsf{H}_2^{(i)}}\rangle \| \\
&\leq (24n+8)(t+8n) \cdot 2^{-(m-3)/2} + 2^{-(m-11)/2}(t+8n)\sqrt{w_2+4n} + 4 \cdot 2^{-m/2} \ .
\end{aligned}
$$

Proof. The only difference between the games $\mathsf{H}_2^{(i-1)}$ and $\mathsf{H}_2^{(i)}$ is in how the location ℓ_i is handled after the game implements the query. In $\mathsf{H}_2^{(i)}$, the commitment and the opening of the location ℓ_i are obtained by initializing the ancilla qubits as $|\psi_{\mathsf{Check}}\rangle$ and applying the commitment scheme; the working registers $(\mathrm{W}_1[\ell_i], \mathrm{W}_2[\ell_i])$ are then initialized to all-zero states. In $\mathsf{H}_2^{(i-1)}$, the commitment and the opening of the location ℓ_i are instead obtained by initializing $(\mathrm{O}[\ell_i], \mathrm{Aux}[\ell_i])$ as the state $|\psi_{\mathsf{AltCheck}}\rangle$, then applying $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}_{\text{NoCollision}}$ while keeping the working registers $(\mathrm{W}_1[\ell_i], \mathrm{W}_2[\ell_i])$. This difference is exactly captured by

$$
\|(V_{\mathsf{Check}}^{\dagger} |\psi_{\mathsf{Check}}\rangle |0\rangle_{\mathrm{W}_1\mathrm{W}_2} - V_{\mathsf{ExtMsg}}^{\dagger} |\psi_{\mathsf{AltCheck}}\rangle) |\psi\rangle_{\mathrm{MED}}\| \ ,
$$

where $|\psi\rangle_{\mathrm{MED}}$ is the joint state when the game handles the location ℓ_i after the query is implemented, $V_{\mathsf{Check}}^{\dagger}$ is exactly what $\mathsf{CM}.\mathsf{Com}_{\text{NoCollision}}$ does, and $V_{\mathsf{ExtMsg}}^{\dagger}$ is exactly what $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}_{\text{NoCollision}}$ does.

By Claim 5.10,

$$
\begin{aligned}
&\|(V_{\mathsf{Check}}^{\dagger} |\psi_{\mathsf{Check}}\rangle |0\rangle_{\mathrm{W}_1\mathrm{W}_2} - V_{\mathsf{ExtMsg}}^{\dagger} |\psi_{\mathsf{AltCheck}}\rangle) |\psi\rangle_{\mathrm{MED}}\| \\
&\leq \frac{4}{\sqrt{2^m}} + \sqrt{2}\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\psi_1\rangle\| + \sqrt{2}\|(I_{\mathrm{D}} - \Pi_{\mathsf{NoHatZero}}) |\psi_2\rangle\| \ ,
\end{aligned}
$$

where $|\psi_1\rangle \coloneqq H_{\mathrm{M}}^{\otimes n}\Pi_{\mathsf{NoCol}}O_{\mathrm{MZ}_1\mathrm{D}}\Pi_{\mathsf{NoCol}} |\psi\rangle_{\mathrm{MED}} |\psi_{\mathsf{Check}}\rangle$ and $|\psi_2\rangle \coloneqq \Pi_{\mathsf{NoCol}} |\psi\rangle_{\mathrm{MED}}$.

Therefore, by the same reasoning as Claim 7.6,

$$
\begin{aligned}
&\| |\phi_{\mathsf{H}_2^{(i-1)}}\rangle - |\phi_{\mathsf{H}_2^{(i)}}\rangle \| \\
&\leq (24n+8)(t+8n) \cdot 2^{-(m-3)/2} + 2^{-(m-11)/2}(t+8n)\sqrt{w_2+4n} + 4 \cdot 2^{-m/2} \ .
\end{aligned}
$$

□

Claim 7.8. *For* $i \in [2^d - 1]$, $\| \, |\phi_{\mathsf{H}_3^{(i-1)}}\rangle - |\phi_{\mathsf{H}_3^{(i)}}\rangle \, \| \leq 2^{-m/2} \left((16n+8)\sqrt{6(t+8n)} + 64\sqrt{2}n \right)$.

Proof. The difference between the games $\mathsf{H}_3^{(i-1)}$ and $\mathsf{H}_3^{(i)}$ is that $\mathsf{H}_3^{(i-1)}$ does an extra $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}_{\text{NoCollision}}(\mathrm{C}[\ell])$ before the game implements the query, and $\mathsf{H}_3^{(i-1)}$ does an extra $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}_{\text{NoCollision}}(\mathrm{Aux}[\ell], \mathrm{M}[\ell])$ after the game implements the query, while $\mathsf{H}_3^{(i)}$ initializes $(\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])$ as all-zeros in the computational basis.

Notice that $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}$ does exactly the inverse of $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}$. The claim follows from the following two facts:

- Replace this extra $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}_{\text{NoCollision}}(\mathrm{C}[\ell])$ with $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}(\mathrm{C}[\ell])$, and $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}_{\text{NoCollision}}(\mathrm{Aux}[\ell], \mathrm{M}[\ell])$ with $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}(\mathrm{Aux}[\ell], \mathrm{M}[\ell])$ in game $\mathsf{H}_3^{(i-1)}$ to form a game $\mathsf{H}'^{(i-1)}_3$. As the database size is always bounded by $t+8n$ and in each game, we always check whether the database has a collision at the beginning and at the end, this step would incur an error

$$\| \, |\phi_{\mathsf{H}_3^{(i-1)}}\rangle - |\phi_{\mathsf{H}'^{(i-1)}_3}\rangle \, \| \leq 4 \| (I_{\mathrm{D}} - \Pi_{\mathsf{NoCol}}) \, (\Pi_{t+8n} O \Pi_{t+8n}) \, \Pi_{\mathsf{NoCol}} \| \leq 4 \cdot \sqrt{6(t+8n)} \cdot 2^{-m/2} \ .$$

- Moving the operation $\mathcal{E}_{\mathsf{CM}}.\mathsf{AltCommit}(\mathrm{Aux}[\ell], \mathrm{M}[\ell])$ in game $\mathsf{H}'^{(i-1)}_3$ to just after the operation $\mathcal{E}_{\mathsf{CM}}.\mathsf{ExtMsg}(\mathrm{C}[\ell])$ results in exactly the same game as $\mathsf{H}_3^{(i)}$, where $(\mathrm{W}_1[\ell], \mathrm{W}_2[\ell])$ is initialized as all-zero states. This step would incur an error

$$\begin{aligned}
&\| \, |\phi_{\mathsf{H}_3^{(i)}}\rangle - |\phi_{\mathsf{H}'^{(i-1)}_3}\rangle \, \| \\
&\leq 8T \cdot \| \, [\mathsf{Inv}, O] \, \| + (8T+2) \cdot \| \, [\Pi_{\mathsf{NoCol}}, \Pi_{t+8n} O \Pi_{t+8n}] \, \| \\
&\leq 8T \cdot 2^{-(m-7)/2} + (16T+4) \cdot \sqrt{6(t+8n)} \cdot 2^{-m/2} \ .
\end{aligned}$$

Notice that $T \leq n$. A triangle inequality concludes the proof. □

Combining Claims 7.5 to 7.8, we obtain

$$\begin{aligned}
&\| \, |\phi_{\mathsf{QSVCSimWorld1Open}^*}\rangle - |\phi_{\mathsf{QSVCOffExtWorld1Open}^*}\rangle \, \|^2 \\
&\leq \Big(n(40n+13)(t+8n) \cdot 2^{-(m-3)/2} + 2^{-m/2+7} n(t+8n)\sqrt{w_2+4n} + 12n \cdot 2^{-m/2} \\
&+ 2^{-m/2} n(16n+8)\sqrt{6(t+8n)} + 2^{-(m-13)/2} n^2 \Big)^2 \\
&\leq 2^{-m+2} \left(8n^2(40n+13)^2(t+8n)^2 + 2^{14} n^2 (t+8n)^2 (w_2+4n) + 6n^2(16n+8)^2(t+8n) + 2^{15} n^4 \right) \\
&\leq 2^{-m+20} n^2 (t+8n)^2 \left(n^2 + w_2 \right) \\
&\leq \xi_{\mathsf{1Open}}(m, n, t, w_1, w_2) \ ,
\end{aligned}$$

where we use the triangle inequality in the second line, and the Cauchy–Schwarz inequality in the third line. □

8 Quantum interactive arguments based on QIOPs

We show how to obtain a quantum-communication succinct interactive argument from public-query QIOPs.

Theorem 8.1. *Let* QIOP *be a quantum interactive oracle proof for relation* R *with public-query soundness* s_{pq}*. Let* QSVC *be an* $(\xi^{(1)}_{\mathsf{Ext}}, \xi^{(2)}_{\mathsf{Ext}}, \xi^{(3)}_{\mathsf{Ext}})$*-extractable quantum state vector commitment scheme (Definition 6.3). Let* m *be the output length of the random oracle. Then the protocol* $(\mathcal{P}, \mathcal{V}) = \mathsf{IBCS}[\mathsf{QIOP}, \mathsf{QSVC}]$ *in Construction 8.2 is a quantum-communication interactive argument for relation* R *with soundness*

$$\xi^{(1)}_{\mathsf{Ext}}(t + \mathsf{vq}, m) + 2s_{\mathsf{pq}}(\nu) + 2\mathsf{k} \cdot \sum_{i \in [\mathsf{k}]} \xi^{(3)}_{\mathsf{Ext}}(t, \mathsf{d}, t, 0, m, \mathsf{l}_i) \ ,$$

where vq *is the number of queries that the argument verifier* $\mathcal{V}$ *makes to the random oracle* RO*. Furthermore, when* QSVC *is instantiated with the construction* QSTC *in Construction 7.1,* $\mathsf{vq} = O(\mathsf{q} \log \mathsf{l}_{\max})$.

8.1 Our transformation

We describe below our construction of the quantum-communication succinct interactive argument in the random oracle model, which we denote $(\mathcal{P}, \mathcal{V}) \coloneqq \mathsf{IBCS}[\mathsf{QIOP}, \mathsf{QSVC}]$. This is a quantum analogue of the IBCS transformation for IOPs [CDGS23].

Construction 8.2. Let $(\mathbf{P}, \mathbf{V}) \coloneqq \mathsf{QIOP}$. The argument prover $\mathcal{P}$ receives an instance $\mathbb{x}$ and a quantum witness $\boldsymbol{\rho}$, and the argument verifier receives the same instance $\mathbb{x}$. $\mathcal{P}$ and $\mathcal{V}$ both have access to a random function RO sampled from $\mathcal{U}(m)$. We do domain separation on RO to obtain k random functions $\{\mathsf{RO}_i\}_{i \in [\mathsf{k}]}$, where $\mathsf{RO}_i(x) = \mathsf{RO}(i, x)$ for $x \in \{0,1\}^*$ and $i \in [\mathsf{k}]$. Then $\mathcal{P}$ and $\mathcal{V}$ interact as follows.

1. $\mathcal{P}$'s initialization: Initialize P_1 to the quantum witness $\boldsymbol{\rho}$, padded with all-zero states, let Mv_0 be an empty register, and set $I_1 \coloneqq \{1\}, J_1 \coloneqq \emptyset$.
2. $\mathcal{V}$'s initialization: Initialize V_1 to all-zero states and set $I_1 \coloneqq \{1\}, J_1 \coloneqq \emptyset$.
3. For $i \in [\mathsf{k}]$:
   (a) $\mathcal{P}$'s i-th commitment.
      i. Compute the returned QIOP proofs: for $j \in J_i$, $\mathrm{Mp}_j \leftarrow \mathsf{QSVC.Recover}^{\mathsf{RO}_j}(1^\lambda, \mathrm{C}_j, \mathrm{T}_j)$.
      ii. Compute the i-th QIOP state: $(\mathrm{Mp}_i, \mathrm{P}_{i+1}) \leftarrow \mathbf{P}(\mathbb{x}, \mathrm{Mv}_{i-1}, \mathrm{P}_i, (\mathrm{Mp}_j)_{j \in J_i})$.
      iii. Parse Mp_i as l_i subregisters: $(\mathrm{Mp}_i[j])_{j \in [\mathsf{l}_i]} \coloneqq \mathrm{Mp}_i$.
      iv. Compute a QVC commitment to the QIOP state where the message length is set to l_i: $(\mathrm{C}_i, \mathrm{T}_i) \leftarrow \mathsf{QSVC.Com}^{\mathsf{RO}_i}(\mathrm{Mp}_i)$.
      v. Send the QVC commitment C_i to $\mathcal{V}$.
   (b) $\mathcal{P}$ answers the queries of $\mathbf{V}$ that the argument verifier $\mathcal{V}$ simulates.
      i. $\mathcal{V}$ initializes $(\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}$ as the all-zero states and sets $\mathrm{V}_i^{(0)} \coloneqq \mathrm{V}_i$.
      ii. For $q \in [\mathsf{d}_i]$:
         A. $\mathcal{V}$ computes the query locations: $((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, \mathrm{V}_i^{(q)}) \leftarrow \mathbf{V}_i^{(q-1)}(\mathbb{x}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, \mathrm{V}_i^{(q-1)})$.
         B. $\mathcal{V}$ sends $(\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}$ to $\mathcal{P}$.
         C. $\mathcal{P}$ computes an opening: $((\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{O}_{i'}, \mathrm{R}_{i'})_{i' \in I_i}) \leftarrow \mathsf{QSVC.Open}^{(\mathsf{RO}_{i'})_{i' \in I_i}}(1^\lambda, (\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{T}_{i'})_{i' \in I_i})$.
         D. $\mathcal{P}$ sends $((\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{O}_{i'})_{i' \in I_i})$ to $\mathcal{V}$.
         E. $\mathcal{V}$ implements the query for $\mathbf{V}$: $(b_{i,q}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{C}_{i'}, \mathrm{O}_{i'})_{i' \in I_i}) \leftarrow \mathsf{QSVC.Query}^{(\mathsf{RO}_{i'})_{i' \in I_i}}(1^\lambda, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{C}_{i'}, \mathrm{O}_{i'})_{i' \in I_i})$.

F. $\mathcal{V}$ sends $((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}, (\mathrm{O}_{i'})_{i'\in I_i})$ to $\mathcal{P}$.

G. $\mathcal{P}$ erases the information about the query locations:
$((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}, (\mathrm{T}_{i'})_{i'\in I_i}) \leftarrow \mathsf{QSVC.Update}^{(\mathsf{RO}_{i'})_{i'\in I_i}}(1^\lambda, (\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}, (\mathrm{O}_{i'}, \mathrm{R}_{i'})_{i'\in I_i})$.

H. $\mathcal{P}$ sends $(\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}$ to $\mathcal{V}$.

(c) If $i \neq \mathsf{k}$, $\mathcal{V}$ generates the i-th message and returns some of the previous proofs.

i. $\mathcal{V}$ computes the next message and the indices of proofs to be returned:
$(J_{i+1}, \mathrm{Mv}_i, \mathrm{V}_{i+1}) \leftarrow \mathbf{V}_i^{(\mathsf{d}_i)}(\mathbb{x}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}_i]}, \mathrm{V}_i^{(\mathsf{d}_i)})$.

ii. $\mathcal{V}$ sends $(J_{i+1}, \mathrm{Mv}_i, (\mathrm{C}_{i'})_{i'\in J_{i+1}})$ to $\mathcal{P}$.

iii. Both $\mathcal{P}$ and $\mathcal{V}$ set $I_{i+1} \coloneqq (I_i \cup \{i+1\}) \setminus J_{i+1}$.

(d) If $i = \mathsf{k}$, $\mathcal{V}$ decides whether to accept:

i. $\mathcal{V}$ computes whether $\mathbf{V}$ accepts: $b_{\mathsf{QIOP}} \leftarrow \mathbf{V}_\mathsf{k}^{(\mathsf{d}_\mathsf{k})}(\mathbb{x}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota\in[\mathsf{w}_\mathsf{k}]}, \mathrm{V}_\mathsf{k}^{(\mathsf{d}_\mathsf{k})})$.

ii. Compute $b_{\mathsf{QSVC}} \coloneqq \wedge_{i\in[\mathsf{k}]}\left(\wedge_{q\in[\mathsf{d}_i]} b_{i,q}\right)$.

iii. Output $b_{\mathsf{QSVC}} \wedge b_{\mathsf{QIOP}}$.

Here in the construction, we overload QSVC.Open, QSVC.Query, and QSVC.Update to handle multiple proofs coherently, using their respective random oracles, and combining validity bits by conjunction.

When QSVC satisfies perfect completeness, the protocol $(\mathcal{P}, \mathcal{V}) = \mathsf{IBCS}[\mathsf{QIOP}, \mathsf{QSVC}]$ in Construction 8.2 has the same completeness guarantee as the quantum interactive oracle proof QIOP.

Moreover, when we instantiate the quantum state vector commitment scheme with QSTC from Construction 7.1, the efficiency measures of the interactive argument $(\mathcal{P}, \mathcal{V}) \coloneqq \mathsf{IBCS}[\mathsf{QIOP}, \mathsf{QSTC}]$ satisfy the following:

- The *round complexity* $k_{\mathsf{IBCS}} = 4\sum_{i\in[\mathsf{k}]} \mathsf{d}_i + 2\mathsf{k} - 1$.
- The *prover-to-verifier communication* $\mathsf{pc}_{\mathsf{IBCS}} = O(m\mathsf{k} + \sum_{i\in[\mathsf{k}]} \mathsf{d}_i \cdot m \cdot \mathsf{w}_i \cdot \log \mathsf{l}_{\max}) = O(\lambda\mathsf{k} + \lambda\mathsf{q}\log \mathsf{l}_{\max})$.
- The *verifier-to-prover communication* $\mathsf{vc}_{\mathsf{IBCS}} = \mathsf{vc} + \mathsf{pc}_{\mathsf{IBCS}} = O(\mathsf{vc} + \lambda\mathsf{k} + \lambda\mathsf{q}\log \mathsf{l}_{\max})$.
- The *verifier's query complexity* to the i-th random oracle $\mathsf{vq}_i = O(\mathsf{d}_i\mathsf{w}_i \log \mathsf{l}_i)$ and the *verifier's query complexity* $\mathsf{vq} = O(\mathsf{q}\log \mathsf{l}_{\max})$.

Therefore, when we instantiate the quantum state vector commitment scheme with QSTC, the resulting protocol $\mathsf{IBCS}[\mathsf{QIOP}, \mathsf{QSTC}]$ is a quantum-communication succinct interactive argument in the quantum random oracle model as long as the underlying QIOP is efficient.

Corollary 8.3. *Let* QIOP *be a quantum interactive oracle proof for relation R with public-query soundness s_{pq}. Let* QSTC *be the quantum state vector commitment scheme in Construction 7.1. Then the protocol $(\mathcal{P}, \mathcal{V}) = \mathsf{IBCS}[\mathsf{QIOP}, \mathsf{QSTC}]$ is a quantum-communication interactive argument for relation R with soundness*

$$2s_{\mathsf{pq}}(\nu) + 2\mathsf{k} \cdot \sum_{i\in[\mathsf{k}]} \xi_{\mathsf{Ext}}^{(3)}(t, \mathsf{d}, t, 0, \lambda, \mathsf{l}_i) \ ,$$

where $\xi_{\mathsf{Ext}}^{(3)}(t, p, w_1, w_2, m, n) \coloneqq 2^{-m+22}p^2n^2(t+8np)^2\left(n^2 + w_1 + w_2 + p + 1\right)$.

Furthermore, if QIOP *satisfies that* $s_{\mathsf{pq}}(\nu) = \mathsf{negl}(\nu)$, $\sum_{i\in[\mathsf{k}]} \mathsf{d}_i(\nu) = O(\mathsf{poly}(\log\nu))$, $\mathsf{q}(\nu) = O(\mathsf{poly}(\log\nu))$, $\mathsf{l}_i(\nu) = O(\mathsf{poly}(\nu))$, $\mathsf{k}(\nu) = O(\mathsf{poly}(\log\nu))$, *and* $\mathsf{vc}(\nu) = O(\mathsf{poly}(\log\nu))$, *then* $(\mathcal{P}, \mathcal{V}) = \mathsf{IBCS}[\mathsf{QIOP}, \mathsf{QSTC}]$ *is a quantum-communication succinct interactive argument for the same relation R with round complexity* $O(\mathsf{poly}(\log\nu))$, *total communication complexity* $O(\mathsf{poly}(\log\nu, \lambda))$, *and soundness error* $\mathsf{negl}(\nu) + 2^{-\lambda}t^3\mathsf{poly}(\nu)$ *against t-query quantum adversaries. $(\mathcal{P}, \mathcal{V})$ also has the same completeness guarantee as the quantum interactive oracle proof* QIOP.

Proof. Corollary 8.3 follows from Theorem 8.1, Theorem 7.3, and the efficiency of the quantum-state vector commitment scheme QSTC from Construction 7.1. □

8.2 The malicious QIOP prover

We construct a malicious QIOP prover $\widetilde{\mathbf{P}}$ to attack the public-query soundness of the underlying quantum interactive oracle proof system QIOP based on a malicious prover $\widetilde{\mathcal{P}}$ for the argument $(\mathcal{P}, \mathcal{V}) = \mathsf{IBCS}[\mathsf{QIOP}, \mathsf{QSVC}]$. Here $\widetilde{\mathcal{P}}$ only has oracle access to k random functions $\{\mathsf{RO}_i\}_{i\in[\mathsf{k}]}$.

Construction 8.4. By Definition 6.3, QSVC has a quantum extractor $\mathcal{E}$ with query access to the quantum simulator U_{Sim} and U_{Ext}. We construct $\widetilde{\mathbf{P}}(\widetilde{\mathcal{P}})$ as follows.

$\widetilde{\mathbf{P}}(\widetilde{\mathcal{P}})$:

(a) Initialize the internal state register S_i for the simulator U_{Sim} for the random oracle in QSVC: for $i \in [\mathsf{k}]$, $\mathrm{S}_i \leftarrow |\perp\rangle$.

(b) Set $I_1 \coloneqq \{1\}, J_1 \coloneqq \emptyset$.

(c) Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} and the internal state registers $(\mathrm{S}_i)_{i\in[\mathsf{k}]}$ until $\widetilde{\mathcal{P}}$ outputs an instance $\mathbb{x}$. Send the instance $\mathbb{x}$ to the QIOP verifier $\mathbf{V}$.

(d) For $i \in [\mathsf{k}]$:

i. Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} and the internal state registers $(\mathrm{S}_i)_{i\in[\mathsf{k}]}$ until $\widetilde{\mathcal{P}}$ outputs a commitment C_i.

ii. Run the extractor $\mathcal{E}$ on C_i to get the i-th prover's message: $(\mathrm{Mp}_i, \mathrm{Aux}_i) \leftarrow \mathcal{E}.\mathsf{ExtMsg}^{U_{\mathsf{Sim}}(\mathrm{S}_i), U_{\mathsf{Ext}}(\mathrm{S}_i)}(\mathrm{C}_i)$.

iii. Send the i-th prover's message Mp_i to the trusted third party $\mathbf{O}$ that implements the queries to the prover's messages for $\mathbf{V}$.

iv. For $q \in [\mathsf{d}_i]$:

A. On receiving the leaked query locations $(\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}$ from $\mathbf{V}$, send $(\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}$ to $\widetilde{\mathcal{P}}$.

B. Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} and the internal state registers $(\mathrm{S}_i)_{i\in[\mathsf{k}]}$ until $\widetilde{\mathcal{P}}$ outputs $((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}, (\mathrm{O}_{i'})_{i'\in I_i})$.

C. Check the opening: $(b_i, (\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}, (\mathrm{O}_{i'}, \mathrm{Aux}_{i'})_{i'\in I_i}) \leftarrow \mathcal{E}.\mathsf{AltCheck}((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}, (\mathrm{O}_{i'}, \mathrm{Aux}_{i'})_{i'\in I_i})$.

D. Abort if $b_i = 0$.

E. Send $(\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}$ to the QIOP verifier $\mathbf{V}$, who asks the trusted third party $\mathbf{O}$ to implement the query with $(\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}$.

F. On receiving $(\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}$ from the QIOP verifier $\mathbf{V}$, send $((\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}, (\mathrm{O}_{i'})_{i'\in I_i})$ to $\widetilde{\mathcal{P}}$.

G. Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} and the internal state registers $(\mathrm{S}_i)_{i\in[\mathsf{k}]}$ until $\widetilde{\mathcal{P}}$ outputs $(\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}$.

H. Send $(\mathrm{Loc}_\iota)_{\iota\in[\mathsf{w}_i]}$ to the QIOP verifier $\mathbf{V}$.

v. If $i \neq \mathsf{k}$:

A. On receiving the returned index set and the i-th verifier's message (J_{i+1}, Mv_i) from $\mathbf{V}$, and the corresponding prover's messages $(\mathrm{Mp}_{i'})_{i'\in J_{i+1}}$ from the trusted third party $\mathbf{O}$, compute the corresponding commitments in J_{i+1} with the extractor $\mathcal{E}$: for $i' \in J_{i+1}$, $\mathrm{C}_{i'} \leftarrow \mathcal{E}.\mathsf{AltCommit}^{U_{\mathsf{Sim}}(\mathrm{S}_{i'}), U_{\mathsf{Ext}}(\mathrm{S}_{i'})}(\mathrm{Aux}_{i'}, \mathrm{Mp}_{i'})$.

B. Send $(J_{i+1}, \mathrm{Mv}_i, (\mathrm{C}_{i'})_{i'\in J_{i+1}})$ to $\widetilde{\mathcal{P}}$.

C. Set $I_{i+1} \coloneqq (I_i \cup \{i+1\}) \setminus J_{i+1}$.

8.3 Proof of Theorem 8.1

We first show that in the public-query soundness game, the malicious QIOP prover $\widetilde{\mathbf{P}}(\widetilde{\mathcal{P}})$ in Construction 8.4 has a similar winning probability as the malicious argument prover $\widetilde{\mathcal{P}}$.

Lemma 8.5. *For every integer ν, t, m, and a t-query argument adversary $\widetilde{\mathcal{P}}$,*

$$\Pr\left[\begin{array}{l|l} |\mathbb{x}| \le \nu & \mathsf{RO} \leftarrow \mathcal{U}(m) \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathcal{P}}^{\mathsf{RO}} \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathcal{P}}^{\mathsf{RO}}, \mathcal{V}^{\mathsf{RO}}(\mathbb{x})\rangle \end{array}\right]$$
$$\le 2 \cdot \Pr\left[\begin{array}{l|l} |\mathbb{x}| \le \nu & \mathbb{x} \leftarrow \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}) \\ \wedge\, \mathbb{x} \notin L(R) & b \leftarrow \langle \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}), \mathbf{V}(\mathbb{x})\rangle_{\mathsf{pq}} \\ \wedge\, b = 1 \end{array}\right] + \xi^{(1)}_{\mathsf{Ext}}(t + \mathsf{vq}, m) + 2\mathsf{k} \cdot \sum_{i\in[\mathsf{k}]} \xi^{(3)}_{\mathsf{Ext}}(t, \mathsf{d}, t, 0, m, \mathsf{l}_i) \ .$$

We first show how Lemma 8.5 implies Theorem 8.1.

Proof of Theorem 8.1. Theorem 8.1 follows from the definition of public-query soundness (Definition 3.2). Specifically, if QIOP has public-query soundness s_{pq}, then

$$\Pr\left[\begin{array}{l|l} |\mathbb{x}| \le \nu & \mathbb{x} \leftarrow \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}) \\ \wedge\, \mathbb{x} \notin L(R) & b \leftarrow \langle \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}), \mathbf{V}(\mathbb{x})\rangle_{\mathsf{pq}} \\ \wedge\, b = 1 \end{array}\right] \le s_{\mathsf{pq}}(\nu) \ .$$

□

Next we prove Lemma 8.5 via two claims.

The first claim replaces the random oracle with the simulator U_{Sim}.

Claim 8.6. *For every integer ν, t, m, and a t-query argument adversary $\widetilde{\mathcal{P}}$,*

$$\Pr\left[\begin{array}{l|l} |\mathbb{x}| \le \nu & \mathsf{RO} \leftarrow \mathcal{U}(m) \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathcal{P}}^{\mathsf{RO}} \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathcal{P}}^{\mathsf{RO}}, \mathcal{V}^{\mathsf{RO}}(\mathbb{x})\rangle \end{array}\right]$$
$$\le \Pr\left[\begin{array}{l|l} |\mathbb{x}| \le \nu & \mathrm{S} \leftarrow |\bot\rangle \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathcal{P}}^{U_{\mathsf{Sim}}(\mathrm{S})} \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathcal{P}}^{U_{\mathsf{Sim}}(\mathrm{S})}, \mathcal{V}^{U_{\mathsf{Sim}}(\mathrm{S})}(\mathbb{x})\rangle \end{array}\right] + \xi^{(1)}_{\mathsf{Ext}}(t + \mathsf{vq}, m) \ .$$

Proof. This follows from the definition of quantum simulator (Definition 4.4), and that the interaction between $\widetilde{\mathcal{P}}$ and $\mathcal{V}$, and the final check on $\mathbb{x}$ can be combined into a single distinguisher with $t + \mathsf{vq}$ queries to the random oracle. □

The second claim replaces all the queries implemented by $\mathsf{QSVC.Query}$ to the i-th prover's message during the interaction between the malicious argument prover $\widetilde{\mathcal{P}}$ and the argument verifier $\mathcal{V}$ with queries implemented by extractors one by one for $i \in [\mathsf{k}]$. We first define the hybrid games.

$\mathsf{H}_{i^*}(\widetilde{\mathcal{P}})$:

1. Initialize the internal state register S_i: for $i \in [\mathsf{k}]$, $\mathrm{S}_i \leftarrow |\bot\rangle$.
2. Initialize the internal state of $\mathcal{V}$, the register V_1, to all-zero states, set Mv_0 to be the empty register, and set $I_1 \coloneqq \{1\}, J_1 \coloneqq \emptyset$.
3. Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} and the internal state registers $(\mathrm{S}_i)_{i\in[\mathsf{k}]}$ until $\widetilde{\mathcal{P}}$ outputs an instance $\mathbb{x}$.
4. Simulate the interaction of $\widetilde{\mathcal{P}}$ and $\mathcal{V}$ for the first i^* commitments of $\widetilde{\mathcal{P}}$ with QSVC.
   For $i \in [i^*]$:

(a) Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} on the input $(J_i, \mathrm{Mv}_{i-1}, (\mathrm{C}_{i'})_{i' \in J_i})$ until $\widetilde{\mathcal{P}}$ outputs a commitment C_i.
(b) Initialize $\mathcal{V}$'s registers $(\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}$ as the all-zero states and set $\mathrm{V}_i^{(0)} := \mathrm{V}_i$.
(c) For $q \in [\mathsf{d}_i]$:
   i. Simulate $\mathcal{V}$ to get the query locations: $((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, \mathrm{V}_i^{(q)}) \leftarrow \mathbf{V}_i^{(q-1)}(\mathbb{x}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, \mathrm{V}_i^{(q-1)})$.
   ii. Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} on the leaked query locations $(\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}$ until $\widetilde{\mathcal{P}}$ outputs $((\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{O}_{i'})_{i' \in I_i})$.
   iii. Implement the access to the underlying message for $\mathcal{V}$: $(b_{i,q}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{C}_{i'}, \mathrm{O}_{i'})_{i' \in I_i}) \leftarrow \mathsf{QSVC.Query}^{(U_{\mathsf{Sim}}(\mathrm{S}_{i'}))_{i' \in I_i}}(1^\lambda, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{C}_{i'}, \mathrm{O}_{i'})_{i' \in I_i})$.
   iv. Output 0 and abort if $b_{i,q} = 0$.
   v. Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} on the registers $((\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{O}_{i'})_{i' \in I_i})$ until $\widetilde{\mathcal{P}}$ outputs the updated $(\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}$.
(d) If $i \neq \mathsf{k}$:
   i. Compute the i-th $\mathcal{V}$'s message and the index set for the proofs to be returned: $(J_{i+1}, \mathrm{Mv}_i, \mathrm{V}_{i+1}) \leftarrow \mathbf{V}_i^{(\mathsf{d}_i)}(\mathbb{x}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, \mathrm{V}_i^{(\mathsf{d}_i)})$.
   ii. Set $I_{i+1} := (I_i \cup \{i+1\}) \setminus J_{i+1}$.
(e) If $i = \mathsf{k}$, simulate $\mathcal{V}$ and check whether $\widetilde{\mathcal{P}}$ wins:
   i. Compute $b_{\mathsf{QIOP}} \leftarrow \mathbf{V}_{\mathsf{k}}^{(\mathsf{d}_\mathsf{k})}(\mathbb{x}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_\mathsf{k}]}, \mathrm{V}_{\mathsf{k}}^{(\mathsf{d}_\mathsf{k})})$.
   ii. Output 1 if $b_{\mathsf{QIOP}} = 1$, $|\mathbb{x}| \leq \nu$, and $\mathbb{x} \notin L(R)$; output 0 otherwise.

5. Simulate the interaction of $\widetilde{\mathbf{P}}(\widetilde{\mathcal{P}})$ and $\mathbf{V}$ for the remaining $\mathsf{k} - i^*$ commitments of $\widetilde{\mathcal{P}}$ with the extractor. For $i \in \{i^*+1, i^*+2, \cdots, \mathsf{k}\}$:
(a) Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} on the input $(J_i, \mathrm{Mv}_{i-1}, (\mathrm{C}_{i'})_{i' \in J_i})$ until $\widetilde{\mathcal{P}}$ outputs a commitment C_i.
(b) Run the extractor $\mathcal{E}$ on C_i to get the i-th prover's message: $(\mathrm{Mp}_i, \mathrm{Aux}_i) \leftarrow \mathcal{E}.\mathsf{ExtMsg}^{U_{\mathsf{Sim}}(\mathrm{S}_i), U_{\mathsf{Ext}}(\mathrm{S}_i)}(\mathrm{C}_i)$.
(c) Initialize $\mathcal{V}$'s registers $(\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}$ as the all-zero states and set $\mathrm{V}_i^{(0)} := \mathrm{V}_i$.
(d) For $q \in [\mathsf{d}_i]$:
   i. Simulate $\mathcal{V}$ to get the query locations: $((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, \mathrm{V}_i^{(q)}) \leftarrow \mathbf{V}_i^{(q-1)}(\mathbb{x}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, \mathrm{V}_i^{(q-1)})$.
   ii. Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} on the leaked query locations $(\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}$ until $\widetilde{\mathcal{P}}$ outputs $((\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{O}_{i'})_{i' \in I_i})$.
   iii. Simulate the trusted third party $\mathbf{O}$ that implements the access to prover's message registers: for $i' \in I_i \cap \{i^*+1, \cdots, \mathsf{k}\}$, $((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, \mathrm{Mp}_{i'}) \leftarrow \mathbf{O}((\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, \mathrm{Mp}_{i'})$.
   iv. Check the opening: for $i' \in I_i \cap \{i^*+1, \cdots, \mathsf{k}\}$, $(b_{i,q,i'}, (\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{O}_{i'}, \mathrm{Aux}_{i'})) \leftarrow \mathcal{E}.\mathsf{AltCheck}((\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{O}_{i'}, \mathrm{Aux}_{i'}))$.
   v. Output 0 and abort if $\wedge_{i' \in I_i \cap \{i^*+1, \cdots, \mathsf{k}\}} b_{i,q,i'} = 0$.
   vi. Implement the access to the first i^* messages for $\mathcal{V}$: $(b_{i,q}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{C}_{i'}, \mathrm{O}_{i'})_{i' \in I_i \cap [i^*]}) \leftarrow \mathsf{QSVC.Query}^{(U_{\mathsf{Sim}}(\mathrm{S}_{i'}))_{i' \in I_i \cap [i^*]}}(1^\lambda, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{C}_{i'}, \mathrm{O}_{i'})_{i' \in I_i \cap [i^*]})$.
   vii. Output 0 and abort if $b_{i,q} = 0$.
   viii. Simulate $\widetilde{\mathcal{P}}$ by answering the random oracle queries with U_{Sim} on the registers $((\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}, (\mathrm{O}_{i'})_{i' \in I_i})$ until $\widetilde{\mathcal{P}}$ outputs the updated $(\mathrm{Loc}_\iota)_{\iota \in [\mathsf{w}_i]}$.
(e) If $i \neq \mathsf{k}$:
   i. Compute the i-th $\mathcal{V}$'s message and the index set for the proofs to be returned: $(J_{i+1}, \mathrm{Mv}_i, \mathrm{V}_{i+1}) \leftarrow \mathbf{V}_i^{(\mathsf{d}_i)}(\mathbb{x}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w}_i]}, \mathrm{V}_i^{(\mathsf{d}_i)})$.
   ii. Compute the corresponding commitments in J_{i+1} with the extractor $\mathcal{E}$:

for $i' \in J_{i+1} \cap \{i^*+1, \cdots, \mathsf{k}\}$, $\mathrm{C}_{i'} \leftarrow \mathcal{E}.\mathsf{AltCommit}^{U_{\mathsf{Sim}}(\mathrm{S}_{i'}), U_{\mathsf{Ext}}(\mathrm{S}_{i'})}(\mathrm{Aux}_{i'}, \mathrm{Mp}_{i'})$.

iii. Set $I_{i+1} \coloneqq (I_i \cup \{i+1\}) \setminus J_{i+1}$.

(f) If $i = \mathsf{k}$, simulate $\mathcal{V}$ and check whether $\widetilde{\mathcal{P}}$ wins:

i. Compute $b_{\mathsf{QIOP}} \leftarrow \mathbf{V}_{\mathsf{k}}^{(\mathsf{d_k})}(\mathbb{x}, (\mathrm{Loc}_\iota, \mathrm{Y}_\iota)_{\iota \in [\mathsf{w_k}]}, \mathrm{V}_{\mathsf{k}}^{(\mathsf{d_k})})$.

ii. Output 1 if $b_{\mathsf{QIOP}} = 1$, $|\mathbb{x}| \leq \nu$, and $\mathbb{x} \notin L(R)$; output 0 otherwise.

Then by Construction 8.4 and the construction of hybrid games $\mathsf{H}_{i^*}(\widetilde{\mathcal{P}})$,

$$\Pr\left[\mathsf{H}_{\mathsf{k}}(\widetilde{\mathcal{P}})\right] = \Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq \nu & \mathrm{S} \leftarrow |\bot\rangle \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathcal{P}}^{U_{\mathsf{Sim}}(\mathrm{S})} \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathcal{P}}^{U_{\mathsf{Sim}}(\mathrm{S})}, \mathcal{V}^{U_{\mathsf{Sim}}(\mathrm{S})}(\mathbb{x})\rangle \end{array}\right], \tag{17}$$

and

$$\Pr\left[\mathsf{H}_0(\widetilde{\mathcal{P}})\right] = \Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq \nu & \mathbb{x} \leftarrow \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}) \\ \wedge\, \mathbb{x} \notin L(R) & b \leftarrow \langle \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}), \mathbf{V}(\mathbb{x})\rangle_{\mathsf{pq}} \\ \wedge\, b = 1 & \end{array}\right]. \tag{18}$$

Furthermore, the only difference between two consecutive hybrid games is whether the message underlying a commitment is accessed through the QSVC interfaces or through the extractor, and thus can be bounded by the extractability of the quantum-state vector commitment scheme QSVC.

Claim 8.7. *For every integer ν, t, m, $i^* \in [\mathsf{k}]$, and a t-query argument adversary $\widetilde{\mathcal{P}}$,*

$$\left|\sqrt{\Pr\left[\mathsf{H}_{i^*}(\widetilde{\mathcal{P}})\right]} - \sqrt{\Pr\left[\mathsf{H}_{i^*-1}(\widetilde{\mathcal{P}})\right]}\right|^2 \leq \xi_{\mathsf{Ext}}^{(3)}(t, \mathsf{d}, t, 0, m, \mathsf{l}_{i^*}) \ .$$

Proof. The only difference between games $\mathsf{H}_{i^*}(\widetilde{\mathcal{P}})$ and $\mathsf{H}_{i^*-1}(\widetilde{\mathcal{P}})$ is how they handle the i^*-th commitment. In the game $\mathsf{H}_{i^*}(\widetilde{\mathcal{P}})$, all the queries of the underlying QIOP verifier $\mathbf{V}$ to the i^*-th prover message are implemented by the interfaces of QSVC, while in the game $\mathsf{H}_{i^*-1}(\widetilde{\mathcal{P}})$, all the queries of the underlying QIOP verifier $\mathbf{V}$ to the i^*-th prover message are implemented by the interfaces of the extractor.

Consider an adversary A of QSVC that does the following:

1. Simulate $\mathsf{H}_{i^*-1}(\widetilde{\mathcal{P}})$ where the queries to $U_{\mathsf{Sim}}(\mathrm{S}_{i^*})$ are forwarded to the oracles (instead of implementing the access on its own) until the i^*-th commitment C_{i^*} is outputted.
2. Send C_{i^*} to the game as the commitment, and keep all other registers in the register E as the internal state.
3. For $i \in \{i^*, \cdots, \mathsf{k}\}$:
   (a) For $q \in [\mathsf{d}_i]$:
      i. Continue to simulate $\mathsf{H}_{i^*-1}(\widetilde{\mathcal{P}})$ until queries to the i^*-th prover's message register are about to be implemented.
      ii. Compute the query set to the i^*-th prover's message and denote it as Q, denote the corresponding answer register for these queries as Y, and denote the opening register for these queries as O_{i^*}. Send $(\mathrm{Q}, \mathrm{O}_{i^*}, \mathrm{Y})$ to the game, and keep all other registers in the register E as the internal state.
   (b) Continue to simulate $\mathsf{H}_{i^*-1}(\widetilde{\mathcal{P}})$ until $\mathbf{V}$ outputs J_{i+1} or $\mathbf{V}$ outputs the decision bit.
   (c) If $i^* \in J_{i+1}$ or $\mathbf{V}$ outputs the decision bit, inform the game that all queries to the underlying message are done.

Consider a distinguisher D that continues to simulate $\mathsf{H}_{i^*-1}(\widetilde{\mathcal{P}})$ from where the adversary A stops.

Then the game $\mathsf{H}_{i^*}(\widetilde{\mathcal{P}})$ is exactly $\mathsf{QSVCSimWorld}(A, D)$, and the game $\mathsf{H}_{i^*-1}(\widetilde{\mathcal{P}})$ is exactly $\mathsf{QSVCExtWorld}(A, D)$. By the extractability of QSVC,

$$\begin{aligned}
&\left|\sqrt{\Pr\left[\mathsf{H}_{i^*}(\widetilde{\mathcal{P}})\right]} - \sqrt{\Pr\left[\mathsf{H}_{i^*-1}(\widetilde{\mathcal{P}})\right]}\right|^2 \\
&= \left|\sqrt{\Pr\left[\mathsf{QSVCExtWorld}(A, D)\right]} - \sqrt{\Pr\left[\mathsf{QSVCSimWorld}(A, D)\right]}\right|^2 \\
&\leq \xi^{(3)}_{\mathsf{Ext}}(t, \mathsf{d}, t, 0, m, \mathsf{l}_{i^*}) \ ,
\end{aligned}$$

where we use the fact that the query depth of A to the underlying message is at most $\sum_{i \geq i^*} \mathsf{d}_i \leq \mathsf{d}$, and moreover, A has at most t queries and at most query mass t to $U_{\mathsf{Sim}}(\mathrm{S}_{i^*})$ since simulating $\mathbf{V}$ does not require query access to $U_{\mathsf{Sim}}(\mathrm{S}_{i^*})$, and the queries of QSVC and the extractor to $U_{\mathsf{Sim}}(\mathrm{S}_{i^*})$ and $U_{\mathsf{Ext}}(\mathrm{S}_{i^*})$ are all performed by the game. □

We combine Claims 8.6 and 8.7 to prove Lemma 8.5.

Proof of Lemma 8.5. By Claim 8.7 and Eqs. (17) and (18) and the triangle inequality,

$$\begin{aligned}
&\left|\sqrt{\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq \nu & \mathrm{S} \leftarrow |\bot\rangle \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathcal{P}}^{U_{\mathsf{Sim}}(\mathrm{S})} \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathcal{P}}^{U_{\mathsf{Sim}}(\mathrm{S})}, \mathcal{V}^{U_{\mathsf{Sim}}(\mathrm{S})}(\mathbb{x})\rangle \end{array}\right]} - \sqrt{\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq \nu & \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}) \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}), \mathbf{V}(\mathbb{x})\rangle_{\mathsf{pq}} \end{array}\right]}\right|^2 \\
&\leq \left(\sum_{i \in [\mathsf{k}]} \sqrt{\xi^{(3)}_{\mathsf{Ext}}(t, \mathsf{d}, t, 0, m, \mathsf{l}_i)}\right)^2 \\
&\leq \mathsf{k} \cdot \sum_{i \in [\mathsf{k}]} \xi^{(3)}_{\mathsf{Ext}}(t, \mathsf{d}, t, 0, m, \mathsf{l}_i) \ . && \text{(By Cauchy–Schwarz inequality)}
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq \nu & \mathsf{RO} \leftarrow \mathcal{U}(m) \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathcal{P}}^{\mathsf{RO}} \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathcal{P}}^{\mathsf{RO}}, \mathcal{V}^{\mathsf{RO}}(\mathbb{x})\rangle \end{array}\right] \\
&\leq \Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq \nu & \mathrm{S} \leftarrow |\bot\rangle \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathcal{P}}^{U_{\mathsf{Sim}}(\mathrm{S})} \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathcal{P}}^{U_{\mathsf{Sim}}(\mathrm{S})}, \mathcal{V}^{U_{\mathsf{Sim}}(\mathrm{S})}(\mathbb{x})\rangle \end{array}\right] + \xi^{(1)}_{\mathsf{Ext}}(t + \mathsf{vq}, m) && \text{(By Claim 8.6)} \\
&\leq \left(\sqrt{\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq \nu & \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}) \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}), \mathbf{V}(\mathbb{x})\rangle_{\mathsf{pq}} \end{array}\right]} + \sqrt{\mathsf{k} \cdot \sum_{i \in [\mathsf{k}]} \xi^{(3)}_{\mathsf{Ext}}(t, \mathsf{d}, t, 0, m, \mathsf{l}_i)}\right)^2 + \xi^{(1)}_{\mathsf{Ext}}(t + \mathsf{vq}, m) \\
&\leq 2\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq \nu & \\ \wedge\, \mathbb{x} \notin L(R) & \mathbb{x} \leftarrow \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}) \\ \wedge\, b = 1 & b \leftarrow \langle \widetilde{\mathbf{P}}(\widetilde{\mathcal{P}}), \mathbf{V}(\mathbb{x})\rangle_{\mathsf{pq}} \end{array}\right] + 2\mathsf{k} \cdot \sum_{i \in [\mathsf{k}]} \xi^{(3)}_{\mathsf{Ext}}(t, \mathsf{d}, t, 0, m, \mathsf{l}_i) + \xi^{(1)}_{\mathsf{Ext}}(t + \mathsf{vq}, m) \ . \\
&&& \text{(By Cauchy–Schwarz inequality)}
\end{aligned}$$

□

A The QIOP that we use

We use the following QIOP in our transformation.

Lemma A.1. *There exists a QIOP for* QMA *with total proof length* $\mathsf{l} = \eta \cdot \mathsf{poly}(\nu)$*, round complexity* $\mathsf{k} = \eta \cdot \mathsf{poly}(\log \nu)$*, query complexity* $\mathsf{q} = \eta \cdot \mathsf{poly}(\log \nu)$*, verifier-to-prover communication complexity* $\mathsf{vc} = \eta \cdot \mathsf{poly}(\log \nu)$*, completeness* $c = 1 - \eta \cdot \mathsf{negl}(\nu)$*, and public-query soundness* $s_{\mathsf{pq}} = 3^{-\eta}$ *for instance size* ν *and positive integer* η*.*

In particular, taking $\eta = \lceil (\log \nu)^2 \rceil$ *yields a QIOP with polynomial total proof length, polylogarithmic round complexity, polylogarithmic query complexity, polylogarithmic verifier-to-prover communication, completeness negligibly close to* 1*, and negligible public-query soundness.*

Proof sketch. We show how to construct a *public-coin* QIOP for QMA with $\mathsf{l} = \mathsf{poly}(\nu)$, $\mathsf{k} = \mathsf{poly}(\log \nu)$, $\mathsf{q} = \mathsf{poly}(\log \nu)$, $\mathsf{vc} = \mathsf{poly}(\log \nu)$, $c = 1 - \mathsf{negl}(\nu)$, and $s = \frac{1}{3}$ for instance size ν. The lemma then follows by repeating this QIOP sequentially η times, accepting only if all executions accept, and applying Lemma 3.3.

Our construction is obtained by making three minor modifications to the protocol of [SV26]. This protocol satisfies all our requirements except that it allows the verifier to return only part of the prover's first message register, has verifier-to-prover communication complexity $\mathsf{vc} = \mathsf{poly}(\nu)$, and is written as a private-coin protocol.

Splitting the prover's message. The verifier in [SV26] partitions the prover's first message register into blocks and returns one block in each round. This partition is fixed in advance and publicly known. Therefore, we can modify the protocol by having the prover send these blocks in separate rounds, with the verifier sending dummy messages between consecutive prover messages. Then each return in the modified protocol consists of an entire prover's message register from a single round.

Reducing communication. Returning prover's message registers do not contribute to vc. The only remaining long message is the final classical challenge specifying the Hamiltonian term used in the verification. As observed in Section 1.2.1 of [SV26], this communication can be reduced to $O(\log \nu)$ bits by using the derandomized amplification technique of [BMVZ26].

Public-coin. The measurement basis and Hamiltonian-term challenges are already public in [SV26]. However, the choice of test branch and round, the sampled code stabilizers, and the PCPP verifier's randomness are private in their formulation.

We modify the protocol to make it public-coin as follows. Instead of choosing the test branch and round in advance, the verifier samples fresh public coins at each checkpoint to decide whether to continue or test and terminate, using conditional stopping probabilities that preserve the original distribution over tests. Each decision is made after all registers needed for the selected test have been received. The verifier then publicly samples the randomness needed for that test (e.g., the code stabilizers and PCPP randomness), performs the queries, and immediately accepts or rejects, without using any subsequent prover message. Thus, revealing the test randomness gives the prover neither an opportunity to alter the registers being tested nor an advantage in preparing subsequent proofs. The original soundness bound therefore continues to apply. □

Acknowledgments

We thank Ziyi Guan, Ignacio Manzur, Baocheng Sun, and Thomas Vidick for helpful discussions. The authors are supported in part by the Ethereum Foundation and the Global Chinese Community of Universal Digital Commons. The authors had access to OpenAI models through the ChatGPT for Academic Researchers program.

AI disclosure

The authors used GPT-6 Astra to assist with identifying related work, improving the writing, and reviewing the proofs. The authors developed the proof ideas and wrote the final proofs, and take full responsibility for the paper.

References

[AALV09] Dorit Aharonov, Itai Arad, Zeph Landau, and Umesh Vazirani. "The Detectability Lemma and Quantum Gap Amplification". In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*. STOC '09. ACM, 2009, pp. 417–426. DOI: 10.1145/1536414.1536472.

[AAV13] Dorit Aharonov, Itai Arad, and Thomas Vidick. "Guest Column: The Quantum PCP Conjecture". In: *ACM SIGACT News* 44.2 (2013), pp. 47–79. DOI: 10.1145/2491533.2491549.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. "Interactive Oracle Proofs". In: *Proceedings of the 14th Theory of Cryptography Conference*. TCC '16-B. 2016, pp. 31–60. DOI: 10.1007/978-3-662-53644-5_2.

[BDFLSZ11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random Oracles in a Quantum World". In: *Proceedings of the 17th International Conference on the Theory and Application of Cryptology and Information Security*. ASIACRYPT '11. 2011, pp. 41–69. DOI: 10.1007/978-3-642-25385-0_3.

[BG08] Boaz Barak and Oded Goldreich. "Universal Arguments and Their Applications". In: *SIAM Journal on Computing* 38.5 (2008). Preliminary version appeared in CCC '02., pp. 1661–1694. DOI: 10.1137/070709244.

[BLM26] James Bartusek, Jiahui Liu, and Giulio Malavolta. "A Modular Approach to Succinct Arguments for QMA". In: *Proceedings of the 45th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '26. 2026, pp. 446–474. DOI: 10.1007/978-3-032-25336-1_16.

[BM26] James Bartusek and Giulio Malavolta. *Succinct Arguments for QMA from Collapsing Hash Functions*. Cryptology ePrint Archive, Report 2026/2040. 2026. URL: https://eprint.iacr.org/2026/2040.

[BMVZ26] Thiago Bergamaschi, Tony Metger, Thomas Vidick, and Tina Zhang. "Derandomised Tensor Product Gap Amplification for Quantum Hamiltonians". In: *Proceedings of the 41st Annual IEEE Conference on Computational Complexity*. CCC '26. 2026, 15:1–15:22. DOI: 10.4230/LIPICS.CCC.2026.15. URL: https://doi.org/10.4230/LIPIcs.CCC.2026.15.

[Bar+22] James Bartusek, Yael Tauman Kalai, Alex Lombardi, Fermi Ma, Giulio Malavolta, Vinod Vaikuntanathan, Thomas Vidick, and Lisa Yang. "Succinct Classical Verification of Quantum Computation". In: *Proceedings of the 42nd Annual International Cryptology Conference*. CRYPTO '22. 2022, pp. 195–211. DOI: 10.1007/978-3-031-15979-4_7.

[CDDGS25] Alessandro Chiesa, Marcel Dall'Agnol, Zijing Di, Ziyi Guan, and Nicholas Spooner. "Quantum Rewinding for IOP-Based Succinct Arguments". In: *Proceedings of the 23rd Theory of Cryptography Conference*. TCC '25. 2025, pp. 460–479. DOI: 10.1007/978-3-032-12296-4_16.

[CDGS23] Alessandro Chiesa, Marcel Dall'Agnol, Ziyi Guan, and Nicholas Spooner. *On the Security of Succinct Interactive Arguments from Vector Commitments*. Cryptology ePrint Archive, Report 2023/1737. 2023. URL: https://eprint.iacr.org/2023/1737.

[CDHZ26] Alessandro Chiesa, Zijing Di, Zihan Hu, and Yuxi Zheng. "How to Prove Post-Quantum Security for Succinct Non-Interactive Reductions". In: *Proceedings of the 45th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '26. 2026, pp. 389–415. DOI: 10.1007/978-3-032-25336-1_14.

[CGMV26] Alessandro Chiesa, Ziyi Guan, Ignacio Manzur, and Thomas Vidick. *Succinctness Requires Probabilistic Checking in the Quantum World*. Unpublished manuscript. 2026.

[CM24] Lijie Chen and Ramis Movassagh. "Quantum Merkle Trees". In: *Quantum* 8 (2024), p. 1380. DOI: 10.22331/q-2024-06-18-1380.

[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. "Succinct Arguments in the Quantum Random Oracle Model". In: *Proceedings of the 17th Theory of Cryptography Conference*. TCC '19. 2019, pp. 1–29. DOI: 10.1007/978-3-030-36033-7_1.

[CMSZ21] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. "Post-Quantum Succinct Arguments: Breaking the Quantum Rewinding Barrier". In: *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science*. FOCS '21. IEEE, 2021, pp. 49–58. DOI: 10.1109/FOCS52979.2021.00014.

[CY24] Alessandro Chiesa and Eylon Yogev. *Building Cryptographic Proofs from Hash Functions*. 2024. URL: https://github.com/hash-based-snargs-book/hash-based-snargs-book.

[DFMS22] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Online-Extractability in the Quantum Random-Oracle Model". In: *Proceedings of the 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '22. 2022, pp. 677–706. DOI: 10.1007/978-3-031-07082-2_24.

[Fai24] Islam Faisal. "Interactive Oracle Arguments in the QROM and Applications to Succinct Verification of Quantum Computation". In: *Proceedings of Cryptographers' Track at the RSA Conference 2024*. CT-RSA '24. 2024, pp. 404–429. DOI: 10.1007/978-3-031-58868-6_16. URL: https://doi.org/10.1007/978-3-031-58868-6_16.

[GJMZ23] Sam Gunn, Nathan Ju, Fermi Ma, and Mark Zhandry. "Commitments to Quantum States". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. STOC '23. ACM, 2023, pp. 1579–1588. DOI: 10.1145/3564246.3585198.

[GKNV25] Sam Gunn, Yael Tauman Kalai, Anand Natarajan, and Ági Villányi. "Classical Commitments to Quantum States". In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*. STOC '25. ACM, 2025, pp. 234–244. DOI: 10.1145/3717823.3718264.

[Kil92] Joe Kilian. "A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract)". In: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*. STOC '92. ACM, 1992, pp. 723–732. DOI: 10.1145/129712.129782.

[MNZ24] Tony Metger, Anand Natarajan, and Tina Zhang. "Succinct Arguments for QMA from Standard Assumptions via Compiled Nonlocal Games". In: *Proceedings of the 65th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '24. IEEE, 2024, pp. 1193–1201. DOI: 10.1109/FOCS61266.2024.00078.

[Mic00] Silvio Micali. "Computationally Sound Proofs". In: *SIAM Journal on Computing* 30.4 (2000). Preliminary version appeared in FOCS '94., pp. 1253–1298. DOI: 10.1137/S0097539795284959.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. 10th Anniversary Edition. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[Pas03] Rafael Pass. "On Deniability in the Common Reference String and Random Oracle Model". In: *Proceedings of the 23rd Annual International Cryptology Conference*. CRYPTO '03. 2003, pp. 316–337. DOI: 10.1007/978-3-540-45146-4_19.

[RRR16] Omer Reingold, Ron D. Rothblum, and Guy N. Rothblum. "Constant-Round Interactive Proofs for Delegating Computation". In: *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*. STOC '16. ACM, 2016, pp. 49–62. DOI: 10.1145/2897518.2897652.

[SV25] Baocheng Sun and Thomas Vidick. "Quantum Interactive Oracle Proofs". In: *Proceedings of the 23rd Theory of Cryptography Conference*. TCC '25. 2025, pp. 409–426. DOI: 10.1007/978-3-032-12296-4_14.

[SV26] Baocheng Sun and Thomas Vidick. "Probabilistically Checking Quantum Proofs, with Interaction". In: *Proceedings of the 41st Annual IEEE Conference on Computational Complexity*. CCC '26. 2026, 4:1–4:49. DOI: 10.4230/LIPIcs.CCC.2026.4.

[Yan22] Jun Yan. "General Properties of Quantum Bit Commitments (Extended Abstract)". In: *Proceedings of the 28th International Conference on the Theory and Application of Cryptology and Information Security*. ASIACRYPT '22. 2022, pp. 628–657. DOI: 10.1007/978-3-031-22972-5_22.

[Zha19] Mark Zhandry. "How to Record Quantum Queries, and Applications to Quantum Indifferentiability". In: *Proceedings of the 39th Annual International Cryptology Conference*. CRYPTO '19. 2019, pp. 239–268. DOI: 10.1007/978-3-030-26951-7_9.