Post-Quantum Zero Knowledge, Revisited

or: How to Do Quantum Rewinding Undetectably

Alex Lombardi
alexjl@mit.edu
MIT

Fermi Ma
fermima@alum.mit.edu
Simons Institute & UC Berkeley

Nicholas Spooner
nspooner@bu.edu
Boston University

November 23, 2021

Abstract

A major difficulty in quantum rewinding is the fact that measurement is destructive: extracting information from a quantum state irreversibly changes it. This is especially problematic in the context of zero-knowledge simulation, where preserving the adversary's state is essential.

In this work, we develop new techniques for quantum rewinding in the context of extraction and zero-knowledge simulation:

1. We show how to extract information from a quantum adversary by rewinding it *without disturbing its internal state*. We use this technique to prove that important interactive protocols, such as the Goldreich-Micali-Wigderson protocol for graph non-isomorphism and the Feige-Shamir protocol for NP, are zero-knowledge against quantum adversaries.
2. We prove that the Goldreich-Kahan protocol for NP is post-quantum zero knowledge using a simulator that can be seen as a natural quantum extension of the classical simulator.

Our results achieve (constant-round) black-box zero-knowledge with *negligible* simulation error, appearing to contradict a recent impossibility result due to Chia-Chung-Liu-Yamakawa (FOCS 2021). This brings us to our final contribution:

3. We introduce *coherent-runtime* expected quantum polynomial time, a computational model that (1) captures all of our zero-knowledge simulators, (2) cannot break any polynomial hardness assumptions, and (3) is not subject to the CCLY impossibility. In light of our positive results and the CCLY negative results, we propose coherent-runtime simulation to be the right quantum analogue of classical expected polynomial-time simulation.

Contents

1 Introduction

Zero-knowledge protocols [GMR85] are a fundamental tool in modern cryptography in which a prover convinces a verifier that some statement is true without revealing any additional information. This security property is formalized via *simulation*: the view of any (even malicious) verifier V^* can be simulated in polynomial time (without access to, e.g., an NP witness for the statement).

Although the zero-knowledge property sounds almost paradoxical, it is achieved by designing a simulator S^{V^*} that makes use of V^* in ways that the honest protocol execution cannot, thereby resolving the apparent paradox. In the simplest and most common setting, the key simulation technique is *rewinding*. Given an interactive adversary A, an oracle algorithm S^A is said to rewind the adversary if it saves the state of A midway through an execution in order to run A *multiple times* on different inputs. Rewinding is ubiquitous in the analysis of interactive proof systems, establishing properties such as zero-knowledge [GMR85, GMW86], soundness [Kil92], and knowledge-soundness [GMR85, BG93].

However, since the foundational techniques of interactive proof systems were established, our conception of what constitutes efficient computation has fundamentally changed. Both in theory [Sho94] and in practice [AAB+19], quantum computers appear to have capabilities beyond that of any efficient classical computer. Thus, it is imperative to analyze security against quantum adversaries. In this work, we consider this question for zero-knowledge protocols.

When do classical zero-knowledge protocols remain secure against quantum adversaries?

At a minimum, such protocols must be based on post-quantum cryptographic assumptions. However, since zero-knowledge is typically proved via rewinding, resolving this question also entails understanding *to what extent we can rewind quantum adversaries.* Unfortunately, rewinding quantum adversaries is notoriously difficult because an adversary's internal state may be disturbed if any classical information is recorded about its response, potentially rendering it useless for subsequent executions [vdG97, Wat06, Unr12, ARU14].

By now, a few techniques exist to rewind quantum adversaries [Wat06, Unr12, Unr16b, CCY21, ACL21, CMSZ21], but the range of protocols to which these techniques apply remains quite limited. As a basic example, Watrous's zero-knowledge simulation technique [Wat06] applies to the standard [GMW86] zero-knowledge proof system for graph isomorphism but (as noted in [Unr12, ARU14]) does *not* apply to the related [GMW86] zero-knowledge proof system for graph *non*-isomorphism (GNI). Recall that in the GNI protocol, the prover P wants to convince the verifier V that two graphs G_0, G_1 are not isomorphic. To do so, the verifier sends a random isomorphic copy H of G_b for a uniformly random bit b, to which the prover returns b.[1] However, to ensure zero-knowledge, the verifier first gives a proof of knowledge (PoK) that H is isomorphic to either G_0 or G_1 via a variant of the parallel-repeated graph *isomorphism* Σ-protocol. Intuitively, this ensures that a malicious verifier V^* already *knows* b and hence does not learn anything new from the interaction.

The classical zero-knowledge simulator for the GNI protocol has two steps:

1. **Extract** an isomorphism π satisfying $\pi(H) = G_b$ for some b using *multiple* valid PoK responses from the malicious verifier V^*.

2. **Simulate** the view of V^* in an real interaction by returning b (computed efficiently from π).

[1] For this overview, we focus on the soundness 1/2 case, but appropriate parallel repetition of this step reduces the soundness error.

It turns out that this kind of extract-and-simulate approach is beyond the reach of existing quantum rewinding techniques, because all known techniques for extracting information from *multiple executions* of the adversary [Unr12, Unr16b, CMSZ21] fundamentally disturb the state. While this particular example just concerns the GNI protocol, this extract-and-simulate approach is very widely applicable, especially in the context of *composition*, including protocols that follow the "FLS trapdoor paradigm" [FS90] or make use of extractable commitments [RK99, PRS02, Ros04]. Given this state of affairs, we ask:

When is it possible to undetectably *extract information from a quantum adversary?*

As per the above discussion, if it is possible to *undetectably* extract from the proof-of-knowledge subroutine in the [GMW86] GNI protocol, then the full protocol is zero-knowledge against quantum adversaries.

1.1 This Work

In this work, we develop new techniques for quantum rewinding in the context of extraction and zero-knowledge simulation:

(1) Our first contribution is to give a quantum analogue of the *extract-and-simulate* paradigm used in many classical zero-knowledge protocols, in which a simulator uses information extracted from *multiple protocol transcripts* to simulate the verifier's view. The key difficulty in the quantum setting is to extract this information without causing any noticeable disturbance to the verifier's quantum state — beyond the disturbance caused by a single protocol execution.

While the recent techniques of [CMSZ21] allow extracting from multiple protocol transcripts, a major problem is that their extractor noticeably disturbs the adversary's state. We revisit the [CMSZ21] approach for extraction and, using several additional ideas, construct an *undetectable* extractor for a broad class of protocols. Using this extraction technique, we prove that the original [GMW86] protocol for graph non-isomorphism and some instantiations of the [FS90] protocol for NP are zero-knowledge against quantum adversaries.

(2) We next turn our attention to the Goldreich-Kahan [GK96] zero-knowledge proof system for NP. Informally, analyzing the [GK96] proof system presents different challenges as compared to [GMW86, FS90] because in the latter protocols, rewinding is used for *extraction* (after which simulation is straight-line), while in the [GK96] protocol, rewinding is used for the *simulation* step (while extraction is trivial/straight-line).

Nevertheless, we show that some of our techniques are also applicable in this setting. We prove that the [GK96] protocol is zero-knowledge against quantum adversaries. Our simulator can be viewed as a natural quantum extension of the classical simulator.

Previously, [CCY21] used different techniques to show that the [GK96] protocol is ε-zero-knowledge against quantum adversaries, but their simulation strategy cannot achieve negligible accuracy.

Isn't this impossible? As stated above, our results (both (1) and (2)) achieve constant-round black-box zero-knowledge with negligible simulation accuracy. Recently, [CCLY21] showed that there do not exist black-box expected quantum polynomial time (EQPT) simulators for constant-round protocols for any language $L \notin \mathsf{BQP}$. The source of the disconnect between our results and [CCLY21] is an ambiguity in the definition of EQPT. This brings us to our final contribution.

(3) We formally study the notion of expected runtime for quantum machines and formulate a model of expected quantum polynomial time simulation that avoids the [CCLY21] impossibility result.

We now discuss these contributions in more detail. To avoid confusion about the formal statements of (1) and (2), we begin by describing (3).

1.2 Coherent-Runtime Expected Quantum Polynomial Time

While [CCLY21] do not formally define EQPT,[2] implicit in their result is the following computational model, which we call *measured-runtime* EQPT (EQPT_m). In this model, a computation on input $|\psi\rangle_{\mathcal{A}}$ is the following process, for a fixed "transition" unitary U_δ (corresponding to a quantum Turing machine transition function δ) and time bound $T = \exp(\lambda)$:[3]

1. Initialize a fresh memory/work register $\mathcal{W}$ to $|0^T\rangle_{\mathcal{W}}$ and a designated "halt" qubit $\mathcal{Q}$ to $|0\rangle$;
2. Repeat for at most T steps:
   (a) measure $\mathcal{Q}$ and halt if it is 1;
   (b) apply U_δ to $\mathcal{A} \otimes \mathcal{Q} \otimes \mathcal{W}$.

The result of the computation is the residual state on $\mathcal{A}$ once the computation has halted. We say that a computation is EQPT_m if for *all* states $|\psi\rangle_{\mathcal{A}}$, the expected running time of this computation is $\mathrm{poly}(\lambda)$. Using this model, we can give a more precise formulation of the [CCLY21] theorem: black-box EQPT_m zero-knowledge simulators for constant-round protocols do not exist. The key feature of the EQPT_m model that enables the [CCLY21] result is that the runtime is *measured*.

In this work, we consider a different computational model for EQPT simulation called *coherent-runtime EQPT* (EQPT_c). In our model, simulators have the ability to run EQPT_m procedures *coherently* — which yields a superposition over computations with different runtimes — and then later *uncompute* the runtime by running the same computation in reverse.

Our notion of EQPT_c (see Definition 9.3) captures (as a special case) computations of the form depicted in Fig. 1 on an input $|\phi\rangle_{\mathcal{X}}$, where the result of the computation is the residual state on $\mathcal{X}$. In Fig. 1, C_1, C_2, C_3 are arbitrary polynomial-size quantum circuits and U is a unitary that *coherently implements* an EQPT_m computation.[4] In slightly more detail, any EQPT_m computation with transition unitary U_δ and runtime bound T can be expressed as a unitary circuit $U = V_T \cdots V_2 V_1$ where each V_i consists of two steps: (1) CNOT the halt qubit onto a register $\mathcal{B}_i$, and then (2) apply U_δ controlled on $\mathcal{B}_i = 0$. The unitary U acts on $\mathcal{A} \otimes \mathcal{Q} \otimes \mathcal{W} \otimes \mathcal{B}$ where $\mathcal{B} := \mathcal{B}_1 \otimes \cdots \otimes \mathcal{B}_T$. While

[2] When defining quantum zero-knowledge simulation, [CCLY21, Page 12] requires that the simulator is a quantum Turing machine with *expected* polynomial runtime, and refers to [BBBV97] (which uses the [BV97] definition of a quantum Turing machine) for the quantum Turing machine model. However, as we discuss in Section 2.1, [BV97] restricts quantum Turing machines to have a *fixed* running time (see [BV97, Def 3.11]) in order to avoid difficult-to-resolve subtleties about quantum Turing machines with variable running time [Mye97, Oza98a, LP98, Oza98b].

[3] The exponential time bound is simply for convenience; by Markov's inequality, for any expected polynomial time computation truncating the computation after an exponential number of steps has only a negligible effect on the output state.

[4] Our actual definition is for *uniform* computation, so (U, C_1, C_2, C_3) will have a uniform description.

our full definition of EQPT_c is more general, all the simulators we give can be written in the form of Fig. 1. In Fig. 1, the input register is of the form $\mathcal{X} = \mathcal{X}_1 \otimes \mathcal{X}_2$ where $\mathcal{X}_2$ is isomorphic to $\mathcal{A}$.

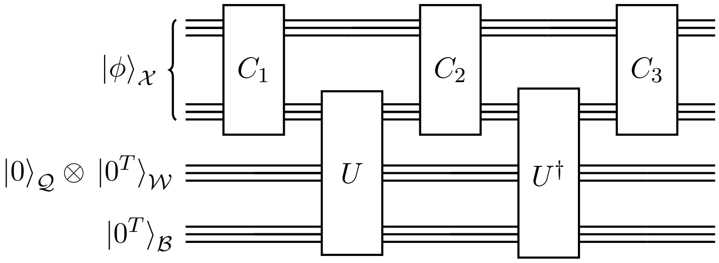

$|\phi\rangle_{\mathcal{X}}$
$|0\rangle_{\mathcal{Q}} \otimes |0^T\rangle_{\mathcal{W}}$
$|0^T\rangle_{\mathcal{B}}$
C_1
U
C_2
$U^\dagger$
C_3

Figure 1: An example of an EQPT_c circuit.

We discuss and motivate the definition of EQPT_c in detail in Section 2.1. For now, we note two key properties of the model. First, the ability to apply $U^\dagger$ is what enables us to circumvent the [CCLY21] impossibility for EQPT_m. Second, we show a result analogous to the statement that any expected *classical* polynomial time computation can be truncated to *fixed* polynomial-time with small error.

Lemma 1.1 (informal, see Claim 9.4)**.** *Any* EQPT_c *computation can be approximated with* ε *accuracy by a quantum circuit of size* $\mathrm{poly}(\lambda, 1/\varepsilon)$*.*

Importantly, this lemma ensures that EQPT_c computations cannot break post-quantum polynomial hardness assumptions (unless the assumptions are false). We also note that this lemma implies that black-box zero-knowledge with EQPT_c simulation implies ε-zero-knowledge with strict quantum polynomial time simulation.[5]

What does this mean for post-quantum zero knowledge? Since the introduction of zero-knowledge protocols [GMR85], expected polynomial-time simulation has been the *de facto* model for classical zero knowledge. Although expected polynomial-time simulators cannot actually be run "in real life" (to negligible accuracy), the security notion captures negligible accuracy simulation in a computational model that *cannot break polynomial hardness assumptions*. Moreover, since [BL02] rules out *strict* polynomial-time simulation for constant-round protocols, expected polynomial-time simulation captures the strongest[6] provable zero-knowledge properties of many fundamental protocols such as quadratic non-residuosity [GMR85], graph non-isomorphism [GMW86], Goldreich-Kahan [GK96], and Feige-Shamir [FS90].

The combination of our positive results and the [CCLY21] negative results transports this state of affairs entirely to the post-quantum setting. In particular, the conclusion we draw from the [CCLY21] negative result is that one must go beyond the EQPT_m model in order to find the right quantum analogue of classical expected polynomial-time zero knowledge simulation. We propose EQPT_c to be that quantum analogue.

[5] We also note that all of our simulators can be truncated to run in time $\mathsf{poly}(\lambda) \cdot 1/\varepsilon$ and achieve ε-ZK, matching the ε-dependence of the classical simulator.

[6] Other models of efficient simulation [Lev86, Klo20] have been proposed (in the classical setting), but they are *relaxations* of expected polynomial time simulation. It should be possible to define similar relaxations in the quantum setting, but we focus on obtaining an analogue to "standard" expected polynomial time simulation.

With this discussion in mind, we proceed to describe our results on post-quantum zero-knowledge and extraction in more detail.

1.3 Results on Zero Knowledge

Our main results regarding post-quantum zero knowledge are as follows. First, we show that the [GMW86] graph non-isomorphism protocol is zero knowledge against quantum verifiers.

Theorem 1.2. *The [GMW86] 4-message proof system for graph non-isomorphism is a post-quantum statistical zero knowledge proof system. The zero-knowledge simulator is black-box and runs in* EQPT_c.

The [GMW86] GNI protocol follows a somewhat general template using instance-dependent commitments [BMO90, IOS97, MV03]; we believe Theorem 1.2 should extend to other instantiations of this paradigm (e.g. for lattice problems).

With some additional work, we use similar techniques to show how to instantiate the "extract-and-simulate" paradigm of Feige-Shamir [FS90] in the post-quantum setting.

Theorem 1.3. *Assuming super-polynomially secure non-interactive commitments, a particular instantiation of the [FS90] 4-message argument system for* NP *is (sound and) zero-knowledge against quantum verifiers. The zero-knowledge simulator is black-box and runs in* EQPT_c.

Finally, using a different approach, we show that the Goldreich-Kahan [GK96] zero-knowledge proof system remains ZK against quantum adversaries.

Theorem 1.4. *When instantiated using a collapse-binding and statistically-hiding commitment scheme, the [GK96] protocol is zero-knowledge with a black-box* EQPT_c *simulator.*

As a bonus, the simulator we construct in Theorem 1.4 bears a strong resemblance to the *classical* Goldreich-Kahan simulator, giving a clean conceptual understanding of constant-round zero knowledge in the quantum setting.

1.4 Results on Extraction

As alluded to in the introduction, Theorem 1.2 and Theorem 1.3 are proved using new results on post-quantum *extraction*. We achieve "undetectable extraction" under the following definition of a *state-preserving proof of knowledge*.[7]

Definition 1.5. An interactive protocol Π is defined to be a state-preserving argument (resp. proof) of knowledge if there exists an extractor $\mathsf{Ext}^{(\cdot)}$ with the following properties:

- **Syntax**: For any quantum algorithm P^* and auxiliary state $|\psi\rangle$, $\mathsf{Ext}^{P^*,|\psi\rangle}$ outputs a protocol transcript τ, prover state $|\psi'\rangle$, and witness w.
- **Extraction Efficiency**: If P^* is a QPT algorithm, $E^{P^*(\cdot),|\psi\rangle}$ runs in expected quantum polynomial time (EQPT_c).

[7] This is a quantum analogue of *witness-extended emulation* [BG02]. Our definition is also similar to a definition appearing in [ACL21], although they only consider the setting of *statistical* state preservation.

- **Extraction Correctness**: the probability that τ is an accepting transcript but w is an invalid NP witness is negligible.

- **State-Preserving**: the pair $(\tau, |\psi'\rangle)$ is computationally (resp. statistically) indistinguishable from a transcript-state pair $(\tau^*, |\psi^*\rangle)$ obtained through an honest one-time interaction with $P^*(\cdot, |\psi\rangle)$ (where $|\psi^*\rangle$ is the prover's residual state).

Proofs/arguments of knowledge are typically used (rather than just sending an NP witness) to achieve either *succinctness* [Kil92] or *security against the verifier* (e.g., witness indistinguishability) [GMW86, FS90]. We show that standard 3- and 4-message protocols in both of these settings are state-preserving proofs/arguments of knowledge.

Theorem 1.6 (State-preserving succinct arguments)**.** *Assuming collapsing hash functions exist, there exists a 4-message public-coin state-preserving succinct argument of knowledge for* NP*.*

For witness indistinguishability (WI), we have three related constructions achieving slightly different properties under different computational assumptions.

Theorem 1.7 (State-preserving WI arguments)**.** *Assuming collapsing hash functions or* super-polynomially secure *one-way functions, there exists a 4-message public-coin state-preserving witness-indistinguishable argument (in the case of collapsing)/proof (in the case of OWFs) of knowledge. Assuming* super-polynomially secure *non-interactive commitments, there exists a 3-message PoK achieving the same properties.*

In fact, as we will explain in the technical overview, we give explicit conditions under which any proof/argument of knowledge is also state-preserving.

One special case of Theorem 1.7 that we would like to highlight is that of *extractable commitments* [PRS02, PW09]. An extractable commitment scheme ExtCom is a commitment scheme with the property that a committed message m can be extracted given black-box access to an adversarial sender (provided that the adversary is sufficiently convincing). Analogously to the setting of proofs-of-knowledge, we consider "state-preserving" extractable commitments (see, e.g., [BS20, GLSV21, BCKM21]), in which the extractor must simulate the entire view of the adversarial committer in addition to extracting the message. This variant of extractable commitments is quite natural; for example, it is exactly the property necessary to prove the post-quantum security of the [Ros04] zero-knowledge proof system for NP. An immediate corollary of Theorem 1.7 is a new construction of state-preserving extractable commitments.

Corollary 1.8 (Extractable commitments)**.** *Assuming super-polynomially secure non-interactive commitments, there exists a* 3*-message public-coin post-quantum statistically-binding* extractable commitment scheme. *Assuming super-polynomially secure one-way functions, there exists a* 4*-message scheme with the same properties. Finally, assuming (polynomially secure) collapsing hash functions, there exists a* 4*-message public-coin collapse-binding extractable commitment scheme.*

We leave open the problem of using these techniques to achieve a statistically-binding extractable commitment scheme from polynomial assumptions.

More generally, we expect our state-preserving extraction results to be useful for future applications, both in the context of zero-knowledge and beyond.

2 Technical Overview

In this section, we describe our techniques for proving our results on state-preserving extraction (Theorems 1.6 and 1.7) and post-quantum zero knowledge (Theorem 1.2, Theorem 1.3, and Theorem 1.4). Finally, we discuss related work in Section 2.5. .

2.1 Defining Expected Quantum Polynomial Time Simulation

In order to clearly present our results on zero knowledge, we begin with a detailed discussion of our model of expected quantum polynomial time simulation and how it relates to the [CCLY21] impossibility result.

Why is EQPT simulation hard to define? Recall from Section 1.2 that [CCLY21] rules out zero-knowledge simulators in a class of computations that we formalize as measured-runtime expected quantum polynomial-time (EQPT_m). An EQPT_m computation takes as input a state $|\psi\rangle_{\mathcal{A}}$, initializes a large ancilla/workspace register $|0\rangle_{\mathcal{W},\mathcal{B}}$ and state register $|q_0\rangle_{\mathcal{Q}}$ (where $|q_0\rangle$ denotes the initial state of a quantum Turing machine), then repeatedly applies some fixed transition unitary U_δ to $\mathcal{A} \otimes \mathcal{W} \otimes \mathcal{B} \otimes \mathcal{Q}$. After each application of U_δ, $\mathcal{Q}$ is measured (applying some $(\Pi_f, \mathbf{I} - \Pi_f)$) to determine if the computation is in the "halt state" $|q_f\rangle$; the computation halts if the outcome of this measurement is 1. A computation is EQPT_m if the expected number of steps before halting is polynomial for all inputs $|\psi\rangle_{\mathcal{A}}$.

Our EQPT_m definition is based on the definition of a quantum Turing machine (QTM) given in the seminal work of Deutsch [Deu85] (though we use a halt state [Oza98a] in place of Deutsch's halt qubit). Note that the operation of a QTM is unitary *except* for the measurement of whether the machine has halted. The validity of this "halting scheme" was the subject of some debate in a sequence of later works [Mye97, Oza98a, LP98, Oza98b].

While the particulars of this debate are not so important here, there was a clear message: the reversibility of a QTM implies that the runtime of any QTM computation is *always* effectively measured, even if there is no explicit monitoring of the halt state. Intuitively, this is because a QTM that has halted must, when reversed, know when to "un-halt"; this requires counting the number of computation steps since the machine halted.

It was observed by [LP98] that this prevents "useful interference" between branches of a QTM computation with different runtimes. That is, each branch of the computation is entangled with a description of its runtime, which prevents the branches from interfering with one another. Because interference is crucial in the design of efficient quantum algorithms, this is considered a major drawback of the QTM model. The now-standard definitions of *efficient* quantum computation [BV97, BBBV97] deliberately avoid this problem by restricting quantum Turing machines to have a *fixed* runtime; these QTMs are effectively uniform quantum circuit families.

This phenomenon underpins the [CCLY21] impossibility result. Both in the classical [BL02] and quantum [CCLY21] settings, there do not exist *strict* polynomial time black-box simulators for constant-round protocols. It follows that such a simulator must have a variable runtime. By the observation of [LP98], simulation branches with different runtimes do not interfere. [CCLY21] leverage this by designing an adversary which can *detect* this absence of interference.

Can we avoid measuring the runtime? The above discussion suggests that the EQPT_m model (i.e., quantum Turing machines in Deutsch's model [Deu85] with expected polynomial runtime) may not capture arbitrary efficient quantum computation. In particular, we ask whether it is possible to formalize a model in which the runtime is *not* measured. Such a model could potentially avoid the [CCLY21] impossibility result.

Our solution is to formalize computations in which the runtime of an EQPT_m subcomputation is left *in superposition* and can later be *uncomputed*. To describe our formalism in more detail, we first briefly discuss coherent computation.

Coherent computation. It is well known that any quantum operation Φ on a state $|\psi\rangle$ can be realized in three steps: (1) prepare some ancilla qubits in a fixed state $|0\rangle$; (2) apply a unitary operation U_Φ to both $|\psi\rangle$ and the ancilla; (3) discard (trace out) the ancilla. We refer to U_Φ as a *unitary dilation* of Φ. U_Φ is not uniquely determined by Φ, but all such dilations are related by an isometry acting only on the ancilla system.

Since an EQPT_m computation is a quantum operation, it has a unitary dilation. In fact, we can choose a unitary dilation with a natural explicit form that we call a "coherent implementation", as shown in Fig. 2.

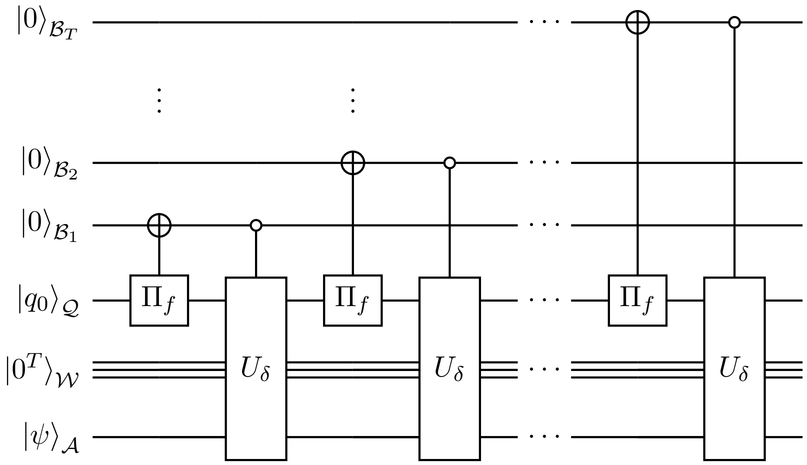

$|0\rangle_{\mathcal{B}_T}$
$|0\rangle_{\mathcal{B}_2}$
$|0\rangle_{\mathcal{B}_1}$
$|q_0\rangle_{\mathcal{Q}}$
$|0^T\rangle_{\mathcal{W}}$
$|\psi\rangle_{\mathcal{A}}$
Π_f
U_δ
Π_f
U_δ
Π_f
U_δ
$\cdots$

Figure 2: A coherent implementation U (unitary dilation) of a quantum Turing machine with transition function δ. The open circles indicate that the ith U_δ is applied when $\mathcal{B}_i$ contains $|0\rangle$. See Section 9.1 for more details.

Note that since we think of T as being exponentially large, the unitary U (corresponding to Fig. 2) is of exponential size. However, as long as the ancilla $\mathcal{W} \otimes \mathcal{B}$ (where $\mathcal{B} := \mathcal{B}_1 \otimes \cdots \otimes \mathcal{B}_T$) is initialized to zero and $\mathcal{Q}$ is initialized to $|q_0\rangle$, the effect of U on $\mathcal{A}$ is identical to the original EQPT_m computation. Indeed, the only difference from the original computation is that the runtime is written (in unary) on $\mathcal{B}$ and left in superposition. This means, in particular, that circuits making a single black-box query to a coherent implementation U of an EQPT_m computation (and that cannot otherwise access $\mathcal{B}$) can only perform EQPT_m computations.

Our formalism: coherent-runtime EQPT. The advantage of moving to coherent implementations is that, unlike the original computation, U has an *inverse* $U^\dagger$. A coherent-runtime EQPT computation is allowed to invoke both U and $U^\dagger$ in a restricted way, as we specify next.

Definition 2.1 (Coherent-runtime EQPT (informal))**.** A computation on a register $\mathcal{X}$ is coherent-runtime EQPT (EQPT_c) if it can be implemented by a procedure that can perform the following operations any polynomial number of times:

(1) apply any polynomial-size quantum circuit to $\mathcal{X}$;

(2) initialize fresh ancillas $\mathcal{W}^{(i)} \otimes \mathcal{B}^{(i)}$ to zero and $\mathcal{Q}^{(i)}$ to $|q_0^{(i)}\rangle$ and apply a coherent implementation U_i of an EQPT_m computation to $\mathcal{W}^{(i)} \otimes \mathcal{B}^{(i)} \otimes \mathcal{Q}^{(i)}$ and a subregister of $\mathcal{X}$;

(3) apply $U_i^\dagger$ to ancillas $\mathcal{W}^{(i)} \otimes \mathcal{B}^{(i)} \otimes \mathcal{Q}^{(i)}$ and a subregister of $\mathcal{X}$, then discard $(\mathcal{W}^{(i)}, \mathcal{B}^{(i)}, \mathcal{Q}^{(i)})$. For each U_i, this operation may occur at any time after performing (2) with respect to U_i.

The output of the computation is the residual state on $\mathcal{X}$.

Note that, because all unitary dilations are equivalent up to local isometry, the map computed by an EQPT_c computation is independent of the particular implementation of U_i.

What is the runtime of an EQPT_c computation? While procedures performing only (1) and (2) are clearly efficient (they are EQPT_m), the efficiency of performing $U_i^\dagger$ in (3) is less immediate. We analyze this in two ways:

- We prove (Claim 9.4) that any EQPT_c computation has strict polynomial-time approximations (obtained by simultaneously truncating each U_i and $U_i^\dagger$ to the same fixed runtime). This tells us that EQPT_c algorithms do not implement "inefficient" computations.

- We give a natural interpretation of "expected runtime" under which the expected runtime of $U^\dagger$ (as applied in (3)) is equal to the expected runtime of U.

Together, these give us a motivated definition of the expected runtime of an EQPT_c computation.

Claim 9.4 is proved in Section 9. In this overview, we focus on the expected runtime interpretation. For simplicity, we consider the basic form of EQPT_c computation depicted in Fig. 1; we assume in addition that $C_1 = C_3 = \mathbf{I}$ and that the computation always halts in at most T steps.

Let $U^{(t)}$ be the truncation of U to just after the t-th controlled application of U_δ. Observe that after applying U followed by C_2 on input state $|\phi\rangle$, the state of the system can be described as a superposition over t of the applications of the $U^{(t)}$:

$$|\Psi\rangle = \sum_{t=0}^{T} \alpha_t \, |0^t 1^{T-t}\rangle_{\mathcal{B}} \, |q_f\rangle_{\mathcal{Q}} \, |\phi_t\rangle_{\mathcal{W},\mathcal{X}} = \sum_{t=0}^{T} \alpha_t (\mathbf{I} \otimes C_2) U^{(t)} \, |0^t 1^{T-t}\rangle_{\mathcal{B}} \, |q_0\rangle_{\mathcal{Q}} \, |0\rangle_{\mathcal{W}} \, |\phi\rangle_{\mathcal{X}}$$

for some states $|\phi_t\rangle$ and $\alpha_t \in \mathbb{C}$ where $|\alpha_t|^2$ is the probability that the EQPT_m computation halts in t steps. The latter equality holds because if the computation halts at step t, the effect of the last $T - t$ steps of U is only to flip $\mathcal{B}_{t+1}, \ldots, \mathcal{B}_T$ from $|0\rangle$ to $|1\rangle$ (and C_2 does not act on $\mathcal{B}$).

We emphasize that since U is a coherent implementation of an EQPT_m computation, we know that

$$\sum_t |\alpha_t|^2 \cdot t = \mathrm{poly}(\lambda),$$

as the left-hand side is equal to the expected runtime of U as an EQPT_m computation.

Now, for any state $|\psi\rangle$ on $\mathcal{W} \otimes \mathcal{X}$, we can also express an application of $U^\dagger$ to $\mathcal{B} \otimes \mathcal{Q} \otimes \mathcal{W} \otimes \mathcal{X}$ entirely in terms of the unitary $U^{(t)}$, where t is the contents of the $\mathcal{B}$ register. Specifically, we have

$$U^\dagger \, |0^t 1^{T-t}\rangle_{\mathcal{B}} \, |q_f\rangle_{\mathcal{Q}} \, |\psi\rangle = (U^{(t)})^\dagger \, |0^T\rangle_{\mathcal{B}} \, |q_f\rangle_{\mathcal{Q}} \, |\psi\rangle$$

because the effect of the first $T-t$ steps of $U^\dagger$ on this state is only to flip $\mathcal{B}_{t+1},\ldots,\mathcal{B}_T$ from $|1\rangle$ to $|0\rangle$. As a result, the final state of the system (after the entire EQPT_c computation) is

$$U^\dagger |\Psi\rangle = \sum_{t=0}^{T} \alpha_t U^\dagger (\mathbf{I} \otimes C_2) U^{(t)} |0^t 1^{T-t}\rangle_{\mathcal{B}} |\phi\rangle = \sum_{t=0}^{T} \alpha_t (U^{(t)})^\dagger (\mathbf{I} \otimes C_2) U^{(t)} |0^T\rangle_{\mathcal{B}} |\phi\rangle .$$

We can interpret this to mean that within the "branch" of the superposition where U ran in time t, the running time of $U^\dagger$ is also t, even if an arbitrary computation C_2 has been applied to $\mathcal{X}$ in between the applications of U and $U^\dagger$. This gives an intuitive explanation for how EQPT_c computations are efficient: they simply compute an $(\alpha_t)_t$-superposition over branches in which U and $U^\dagger$ together ran for $2t$ steps, such that the expectation $\sum_t |\alpha_t|^2 \cdot t$ is polynomial! Curiously, the [LP98] reversibility issue indicates that such a computation cannot be implemented by an EQPT_m quantum Turing machine, which is what necessitates our new EQPT_c definition.

With all of this as motivation, we define the expected running time of an EQPT_c computation of the form $(U, C_1 = \mathbf{I}, C_2, C_3 = \mathbf{I})$ to be the appropriate linear combination of the branch runtimes, which is

$$\sum_t |\alpha_t|^2 \cdot (2t + \mathsf{time}(C_2)) = 2 \cdot \mathsf{time}(U) + \mathsf{time}(C_2),$$

where $\mathsf{time}(U)$ is the expected running time of U as an EQPT_m computation and $\mathsf{time}(C_2)$ is the (strict) running time of C_2. Claim 9.4 (whose proof makes use of this analysis) provides additional justification for this definition.

Extension to multiple U_i. Everything we have discussed so far extends to the general case of Definition 2.1. However, we emphasize that the above analysis crucially relies on the ancilla being well-formed. This is the reason that EQPT_c algorithms have restricted access to $U_i, U_i^\dagger$: removing any of these restrictions could lead to applying these operations on malformed ancillas. Indeed, one can show that allowing an algorithm to apply $U_i, U_i^\dagger, U_i$ to the same ancilla register would enable it to perform exponential-time computations.

Having established our computational model for simulation/extraction, we now give a detailed overview of our simulation and extraction techniques.

2.2 Post-Quantum ZK for [GMW86] and [FS90] from Guaranteed Extraction

The central idea behind our proofs of post-quantum ZK for the [GMW86] GNI protocol (Theorem 1.2) and a variant of the [FS90] protocol for NP (Theorem 1.3) is *state-preserving extraction* (Definition 1.5). Given a state-preserving extractor of the appropriate "one-out-of-two graph isomorphism" subroutine, proving the post-quantum ZK for the [GMW86] GNI protocol (Theorem 1.2) follows easily, as simulating a cheating verifier immediately reduces to performing a state-preserving extraction of the verifier's (uniquely determined) bit b such that $H \simeq G_b$. Proving post-quantum ZK for the [FS90] protocol (Theorem 1.3) is more complicated because the Feige–Shamir protocol is a concurrent composition of two different protocols; we refer the reader to Section 12 for details on its analysis.

In this subsection, we show that state-preserving extraction reduces to a related task that we call *guaranteed extraction*; achieving the latter will be the focus of Section 2.3.

Consider a 3-message[8] public coin classical proof of knowledge (P_Σ, V_Σ) satisfying *special soundness*:[9] for any prover first message a and any *pair* of accepting transcripts $(a, r, z), (a, r', z')$ on different challenges $r \neq r'$, it is possible to extract a witness from (a, r, z, r', z'). For any such protocol, in the classical setting, it is possible to extract a witness from a cheating prover P^* as follows:

- Given a cheating prover P^*, the extractor first generates a single transcript (a, r, z) by running P^* to obtain a, and then running it on a random r to get z. If the transcript is rejecting, the extractor gives up.
- If the transcript is accepting, the extractor rewinds P^* to the point after a was sent, and then repeatedly sends i.i.d. challenges $r_1, r_2, \ldots$ until P^* produces *another* accepting transcript.

As long as the prover has significantly greater than $2^{-\lambda}$ probability of convincing the verifier, the second accepting transcript (a, r', z') produced will satisfy $r \neq r'$ with all but negligible probability, and thus a witness can be computed. In other words, this extractor *guarantees* (with all but negligible probability) that a witness is extracted conditioned on an initial accepting execution. Moreover, for *any* efficient P^*, the expected runtime of this procedure is $\mathrm{poly}(\lambda)$, since if P^* (with some fixed random coins) is convincing with probability p, the expected number of rewinds in this procedure is $\frac{1}{p}$ and thus the overall expected number of rewinds is $p \cdot \frac{1}{p} = 1$.

In the quantum setting, one might hope for a similar "guaranteed" extractor, but prior works [Unr12, Unr16b, CMSZ21] fail to achieve this. Indeed, [Unr12, Page 32] explicitly asks whether something of this nature is possible.

Our first idea is to abstractly define a quantum analogue of this "guaranteed" extraction property and show that under certain conditions, it generically implies state-preserving extraction. Since the classical problem can only be solved in *expected* polynomial time, there is again an ambiguity in what the quantum efficiency notion should be. However, it turns out that there is no [CCLY21]-type impossibility result for the problem of guaranteed extraction, so we demand the stronger EQPT_m extraction efficiency notion.

Definition 2.2. (P_Σ, V_Σ) is a post-quantum proof of knowledge with *guaranteed extraction* if it has an extractor $\mathsf{Extract}^{P^*}$ of the following form.

- $\mathsf{Extract}^{P^*}$ first runs the cheating prover P^* to generate a (classical) first message a.
- $\mathsf{Extract}^{P^*}$ runs P^* coherently on the superposition $\sum_{r \in R} |r\rangle$ of all challenges to obtain a superposition $\sum_{r,z} \alpha_{r,z} |r, z\rangle$ over challenge-response pairs.[10]
- $\mathsf{Extract}^{P^*}$ then computes (in superposition) the verifier's decision $V(x, a, r, z)$ and measures it. If the measurement outcome is 0, the extractor gives up.
- If the measurement outcome is 1, run some quantum procedure $\mathsf{FindWitness}^{P^*}$ that outputs a string w.

We require that the following two properties hold.

[8] Throughout our discussion of proofs of knowledge, we focus on the case of 3- and 4-message protocols. We sometimes ignore the first verifier message vk in a 4-message protocol for notational convenience.

[9] This particular special soundness assumption is also for convenience; we later describe generalizations of special soundness for which we have results.

[10] In general, the response z will be entangled with the prover's state; here we suppress this dependence.

- **Correctness (guaranteed extraction):** The probability that the initial measurement returns 1 but the output witness w is invalid is $\mathrm{negl}(\lambda)$.
- **Efficiency:** For any QPT P^*, the procedure $\mathsf{Extract}^{P^*}$ is in EQPT_m.

We claim that under suitable conditions, this kind of guaranteed extraction *generically* implies state-preserving extraction, where the extractor will be EQPT_c rather than EQPT_m. We describe the simplest example of these conditions: when the NP language itself is in UP (i.e. witnesses are unique).

Lemma 2.3 (see Lemma 10.3)**.** *If (P_Σ, V_Σ) is a post-quantum proof of knowledge with guaranteed extraction for a language with unique witnesses, then (P_Σ, V_Σ) is a state-preserving proof of knowledge with EQPT_c extraction.*

Lemma 2.3 can be extended to higher generality. For example, informally:

1. We can also extract "partial witnesses" that are uniquely determined by the instance x.
2. We can extract undetectably when the first message a "binds" the prover to a single witness in the sense that the guaranteed extractor will only output this one witness (even if many others exist).
3. This can also be extended to certain protocols whose first messages are informally "collapse-binding" [Unr16b] to the witness.

These generalizations are formalized in Section 10 using the notion of a "witness-binding protocol" (Definition 10.2). In this overview, we give a proof for the "unique witness" setting.

Proof sketch. Let $\mathsf{Extract}^{P^*}$ be a post-quantum guaranteed extractor with associated subroutine $\mathsf{FindWitness}^{P^*}$. We will present an EQPT_c extractor $\overline{\mathsf{Extract}}^{P^*}$ that has the form of an EQPT_c computation (see Fig. 1) where the unitary U is a coherent implementation of $\mathsf{FindWitness}^{P^*}$.

Remark 2.4. *This is an oversimplification of our real state-preserving extractor. In particular, $\mathsf{Extract}^{P^*}$ as described in this overview does not fit the EQPT_c model because $\mathsf{FindWitness}^{P^*}$ is not necessarily an EQPT_m computation — its running time is only expected polynomial when viewed as a subroutine of $\mathsf{Extract}^{P^*}$, which runs $\mathsf{FindWitness}^{P^*}$ with some probability (which may be negligible) and moreover, only runs it on inputs consistent with the verifier decision $V(x, a, r, z) = 1$. In Section 10, we formally demonstrate that our state-preserving extractor is EQPT_c by showing that it can be written in the form of Fig. 1 where the unitary U is a coherent implementation of the EQPT_m procedure $\mathsf{Extract}^{P^*}$.*

Our (simplified) EQPT_c extractor $\overline{\mathsf{Extract}}^{P^*}$ is defined as follows.

- Given P^*, generate a first message a and superposition $\sum_{r,z} \alpha_{r,z} |r, z\rangle$ as in $\mathsf{Extract}^{P^*}$.
- Compute the verifier's decision bit $V(x, a, r, z)$ in superposition and then measure it. If the measurement outcome is 0, measure r, z and terminate, outputting $(a, r, z, w = \bot)$ along with the current prover state.
- If the measurement outcome is 1, let $|\psi\rangle_{\mathcal{H}}$ denote the current prover state. For simplicity, assume that $|\psi\rangle_{\mathcal{H}}$ includes the superposition over (r, z) and space to write the extracted witness. The next steps are:

- Run U on input $|\psi\rangle_{\mathcal{H}} \otimes |0\rangle_{\mathcal{B},\mathcal{W}}$.
- Measure the sub-register of $\mathcal{H}$ containing the witness w.
- Run $U^\dagger$.
- Measure the sub-register of $\mathcal{H}$ containing the current transcript r, z.
- Return (a, r, z, w) and the residual prover state (i.e., the rest of $\mathcal{H}$).

Extraction correctness follows from the correctness of $\mathsf{FindWitness}^{P^*}$. Moreover, one can see that $\overline{\mathsf{Extract}}^{P^*}$ is state-preserving by considering two cases:

- **Case 1:** The initial measurement returns 0. In this case, the transcript (r, z) is immediately measured, and the resulting (sub-normalized) state exactly matches the component of the post-interaction P^* view corresponding to when the verifier rejects.
- **Case 2:** The initial measurement returns 1. In this case, the procedure $\mathsf{FindWitness}^{P^*}$ would output a valid witness with probability $1 - \mathsf{negl}$, so the output register of $U(|\psi\rangle_{\mathcal{A}} \otimes |0\rangle_{\mathcal{B},\mathcal{W}})$ contains a valid witness with probability $1 - \mathsf{negl}$. Since we assumed that the language L is in UP, this witness register is actually *deterministic*, so measuring it is computationally (even statistically!) undetectable, and hence after applying $U^\dagger$ the resulting state $|\psi'\rangle$ is computationally indistinguishable from $|\psi\rangle$. Thus, the output of the extractor the measured witness w along with a view that is computationally indistinguishable from the view of P^* corresponding to when the verifier accepts.

This completes the proof sketch. □

How do we apply Lemma 2.3? We now describe how to instantiate Σ-protocols so that the reduction in Lemma 2.3 applies (see Section 10.2).

First, we note that the *un-repeated* variants of standard proofs of knowledge [GMW86, Blu86] are "witness-binding" in the informal sense of the generalization (2); an extractor run on such protocols will only output a witness consistent with the commitment string a. However, since the un-repeated protocols only have constant (or worse) soundness error, there is no guaranteed extraction procedure for them (even in the classical setting).

In order to obtain negligible soundness error, these protocols are typically repeated in parallel; in this case, we *do* show guaranteed extraction procedures, but the protocols *lose* the witness-binding property (2). This is because each "slot" of the parallel repetition may be consistent with a different witness, and the extractor has no clear way of outputting a canonical one. In this case, measuring the witness potentially disturbs the prover's state by collapsing it to be consistent with the measured witness, which would not happen in the honest execution.

We resolve this issue using *commit-and-prove*. Given a generic Σ-protocol for which we have a guaranteed extractor, we consider a modified protocol in which the prover sends a (collapsing or statistically binding) commitment $\mathsf{com} = \mathsf{Com}(w)$ to its NP-witness along with a Σ-protocol proof of knowledge of an opening of com to a valid NP-witness. When the extractor $\overline{\mathsf{Extract}}^{P^*}$ of Lemma 2.3 is applied to this protocol composition, the procedure $\mathsf{FindWitness}^{P^*}$ (which is run coherently as U) actually obtains *both* an NP witness w *and* an opening of com to w. Therefore, the collapsing property of Com says that w can be measured undetectably. In other words, the commit-and-prove compiler enforces a computational uniqueness property sufficient for Lemma 2.3 to apply.

It also turns out that the (original, unmodified) [GMW86] graph-nonisomorphism protocol can be viewed as using this commit-and-prove paradigm,[11] which is one way to understand the proof of Theorem 1.2.

Finally, we remark that this commit-and-prove compiler is the cause of the super-polynomial assumptions in Theorems 1.3 and 1.7. This is because in order to show that a commit-and-prove protocol remains witness-indistinguishable, it must be argued that the proof of knowledge does not compromise the hiding of Com, which we only know how to argue by simulating the proof of knowledge in superpolynomial time (and assuming that Com is superpolynomially secure). This issue does not arise when Com is statistically hiding and the Σ-protocol is statistically witness-indistinguishable.

2.3 Achieving Guaranteed Extraction

So far, we have reduced from state-preserving extraction to the problem of guaranteed extraction. We now describe how we achieve guaranteed extraction for a wide class of Σ-protocols. Informally, we require that the protocol satisfies two important properties in order to perform guaranteed extraction:

- **Collapsing:** Prover responses can be measured undetectably provided that they are valid.
- **k-special soundness:** It is possible to obtain a witness given k accepting protocol transcripts $(a, r_1, z_1, \ldots, r_k, z_k)$ with distinct r_i (for the same first prover message a).

Both of these restrictions can be relaxed substantially[12] (see Sections 3.5, 5 and 8.4 for more details), but we focus on this case for the technical overview.

Theorem 2.5 (See Theorem 8.2)**.** *Any public-coin interactive argument satisfying collapsing and k-special soundness is a post-quantum proof of knowledge with guaranteed extraction (in EQPT_m).*

We consider Theorem 2.5 to be an interesting result in its own right and expect it to be useful in future work. We now describe our proof of Theorem 2.5 over the course of several steps:

- We begin by describing an abstract template that generalizes the [CMSZ21] extraction procedure in Section 2.3.1. In this template, the extractor repeatedly (1) queries the adversary on i.i.d. random challenges and then (2) applies a "repair procedure" to restore the adversary's success probability.
- In Section 2.3.2, we describe a natural "first attempt" at guaranteed extraction based on the [CMSZ21] template.
- We then observe in Section 2.3.3 that the entire template is unlikely to achieve guaranteed extraction in expected polynomial time. Perhaps surprisingly (and unlike the classical setting), querying the adversary on i.i.d. challenges appears *too slow* for this extraction task.

[11] The verifier sends an instance-dependent commitment [BMO90, IOS97, MV03] of a bit to the prover (which is perfectly binding in the proof of ZK) and demonstrates knowledge of the bit and its opening.

[12] We highlight that the PoK subroutine in the [GMW86] graph non-isomorphism protocol is *not* collapsing; it is only collapsing onto its responses of 0 challenge bits; however, it turns out that this property is still sufficient to obtain guaranteed extraction for the subroutine (see Sections 5.3 and 8.4).

- In Section 2.3.4, we introduce a new extraction template in which the adversary is *entangled* with a superposition of challenges, and the challenge is only measured once the adversary is guaranteed to give an accepting response.

- While this new template is a promising idea, we are still far from achieving guaranteed extraction. For the rest of the overview (Sections 2.3.5 to 2.3.7), we outline several technical challenges in instantiating this approach, eventually leading to our final extraction procedure and analysis.

2.3.1 An Abstract [CMSZ21] Extraction Template

[CMSZ21] recently showed that protocols satisfying collapsing and k-special soundness are post-quantum proofs of knowledge. Unlike our setting of guaranteed extraction, the [CMSZ21] extractor $\mathsf{Extract}^{P^*}(x, \gamma)$ is given *as advice* an error parameter γ and and extracts from cheating provers P^* (that may have some initial quantum state) that are convincing with probability $\gamma^* \geq \gamma$. The extractor's success probability is roughly $\frac{\gamma}{2}$.

At a high level, our abstract template makes use of two core subroutines that we call $\mathsf{Estimate}$ and $\mathsf{Transform}$. We describe the correctness properties required of $\mathsf{Estimate}$ and $\mathsf{Transform}$ below, and also describe their particular instantiations in [CMSZ21].

Jordan's lemma and singular vector algorithms. Let $\Pi_{\mathsf{A}}, \Pi_{\mathsf{B}}$ be projectors on a Hilbert space $\mathcal{H}$ with corresponding binary projective measurements $\mathsf{A} = (\Pi_{\mathsf{A}}, \mathbf{I} - \Pi_{\mathsf{A}})$ and $\mathsf{B} = (\Pi_{\mathsf{B}}, \mathbf{I} - \Pi_{\mathsf{B}})$. Recall that Jordan's lemma [Jor75] states that $\mathcal{H}$ can be decomposed as a direct sum $\mathcal{H} = \bigoplus \mathcal{S}_j$ of two-dimensional invariant subspaces $\mathcal{S}_j$, where in each $\mathcal{S}_j$, the projectors Π_{A} and Π_{B} act as rank-one projectors $|v_{j,1}\rangle\langle v_{j,1}|$ and $|w_{j,1}\rangle\langle w_{j,1}|$.[13] The vectors $|v_{j,1}\rangle$ and $|w_{j,1}\rangle$ are also left and right singular vectors of $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$ with singular value $\sqrt{p_j}$, where $p_j := |\langle v_{j,1}|w_{j,1}\rangle|^2$. This decomposition allows us to define on $\mathcal{H}$ the projective measurement $\mathsf{Jor} = (\Pi_j^{\mathsf{Jor}})$ onto the Jordan subspaces $\mathcal{S}_j$ (i.e., image$(\Pi_j^{\mathsf{Jor}}) = \mathcal{S}_j$). For an arbitrary state $|\psi\rangle$, we define the $\mathsf{Jordan\ spectrum}$ of $|\psi\rangle$ to be the distribution of p_j induced by Jor.

We will make use of procedures $\mathsf{Estimate}, \mathsf{Transform}$ satisfying the following properties.

- The Jordan subspaces $\mathcal{S}_j$ are invariant[14] under $\mathsf{Estimate}^{\mathsf{A},\mathsf{B}}$ and $\mathsf{Transform}^{\mathsf{A},\mathsf{B}}$. Equivalently, $\mathsf{Estimate}^{\mathsf{A},\mathsf{B}}$ and $\mathsf{Transform}^{\mathsf{A},\mathsf{B}}$ should *commute* with Jor. This property is important for arguing about the output behavior of $\mathsf{Estimate}$ and $\mathsf{Transform}$ on arbitrary states.

- $\mathsf{Estimate}^{\mathsf{A},\mathsf{B}}$: on input $|S_j\rangle \in \mathcal{S}_j$, output $p \approx p_j$; the residual state remains in $\mathcal{S}_j$.

- $\mathsf{Transform}^{\mathsf{A},\mathsf{B}}$ maps each $|v_{j,1}\rangle$ to $|w_{j,1}\rangle$. We have no requirements on any other state in $\mathcal{S}_j$ except that it remains in $\mathcal{S}_j$.

[CMSZ21] implement a version of $\mathsf{Estimate}^{\mathsf{A},\mathsf{B}}$ (following [MW05]) with ε accuracy by alternating A and B for $t = \mathrm{poly}(\lambda)/\varepsilon^2$ steps. The output is $p = d/(t-1)$ where d is the number of occurrences of $b_i = b_{i+1}$ among the outcomes $b_1, b_2, \ldots, b_t$. With probability $1 - 2^{\lambda}$, we have $|p - p_j| \leq \varepsilon$. They

[13] There will also be one-dimensional subspaces, which we ignore in this overview since they can be viewed as "degenerate" two-dimensional subspaces.

[14] We allow for decoherence, so we ask that every element of $\mathcal{S}_j$ is mapped to a *mixed state* where every component is in $\mathcal{S}_j$.

(implicitly) implement $\mathsf{Transform}^{\mathsf{A},\mathsf{B}}$ by alternating measurements A and B back and forth until $\mathsf{B} \to 1$, with an expected running time of $O(1/p_j)$ on $\mathcal{S}_j$.

The [CMSZ21] Extractor. We now use the abstract procedures $(\mathsf{Estimate}, \mathsf{Transform})$ to describe (a slightly simplified version of) the [CMSZ21] extractor. Let $|+_R\rangle_{\mathcal{R}}$ denote the uniform superposition over challenges on register $\mathcal{R}$ and let $\mathcal{H}$ denote the register containing the prover's state. Let $\mathsf{V}_r = (\Pi_{V,r}, \mathbf{I} - \Pi_{V,r})$ denote a binary projective measurement on $\mathcal{H}$ that measures whether P^* returns a valid response on r.

The extraction technique makes crucial use of two measurements: the first is $\mathsf{U} = (\Pi_{\mathsf{U}}, \mathbf{I} - \Pi_{\mathsf{U}})$, where $\Pi_{\mathsf{U}} := \mathbf{I}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|_{\mathcal{R}}$ is the projective measurement of whether the challenge register $\mathcal{R}$ is *uniform*. The second is $\mathsf{C} = (\Pi_{\mathsf{C}}, \mathbf{I} - \Pi_{\mathsf{C}})$, where $\Pi_{\mathsf{C}} := (\Pi_{V,r_i})_{\mathcal{H}} \otimes \sum_{r \in R} |r\rangle\langle r|_{\mathcal{R}}$ is the projective measurement that runs the prover on the challenge on $\mathcal{R}$ and *checks* whether the prover wins. The extraction procedure is described in Fig. 3 below.

Figure 3: The [CMSZ21] extractor with generic procedures $\mathsf{Estimate}, \mathsf{Transform}$

1. Generate a first verifier message vk and run $P^*(\mathsf{vk}) \to a$ to obtain a classical first prover message a once and for all. Let $|\psi\rangle$ denote the state of P^* after it returns a.

2. Run $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ to accuracy $\gamma/4$ on $|\psi\rangle\,|+_R\rangle$, which outputs an estimate p of the adversary's success probability and then discard $\mathcal{R}$;[a] abort if $p < \gamma/2$ (this occurs with probability at most $1-\gamma/2$). Subtract $\gamma/4$ from p so that p represents a reasonable lower bound on the success probability. Set an error parameter $\varepsilon = \frac{\gamma^2}{2\lambda k}$ for the rest of the procedure and fix $N = \lambda k/p$.

3. We now want to generate k accepting transcripts. For i from 1 to N:

   (a) Sample a uniformly random challenge r_i and apply V_{r_i} to the current state $|\psi_i\rangle$.

   (b) If the output is $b_i = 1$, measure the response z. This is (computationally) undetectable by the protocol's collapsing property, so we ignore this step **for now**.

   (c) Let E be a unitary such that applying E to $\mathcal{H} \otimes \mathcal{W}$ (where $\mathcal{W}$ is an appropriate-size ancilla initialized to $|0\rangle_{\mathcal{W}}$) and then discarding $\mathcal{W}$ is equivalent to running $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ for $\lambda p/\varepsilon^2$ steps on $\mathcal{H} \otimes \mathcal{R}$ (where $\mathcal{R}$ is initialized to $|+_R\rangle_{\mathcal{R}}$) and then discarding $\mathcal{R}$.

   We *repair* the success probability by initializing $\mathcal{W} = |0\rangle_{\mathcal{W}}$ and then running $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ on $\mathcal{H} \otimes \mathcal{W}$ where, roughly speaking, D is a projective measurement corresponding to the *disturbance* caused by step (a), and G is a projective measurement that determines whether the adversary's success probability is *good*, meaning at least $p - \varepsilon$. More precisely:

   - $\mathsf{G} = (\Pi_{p,\varepsilon}, \mathbf{I} - \Pi_{p,\varepsilon})$ returns 1 if, after applying E, the estimate is at least $p - \varepsilon$.[b]
   - $\mathsf{D} = (\Pi_{r_i,b_i}, \mathbf{I} - \Pi_{r_i,b_i})$ returns 1 if $\mathcal{W} = |0\rangle_{\mathcal{W}}$ *and* applying V_{r_i} returns b_i.

   If $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ has not terminated within T calls to D and G, abort (this occurs with probability at most $O(1/T)$). Otherwise, apply E, trace out $\mathcal{W}$, re-initialize

$\mathcal{R}$ to $|+_R\rangle$ and then run $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ for $\lambda p/\varepsilon^2$ steps to obtain a new probability estimate p'. If $p' < p - 2\varepsilon$, abort. Finally, discard $\mathcal{R}$ and re-define $p := p'$.

[a] We assume here that when we run $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ on a state $|\phi\rangle \in \text{image}(\Pi_{\mathsf{U}})$, the residual state is in $\text{image}(\Pi_{\mathsf{U}})$. Then we are guaranteed that $\mathcal{R}$ is unentangled from $\mathcal{H}$, which allows us to discard it. In our eventual construction, this assumption is enforced in Theorem 6.2.

[b] In our actual construction/proof, we replace this call to $\mathsf{Estimate}$ (and the additional call at the end of Step 3c) with a weaker primitive that *only* computes the threshold instead of fully estimating p. This change makes it easier to instantiate the primitive.

2.3.2 Guaranteed extraction, first attempt

The [CMSZ21] algorithm, interpreted in terms of the abstract procedures $\mathsf{Estimate}, \mathsf{Transform}$, will serve as our initial template for extraction. We now consider whether it can be modified to achieve *guaranteed* extraction.

Syntactic Changes. The first issues with the [CMSZ21] extraction procedure are syntactic in nature. Namely, we want an extraction procedure that works for *any* P^*, with no a priori lower bound γ on the success probability of P^*. Of course, an extractor $\mathsf{Extract}^{P^*}$ that extracts with probability close to 1 given an arbitrary P^* is impossible to achieve (imagine a P^* with negligible success probability), so the game is also changed as described in Definition 2.2. In terms of the [CMSZ21] template, the change is as follows:

- After obtaining $(a, |\psi\rangle)$, measure C on $|\psi\rangle\, |+_R\rangle$ and terminate if the outcome is 0.
- Otherwise, the state is (re-normalized) $\Pi_{\mathsf{C}}(|\psi\rangle\, |+_R\rangle)$, and the goal is to extract with probability $1 - \text{negl}$.

Variable-Runtime Estimation. Since we are given no *a priori* lower bound γ on the success probability of P^*, there is no fixed additive precision ε for which the initial $\mathsf{Estimate}$ in Step 2 guarantees successful extraction — the initial state $|\psi\rangle\, |+_R\rangle$ could be concentrated on subspaces $\mathcal{S}_j$ such that $p_j \ll \varepsilon$, in which case the estimation procedure almost certainly returns 0.

To remedy this issue, we define a *variable-length* variant of $\mathsf{Estimate}^{\mathsf{A},\mathsf{B}}$ with the guarantee that for every j and every state in $\mathcal{S}_j$, $\mathsf{Estimate}^{\mathsf{A},\mathsf{B}}$ returns p_j to within constant (factor 2) *multiplicative* accuracy with probability $1-2^{-\lambda}$. With regard to instantiation, we note that the [MW05, CMSZ21] implementation of $\mathsf{Estimate}^{\mathsf{A},\mathsf{B}}$ can be modified to be variable-length: simply continue alternating Π_A, Π_B until sufficiently many ($d = \text{poly}(\lambda)$) $b_i = b_{i+1}$ occur, so that the estimate $\frac{d}{t-1}$ (where t is the number of measurements performed) is reasonably concentrated around its expectation.

Thus, we begin with the natural idea that Step 2 should be modified to use this variable-length $\mathsf{Estimate}$. We remark that variable-length $\mathsf{Estimate}$ is *not* required in later steps: the output p of Step 2 can be used to set the parameters (ε, N) for the rest of the procedure.

With this modification, our extractor *never* aborts in Step 2, but it also no longer runs in strict polynomial time. How do we analyze its runtime? First, one can compute that when run on a state in $\mathcal{S}_j$, the expected running time of this procedure is (up to factors of $\text{poly}(\lambda)$) roughly $\frac{1}{p_j}$. This might seem concerning, because this expectation could be large (even superpolynomial) if p_j is very small. However, what we care about is the runtime of $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ on the (re-normalized) state $\Pi_{\mathsf{C}}(|\psi\rangle\, |+_R\rangle)$. Writing $|\psi\rangle\, |+_R\rangle = \sum_j \alpha_j\, |v_{j,1}\rangle$, we see that $\Pi_{\mathsf{C}}(|\psi\rangle\, |+_R\rangle) = \sum_j \alpha_j \sqrt{p_j}\, |w_{j,1}\rangle$.

To calculate the overall expected runtime, we use the fact that $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ commutes with the projective measurement Jor that outputs j on each subspace $\mathcal{S}_j$. This implies that the expected runtime of $\mathsf{Estimate}$ on our state is the weighted linear combination of its expected runtime on the eigenstates $|v_{j,1}\rangle$, namely

$$\frac{1}{\gamma^*}\sum_j |\alpha_j|^2 p_j \cdot \frac{1}{p_j} = \frac{1}{\gamma^*},$$

where $\gamma^* = ||\Pi_{\mathsf{C}}(|\psi\rangle\, |+_R\rangle)||^2$ is the probability that $\mathsf{C} \to 1$ in the initial execution.[15] Thus, the overall expected runtime equals $\gamma^* \cdot \frac{1}{\gamma^*} = 1$, so Step 2 of the procedure is efficient!

Our first attempt. With the changes above, Step 2 of the extraction procedure now has zero error and runs in expected polynomial time (EQPT_m).

The other source of non-negligible extraction error from [CMSZ21] is in the cutoff T imposed on $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$. By removing this cutoff, we obtain a procedure that is somewhat closer to the goal of guaranteed extraction in expected polynomial time, described in Fig. 4 below.

Figure 4: Guaranteed extraction (Attempt 1)

1. After obtaining $(a, |\psi\rangle)$, apply C to $|\psi\rangle\, |+_R\rangle$ and terminate if the measurement returns 0. Otherwise, let $|\phi\rangle$ denote the resulting state on $\mathcal{H} \otimes \mathcal{R}$.
2. Run the variable-length $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ on $|\phi\rangle$, obtaining output p. Divide p by 2 to obtain a lower bound on the resulting success probability. Set $\varepsilon = \frac{p^2}{2\lambda k}$ and $N = \lambda k/p$.
3. Run Step 3 of the original [CMSZ21] extractor as in Fig. 3, with the parameters p, ε, N. Instead of imposing a time limit T, the procedure $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ is allowed to run until completion[a] ($\mathsf{G} \to 1$).

[a] To avoid a computation that runs for infinite time, one should at the very least impose an exponential 2^λ time cutoff, which can be shown to incur only a $2^{-\lambda}$ correctness error.

2.3.3 Problem: Step 3 is not expected poly-time.

Unfortunately, the "first attempt" above does *not* satisfy Definition 2.2. The issue lies in its runtime: we argued before that over the randomness of $\mathsf{Extract}^{P^*}$, Step 2 runs in expected polynomial time. However, we did not analyze Step 3, which is the main loop for generating transcripts. Here is a rough estimate for its runtime.

Recall that Step 3 loops the following steps for each $i = 1, \ldots, \lambda k/p$:

- Run the prover P^* on a random challenge r_i. This takes a fixed $\mathrm{poly}(\lambda)$ amount of time.
- Then, *regardless* of whether P^* was successful, the residual prover state $|\phi_i\rangle$ must be *repaired* to have success probability $\approx p$.

It turns out that as currently written, the expected runtime of the repair step is (up to $\mathrm{poly}(\lambda)$ factors) equal to the runtime of a fixed-length $\mathsf{Estimate}$ procedure with precision $\approx p^2$ (this ensures

[15] One way to see this is to notice that applying Jor after running $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ clearly cannot affect the runtime of $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$. Then Jor can be commuted to occur before $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$.

that after $1/p$ repair steps, the total success probability loss must be at most p). Moreover, this runtime is intuitively necessary for *any* possible repair procedure, since repairing the success probability should be at least as hard as computing whether it is above the acceptable threshold.

In our setting, the [MW05, CMSZ21] estimation procedure requires $1/p^3$ time to obtain a p^2-accurate estimate in the relevant parameter regime.[16] Since Step 3 performs this loop $\frac{1}{p}$ times (omitting the λk factor), the total runtime will be at least $\frac{1}{p^4}$. This is too long for the "conditioning" of Step 1 to save us: if the initial state at the beginning of Step 1 is $|\psi\rangle \, |+_R\rangle \in \mathcal{S}_j$, the expected runtime of Step 3 is $p_j \cdot \frac{1}{p_j^4} = \frac{1}{p_j^3}$, which can be arbitrarily large (when p_j is small).

Idea: Use a faster $\mathsf{Estimate}$? Given how we have phrased the extractor in terms of abstract $(\mathsf{Estimate}, \mathsf{Transform})$ algorithms, a natural idea for improving the runtime is to use an implementation of the abstract $\mathsf{Estimate}$ algorithm that is faster than the [MW05]-based one used in [CMSZ21]. Indeed, if we use the procedure described in [NWZ09] to implement $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$, we obtain a quadratic speedup: the runtime of $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ in Step 3c can be improved from $\frac{1}{p^3}$ to $\frac{1}{p^{3/2}}$.

This speedup will be relevant to our eventual solution, but it does not resolve the problem. The back-of-the envelope calculation now just says that the expected runtime of Step 3 on a state $|\psi\rangle \, |+_R\rangle \in \mathcal{S}_j$ is $p_j \cdot p_j^{-5/2} = p_j^{-3/2}$, which is still unbounded.

So are we doomed? Indeed, this runtime calculation seems problematic for the *entire* [CMSZ21] template that we abstracted, by the following reasoning:

- On a state with initial estimate p, each choice of r_i will only produce an accepting transcript with probability $\approx p$, so we must try $\approx k/p$ choices of i.i.d. r_i to obtain k accepting transcripts.
- Therefore, as long as the repair step takes **super-constant time** (as a function of $1/p$), the overall extraction procedure will take too long.

This seems to indicate a dead end for extractors that follow the standard rewinding template of repeatedly running P^* on random r to obtain accepting transcripts.

2.3.4 Solution: A New Rewinding Template

We solve our unbounded runtime issue by abandoning "classical" rewinding, in the following sense: unlike prior extraction procedures [Unr12, CMSZ21], our extractor will *not* follow the standard approach of obtaining transcripts by feeding uniformly random r_i to P^*. Instead, we will *generate* accepting transcripts (r_i, z_i) via an inherently quantum procedure so that *every* generated transcript is accepting (as opposed to only a p fraction of them).

We accomplish this by using the procedure $\mathsf{Transform}$, which was previously only used for state repair, to *generate* the transcripts. Consider a prover state $|\psi_i\rangle$ at the beginning of Step 3. By definition, $|\psi_i\rangle \, |+_R\rangle \in \mathrm{image}(\Pi_{\mathsf{U}})$, so applying $\mathsf{Transform}^{\mathsf{U},\mathsf{C}}$ to $|\psi_i\rangle \, |+_R\rangle$ produces a state in $\mathrm{image}(\Pi_{\mathsf{C}})$. Now if the challenge register $\mathcal{R}$ is *measured* (obtaining a string r_i), the residual prover state is *guaranteed* to produce an accepting response on r_i!

[16] As written in [CMSZ21], the estimation procedure runs in $1/p^4$ time, but a factor of p can be saved because (roughly speaking) the estimate only needs to achieve p^2 accuracy when p_j is close to p.

Moreover, the extraction procedure can afford to run $\mathsf{Transform}^{\mathsf{U},\mathsf{C}}$: since $|\psi_i\rangle\,|+_R\rangle$ has been constructed to lie almost entirely in subspaces $\mathcal{S}_j$ such that $p_j < p - \varepsilon$, the expected running time of $\mathsf{Transform}^{\mathsf{U},\mathsf{C}}$ can be shown to be roughly[17] $\frac{1}{p}$.

This gives us a potential *new* template for extraction: we modify the main loop (Step 3) as in Fig. 5.

Figure 5: Our new extraction template

1. After obtaining $(a, |\psi\rangle)$, apply C to $|\psi\rangle\,|+_R\rangle$ and terminate if the measurement returns 0. Otherwise, let $|\phi\rangle$ denote the resulting state on $\mathcal{H} \otimes \mathcal{R}$.
2. Run the variable-length $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ on $|\phi\rangle$, obtaining output p. Divide p by 2 to obtain a lower bound on the resulting success probability. Set $\varepsilon = \frac{p}{4k}$.
3. For i from 1 to k:
   (a) Given current prover state $|\psi_i\rangle$, apply $\mathsf{Transform}^{\mathsf{U},\mathsf{C}}$ to $|\psi_i\rangle\,|+_R\rangle$. Call the resulting state $|\phi_{\mathsf{C}}\rangle$.
   (b) Obtain a *guaranteed accepting* transcript (r_i, z_i) by measuring the $\mathcal{R}$ register of $|\phi_{\mathsf{C}}\rangle$ and then running P^* on r_i. As before, measuring z_i is computationally undetectable.
   (c) Run the Repair Step (3c) as in Fig. 3 by calling $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ and re-estimating p.

We emphasize two crucial efficiency gains from this new extraction template:

- As already mentioned, the main loop now has k steps instead of k/p, since each transcript is now guaranteed to be accepting.
- Since only k repair operations are now required, the *error* parameter ε for $\Pi_{p,\varepsilon}$ can be set to $\approx p$ instead of $\approx p^2$.

Correctness Analysis. We remark that even the correctness of this new extraction procedure is unclear. In the case of k-special sound protocols, we need the extraction procedure to produce k accepting transcripts with distinct r_i; previously, this was guaranteed because each r_i was sampled i.i.d., so (w.h.p.) no pair of them coincide. Here, r_i is *not* uniformly random — it has been sampled by measuring the $\mathcal{R}$ register of some state in Π_{C}.

In order to analyze the behavior of this extractor, it is important to understand the state $|\phi_{\mathsf{C}}\rangle$ obtained after applying $\mathsf{Transform}^{\mathsf{U},\mathsf{C}}$. Of course, we have an explicit representation $\sum_j \alpha_j \sqrt{p_j}\,|w_{j,1}\rangle$ for it, but it is not clear a priori how this helps.

To prove correctness, we analyze the state $|\phi_{\mathsf{C}}\rangle$ using what we call the Pseudoinverse Lemma (Lemma 7.1), which states that $|\phi_{\mathsf{C}}\rangle$ can be viewed as a *conditional* state obtained by starting with a state $|\phi_{\mathsf{U}}\rangle = |\psi_{\mathsf{U}}\rangle\,|+_R\rangle \in \text{image}(\Pi_{\mathsf{U}})$ and *post-selecting* (i.e., conditioning) on a C-measurement of $|\phi_{\mathsf{U}}\rangle$ outputting 1. Crucially, this pseudoinverse state has a precisely characterized (U,C)-Jordan spectrum related to the Jordan spectrum of $|\phi_C\rangle$. We emphasize that the state $|\phi_{\mathsf{U}}\rangle$ does not actually exist in the extraction procedure; it is just a tool for the analysis.

[17] For technical reasons, we cut off $\mathsf{Transform}$ after an exponential number of steps so that the component of $|\psi_i\rangle\,|+_R\rangle$ lying in "bad" $\mathcal{S}_j$ (i.e., where p_j is tiny) does not ruin the expected running time.

Using the pseudoinverse lemma, one can show that the probability a C-measurement of $|\phi_{\mathsf{U}}\rangle$ returns 1 is $\approx p$, which implies that the joint distribution of $(r_1, \ldots, r_k)$ comes from a "random enough" distribution that we formalize as "admissible" (Definition 5.5). This is shown by the following reasoning: since measuring $\mathcal{R}$ commutes with C, it is as if we have an initially uniformly random r_i (obtained from measuring $\mathcal{R}$ of $|\phi_{\mathsf{U}}\rangle$) that is "output" with probability $\approx p$ (when C returns 1). This is sufficient to argue about correctness properties of the extractor.

Runtime Analysis Idea. Analyzing the runtime of $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ also turns out to be significantly more subtle than in the [CMSZ21] setting. The basic idea is to show that (within a reasonable amount of time) $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ *returns* a state on $\mathcal{H} \otimes \mathcal{W}$ to image$(\Pi_{p,\varepsilon})$ after it was "initially" disturbed by the binary measurement D. In [CMSZ21], this is literally true: the disturbance is measuring $(\Pi_{V,r}, \mathbf{I} - \Pi_{V,r})$ for randomly sampled r on the prover state $|\psi_i\rangle$. One can then show that an expected constant number of (D, G)-measurements returns the state to G by appealing to the statistics of the (D, G) Marriott-Watrous distribution.

However, in our setting, the "disturbance" is quite different: the amplified state $|\phi_{\mathsf{C}}\rangle \in$ image(Π_{C}) consists of a prover state *entangled with* the challenge register $\mathcal{R}$ in a way that is *guaranteed* to produce an accepting transcript. $|\phi_{\mathsf{C}}\rangle$ is then disturbed by measuring its $\mathcal{R}$ register, and the measurement D being applied in $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ depends on this $\mathcal{R}$ measurement outcome. Since the $\mathcal{R}$ measurement can disturb $|\phi_{\mathsf{C}}\rangle$ by a large amount (unlike D), it is not a priori clear why $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ should return the state to image$(\Pi_{p,\varepsilon})$.

At a high level, we show how to bound the runtime of this new procedure by appealing to the pseudoinverse state $|\phi_{\mathsf{U}}\rangle$, again! In more detail, using the pseudoinverse lemma, the state on $\mathcal{H} \otimes \mathcal{W}$ obtained after measuring $\mathcal{R}$ on $|\phi_{\mathsf{C}}\rangle$ (along with initializing $\mathcal{W}$ to $|0\rangle$) can be alternatively thought of as the state obtained by:

- Sampling r_i proportional to the probability ζ_{r_i} of $|\psi_{\mathsf{U}}\rangle$ successfully answering r_i, and
- Outputting (normalized) $\Pi_{r_i}(|\psi_{\mathsf{U}}\rangle \otimes |0\rangle_{\mathcal{W}})$, where $\Pi_{r_i} := \Pi_{r_i,1}$.

This conditioning argument allows us to appeal to the same "return to $\Pi_{p,\varepsilon}$" principle to show that $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ indeed "returns" the state to image$(\Pi_{p,\varepsilon})$, as if it had "started out" as the state $|\phi_{\mathsf{U}}\rangle\,|0\rangle_{\mathcal{W}}$, which only exists in the analysis!

2.3.5 Problem: Step 3 is *still* not expected poly-time.

The premise of our new extraction template was to speed up the extraction process by getting rid of excess work from running state repair in situations where no accepting transcript was obtained. Previously, we computed the expected runtime to perform $N \approx k/p$ repair steps in Fig. 3 (conditioned on a successful initial execution and initial estimate p) to be $pN/\varepsilon^2 \approx 1/p^4$, since the runtime of each repair step was equivalent (up to a constant factor) to the runtime of G, which was p/ε^2, and $\varepsilon \approx p^2$. As noted above, with our new template we now only have to perform $N = k$ repair steps, and the error parameter ε can now be $\approx p$. With these improvements alone, one might hope to perform N repair steps in $pN/\varepsilon^2 = p(k)(1/p^2) \approx 1/p$ time. This would result in expected polynomial runtime for the overall extractor when factoring in the conditioning.

Perhaps surprisingly, the above reasoning is incorrect! This new extraction procedure is *still* not expected QPT: the expected runtime of N repair steps will be $\approx \frac{1}{p^2}$, not $\frac{1}{p}$.

Why does this happen? It turns out that in this new extraction template, each repair step (which previously made expected $O(1)$ calls to G) must now make an expected $O(1/p)$ calls to G, cancelling out the factor-$1/p$ savings in N obtained by using $\mathsf{Transform}^{\mathsf{U},\mathsf{C}}$ to generate transcripts.

Indeed, the pseudoinverse-based runtime analysis above for $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ implies that each repair step must now make

$$\frac{1}{\zeta_R}\sum_r \zeta_r \cdot \frac{1}{\zeta_r} = \frac{1}{\zeta_R} \approx 1/p$$

calls to G (where $\zeta_R = \sum_r \zeta_r \approx p$ is the normalization factor for the r_i-distribution). This results in an overall expected running time of $\frac{1}{p}$ calls to G if p was initially measured. Essentially, this is saying that while obtaining an accepting transcript (r_i, z_i) causes *limited enough* disturbance that repair can work, it causes more disturbance than a binary measurement, resulting in a factor of $1/p$ increase in the repair time.

2.3.6 Solution: Use faster Estimate and Transform

Despite the less-than-expected speedup observed in Section 2.3.5, it turns out that we nevertheless made significant progress. The reason is that the bottleneck to obtaining a faster extraction procedure is now in the running times of $\mathsf{Estimate}$ and $\mathsf{Transform}$, so we can hope to obtain an expected polynomial time procedure by using faster algorithms for $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ and $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$.

As discussed above, speeding up the fixed-length $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ in G is relatively straightforward by appealing to [NWZ09];[18] this results in an expected running time of $\frac{1}{\sqrt{p}}$ for G.

However, implementing a *fast* version of $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ achieving $1-\mathrm{negl}(\lambda)$ correctness (which is required for our extraction procedure to have negligible error) is less straightforward. Some implementations in the literature (e.g., [GSLW19]) achieve this correctness guarantee, but only given a known (inverse polynomial) lower bound on the eigenvalue q_j (associated with (D,G)-Jordan subspace $\mathcal{T}_j$). We have no such lower bound for our state $\frac{1}{\zeta_r}\Pi_r(|\phi_{\mathsf{U}}\rangle\,|0\rangle_{\mathcal{W}})$. Our resolution is to first apply a variable-length fast phase estimation algorithm (implemented by repeatedly running [NWZ09] to increasing precision, or singular value discrimination [GSLW19] with decreasing thresholds, until we obtain a multiplicative estimate of the phase) and then run a fixed-length fast $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ using the estimated phase to lower bound the eigenvalue. The fixed-length fast $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ can be done using [GSLW19]; it is also possible to use a more elementary algorithm combining fast amplitude amplification [BHMT02] with ideas from [Wat06] for achieving $1-\mathrm{negl}(\lambda)$ correctness.

To summarize, we obtain a final $1/p$ speedup by combining a $1/\sqrt{p}$ speedup from using a faster $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ with a $1/\sqrt{p}$ speedup from using a faster $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$. The fact that the latter speedup is actually realized turns out to be subtle to argue.

2.3.7 Last Problem: Measuring z ruins the runtime guarantee

Unfortunately, we are *still* not done! There is one subtle issue with our extractor that we have ignored so far: our runtime analysis was only valid ignoring the effect of measuring the prover response z. Since all transcripts after running $\mathsf{Transform}^{\mathsf{U},\mathsf{C}}$ are accepting by construction, the collapsing property of the protocol implies that measuring z is computationally undetectable, so one might assume that the runtime analysis extends immediately.

[18] For technical reasons, we use a different algorithm due to [GSLW19], but a variant of [NWZ09] would also suffice.

However, the *expected running time* of an algorithm is not an efficiently testable property of the input state. This is not just an issue with our proof strategy: the version of the above extractor where z is measured does not run in expected polynomial time.

In a nutshell, the issue is that a computationally undetectable measurement can still cause a state's eigenvalues (either $\{p_j\}$, in $\mathsf{Jor}^{\mathsf{U},\mathsf{C}}$, or $\{q_j\}$, in $\mathsf{Jor}^{\mathsf{G},\mathsf{D}}$) to change by a negligible but nonzero amount, affecting the subsequent runtime of $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$. This negligible change can have an enormous effect on the expected runtime of the extractor, because if the runtime of a procedure is inversely proportional to the disturbed eigenvalue $\tilde{p} = p - \mathrm{negl}$, an overall expected runtime expression can now contain terms of the form $\frac{p}{p-\mathrm{negl}}$, which can be unbounded when p is also negligible. Interestingly, such issues have long been known to exist in the *classical* setting: these $\frac{p}{p-\mathrm{negl}}$ terms are the major technical difficulty in obtaining a classical simulator for the [GK96] protocol. This classical analogy inspires our resolution.

Solution: Estimate repair time before measuring z. We modify our extractor so that in each loop iteration, all procedures occurring after the z-measurement have a *pre-determined* runtime. Previously, after z was measured, we ran a fast variable-length $\mathsf{Transform}$ by running a the variable-length $\mathsf{Estimate}^{\mathsf{D},\mathsf{G}}$ to determine a time bound t, and then running a t-time $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$. Instead of this, we will run $\mathsf{Estimate}^{\mathsf{D},\mathsf{G}}$ *before* z is measured. This allows us to compute a runtime bound for $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ before the z measurement disturbs the state, preserving the expected running time of the entire procedure. This results in the final extraction procedure described in Fig. 6 below.

Figure 6: Our final extraction procedure

1. After obtaining $(a, |\psi\rangle)$, apply C to $|\psi\rangle\,|+_R\rangle$ and terminate if the measurement returns 0. Otherwise, let $|\phi\rangle$ denote the resulting state on $\mathcal{H} \otimes \mathcal{R}$.
2. Run the variable-length $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ on $|\phi\rangle$, obtaining output p. Divide p by 2 to obtain a lower bound on the resulting success probability. Set $\varepsilon = \frac{p}{4k}$ and $N = k$.
3. For i from 1 to N:
   (a) Given prover state $|\psi_i\rangle$, apply $\mathsf{Transform}^{\mathsf{U},\mathsf{C}}\,|\psi_i\rangle\,|+_R\rangle$. Call the resulting state $|\phi_{\mathsf{C}}\rangle$.
   (b) Measure (and discard) the $\mathcal{R}$ register of $|\phi_{\mathsf{C}}\rangle$ to obtain a classical challenge r_i.
   (c) Initialize $\mathcal{W}$ to $|0\rangle_{\mathcal{W}}$ and call the variable-length $\mathsf{Estimate}^{\mathsf{D},\mathsf{G}}$, which outputs a value q. We require that the output state is in the image of Π_{r_i}.
   (d) Measure the response z_i.
   (e) We repair the success probability by running $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ on $\mathcal{H} \otimes \mathcal{W}$ for $\frac{\lambda}{\sqrt{q}}$ oracle steps. If the resulting state is not in the image of $\Pi_{p,\varepsilon}$, abort.
   Trace out $\mathcal{W}$ and run $\mathsf{Estimate}^{\mathsf{U},\mathsf{C}}$ for $\lambda\sqrt{p}/\varepsilon$ steps to obtain a new probability estimate p'. If $p' < p - 2\varepsilon$, abort. Finally, discard $\mathcal{R}$ and re-define $p := p'$.

By making this change, we incur an additional *correctness* error for the extractor, because the collapsing measurement may decrease the probability that $\mathsf{Transform}^{\mathsf{D},\mathsf{G}}$ successfully maps the state to $\Pi_{p,\varepsilon}$. However, this error is negligible because this correctness property is efficiently checkable (unlike the expected runtime). Thus, this procedure achieves both expected polynomial runtime[19]

[19] It remains to be argued that measuring z_j does not affect the running time of *subsequent* variable-runtime steps.

and the desired correctness guarantees.

2.3.8 Putting everything together

To summarize, we gave a new extraction template along with a particular instantiation that achieves expected polynomial runtime, by leveraging four different algorithmic improvements:

1. By generating accepting transcripts with $\mathsf{Transform}^{\mathsf{U},\mathsf{C}}$, we now only have to generate k transcripts and repair k prover states (instead of k/p).
2. (1) allows us to relax the error parameter ε by a factor of $1/p$ (speeding up G).
3. Using a fast algorithm for $\mathsf{Estimate}$ from the literature [NWZ09, GSLW19] saves a factor of $1/\sqrt{p}$ runtime.
4. Using a new fast, variable-runtime algorithm for $\mathsf{Transform}$ saves another factor of $1/\sqrt{p}$.

Finally, we implement the variable-length $\mathsf{Transform}$ in two phases (variable-length phase estimation followed by fixed-length $\mathsf{Transform}$) and interleave the measurement of the response z between them, so that this z-measurement has no effect on the runtime.

We remark that the overall analysis of our extractor is rather involved (as we have omitted additional details in this overview); we refer the reader to Section 8 for a full analysis.

2.4 Post-Quantum ZK for [GK96]

In this section we give an overview of our proof that the Goldreich–Kahan (GK) protocol is post-quantum zero-knowledge (Theorem 1.4). Our simulator makes use of some of the techniques described in Section 2.3, but the simulation strategy is quite different to our other results. In particular, our simulator does *not* make use of state-preserving extraction.

We first recall the Goldreich–Kahan construction of a constant-round zero-knowledge proof system for NP. Let (P_Σ, V_Σ) be a Σ-protocol for NP satisfying special honest verifier zero knowledge (SHVZK)[20] and let Com be a statistically hiding, computationally binding commitment. [GK96] construct a zero knowledge protocol (P, V) as described in Fig. 7.

Soundness of the [GK96] protocol holds against *unbounded* P^* and therefore extends immediately to the quantum setting.

Recap: the naïve classical simulator. As observed by [GK96], there is a natural *naïve simulator* for their protocol that, for reasons analogous to Section 2.3.7, turns out to have an unbounded expected runtime. To build intuition for our quantum simulation strategy, we will first recall the naïve classical simulator and show how to extend it to a naïve quantum simulator (while temporarily ignoring the runtime issue). Then, by using the technique described in Section 2.3.7, we will improve this to a full EQPT_c quantum simulator.

The naïve classical simulator does the following:

This turns out to hold because the runtime of future loop iterations can be guaranteed by the correctness properties of the re-estimation step, which hold for an *arbitrary* re-estimation input state.

[20] Recall that the special honest-verifier zero-knowledge property guarantees the existence of a randomized simulation algorithm $\mathsf{SHVZK.Sim}(r)$ that takes any Σ-protocol challenge $r \in R$ as input and outputs a tuple (a, z) such that the distribution of (a, r, z) is indistinguishable from the distribution of transcripts arising from an honest prover interaction on challenge r.

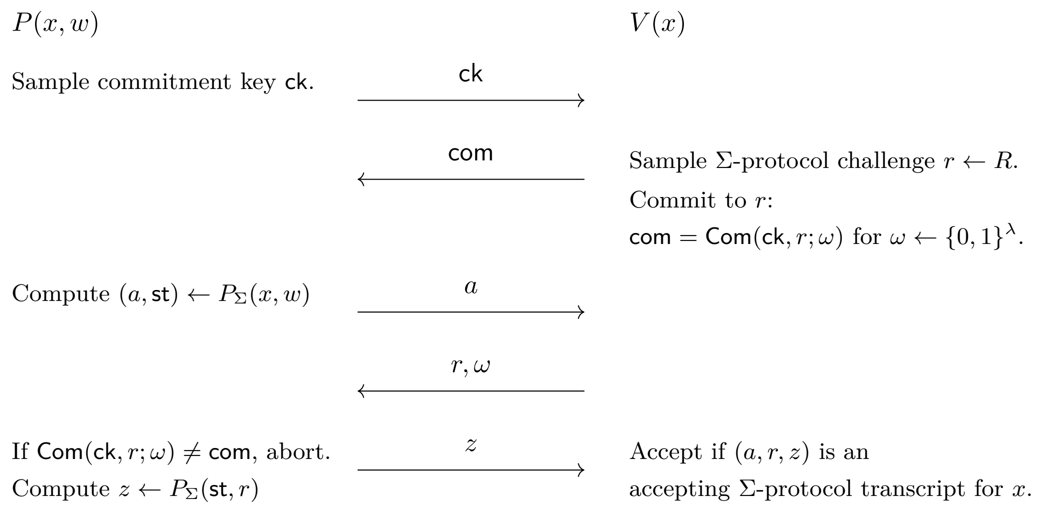

$P(x,w)$
$V(x)$
Sample commitment key ck.
ck
com
Sample Σ-protocol challenge $r \leftarrow R$.
Commit to r:
$\mathsf{com} = \mathsf{Com}(\mathsf{ck}, r; \omega)$ for $\omega \leftarrow \{0,1\}^\lambda$.
Compute $(a, \mathsf{st}) \leftarrow P_\Sigma(x, w)$
a
r, ω
If $\mathsf{Com}(\mathsf{ck}, r; \omega) \neq \mathsf{com}$, abort.
Compute $z \leftarrow P_\Sigma(\mathsf{st}, r)$
z
Accept if (a, r, z) is an
accepting Σ-protocol transcript for x.

Figure 7: The [GK96] Zero Knowledge Proof System for NP.

1. Call V^* on a random commitment key ck to obtain a commitment com.
2. Sample $(a', z') \leftarrow \mathsf{SHVZK.Sim}(0)$.
3. Run V^* on a' to obtain a challenge-opening pair (r', ω'). If ω' is not a valid opening of com to r', terminate the simulation and output the current view of V^*.
4. **Rewinding step.** Sample $(a, z) \leftarrow \mathsf{SHVZK.Sim}(r')$ and run V^* on a. If the output (r, ω) is not a valid message-opening pair, repeat this step from the beginning.
5. Respond with z and output V^*'s view.

To see that this simulator outputs the correct view for V^*, consider two hybrid steps:

- First, switch to a hybrid simulator in which the sample $(a', z') \leftarrow \mathsf{SHVZK.Sim}(0)$ is instead computed by running the honest prover $P(x, w)$. The indistinguishability between this hybrid simulator and the real simulator follows from the fact that a' sampled as $(a', z') \leftarrow \mathsf{SHVZK.Sim}(0)$ is computationally indistinguishable from the honestly generated a'.

- Next, switch to a second hybrid simulator in which the honest prover is also used in the rewinding step to generate the (a, z) samples rather than $\mathsf{SHVZK.Sim}(r')$ (where z is generated by running the honest prover on (a, r')). This is indistinguishable from the previous hybrid simulator by the SHVZK property, and moreover, by the computational binding of the commitment, the r obtained in Step 4 must be r' except with $\mathrm{negl}(\lambda)$ probability. Moreover, conditioned on $r = r'$, the second hybrid produces the same distribution as the honest interaction.

We now show how to extend this simulator to the quantum setting.

Our "naïve" quantum simulator. Step 1 of the naïve classical simulator will be unchanged in the quantum setting, so we focus on devising quantum verisons of Steps 2,3, and 4 while assuming $\mathsf{ck}, \mathsf{com}$ are fixed throughout.

Let $|\psi\rangle_{\mathcal{V}}$ be the state of the malicious verifier immediately after it sends com. We let registers $\mathcal{A}, \mathcal{Z}$ denote registers containing the messages a, z in the Σ-protocol and let $\mathcal{M}$ be a register that will contain the random coins for $\mathsf{SHVZK.Sim}$ (or the honest prover later on). Let $|\mathsf{Sim}_r\rangle$ for any $r \in R$ be the state $|\mathsf{Sim}_r\rangle_{\mathcal{A},\mathcal{Z},\mathcal{M}} = \sum \alpha_\mu |\mathsf{SHVZK.Sim}(r;\mu), \mu\rangle$ obtained by running $\mathsf{SHVZK.Sim}$ on a uniform superposition of its random coins μ.

We define binary projective measurements analogous to the U and C measurements used in our state-preserving extractor. However, instead of a single U measurement, we will have for each $r \in R$ a measurement $\mathsf{S}_r = (\Pi_{\mathsf{S},r}, \mathbf{I} - \Pi_{\mathsf{S},r})$ on $\mathcal{V} \otimes \mathcal{A} \otimes \mathcal{Z} \otimes \mathcal{M}$ where $\Pi_{\mathsf{S},r} \coloneqq \mathbf{I}_\mathcal{V} \otimes |\mathsf{Sim}_r\rangle\langle\mathsf{Sim}_r|_{\mathcal{A},\mathcal{Z},\mathcal{M}}$. The idea behind the $\mathsf{C} = (\Pi_\mathsf{C}, \mathbf{I} - \Pi_\mathsf{C})$ measurement is the same as before: it measures whether the malicious verifier V^* returns a valid opening when run on the challenge $\mathcal{A}$. Note that C acts as identity on $\mathcal{Z}, \mathcal{M}$.

The next steps of the quantum simulator are a direct analogue of the corresponding steps in the classical simulator:

2*. Initialize $\mathcal{A} \otimes \mathcal{Z} \otimes \mathcal{M}$ to $|\mathsf{Sim}_0\rangle$.

3*. Measure $|\psi\rangle_\mathcal{V} \otimes |P\rangle_{\mathcal{A},\mathcal{Z},\mathcal{M}}$ with C. If the outcome of C is 0 (the opening is invalid), terminate the simulation at this step: measure $\mathcal{A}$ to obtain a', compute and measure the verifier's response (r', ω') and return $(\mathsf{ck}, \mathsf{com}, a', (r', \omega'), z = \bot)$ along with $\mathcal{V}$. If the outcome of C is 1, we will have to rewind. First, compute the verifier's response and measure it to obtain r'.

When the opening is invalid (C outputs 0), the SHVZK guarantee informally implies that these steps computationally simulate the view of V^*.

The hard case is when the opening is valid (C outputs 1). At this stage of the simulation, the state on $\mathcal{V} \otimes \mathcal{A} \otimes \mathcal{Z}$ is $\Pi_\mathsf{C}(|\psi\rangle_\mathcal{V} |\mathsf{Sim}_0\rangle_{\mathcal{A},\mathcal{Z}})$ (up to normalization). Intuitively, we want to "swap" $|\mathsf{Sim}_0\rangle_{\mathcal{A},\mathcal{Z},\mathcal{M}}$ for $|\mathsf{Sim}_{r'}\rangle_{\mathcal{A},\mathcal{Z},\mathcal{M}}$, but the application of Π_C has entangled the $\mathcal{A}$ register with $\mathcal{V}$. We will therefore apply an operation to disentangle these registers, then swap $|\mathsf{Sim}_0\rangle$ for $|\mathsf{Sim}_{r'}\rangle$, and then "undo" the disentangling operation. We do this by defining a unitary U that is the coherent implementation of the following variable-length computation on $\mathcal{V} \otimes \mathcal{A} \otimes \mathcal{Z} \otimes \mathcal{M} \otimes \mathcal{R}$: measure $\mathcal{R}$ to obtain r, and then run a variable-length $\mathsf{Transform}^{\mathsf{C},\mathsf{S}_r}$ on $\mathcal{V} \otimes \mathcal{A} \otimes \mathcal{Z} \otimes \mathcal{M}$.[21] Recall that implementing a variable-length computation coherently requires additional ancilla registers $\mathcal{W}, \mathcal{B}, \mathcal{Q}$ (see Section 1.2); we will suppress these registers for this overview, but we emphasize that they must be all be initialized to $|0\rangle$.

The simulator then continues as follows.

4*. Run the following steps:

(a) Initialize $\mathcal{R}$ to $|0\rangle$ and apply U to $\Pi_\mathsf{C}(|\psi\rangle_\mathcal{V} |\mathsf{Sim}_0\rangle_{\mathcal{A},\mathcal{Z},\mathcal{M}}) \otimes |0\rangle_\mathcal{R}$. On $\mathcal{V} \otimes \mathcal{A} \otimes \mathcal{Z} \otimes \mathcal{M}$, this maps image($\Pi_\mathsf{C}$) to image($\Pi_{\mathsf{S},0}$), which yields a state of the form $|\psi'\rangle_\mathcal{V} |\mathsf{Sim}_0\rangle_{\mathcal{A},\mathcal{Z},\mathcal{M}}$. Importantly, this (carefully!) breaks the entanglement between $\mathcal{V}$ and $\mathcal{A}$.

(b) Now the simulator can easily swap $|\mathsf{Sim}_0\rangle$ out for $|\mathsf{Sim}_{r'}\rangle$.

(c) Finally, the simulator changes the $\mathcal{R}$ register from $|0\rangle_\mathcal{R}$ to $|r'\rangle_\mathcal{R}$, and then applies $U^\dagger$ and traces out $\mathcal{R}$. This step maps the state on *back* from image($\Pi_{\mathsf{S},r'}$) to image(Π_C).

5*. Measure $\mathcal{A}$ to obtain a, compute and measure the verifier's response (r, ω), measure $\mathcal{Z}$ to obtain z, and output $(\mathsf{ck}, \mathsf{com}, a, (r, \omega), z)$ along with $\mathcal{V}$.

[21] The register $\mathcal{R}$ is required for the definition of U and should not be confused with the sub-register of $\mathcal{V}$ that we measure to obtain the verifier's response.

This simulator can be written as an EQPT_c computation, but we defer the details of this to our full proof (Section 13). For this overview, we will focus on proving the simulation guarantee.

Inspired by classical proof, we prove that our simulator produces the correct view for V^* by considering two hybrid simulators. To describe the hybrid simulators, we define states $|P\rangle$ and $|P_r\rangle$ for any $r \in R$ corresponding to responses of the honest prover:

- Let $|P\rangle_{\mathcal{A},\mathcal{Z},\mathcal{M}}$ be the state from running the honest prover $P_{x,w}$ on a uniform superposition of random coins μ to generate a first message $P_{x,w}(\mu)$, i.e., $|P\rangle_{\mathcal{A},\mathcal{Z},\mathcal{M}} = \sum_\mu |P_{x,w}(\mu)\rangle_{\mathcal{A}} |0\rangle_{\mathcal{Z}} |\mu\rangle_{\mathcal{M}}$
- For any $r \in R$, let $|P_r\rangle$ be the same as $|P\rangle_{\mathcal{A},\mathcal{Z},\mathcal{M}}$, except $\mathcal{Z}$ additionally contains the honest prover's response to r, i.e., $|P_r\rangle_{\mathcal{A},\mathcal{Z},\mathcal{M}} = \sum_\mu |P_{x,w}(\mu)\rangle_{\mathcal{A}} |P_{x,w}(r;\mu)\rangle_{\mathcal{Z}} |\mu\rangle_{\mathcal{M}}$

The hybrid simulators are essentially quantum versions of the classical ones:

- The first hybrid simulator behaves the same as the original simulator except that everywhere the simulator uses $|\mathsf{Sim}_0\rangle$, the hybrid simulator uses $|P\rangle$ instead. The amplification in Step $4^*(a)$ is now onto image$(|P\rangle\langle P|)$ rather than image(S_0). Moreover, in Step 4^*b, the simulator swaps $|P\rangle$ out for $|\mathsf{Sim}_{r'}\rangle$.
- The second hybrid simulator is the same as the first, except every appearance of $|\mathsf{Sim}_{r'}\rangle$ is replaced with $|P_{r'}\rangle$. In particular, in Step $4^*(b)$, the simulator swaps $|P\rangle$ out for $|P_{r'}\rangle$. The (inverse) amplification in Step $4^*(c)$ is now from image$(|P_{r'}\rangle\langle P_{r'}|)$ onto image(Π_{C}).

We remark that defining these hybrid simulators also requires extending the definition of the unitary U that performs $\mathsf{Transform}$. In particular, U must now support $\mathsf{Transform}^{\mathsf{C},\mathsf{P}}$ where $\mathsf{P} = (\mathbf{I}_{\mathcal{V}} \otimes |P\rangle\langle P|, \mathbf{I} - \mathbf{I}_{\mathcal{V}} \otimes |P\rangle\langle P|)$ and $\mathsf{Transform}^{\mathsf{C},\mathsf{P}_r}$ where $\mathsf{P}_r = (\mathbf{I}_{\mathcal{V}} \otimes |P_r\rangle\langle P_r|, \mathbf{I} - \mathbf{I}_{\mathcal{V}} \otimes |P_r\rangle\langle P_r|)$.

Proving indistinguishability of these hybrids requires some care. Intuitively, we want to invoke the SHVZK property to claim that $|\mathsf{Sim}_0\rangle$ and $|P\rangle$ are indistinguishable given just the reduced density matrices on the $\mathcal{A}$ register (for the first hybrid) and that $|\mathsf{Sim}_{r'}\rangle$ and $|P_{r'}\rangle$ are indistinguishable given just the reduced density matrices on $\mathcal{A} \otimes \mathcal{Z}$ (for the second hybrid). However, we have to ensure that the application of $\mathsf{Transform}$ — which makes use of projections onto these states — does not make this distinguishing task any easier.

We resolve this by proving a general lemma (Lemma 13.1) about quantum computational indistinguishability that may be of independent interest, which we briefly elaborate on here. Consider the states $|\tau_b\rangle := \sum_\mu |\mu\rangle_{\mathcal{X}} |D_b(\mu)\rangle_{\mathcal{Y}}$ where D_0, D_1 are computationally indistinguishable classical distributions with randomness μ. If we are only given access to $\mathcal{Y}$, then distinguishing $|\tau_0\rangle$ from $|\tau_1\rangle$ is clearly hard (since $\mathrm{Tr}_{\mathcal{X}}(|\tau_b\rangle\langle\tau_b|)$ is a random classical sample from D_b). Lemma 13.1 strengthens this claim: it states that guessing b remains hard even given an oracle implementing the corresponding binary-outcome measurement $\left(|\tau_b\rangle\langle\tau_b|_{\mathcal{X},\mathcal{Y}}, \mathbf{I} - |\tau_b\rangle\langle\tau_b|_{\mathcal{X},\mathcal{Y}}\right)$.

By combining this lemma with the fact that our $\mathsf{Transform}$ procedure can always be truncated (in a further hybrid argument) to have strict $\mathrm{poly}(\lambda, 1/\varepsilon)$-runtime with ε-accuracy, we can prove the desired indistinguishability claims.

From the naïve simulator to the full simulator. The problem with both the classical and quantum naïve simulators presented above is that their expected runtime is not polynomial. The issue is conceptually the same as in Section 2.3.7. Consider a malicious verifier V^* that gives a valid response with negligible probability p when run on a sampled as $(a, z) \leftarrow \mathsf{SHVZK.Sim}(0)$, and succeeds with probability $p - \text{negl}$ when run on a sampled as $(a, z) \leftarrow \mathsf{SHVZK.Sim}(r)$. Then the expected running time is $\frac{p}{p-\text{negl}}$, which can be unbounded for small p.

The solution described in [GK96] is therefore to *estimate* the running time of the rewinding step before making the computational switch. That is, if the simulator obtains a valid response before the rewinding step, then it keeps running the V^* on samples from $\mathsf{SHVZK.Sim}(0)$ until it obtains λ additional valid responses. This gives the simulator an accurate estimate of the success probability of V^*, which it uses to bound the running time of the subsequent rewinding step.

We give a quantum simulator in EQPT_c for the [GK96] protocol that implements the analogous quantum version of this estimation trick. As in Section 2.3.7, the idea is to first compute an upper bound on the runtime of the $\mathsf{Transform}$ step (equivalently, a lower bound on the singular values) after measuring C in Step 3* *before* measuring r. This estimate is computed using a variable-length $\mathsf{Estimate}^{\mathsf{S}_0,\mathsf{C}}$ procedure, and since the $\mathsf{Transform}$ step has now been restricted to run in fixed polynomial time, we achieve the desired $p \cdot 1/p = 1$ cancellation in the expected running time.

Implementing this properly requires several tweaks to our simulator. In particular, the simulator no longer measures the verifier's challenge r' directly in Step 3*; recording r' is now delegated to U, since this step must be performed "in between" $\mathsf{Estimate}$ and $\mathsf{Transform}$. That is, we must modify U so that instead of just performing (a coherent implementation of) $\mathsf{Transform}$, it runs the following steps coherently: (1) perform a variable-length $\mathsf{Estimate}$, where $\mathsf{Estimate}$ is parameterized by the same projectors as $\mathsf{Transform}$ (2) compute and measure the verifier's response (3) run $\mathsf{Transform}$ using the time bound computed from $\mathsf{Estimate}$. We defer further details to the full proof (Section 13). We remark that just as in Section 2.3.7, the $\text{negl}(\lambda)$ error incurred by the collapsing measurement moves into the *correctness* error of the simulation.

2.5 Related Work

Post-Quantum Zero-Knowledge. The first construction of a zero-knowledge protocol secure against quantum adversaries is due to Watrous [Wat06]. Roughly speaking, [Wat06] shows that "partial simulators" that succeed with an inverse polynomial probability that is *independent* of the verifier state can be extended to full post-quantum zero-knowledge simulators. This technique handles sequential repetitions of classical Σ-protocols and has been used as a subroutine in other contexts (e.g., [BS20, BCKM21, CCY21, ACL21]), but its applicability is limited to somewhat special situations. Nevertheless, most prior post-quantum zero-knowledge results have relied crucially on the [Wat06] technique.

[BS20, AL20] recently introduced a beautiful *non-black-box* technique that, in particular, achieves constant-round zero knowledge arguments for NP with *strict* polynomial time simulation [BS20]. As discussed above, the use of non-black-box techniques is necessary to achieve strict polynomial time simulation in the classical [BL02] and quantum [CCLY21] settings (and in the quantum setting this extends to EQPT_m simulation).

Finally, recent work [CCY21] showed that the Goldreich–Kahan protocol achieves post-quantum ε-zero knowledge. This is closely related to our Theorem 1.4, and so we present a detailed comparison below.

Comparison with [CCY21]. The post-quantum security of the Goldreich–Kahan protocol was analyzed previously in [CCY21]. Our simulation strategy for Theorem 1.4 is related to that of [CCY21] in that the two simulators both consider the Jordan decomposition for essentially the same pair of projectors, but the two simulators are otherwise quite different.

At a high level, [CCY21] constructs a (non-trivial) quantum analogue of the following classical simulator: given error parameter ε, repeat $\mathrm{poly}(1/\varepsilon)$ times: sample $a \leftarrow \mathsf{Sim}(0)$ and run V^* on a. If V^* ever opens correctly, record its response r. Then, run a single execution of the protocol using $(a, z) \leftarrow \mathsf{Sim}(r)$ and output the result.

More concretely, the [CCY21] simulator first attempts to extract the verifier's challenge r in $\mathrm{poly}(1/\varepsilon)$ time, and then attempts to generate an accepting transcript in a single final interaction with the verifier. However, if the verifier *aborts* in this final interaction, the simulation fails; this is roughly because successfully extracting r skews the verifier's state towards not aborting. To obtain a full simulator, they use an idea from [BS20]: (1) design a "partial simulator" that randomly guesses whether the verifier will abort in its final invocation, then achieves ε-simulation conditioned on a correct guess; (2) apply [Wat06]-rewinding to "amplify" onto executions where the guess is correct.

It is natural to ask whether the above simulation strategy would have sufficed to prove Theorem 1.4 (instead of writing down a new simulator). We remark that this is unlikely; their simulator seems to be tailored to ε-ZK and, moreover, does not address what [GK96] describe as the main technical challenge in the classical setting: handling verifiers that abort with all but negligible probability. In more detail:

- Their non-aborting simulator (like the classical analogue above) *always* tries to extract r. To achieve negligible simulation error, this extraction must succeed with all but negligible probability for any adversary that with inverse polynomial probability does not abort. This would require that the simulator run in superpolynomial time.

  Our simulator, as well as essentially all classical black-box ZK simulators, address this issue by first measuring whether the verifier aborts, and then only proceeding with the simulation in the non-aborting case.

- By Markov's inequality, expected polynomial time simulation implies ε-simulation in time $O(1/\varepsilon)$. As a function of ε, the [CCLY21] simulator runs in some large polynomial time (as currently written, they appear to achieve runtime $1/\varepsilon^6$, although it is likely unoptimized). Thus, even a hypothetical variable-runtime version of their simulator would not be expected polynomial time. In particular, the [Wat06, BS20] "guessing" compiler appears to cause a quadratic blowup in the runtime of their non-aborting simulator (due to a required smaller accuracy parameter).

- The [Wat06, BS20] "guessing" compiler adds an additional layer of complexity onto the [CCY21] simulator that is incompatible with the EQPT_c definition in the sense that given an EQPT_c partial simulator, the [Wat06, BS20] "guessing" compiler would not produce a procedure in EQPT_c.

We also achieve some improvements over [CCY21] unrelated to the simulation accuracy:

- [CCY21] require that the underlying sigma protocol satisfies a *delayed witness* property, which is not required in the classical setting. Our "projector indistinguishability" lemma (Lemma 13.1; see also Section 2.4) enables us to handle arbitrary sigma protocols.

- [CCY21] require that the verifier commit to the sigma protocol challenge r using a *strong collapse-binding* commitment. Using a new proof technique (see Section 4), we show that standard collapse-binding suffices.

Post-Quantum Extraction As previously discussed, there is a line of prior work [Unr12, Unr16b, CMSZ21] that achieves forms of post-quantum extraction that do *not* preserve the prover state. Below we briefly discuss prior work on state-preserving post-quantum extraction.

[BS20] directly constructs a state-preserving extractable commitment with *non-black-box* extraction in order to achieve their zero-knowledge result. Their construction makes use of post-quantum fully homomorphic encryption (for quantum circuits). Their extractor homomorphically evaluates the adversarial sender.

[BS20] also shows that constant-round zero-knowledge arguments and post-quantum secure function evaluation generically imply constant-round state-preserving extractable commitments. Combining this with [Wat06] yields a polynomial-round state-preserving extractable commitment scheme. Since this result also holds in the "ε setting," plugging in [CCY21] implies a constant-round ε state-preserving extractable commitment, although this protocol would have many rounds and is only privately verifiable.

All of the above results achieve computationally state-preserving extraction. [ACL21] constructs a polynomial-round state-preserving extractable commitment scheme with *statistical* state preservation. They use the [Wat06] simulation technique as the core of their extraction procedure, applied to a new construction where statistical state preservation is possible.

3 Preliminaries

The security parameter is denoted by λ. A function $f\colon \mathbb{N} \to [0,1]$ is *negligible*, denoted $f(\lambda) = \mathrm{negl}(\lambda)$, if it decreases faster than the inverse of any polynomial. A probability is *overwhelming* if is at least $1 - \mathrm{negl}(\lambda)$ for a negligible function $\mathrm{negl}(\lambda)$. For any positive integer n, let $[n] := \{1, 2, \dots, n\}$. For a set R, we write $r \leftarrow R$ to denote a uniformly random sample r drawn from R.

3.1 Quantum Preliminaries and Notation

Quantum information. A (pure) *quantum state* is a vector $|\psi\rangle$ in a complex Hilbert space $\mathcal{H}$ with $\| |\psi\rangle \| = 1$; in this work, $\mathcal{H}$ is finite-dimensional. We denote by $\mathbf{S}(\mathcal{H})$ the space of Hermitian operators on $\mathcal{H}$. A *density matrix* is a positive semi-definite operator $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H})$ with $\mathrm{Tr}(\boldsymbol{\rho}) = 1$. A density matrix represents a probabilistic mixture of pure states (a mixed state); the density matrix corresponding to the pure state $|\psi\rangle$ is $|\psi\rangle\langle\psi|$. Typically we divide a Hilbert space into *registers*, e.g. $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$. We sometimes write, e.g., $\boldsymbol{\rho}^{\mathcal{H}_1}$ to specify that $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H}_1)$.

A unitary operation is a complex square matrix U such that $UU^\dagger = \mathbf{I}$. The operation U transforms the pure state $|\psi\rangle$ to the pure state $U|\psi\rangle$, and the density matrix $\boldsymbol{\rho}$ to the density matrix $U\boldsymbol{\rho}U^\dagger$.

A *projector* Π is a Hermitian operator ($\Pi^\dagger = \Pi$) such that $\Pi^2 = \Pi$. A *projective measurement* is a collection of projectors $\mathsf{P} = (\Pi_i)_{i \in S}$ such that $\sum_{i \in S} \Pi_i = \mathbf{I}$. This implies that $\Pi_i \Pi_j = 0$ for distinct i and j in S. The application of P to a pure state $|\psi\rangle$ yields outcome $i \in S$ with probability $p_i = \|\Pi_i |\psi\rangle\|^2$; in this case the post-measurement state is $|\psi_i\rangle = \Pi_i |\psi\rangle / \sqrt{p_i}$. We refer to the post-measurement state $\Pi_i |\psi\rangle / \sqrt{p_i}$ as the result of applying P to $|\psi\rangle$ and *post-selecting* (conditioning) on outcome i. A state $|\psi\rangle$ is an *eigenstate* of P if it is an eigenstate of every Π_i.

A two-outcome projective measurement is called a *binary projective measurement*, and is written as $\mathsf{P} = (\Pi, \mathbf{I} - \Pi)$, where Π is associated with the outcome 1, and $\mathbf{I} - \Pi$ with the outcome 0.

General (non-unitary) evolution of a quantum state can be represented via a *completely-positive trace-preserving (CPTP)* map $T\colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H}')$. We omit the precise definition of these maps in this work; we only use the facts that they are trace-preserving (for every $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H})$ it holds that $\mathrm{Tr}(T(\boldsymbol{\rho})) = \mathrm{Tr}(\boldsymbol{\rho})$) and linear.

For every CPTP map $T\colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H})$ there exists a *unitary dilation* U that operates on an expanded Hilbert space $\mathcal{H} \otimes \mathcal{K}$, so that $T(\boldsymbol{\rho}) = \mathrm{Tr}_{\mathcal{K}}(U(\rho \otimes |0\rangle\langle 0|^{\mathcal{K}})U^{\dagger})$. This is not necessarily unique; however, if T is described as a circuit then there is a dilation U_T represented by a circuit of size $O(|T|)$.

For Hilbert spaces $\mathcal{A}, \mathcal{B}$ the *partial trace* over $\mathcal{B}$ is the unique CPTP map $\mathrm{Tr}_{\mathcal{B}}\colon \mathbf{S}(\mathcal{A} \otimes \mathcal{B}) \to \mathbf{S}(\mathcal{A})$ such that $\mathrm{Tr}_{\mathcal{B}}(\boldsymbol{\rho}_A \otimes \boldsymbol{\rho}_B) = \mathrm{Tr}(\boldsymbol{\rho}_B)\boldsymbol{\rho}_A$ for every $\boldsymbol{\rho}_A \in \mathbf{S}(\mathcal{A})$ and $\boldsymbol{\rho}_B \in \mathbf{S}(\mathcal{B})$.

A *general measurement* is a CPTP map $\mathsf{M}\colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H} \otimes \mathcal{O})$, where $\mathcal{O}$ is an ancilla register holding a classical outcome. Specifically, given measurement operators $\{M_i\}_{i=1}^{N}$ such that $\sum_{i=1}^{N} M_i M_i^{\dagger} = \mathbf{I}$ and a basis $\{\,|i\rangle\}_{i=1}^{N}$ for $\mathcal{O}$, $\mathsf{M}(\boldsymbol{\rho}) \coloneqq \sum_{i=1}^{N}(M_i \boldsymbol{\rho} M_i^{\dagger} \otimes |i\rangle\langle i|^{\mathcal{O}})$. We sometimes implicitly discard the outcome register. A projective measurement is a general measurement where the M_i are projectors. A measurement induces a probability distribution over its outcomes given by $\Pr[i] = \mathrm{Tr}\Big(\,|i\rangle\langle i|^{\mathcal{O}}\,\mathsf{M}(\boldsymbol{\rho})\Big)$; we denote sampling from this distribution by $i \leftarrow \mathsf{M}(\boldsymbol{\rho})$.

The *trace distance* between states $\boldsymbol{\rho}, \boldsymbol{\sigma}$, denoted $d(\boldsymbol{\rho}, \boldsymbol{\sigma})$, is defined as $\frac{1}{2}\mathrm{Tr}\Big(\sqrt{(\boldsymbol{\rho} - \boldsymbol{\sigma})^2}\Big)$. The trace distance is contractive under CPTP maps (for any CPTP map T, $d(T(\boldsymbol{\rho}), T(\boldsymbol{\sigma})) \le d(\boldsymbol{\rho}, \boldsymbol{\sigma})$). It follows that for any measurement M, the statistical distance between the distributions $\mathsf{M}(\boldsymbol{\rho})$ and $\mathsf{M}(\boldsymbol{\sigma})$ is bounded by $d(\boldsymbol{\rho}, \boldsymbol{\sigma})$. We have the following *gentle measurement lemma*, which bounds how much a state is disturbed by applying a measurement whose outcome is almost certain.

Lemma 3.1 (Gentle Measurement [Win99])**.** *Let $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H})$ and $\mathsf{P} = (\Pi, \mathbf{I} - \Pi)$ be a binary projective measurement on $\mathcal{H}$ such that $\mathrm{Tr}(\Pi\boldsymbol{\rho}) \ge 1 - \delta$. Let*

$$\boldsymbol{\rho}' \coloneqq \frac{\Pi\boldsymbol{\rho}\Pi}{\mathrm{Tr}(\Pi\boldsymbol{\rho})}$$

be the state after applying P to $\boldsymbol{\rho}$ and post-selecting on obtaining outcome 1. Then

$$d(\boldsymbol{\rho}, \boldsymbol{\rho}') \le 2\sqrt{\delta}.$$

Definition 3.2. A real-valued measurement M on $\mathcal{H}$ is **(ε, δ)-almost-projective** if applying M twice in a row to any state $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H})$ produces measurement outcomes p, p' where

$$\Pr\big[|p - p'| \le \varepsilon\big] \ge 1 - \delta.$$

Quantum algorithms. In this work, a *quantum adversary* is a family of quantum circuits $\{\mathsf{Adv}_\lambda\}_{\lambda \in \mathbb{N}}$ represented classically using some standard universal gate set. A quantum adversary is *polynomial-size* if there exists a polynomial p and $\lambda_0 \in \mathbb{N}$ such that for all $\lambda > \lambda_0$ it holds that $|\mathsf{Adv}_\lambda| \le p(\lambda)$ (i.e., quantum adversaries have classical non-uniform advice).

3.2 Black-Box Access to Quantum Algorithms

Let A be a polynomial-time quantum algorithm with internal state $\boldsymbol{\rho} \in \mathrm{D}(\mathcal{H})$ whose behavior is specified by a unitary U on $\mathcal{X} \otimes \mathcal{H}$. A quantum oracle algorithm S^A with *black-box access* to $(A, \boldsymbol{\rho})$

is restricted to acting on $\mathcal{H}$ (which is initially set $\boldsymbol{\rho}$) by applying the unitary U or $U^\dagger$, but can freely manipulate $\mathcal{X}$ and an arbitrary external register $\mathcal{Y}$.

Black-box access models sometimes permit the U and $U^\dagger$ gates to be controlled on any external registers (i.e., any registers other than the registers $\mathcal{Z} \otimes \mathcal{H}$ to which U is applied). We note that none of the black-box algorithms in this work require controlled access to $U, U^\dagger$. This is because our black-box use of $U, U^\dagger$ takes the form $U^\dagger(\mathbf{I}_\mathcal{H} \otimes V_{\mathcal{X},\mathcal{Y}_1})U$ where V is a unitary acting only on $\mathcal{X} \otimes \mathcal{Y}_1$, and we can replace $U, U^\dagger$ controlled on $\mathcal{Y}_2$, with V controlled on $\mathcal{Y}_2$.

Algorithms with classical input and output. We also consider the special case of quantum algorithms that take classical "challenge" r and produce classical "response" z. Writing $\mathcal{X} = \mathcal{R} \otimes \mathcal{Z}$, an algorithm of this form is specified by a unitary U on $\mathcal{R} \otimes \mathcal{Z} \otimes \mathcal{H}$ of the form $\sum_r |r\rangle\langle r|_\mathcal{R} \otimes U^{(r)}_{\mathcal{Z},\mathcal{H}}$. For example, S^A can run A on a superposition of inputs by instantiating $\mathcal{R} \otimes \mathcal{Z}$ to $\sum_r |r\rangle_\mathcal{R} \otimes |0\rangle_\mathcal{Z}$ and then applying U.

We note that this definition is consistent with the notions of interactive quantum machines and oracle access to an interactive quantum machine used in e.g. [Unr12] and other works on post-quantum zero-knowledge.

We remark that our formalism is tailored to the two-message challenge-response setting. While the protocols we analyze in this paper will have more than two messages of interaction, our analysis will typically center around two particular messages in the middle of a longer execution, and $\boldsymbol{\rho}$ will be the intermediate state of the interactive algorithm right before the next challenge is sent. We also point out that the unitary U can be treated as independent of the (classical) protocol transcript before challenge r is sent, since we can assume this transcript is saved in $\boldsymbol{\rho}$.

3.3 Jordan's Lemma

We state Jordan's lemma and its relation to the singular value decomposition.

Lemma 3.3 ([Jor75])**.** *For any two Hermitian projectors Π_A and Π_B on a Hilbert space $\mathcal{H}$, there exists an orthogonal decomposition of $\mathcal{H} = \bigoplus_j \mathcal{S}_j$ into one-dimensional and two-dimensional subspaces $\{\mathcal{S}_j\}_j$ (the* Jordan subspaces*), where each $\mathcal{S}_j$ is invariant under both Π_A and Π_B. Moreover:*

- *in each one-dimensional space, Π_A and Π_B act as identity or rank-zero projectors; and*
- *in each two-dimensional subspace $\mathcal{S}_j$, Π_A and Π_B are rank-one projectors. In particular, there exist distinct orthogonal bases $\{|v_{j,1}\rangle, |v_{j,0}\rangle\}$ and $\{|w_{j,1}\rangle, |w_{j,0}\rangle\}$ for $\mathcal{S}_j$ such that Π_A projects onto $|v_{j,1}\rangle$ and Π_B projects onto $|w_{j,1}\rangle$.*

A simple proof of Jordan's lemma can be found in [Reg06].

For each j, the vectors $|v_{j,1}\rangle$ and $|w_{j,1}\rangle$ are corresponding left and right singular vectors of the matrix $\Pi_\mathsf{A}\Pi_\mathsf{B}$ with singular value $s_j = |\langle v_{j,1}|w_{j,1}\rangle|$. The same is true for $|v_{j,0}\rangle$ and $|w_{j,0}\rangle$ with respect to $(\mathbf{I} - \Pi_\mathsf{A})(\mathbf{I} - \Pi_\mathsf{B})$.

3.4 Commitment Schemes

A *commitment scheme* consists of a pair of PPT algorithms $\mathsf{Gen}, \mathsf{Commit}$ with the following properties.

Statistical/computational hiding. For an adversary Adv, define the experiment $\mathsf{Exp}_{\mathsf{hide}}^{\mathsf{Adv}}(\lambda)$ as follows.

1. $\mathsf{Adv}(1^\lambda)$ sends (ck, m_0, m_1) to the challenger.
2. The challenger flips a coin $b \in \{0,1\}$ and returns $\mathsf{com} := \mathsf{Commit}(\mathsf{ck}, m_b)$ to the adversary.
3. The adversary outputs a bit b'. The experiment outputs 1 if $b = b'$.

We say that $(\mathsf{Gen}, \mathsf{Commit})$ is statistically (resp. computationally) hiding if for all unbounded (resp. non-uniform QPT) adversaries Adv,

$$|\Pr\left[\mathsf{Exp}_{\mathsf{hide}}^{\mathsf{Adv}}(\lambda) = 1\right] - 1/2| = \mathrm{negl}(\lambda) \ .$$

Statistical/computational binding. For an adversary Adv, define the experiment $\mathsf{Exp}_{\mathsf{bind}}^{\mathsf{Adv}}(\lambda)$ as follows.

1. The challenger generates $\mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda)$.
2. $\mathsf{Adv}(\mathsf{ck})$ sends $(m_0, \omega_0, m_1, \omega_1)$ to the challenger.
3. The experiment outputs 1 if $\mathsf{Commit}(\mathsf{ck}, m_0, \omega_0) = \mathsf{Commit}(\mathsf{ck}, m_1, \omega_1)$.

We say that $(\mathsf{Gen}, \mathsf{Commit})$ is statistically (resp. computationally) binding if for all unbounded (resp. non-uniform QPT) adversaries Adv,

$$\Pr\left[\mathsf{Exp}_{\mathsf{bind}}^{\mathsf{Adv}}(\lambda) = 1\right] = \mathrm{negl}(\lambda) \ .$$

Collapse binding. For an adversary Adv, define the experiment $\mathsf{Exp}_{\mathsf{cl}}^{\mathsf{Adv}}(\lambda)$ as follows.

1. The challenger generates $\mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda)$.
2. $\mathsf{Adv}(\mathsf{ck})$ sends a commitment com and a quantum state $\boldsymbol{\rho}$ on registers $\mathcal{M} \otimes \mathcal{W}$.
3. The challenger flips a coin $b \in \{0,1\}$. If $b = 0$, the challenger does nothing. Otherwise, the challenger measures $\mathcal{M}$ in the computational basis.
4. The challenger returns registers $\mathcal{M} \otimes \mathcal{W}$ to the adversary, who outputs a bit b'. The experiment outputs 1 if $b = b'$.

We say that Adv is valid if measuring the output of $\mathsf{Adv}(\mathsf{ck})$ in the computational basis yields, with probability 1, $(\mathsf{com}, m, \omega)$ such that $\mathsf{Commit}(\mathsf{ck}, m, \omega) = \mathsf{com}$.

We say that $(\mathsf{Gen}, \mathsf{Commit})$ is collapse-binding if for all *valid* non-uniform QPT adversaries Adv,

$$|\Pr\left[\mathsf{Exp}_{\mathsf{cl}}^{\mathsf{Adv}}(\lambda) = 1\right] - 1/2| = \mathrm{negl}(\lambda) \ .$$

3.5 Preliminaries on Interactive Arguments

An interactive argument for an NP-language L consists of a pair of interactive algorithms P, V:

- The prover algorithm P is given as input an NP statement x and an NP witness w for x.
- The verifier algorithm V is given as input an NP statement x; at the end of the interaction, it outputs a bit b (interpreted as "accept"/"reject").

The minimal requirement we ask of such a protocol is *completeness*, which states that when the honest P, V algorithms are executed on a valid instance-witness pair (x, w), the verifier should accept with probability $1 - \mathrm{negl}(\lambda)$.

We typically consider interactive arguments consisting of either 3 or 4 messages. In many (but not all) settings we assume that the argument system is *public-coin* (in the second-to-last round), meaning that the second-to-last message (or *challenge*) is a uniformly random string r from some domain. We will use the following notation to denote messages in any such protocol:

- For 4-message public-coin protocols, we use vk to denote the first verifier message.
- We denote the first prover message by a.
- We denote the verifier challenge by r.
- We denote the prover response by z.
- We denote the verification predicate by $V(\mathsf{vk}, a, r, z)$.

We consider 3-message protocols as a special case of 4-message protocols in which $\mathsf{vk} = \bot$.

A key property of interactive protocols considered in this work is *collapsing* (and relaxations thereof), defined below.

Definition 3.4 (Collapsing Protocol [Unr16b, LZ19, DFMS19])**.** An interactive protocol (P, V) is *collapsing* if for every polynomial-size interactive quantum adversary A (where A may have an arbitrary polynomial-size auxiliary input quantum advice state),

$$\Big| \Pr[\mathsf{CollapseExpt}(0, A) = 1] - \Pr[\mathsf{CollapseExpt}(1, A) = 1] \Big| \leq \mathrm{negl}(\lambda).$$

For $b \in \{0, 1\}$, the experiment $\mathsf{CollapseExpt}(b, A)$ is defined as follows:

1. The challenger runs the interaction $\langle A, V \rangle$ between A (acting as a malicious prover) and the honest verifier V, stopping just before the measurement the register $\mathcal{Z}$ containing the malicious prover's final message. Let τ' be the transcript up to this point excluding the final prover message.
2. The challenger applies a unitary U to compute the verifier's decision bit $V(\tau', \mathcal{Z})$ onto a fresh ancilla, measures the ancilla, and then applies $U^\dagger$. If the measurement outcome is 0, the experiment aborts.
3. If $b = 0$, the challenger does nothing. If $b = 1$, the challenger measures the $\mathcal{Z}$ register in the computational basis and discards the result.
4. The challenger returns the $\mathcal{Z}$ register to A. Finally A outputs a bit b', which is the output of the experiment.

Definition 3.4 captures the collapsing property of Kilian's interactive argument system [Kil92] (as well as other Σ-protocols that make use of "strongly collapsing commitments" [CCY21]), but does not accurately capture protocols that make use of commitments satisfying statistical binding but not "strict binding" [Unr12]. To capture these protocols, we introduce a partial-collapsing definition.

For a 3 or 4-message interactive protocol (P, V), let T denote the set of transcript prefixes τ_{pre} (i.e., the first message a in a 3-message protocol or the first two messages (vk, a) in a 4-message protocol), let R denote the set of challenges r (the second-to-last message) and let Z denotes the set of possible responses z (the final message). Informally, such a protocol is *partially collapsing* with respect to a function $f : T \times R \times Z \to \{0, 1\}^*$ if the prover cannot detect a measurement of f.

Definition 3.5 (Partially Collapsing Protocol)**.** Let $f : T \times R \times Z \to \{0,1\}^*$ be a public efficiently computable function. A 3 or 4-message interactive protocol (P, V) is partially collapsing with respect to f if for every polynomial-size interactive quantum adversary A (where A may have an arbitrary polynomial-size auxiliary input quantum advice state),

$$\left| \Pr[\mathsf{PCollapseExpt}(0, f, A) = 1] - \Pr[\mathsf{PCollapseExpt}(1, f, A) = 1] \right| \leq \mathrm{negl}(\lambda).$$

For $b \in \{0, 1\}$, the experiment $\mathsf{PCollapseExpt}(b, f, A)$ is defined as follows:

1. The challenger runs the interaction $\langle A, V \rangle$ between A (acting as a malicious prover) and the honest verifier V, stopping just before the measurement the register $\mathcal{Z}$ containing the malicious prover's final message. Let (τ_{pre}, r) be the transcript up to this point (i.e., excluding the final prover message).
2. The challenger applies a unitary U to compute the verifier's decision bit $V(\tau', \mathcal{Z})$ onto a fresh ancilla, measures the ancilla, and then applies $U^\dagger$. If the measurement outcome is 0, the experiment aborts.
3. If $b = 0$, the challenger does nothing. If $b = 1$, the challenger initializes a fresh ancilla $\mathcal{Y}$ to $|0\rangle_{\mathcal{Y}}$, applies the unitary U_f (acting on $\mathcal{Z} \otimes \mathcal{Y}$) that computes $f(\tau_{\text{pre}}, r, \cdot)$ on $\mathcal{Z}$ and XORs the output onto $\mathcal{Y}$, measures $\mathcal{Y}$ and discards the result, and then applies $U_f^\dagger$.
4. The challenger returns the $\mathcal{Z}$ register to A. Finally A outputs a bit b', which is the output of the experiment.

Definition 3.5 captures the collapsing property of standard commit-and-open Σ-protocols [GMW86, Blu86] that make use of statistically binding (or, more generally, standard collapse-binding [Unr16b, CCY21]) commitments by setting f to output the part of z corresponding to the committed message (but not the opening). In some other cases (a subroutine of the [GMW86] graph non-isomorphism protocol, as well as the [LS91] "reverse Hamiltonicity" Σ-protocol) we will use more complicated definitions of f that measure different pieces of information depending on the challenge r.

Finally, we recall the definition of special honest-verifier zero knowledge.

Definition 3.6 (Special honest-verifier zero knowledge)**.** A 3-message sigma protocol (P_Σ, V_Σ) is *special honest verifier zero knowledge* (SHVZK) if there exists an algorithm $\mathsf{SHVZK.Sim}$ such that for all $(x, w) \in \mathfrak{R}$ and challenges $r \in R$, the distributions

$$\mathsf{SHVZK.Sim}(x, r) \quad \text{and} \quad (a, z) \leftarrow P_\Sigma(x, w, r)$$

are computationally indistinguishable.

4 Standard Collapse-Binding Implies Unique Messages

Recall that the standard collapse-binding security property ensures that if an efficient adversary produces a superposition of valid message-opening pairs (m, ω) to a commitment c, then it cannot detect whether a measurement of m is performed. There is an *apparent* deficiency with this definition as compared to the classical binding definition, which Unruh (implicitly) observes in [Unr12, Unr16b]: collapse-binding does not seem to imply that an adversary cannot give valid openings to two different messages *if the openings themselves are not measured.*

This issue has received relatively little attention, in part because circumventing it turns out to be fairly easy in many cases by either modifying the underlying protocol, or by simply assuming "strong" collapse-binding [CCY21] where the measurement of the message *and opening* is undetectable. For example:

- In [Unr12], Unruh introduces the notion of a *strict-binding* commitment, defined so that for any commitment c, there is a unique valid message-opening pair (m, ω). Unruh shows that standard Σ-protocols (such as GMW 3-coloring and Blum Hamiltonicity) are sound when instantiated with strict-binding commitments, but due to the issue described above, is unable to prove that these protocols are sound when instantiated with a statistically-binding commitment.

- In [Unr16b], Unruh gives a generic transformation which converts a classically secure Σ-protocol into a quantum proof of knowledge by committing to the responses to each challenge in advance. However, in many Σ-protocols (e.g. [GMW86, Blu86]) the response already consists of an opening to a commitment; are these protocols secure if the commitment is collapse-binding?

- This issue also arises in [CCY21], which explicitly asks for a strong collapse-binding commitment to instantiate their Σ-protocols. (They do note that a statistically binding commitment also suffices via a different argument.)

We believe this is an unsatisfying state of affairs. Collapse-binding is widely accepted as the quantum analogue of classical computational binding, but as the above examples illustrate, there are many natural settings where it is unclear whether it can be used as a drop-in replacement for classically binding commitments. Given this issue, a natural suggestion would be to treat strong collapse-binding as the quantum analogue of classical binding. However, we suggest that any definition of quantum computationally binding should at least capture statistically binding commitments. Statistically binding commitments do not generically satisfy strong collapse-binding, but are (standard) collapse-binding. Worse, strong collapse-binding is not a "robust" notion: we can make any commitment scheme lose its strong collapse-binding property by adding a single bit to the opening that the receiver ignores.

In this section, we resolve this difficulty and show that standard collapse-binding generically implies that an adversary cannot give two valid openings for two different messages, even when the openings are left unmeasured. This simplifies some of the proofs in this work, and also implies that strong collapse-binding and strict binding are unnecessary in the above examples.

Towards proving this, we first formalize a natural security property that captures the fact that a quantum adversary should only be able to open to a unique message.

Let $\mathsf{Com} = (\mathsf{Gen}, \mathsf{Commit})$ be a non-interactive commitment scheme. Define the following challenger-adversary interaction $\mathsf{Exp}_{uniq}^{\mathsf{Adv}}(\lambda)$ where $\mathsf{Adv} = (\mathsf{Adv}_1, \mathsf{Adv}_2)$ is a two-phase adversary.

1. The challenger generates $\mathsf{ck} \leftarrow \mathsf{Gen}(\lambda)$.

2. Run $\mathsf{Adv}_1(\mathsf{ck})$ to output a classical commitment string com, a classical message m_1 and a superposition of openings on register $\mathcal{W}$. It also returns its internal state $\mathcal{H}$, which is passed onto Adv_2.

3. The challenger measures whether $\mathcal{W}$ contains a valid opening for m_1 with respect to com and aborts (and outputs 0) if not.

4. Run $\mathsf{Adv}_2(\mathsf{ck})$ on $(\mathcal{H}, \mathcal{W})$. It outputs another message m_2 and a superposition of openings on register $\mathcal{W}$. If $m_2 = m_1$ then the expriment aborts and outputs 0.

5. The challenger measures whether $\mathcal{W}$ contains a valid opening for m_1 with respect to com. If so, the experiment outputs 1, otherwise 0.

Definition 4.1. We say that a commitment is *unique-message binding* if it can only be opened to a unique message if for all QPT adversaries Adv,

$$\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{uniq}(\lambda) = 1\right] = \operatorname{negl}(\lambda).$$

Lemma 4.2. *Any collapse-binding commitment* Com *satisfies unique-message binding.*

We remark that the unique-message binding definition and this lemma easily extend to interactive collapse-binding commitments. However, we will focus on the non-interactive case for simplicity. Our proof is reminiscent of the "control qubit" trick used by Unruh in [Unr16a] to prove that collapse-binding implies a notion called sum-binding.

Proof. Suppose that $\mathsf{Adv} = (\mathsf{Adv}_1, \mathsf{Adv}_2)$ satisfies $\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{uniq}(\lambda) = 1\right] = \varepsilon(\lambda) = \varepsilon$. Then we construct an adversary Adv' that obtains advantage $\varepsilon/8$ in the collapsing game for Com as follows:

1. Upon receiving ck from the challenger, Adv' does the following:

   (a) Run $\mathsf{Adv}_1(\mathsf{ck})$ to obtain a classical commitment com, a classical message m_1 (on register $\mathcal{M}$), and registers $\mathcal{W}, \mathcal{H}$.

   (b) Measure whether $\mathcal{W}$ contains a valid opening for m_1 with respect to com; if the opening is invalid, abort and output a random b'.[22]

   (c) Next, prepares an ancilla qubit $\mathcal{B}$ in the state $|+\rangle_{\mathcal{B}}$ and then apply the unitary U defined as

   $$U = |1\rangle\langle 1|_{\mathcal{B}} \otimes U^{\mathsf{Adv}_2}_{\mathcal{H},\mathcal{M},\mathcal{W}} + |0\rangle\langle 0|_{\mathcal{B}} \otimes \mathbf{I}_{\mathcal{H},\mathcal{M},\mathcal{W}}.$$

   where U^{Adv_2} is a unitary description of Adv_2 (the action of Adv_2 on $\mathcal{H} \otimes \mathcal{M} \otimes \mathcal{W}$ is unitary without loss of generality). That is, the unitary U has two branches of computation: it does nothing when $\mathcal{B} = 0$, and it runs Adv_2 when $\mathcal{B} = 1$.

   (d) Next apply the binary projective measurement $(\Pi_{\mathsf{ck},\mathsf{com},m_1}, \mathbf{I} - \Pi_{\mathsf{ck},\mathsf{com},m_1})$ where

   $$\Pi_{\mathsf{ck},\mathsf{com},m_1} \coloneqq |0\rangle\langle 0|_{\mathcal{B}} \otimes \mathbf{I}_{\mathcal{H},\mathcal{M},\mathcal{W}} + |1\rangle\langle 1|_{\mathcal{B}} \otimes \mathbf{I}_{\mathcal{H}} \otimes \sum_{\substack{m,\omega \,:\, m \neq m_1 \wedge \\ \mathsf{Commit}(\mathsf{ck},m,\omega) = \mathsf{com}}} |m,\omega\rangle\langle m,\omega|_{\mathcal{M},\mathcal{W}}\,.$$

   This measurement checks that after applying U, the output of Adv_2 (when $\mathcal{B} = 1$) is a valid message and opening (m, ω) where $m \neq m_1$. If this measurement rejects, abort and output a random b'.

   (e) Finally, send $\mathcal{M} \otimes \mathcal{W}$ to the collapsing challenger.

2. When the collapsing challenger returns $\mathcal{M} \otimes \mathcal{W}$, apply $U^{\dagger}$.

[22] To match the syntax of the collapsing game, the "abort" works as follows: $\mathsf{Adv}'(\mathsf{ck})$ initializes $\mathcal{M} \otimes \mathcal{W}$ to some valid commitment, sends it to the challenger, ignores the registers it gets back, and then outputs a random b'.

3. Perform the binary projective measurement $(\Pi_+, \mathbf{I} - \Pi_+)$ where $\Pi_+ := |+\rangle\langle+|_{\mathcal{B}} \otimes \mathbf{I}_{\mathcal{H},\mathcal{M},\mathcal{W}}$. If the measurement outcome is 1 (corresponding to $|+\rangle$), then Adv' outputs $b' = 0$ (i.e., guesses that the collapsing challenger did not measure the message). Otherwise, it outputs $b' = 1$.

We now compute the probability that Adv' outputs $b' = b$ for each choice of the collapsing challenge bit b.

If $b = 1$, then Adv' guesses correctly (outputs $b' = 1$) with probability exactly 1/2. This is because if Adv' aborts, it outputs 1 with probability 1/2 by definition, and if it does not abort, then it sends the collapsing challenger $\mathcal{M} \otimes \mathcal{W}$ where a measurement of the $\mathcal{M}$ register will completely determine the $\mathcal{B}$ register. In particular, if the outcome of the $\mathcal{M}$ measurement is m_1, $\mathcal{B}$ collapses to $|0\rangle$; otherwise, $\mathcal{B}$ collapses to $|1\rangle$. In either case, the probability that the measurement of $(\Pi_+, \mathbf{I} - \Pi_+)$ returns 0 (making Adv' output $b' = 1$) is exactly 1/2.

We now consider the case $b = 0$. Let $\boldsymbol{\rho} = \sum_{\mathsf{ck},\mathsf{com},m_1} \boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1} + \boldsymbol{\rho}_\perp$ be the state on $\mathcal{M} \otimes \mathcal{W} \otimes \mathcal{H}$ after Step 1b, where $\boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1}$ is the (subnormalized) state corresponding to outcomes $\mathsf{ck}, \mathsf{com}, m_1$ and the outcome "valid" in Step 1b, and $\boldsymbol{\rho}_\perp$ is the (subnormalized) state corresponding to the outcome "invalid" in Step 1b.

Recall that in the case $b = 0$, the collapsing challenger does nothing to $\mathcal{M} \otimes \mathcal{W}$. Thus the effect of Steps 1c, 1d and 2 is to apply a binary projective measurement $\left(\Pi'_{\mathsf{ck},\mathsf{com},m_1}, \mathbf{I} - \Pi'_{\mathsf{ck},\mathsf{com},m_1}\right)$ where $\Pi'_{\mathsf{ck},\mathsf{com},m_1} := U^\dagger \Pi_{\mathsf{ck},\mathsf{com},m_1} U$. From the description of the experiment, it holds that

$$\begin{aligned}
\Pr[b' = 0] &= \frac{1}{2}\operatorname{Tr}(\boldsymbol{\rho}_\perp) + \frac{1}{2}\sum_{\mathsf{ck},\mathsf{com},m_1} \operatorname{Tr}\Big((\mathbf{I} - \Pi'_{\mathsf{ck},\mathsf{com},m_1})(\,|+\rangle\langle+| \otimes \boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1})\Big) \\
&\quad + \sum_{\mathsf{ck},\mathsf{com},m_1} \operatorname{Tr}\Big(\Pi_+ \Pi'_{\mathsf{ck},\mathsf{com},m_1}(\,|+\rangle\langle+| \otimes \boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1})\Pi'_{\mathsf{ck},\mathsf{com},m_1}\Big) \\
&\geq \frac{1}{2} - \frac{1}{2}\sum_{\mathsf{ck},\mathsf{com},m_1} \operatorname{Tr}\Big(\Pi'_{\mathsf{ck},\mathsf{com},m_1}(\,|+\rangle\langle+| \otimes \boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1})\Big) \\
&\quad + \sum_{\mathsf{ck},\mathsf{com},m_1} \operatorname{Tr}(\boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1}) \left(\frac{\operatorname{Tr}\Big(\Pi'_{\mathsf{ck},\mathsf{com},m_1}(\,|+\rangle\langle+| \otimes \boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1})\Big)}{\operatorname{Tr}(\boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1})}\right)^2 \\
&\geq \frac{1}{2} - \frac{1}{2}\sum_{\mathsf{ck},\mathsf{com},m_1} \operatorname{Tr}\Big(\Pi'_{\mathsf{ck},\mathsf{com},m_1}(\,|+\rangle\langle+| \otimes \boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1})\Big) \\
&\quad + \frac{\Big(\sum_{\mathsf{ck},\mathsf{com},m_1} \operatorname{Tr}\Big(\Pi'_{\mathsf{ck},\mathsf{com},m_1}(\,|+\rangle\langle+| \otimes \boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1})\Big)\Big)^2}{\sum_{\mathsf{ck},\mathsf{com},m_1} \operatorname{Tr}(\boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1})}
\end{aligned}$$

where the latter inequality is Jensen, and the former is the following:

Claim 4.3. *If* $\Pi_{\mathsf{A}}\boldsymbol{\rho} = \boldsymbol{\rho}$ *then* $\operatorname{Tr}(\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}\boldsymbol{\rho}\Pi_{\mathsf{B}}) \geq \operatorname{Tr}(\Pi_{\mathsf{B}}\boldsymbol{\rho})^2 / \operatorname{Tr}(\boldsymbol{\rho})$.

Proof. $\operatorname{Tr}(\Pi_{\mathsf{B}}\boldsymbol{\rho}) = \operatorname{Tr}(\Pi_{\mathsf{B}}\boldsymbol{\rho}\Pi_{\mathsf{A}}) = \operatorname{Tr}(\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}\boldsymbol{\rho}) \leq \sqrt{\operatorname{Tr}(\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}\boldsymbol{\rho}\Pi_{\mathsf{B}}\Pi_{\mathsf{A}})\operatorname{Tr}(\boldsymbol{\rho})}$, where the inequality is by Cauchy-Schwarz. □

Let $\gamma := \sum_{\mathsf{ck},\mathsf{com},m_1} \operatorname{Tr}(\boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1})$. Observe that $\sum_{\mathsf{ck},\mathsf{com},m_1} \operatorname{Tr}\Big(\Pi'_{\mathsf{ck},\mathsf{com},m_1}(\,|+\rangle\langle+| \otimes \boldsymbol{\rho}_{\mathsf{ck},\mathsf{com},m_1})\Big) = (\gamma + \varepsilon)/2$. It follows that

$$\Pr[b' = 0] = \frac{1}{2} - \frac{1}{4}(\gamma + \varepsilon) + \frac{1}{4} \cdot \frac{(\gamma + \varepsilon)^2}{\gamma} \geq \frac{1}{2} + \frac{\varepsilon}{4}\ .$$

Thus the overall probability that Adv' guesses a random b correctly in the collapsing experiment is at least $1/2 + \varepsilon(\lambda)/8$. □

5 Generalized Notions of Special Soundness

Let (P, V) denote a 3 or 4-message public-coin interactive proof or argument system. Let T denote the set of transcript prefixes τ_{pre} (i.e., the first message in a 3-message protocol or the first two messages in a 4-message protocol), R denotes the set of challenges r (the second-to-last message) and Z denotes the set of possible responses z (the final message). The instance x is assumed to be part of τ_{pre}, which allows us to capture protocols in which the instance is adaptively chosen by the prover in its first message.

We introduce generalizations of the special soundness property to capturing situations where

1. the special soundness extractor is able to produce a witness given only a function $f(z)$ of the response z, and/or

2. the extractor is only required to succeed (with some $1 - \text{negl}(\lambda)$ probability) when the challenges are sampled from an "admissible distribution."

The second property is related to the notion of probabilistic special soundness due to [CMSZ21].[23]

Throughout this section, k will be a parameter specifying the number of (partial) transcripts required to extract.

5.1 Generalized Special Soundness Definitions

We first recall the standard definition of k special soundness.

Definition 5.1 (k-special soundness)**.** An interactive protocol (P, V) is k-special-sound if there exists an efficient extractor $\mathsf{SSExtract} : T \times (R \times Z)^k \to \{0,1\}^*$ such that given $\tau_{\text{pre}}, (r_i, z_i)_{i \in [k]}$ where each r_i is distinct and for each i, $(\tau_{\text{pre}}, r_i, z_i)$ is an accepting transcript, $\mathsf{SSExtract}(\tau_{\text{pre}}, r_i, z_i)$ outputs a valid witness w for the instance x with probability 1.

In order to generalize this definition, we consider interactive protocols (P, V) with a "consistency" predicate $g : T \times (R \times \{0,1\}^*)^* \to \{0,1\}$. The argument $\{0,1\}^*$ corresponds to some partial information y about a response z. The consistency predicate should have the property that if $g(\tau_{\text{pre}}, (r_i, y_i)_{i \in [k]}) = 1$, then $g(\tau_{\text{pre}}, (r_i, y_i)_{i \in G}) = 1$ for all subsets $G \subset [k]$. For any positive integer k, we define the set $\mathsf{Consistent}_k$ to be the subset of $T \times (R \times \{0,1\}^*)^k$ on which g outputs 1. We can extend k special soundness to allow the $\mathsf{SSExtract}$ algorithm to produce a witness given only partial information y_i of the responses z_i provided that the "partial transcripts" satisfy consistency.

Definition 5.2 ((k, g)-special soundness)**.** An interactive protocol (P, V) is (k, g)-special-sound if there exists an efficient extractor $\mathsf{SSExtract}_g : T \times (R \times \{0,1\}^*)^* \to \{0,1\}$ such that given $(\tau_{\text{pre}}, (r_i, y_i)_{i \in [k]}) \in \mathsf{Consistent}_k$ where each r_i is distinct and for each i, $\mathsf{SSExtract}_g(\tau_{\text{pre}}, r_i, y_i)$ outputs a valid witness w with probability 1.

[23] A similar (but not identical) definition appears in an older version of [CMSZ21]: https://arxiv.org/pdf/2103.08140v1.pdf.

Notice that all k-special-sound protocols with super-polynomial size challenge space are (k, g)-probabilistic-special sound for the "trivial" consistency predicate g that simply checks (interpreting $y_i = z_i$ as a full response) whether all the transcripts are accepting.

Claim 5.3. *For any k-special-sound protocol (P, V), there exists a consistency predicate g such that (P, V) is (k, g)-special-sound.*

Proof. Define g to output 1 on input $\tau_{\text{pre}}, (r_i, y_i)_{i\in[k]}$ if and only if each $(\tau_{\text{pre}}, r_i, y_i)$ is an accepting transcript. It follows that the original $\mathsf{SSExtract}$ in the special soundness definition satisfies the requirements of the (k, g)-special soundness definition. □

When the challenge space R is super-polynomial-size, we can generalize this definition even further so that the extractor need not succeed on worst-case k-tuples of distinct challenges, but only on k-tuples sampled from an "admissible distribution."

Definition 5.4 (Q-admissible distribution)**.** A distribution D_k over R^k is *admissible* if there exists a negligible function $\text{negl}(\lambda)$ and a sampling procedure Samp such that D_k is $\text{negl}(\lambda)$-close to the output distribution of the following process:

- Samp makes, in expectation, $Q(\lambda)$ classical queries to an oracle O_R that outputs a uniformly random challenge $r \leftarrow R$ each time it is queried.
- Samp must produce its outputs as follows. Let Q_{total} be the total number of queries it makes to O_R. Samp specifies a set $\{i_1, \ldots, i_k\} \subseteq [Q_{\text{total}}]$, and its output is defined to be $r_{i_1}, \ldots, r_{i_k}$ where r_i is the ith output of the uniform sampling oracle O_R.

  We stress that Samp may use an arbitrary (e.g., even inefficient) process to select the set $\{i_1, \ldots, i_k\}$. Moreover, the output challenges $r_{i_1}, \ldots, r_{i_k}$ do not necessarily have distinct values (this can occur if the sampling oracle O_R outputs the same challenge more than once).

Definition 5.5 (admissible distribution)**.** A distribution D_k over R^k is *admissible* if there exists $Q = \text{poly}(\lambda)$ such that D_k is a Q-admissible distribution.

Definition 5.6 ((k, g)-probabilistic special soundness)**.** An interactive protocol (P, V) with consistency predicate g is (k, g)-probabilistic-special-sound if there exists an efficient extractor $\mathsf{SSExtract} : T \times (R \times \{0, 1\}^*)^k \to \{0, 1\}^*$ such that for any distribution D supported on $\mathsf{Consistent}_k$ whose marginal distribution on R^k is admissible,

$$\Pr_{(\tau_{\text{pre}},(r_i,y_i)_{i\in[k]})\leftarrow D}[\mathsf{PSSExtract}_g(\tau_{\text{pre}}, (r_i, y_i)_{i\in[k]}) \to w \wedge w \text{ is a valid witness for } x] = 1 - \text{negl}(\lambda)$$

Note that (k, g)-probabilistic special soundness (PSS) is only meaningful when the challenge space R has super-polynomial size. When R is polynomial, an admissible distribution D_k can simply output $(r, \ldots, r)$ (the same challenge repeated k times) since there exists a Samp that simply queries O_R until it outputs the same challenge k times.

However, when $|R|$ is superpolynomial, (k, g)-PSS is a relaxation of (k, g)-special soundness.

Claim 5.7. *When $R = 2^{\omega(\log \lambda)}$, any (k, g)-special-sound protocol is also (k, g)-probabilistic-special-sound.*

Proof. It suffices to prove that the probability any admissible distribution outputs the same challenge r more than once is $\text{negl}(\lambda)$. By the definition of an admissible distribution, its output is negl(λ)-close to the output of an arbitrary sampling algorithm that makes an expected poly(λ) number of queries to a uniform sampling oracle O_R over R, and then outputs a size-k subset of the oracle responses.

Suppose towards a contradiction that there exists constant c such that for infinitely many $\lambda \in \mathbb{N}$, the sampling oracle O_R outputs a repeated challenge with probability $1/\lambda^c$. Let d be a constant such that the expected number of queries to the uniform sampling oracle O_R is $O(\lambda^d)$. If $0 \leq q \leq \lambda^{d+c+1}$ oracle queries have already been made, the probability that the next oracle query allows finding a collision is at most $\lambda^{d+c+1}/|R|$. This implies that finding a collision within λ^{d+c+1} queries is at most $\lambda^{2d+2c+2}/|R|$. Thus, to find a collision with probability at least $1/\lambda^c$, the number of oracle queries must be at least λ^{d+c+1} with probability at least $1/\lambda^c - \lambda^{2d+2c+2}/|R|$, which implies the expected number of oracle queries is at least $\lambda^{d+c+1}(1/\lambda^c - \lambda^{2d+2c+2}/|R|) = \lambda^{d+1} - \lambda^{3d+3c+3}/|R|$. Since $R = 2^{\omega(\log \lambda)}$, there exists a constant λ_0 such that for all $\lambda > \lambda_0$, this expectation is $\lambda^{d+1} - \lambda^{3d+3c+3}/|R| > \lambda^{d+1} - 1$. This contradicts our assumption that the expected number of queries to the sampling oracle is $O(\lambda^d)$. □

5.2 A Special Soundness Parallel Repetition Theorem

Although it is well-known that 2-special soundness is preserved under parallel repetition, the situation is more complicated for generalized special soundness notions (and even k-special soundness for larger values of k). We state and prove a useful theorem about the parallel repetition of special sound protocols.

Lemma 5.8. *If $\Sigma = (P, V)$ is a (k, g)-special-sound protocol, then the $t = \Omega(k^2 \log^2(\lambda))$-fold parallel repetition Σ^t is (k^2, g^t)-probabilistic special sound where g^t outputs 1 if and only (1) the arguments y_i consist of t formally separated components, and (2) g outputs 1 on each of the t components.*

Proof. Let $\mathsf{Consistent}_{k^2}$ be the set of k-tuples of shared-prefix partial transcripts of Σ^t on which g^t outputs 1. Let D be a distribution supported on $\mathsf{Consistent}_{k^2}$ whose marginal distribution on $(R^t)^{k^2}$ is admissible.

We construct $\mathsf{PSSExtract}_{g^t}$ for Σ^t that takes as input

$$(\tau_{\mathsf{pre},j})_{j\in[t]}, ((r_{j,i})_{j\in[t]}, (y_{j,i})_{j\in[t]})_{i\in[k^2]} \leftarrow D$$

and does the following:

1. Look for $j \in [t]$ such that $\{r_{j,i}\}_{i\in[k^2]}$ consists of k distinct challenges. If no such j exists, abort and output $\perp$.

2. If such a j exists, let H be a size-k subset of $[k^2]$ such that $\{r_{j,i}\}_{i\in H}$ consists of k distinct challenges, and let $\mathsf{SSExtract}_g$ be the (k, g)-special-soundness extractor for Σ. Run $\mathsf{SSExtract}_g(\tau_{\mathrm{pre},j}, (r_{j,i}, y_{j,i})_{i\in H}) \rightarrow w$ and output w.

First, we note that

$$g(\tau_{\mathrm{pre},j}, (r_{j,i}, y_{j,i})_{i\in H}) = 1$$

follows from

$$g^t((\tau_{\mathsf{pre},j})_{j\in[t]}, ((r_{j,i})_{j\in[t]}, (y_{j,i})_{j\in[t]})_{i\in[k^2]}) = 1.$$

Thus, it suffices to prove that this extractor aborts with probability $\mathrm{negl}(\lambda)$. Define $\mathsf{BAD} \subset R^{tk^2}$ to be the set of all tk^2-tuples $(r_{j,i})_{j\in[t],i\in[k^2]}$ such that for all $j \in [t]$, the k^2-tuple $(r_{j,i})_{i\in[k^2]}$ does not contain k distinct challenges.

Suppose $(r_{j,i})_{j\in[t],i\in[k^2]}$ is sampled uniformly at random from R^{tk^2}. Then we have

$$\Pr_{(r_{j,i})_{j\in[t],i\in[k^2]}\leftarrow R^{tk^2}}[(r_{j,i})_{j\in[t],i\in[k^2]} \in \mathsf{BAD}] \le \left(\frac{k}{e^k}\right)^t.$$

This follows from the fact that for any fixed j, the probability that $(r_{j,i})_{i\in[k^2]}$ does *not* contain k distinct challenges is at most $k((k-1)/k)^{k^2} \le k/e^k$.

By the definition of an admissible distribution (Definition 5.5), the marginal distribution of D_{Σ^t} on $(R^t)^{k^2}$ is the result of the following process (up to $\mathrm{negl}(\lambda)$ statistical distance): make an expected $\mathrm{poly}(\lambda)$ number of classical queries to a uniform sampling oracle O_{R^t} over R^t, receiving a set of challenges A, and then (using an arbitrary procedure) output any size-k^2 subset $\{(r_{1,1},\ldots,r_{t,1}),\ldots,(r_{1,k^2},\ldots,r_{t,k^2})\}$ of A of k^2 challenges. The extractor aborts if $(r_{j,i})_{j\in[t],i\in[k^2]} \in \mathsf{BAD}$.

Let d be a constant such that the expected number of queries to the uniform sampling oracle O_{R^t} is $O(\lambda^d)$. Suppose towards a contradiction that the extractor aborts with non-negligible probability, i.e., there exists a constant c such that for infinitely many $\lambda \in \mathbb{N}$, the extractor aborts with probability at least $1/\lambda^c$. If $0 \le q \le \lambda^{d+c+1}$ oracle queries have already been made, the probability that the next oracle query allows finding a size-k^2 subset of outputs in BAD is at most

$$(\lambda^{d+c+1})^{k^2}\left(\frac{k}{e^k}\right)^{k^2\log^2(\lambda)}.$$

Moreover, there exists a constant λ_0 such that for all $\lambda > \lambda_0$, this can be upper bounded as

$$(\lambda^{d+c+1})^{k^2}\left(\frac{k}{e^k}\right)^{k^2\log^2(\lambda)} < \left(\frac{2^{k^2+k^2\log(k)}}{e^{k^3}}\right)^{\log^2(\lambda)} < (1/2)^{\log^2(\lambda)} = 1/\lambda^{\log(\lambda)}.$$

Thus for all $\lambda > \lambda_0$, the probability of finding a size-k^2 subset of oracle outputs in BAD within λ^{d+c+1} oracle queries is at most $\lambda^{d+c+1}/\lambda^{\log(\lambda)}$; this implies that finding a size-k^2 subset of oracle outputs in BAD with probability $1/\lambda^c$ requires making at least λ^{d+c+1} oracle queries with probability at least $1/\lambda^c - \lambda^{d+c+1}/\lambda^{\log(\lambda)}$. Then for $\lambda > \lambda_0$, the expected number of queries is at least $(\lambda^{d+c+1})(1/\lambda^c - \lambda^{d+c+1}/\lambda^{\log(\lambda)}) = \lambda^{d+1} - \lambda^{2d+2c+2}/\lambda^{\log(\lambda)}$. Since c and d are constants, there exists λ_0' such that for $\lambda > \lambda_0'$, the expected number of queries is at least $\lambda^{d+1} - 1$. This contradicts our assumption that the number expected number of queries to the uniform sampling oracle is $O(\lambda^d)$. □

5.3 Examples of Probabilistic Special Sound Protocols

We now show that many classical interactive proofs-of-knowledge (or arguments-of-knowledge) satisfy probabilistic special soundness. It was already noted above that (parallel repetitions of) standard special sound protocols satisfy the notion. Here, we highlight three other cases: commit-and-open protocols (where g is only given partial transcripts), Kilian's protocol, and a subroutine of the [GMW86] graph non-isomorphism protocol.

5.3.1 The "one-out-of-two" graph isomorphism subroutine

In order to prove Theorem 1.2, we consider the following proof-of-knowledge subroutine of the [GMW86] graph non-isomorphism protocol:

- The subroutine instance is three graphs G_0, G_1, H. The prover[24] wants to prove that there exists a bit b such that G_b is isomorphic to H. To do so, they execute a parallel repetition of the following protocol.

- The prover picks a random permutations σ_0, σ_1, a random bit c, and sends $(H_0 = \sigma_0(G_c), H_1 = \sigma_1(G_{1-c}))$ to the verifier.

- The verifier sends a random bit r.

- If $r = 0$, the prover sends (c, σ_0, σ_1) and the verifier checks that $(H_0 = \sigma_0(G_c), H_1 = \sigma_1(G_{1-c}))$ was computed correctly.

- If $r = 1$, the prover sends $(c \oplus b, \sigma_{c\oplus b}\pi)$, where π is an isomorphism mapping H to G_b. The verifier then checks that $(\sigma_{c\oplus b}\pi)H = H_{c\oplus b}$.

In the classical setting, this is generally viewed as a proof of knowledge of (b, π). However, we consider it as a proof of knowledge of the bit b, in the situation where G_0 and G_1 are not isomorphic. We will formalize this in *two* different ways: first by showing that the protocol is $(2, g)$-special sound for a natural consistency predicate g, and then by showing that it is $(2, g')$-PSS for a more complicated predicate g' that we have to use to be compatible with the protocol's limited partial collapsing property.

First, we define an (inefficient) consistency predicate g, which is given as input τ_{pre} an arbitrary number of pairs $(\mathbf{r}, \mathbf{c}) \in \{0,1\}^\lambda \times \{0,1\}^\lambda$ (rejecting if the input is not of this form). g outputs 1 if the following conditions hold for all $\ell \in [\lambda]$:

- If $r_\ell = 0$, the graphs $(H_{0,\ell}, H_{1,\ell})$ are isomorphic to $(G_{c_\ell}, G_{1-c_\ell})$.

- If $r_\ell = 1$, the graph $H_{c_\ell,\ell}$ is isomorphic to H.

The following claim then holds immediately by transitivity of graph isomorphism. The extractor, given $(\mathbf{r}, \mathbf{c})$ and $(\mathbf{r}', \mathbf{c}')$ simply chooses an ℓ such that $r_\ell \neq r'_\ell$ and outputs $b = c_\ell \oplus c'_\ell$.

Claim 5.9. *If G_0 and G_1 are not isomorphic, then the [GMW86] subroutine satisfies $(2, g)$-special soundness, where the extractor outputs the bit b.*

Finally, we define the predicate g' to be a slight modification of g: for the *first* pair $(r^{(1)}, c^{(1)})$, g' ignores[25] the bits $c_\ell^{(1)}$ for i such that $r_\ell^{(1)} = 1$. The protocol will then not be $(2, g')$-special sound (e.g. a first transcript with $r^{(1)} = 1^\lambda$ would provide no information), it *will* be $(2, g')$-PSS.

Claim 5.10. *If G_0 and G_1 are not isomorphic, then the [GMW86] subroutine satisfies $(2, g')$-PSS, where the extractor outputs the bit b.*

[24] The [GMW86] verifier acts as the prover in this subroutine.

[25] Alternatively, we could define g' to require inputs with these $c_\ell^{(1)}$ omitted.

Proof. This follows from the claim that if $(r^{(1)}, r^{(2)}) \in \{0,1\}^\lambda \times \{0,1\}^\lambda$ is sampled according to an admissible distribution, then with all but $\mathrm{negl}(\lambda)$ probability, there exists an index ℓ such that $r^{(1)}_\ell = 0$ and $r^{(2)}_\ell = 1$. This can be argued using the same reasoning as in the proof of Claim 5.7, since the probability that two uniformly random λ-bit strings $r^{(1)}$ and $r^{(2)}$ do not have an index $\ell \in [\lambda]$ such that $r^{(1)}_\ell = 0$ and $r^{(2)}_\ell = 1$ is $\mathrm{negl}(\lambda)$. □

5.3.2 Commit-and-Open Protocols

The next class of examples we discuss is that of *commit-and-open protocols.* In particular, we are interested in characterizing a special soundness property where the extractor is only given the opened *messages* in the prover's response (and not their openings).

Definition 5.11. Let Com denote a (possibly keyed) non-interactive commitment scheme. A commit-and-open protocol is a (3 or 4 message) protocol for an NP language L of the following form:

- (Optional first verifier message) If Com is keyed, the verifier samples and sends the commitment key ck for Com.
- The prover, given a witness w for some statement $x \in L$, computes a string $y \in \{0,1\}^N$ and sends a bitwise commitment $a = \mathsf{Com}(\mathsf{ck}, y)$ to the verifier.
- The verifier samples a string r that encodes a subset $S \subset [N]$ and sends r to the prover.
- The prover sends openings to $\{y_i\}_{i \in S}$.
- The verifier checks that each opening to y_i (for $i \in S$ is valid and then computes some function $\mathsf{Check}(y_S)$ on the opened bits.

We say that such a protocol satisfies "commit-and-open k-special soundness" if there exists an extractor $\mathsf{Extract}(x, y)$ satisfying the following property. For every instance x and every collection of k *distinct* sets $S_1, \ldots, S_k$ (represented by strings $(r_1, \ldots, r_k)$, for *any* string y such that $\mathsf{Check}(y_{S_i}) = 1$ for all i, $w = \mathsf{Extract}(x, y)$ is a valid NP-witness for x.

It is not hard to see that the "commit-and-open" k-special soundness property, combined with the (computational/statistical) binding of the commitment scheme, implies a standard (computational/statistical) k-special soundness property of the Σ-protocol. However, we consider "commit-and-open k-special soundness" explicitly in order to satisfy (probabilistic) special soundness with respect to *partial* transcripts.

This definition captures extremely common Σ-protocols, such as:

- The [GMW86] Σ-protocol for 3-coloring.
- A slight variant of the [Blu86] Σ-protocol for Hamiltonicity[26]
- Protocols following the "MPC-in-the-head" paradigm [IKOS07].

[26] In this variant, in addition to committing to a permuted graph $\pi(G)$, the prover commits to the permutation π and the permuted cycle $\pi \circ \sigma$. On the 0 challenge, the prover additionally opens the commitment to π, and on the 1 challenge, the prover additionally opens the commitment to $\pi \circ \sigma$.

To view this in terms of generalized k-special soundness, define a consistency predicate g as follows: on input $(\tau_{\text{pre}}, (r_i, \{m_{i,\ell}\}_{\ell \in S_i})_{i\in[k]})$, output 1 if and only if

- For any pair of sets S_i, S_j (corresponding to challenges r_i, r_j), for any $\ell \in S_i \cap S_j$, we have $m_{i,\ell} = m_{j,\ell}$. That is, the "opened" message subsets are mutually consistent.
- For all $i \in [k]$, $\mathsf{Check}(\{m_{i,\ell}\}_{\ell\in S_i}) = 1$.

. With this formalism in place, the following claim is immediate.

Claim 5.12. *Any protocol satisfying commit-and-open k-special soundness (as described in Definition 5.11) is (k, g)-special sound.*

5.3.3 Kilian's Protocol

We briefly recall Kilian's protocol [Kil92] instantiated with a collapsing hash function:

1. The verifier samples a collapsing hash function $h \leftarrow H_\lambda$ and sends h to the prover.
2. Let h_{Merkle} be the Merkle hash function corresponding to h. The prover uses w to compute a PCP π, and then sends $\mathsf{rt} = h_{\text{Merkle}}(\pi)$ to the verifier.
3. The verifier samples random coins r and sends them to the prover.
4. The prover computes the set of the PCP indices q_r that the PCP verifier with randomness r would check. It sends the corresponding values $\pi[q_r]$ along with the Merkle openings of rt on the positions q_r.
5. Finally, the verifier accepts if all the Merkle openings are valid and $V_{\text{PCP},x}(r, \pi[q_r]) = 1$, i.e., the PCP verifier with randomness r accepts $\pi[q_r]$.

We will instantiate Kilian's protocol with a PCP of knowledge, defined as follows. Let $\mathsf{WIN}_{\mathsf{PCP},x}(\pi)$ denote the probability that π is accepted by the PCP verifier.

Definition 5.13 (PCP of Knowledge)**.** A PCP has knowledge error $\kappa_{\text{PCP}}(\lambda)$ if there is an extractor E_{PCP} such that given any PCP π where $\mathsf{WIN}_{\text{PCP},x}(\pi) > \kappa_{\text{PCP}}$, the extractor $\mathsf{E}_{\text{PCP}}(\pi) \to w$ outputs a valid witness w for x with probability 1.

The following claim is due to [CMSZ21], though we have slightly rewritten it to match our definition of k-PSS.

Claim 5.14. *Kilian's protocol instantiated with a PCP with knowledge error $\kappa_{\text{PCP}}(\lambda) = \text{negl}(\lambda)$ proof length $\ell(\lambda)$, and alphabet-size $\Sigma(\lambda)$ is (k, g)-PSS where $k = \ell \log(|\Sigma|)$ and the consistency function g outputs 1 on $(\tau_{\text{pre}}, (r_i, z_i)_{i\in[k]})$ if (1) for each i, the response z_i contains PCP answers $\pi[q_{r_i}]$ such that $V_{\text{PCP}}(x, r_i, \pi[q_{r_i}]) = 1$, and (2) for every $i \neq i'$ the answers $\pi[q_{r_i}]$ and $\pi[q_{r_{i'}}]$ agree on all indices in $q_{r_i} \cap q_{r_{i'}}$.*

Proof. Our extractor $\mathsf{PSSExtract}_g$ takes as input $(\tau_{\text{pre}}, (r_i, z_i)_{i\in[k]})$ and generates a witness as follows:

1. Generate a PCP string $\pi \in \Sigma^\ell$ as follows. For each $t \in [\ell]$, check if $t \in q_{r_i}$ for any i. If so, pick such an i arbitrarily and set $\pi[t]$ according to the value specified in z_i (the choice of i does not matter since the input satisfies consistency with respect to g). If there is no such i, set $\pi[t]$ arbitrarily.

2. Run $\mathsf{E}_{\text{PCP}}(\pi) \to w$ and output w.

We prove that Step 1 constructs a PCP π where $\mathsf{WIN}_{\text{PCP},x}(\pi) > \kappa_{\text{PCP}}$ with $1 - \text{negl}(\lambda)$ probability whenever $(\tau_{\text{pre}}, (r_i, z_i)_{i \in [k]})$ is sampled from a distribution supported on $\mathsf{Consistent}_k$ (i.e., the subset of $T \times (R \times Z)^k$ where g outputs 1) whose marginal distribution on R^k is admissible.

It suffices to prove that if $(r_1, \ldots, r_k)$ are output by Samp (where Samp makes an expected $\text{poly}(\lambda)$ number of queries to a uniform sampling oracle O_R and then outputs a size-k subset of the outputs of O_r) then the probability there *exists* $\pi \in \Sigma^\ell$ such that (1) $\mathsf{WIN}_{\text{PCP},x}(\pi) \leq \kappa_{\text{PCP}}$ and (2) $V_{\text{PCP},x}(r_i, \pi[q_{r_i}]) = 1$ for all $i \in [k]$ is $\text{negl}(\lambda)$. This follows by invoking the definition of an admissible distribution, and observing that any π resulting from Step 1 satisfies (2) by construction, which means that $\mathsf{WIN}_{\text{PCP},x}(\pi) > \kappa_{\text{PCP}}$ with probability $1 - \text{negl}(\lambda)$.

Let d be a constant such that for all $\lambda > \lambda_d$, Samp makes at most λ^d queries to the sampling oracle O_r. Suppose towards contradiction that there exists a constant c such that for infinitely many λ, the probability that Samp outputs $(r_1, \ldots, r_k)$ such that with probability at least $1/\lambda^c$, there *exists* $\pi \in \Sigma^\ell$ satisfying conditions (1) and (2) above. Thus, for infinitely many λ, the probability that Samp makes $2\lambda^{c+d}$ queries (or more) to its sampling oracle O_R is at most $1/(2\lambda^c)$ by Markov's inequality. This means that even if Samp makes at most $2\lambda^{c+d}$ queries to its sampling oracle, it still succeeds with probability at least $1/(2\lambda^c)$ for infinitely many λ.

Consider any fixed PCP π such that $\mathsf{WIN}_{\text{PCP},x}(\pi) \leq \kappa_{\text{PCP}}$. The probability that the PCP is accepting on at least k challenges out of $2\lambda^{c+d}$ uniformly random challenges is at most

$$\kappa_{\text{PCP}}^k \cdot \binom{2\lambda^{c+d}}{k} \leq \kappa_{\text{PCP}}^k (2\lambda^{c+d})^k.$$

By taking a union bound over all $\pi \in \Sigma^\ell$ we conclude that given $2\lambda^{c+d}$ uniformly random challenges, the probability there *exists* a PCP π such that $\mathsf{WIN}_{\text{PCP},x}(\pi) \leq \kappa_{\text{PCP}}$ and π is accepting on at least k of the $2\lambda^{c+d}$ challenges is at most

$$|\Sigma|^\ell \kappa_{\text{PCP}}^k (2\lambda^{c+d})^k = (|\Sigma| \cdot (2\lambda^{c+d}\kappa_{\text{PCP}})^{\log(|\Sigma|)})^\ell,$$

where we have plugged in $k = \ell \log(|\Sigma|)$. Since $\kappa_{\text{PCP}} = \text{negl}(\lambda)$, there exists λ_0 such that $2\lambda^{c+d}\kappa_{\text{PCP}} < \frac{1}{4}$ for all $\lambda > \lambda_0$. Then for all $\lambda > \lambda_0$, we have

$$(|\Sigma| \cdot (2\lambda^{c+d}\kappa_{\text{PCP}})^{\log(|\Sigma|)})^\ell < \frac{1}{|\Sigma|^\ell}$$

Since the PCP alphabet size is at least $|\Sigma| \geq 2$ and the PCP length is at least $\ell \geq \lambda$, the probability that Samp succeeds when restricted to making at at most $2\lambda^{c+d}$ queries to O_R is at most $O(1/2^\lambda)$, which is a contradiction. □

6 Singular Vector Algorithms

In this section we give algorithms for working with states that are singular vectors of a matrix $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$, where $\Pi_{\mathsf{A}}, \Pi_{\mathsf{B}}$ are projectors. In Section 6.2 we give an algorithm that transforms left singular vectors to right singular vectors with negligible error. The runtime of the algorithm depends on the corresponding singular value.

Notation. Throughout this section we will consider the interaction between two binary projective measurements $\mathsf{A} = (\Pi_{\mathsf{A}}, \mathbf{I} - \Pi_{\mathsf{A}})\,, \mathsf{B} = (\Pi_{\mathsf{B}}, \mathbf{I} - \Pi_{\mathsf{B}})$.

We consider the matrix $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$ and its singular value decomposition $V\Sigma W^\dagger$. Recall that V, W are unitary and Σ is a diagonal matrix. The columns of V (resp. W) are the left (resp. right) singular vectors of $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$, and the entries on the diagonal of Σ are the singular values s_j. Note that the singular value decomposition is not in general unique; for the purposes of this section we fix one arbitrarily.

We denote left (resp. right) singular vectors of $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$ with $s_j > 0$ by $|v_{j,1}\rangle$ (resp. $|w_{j,1}\rangle$). Define $\mathcal{S}_j := \mathrm{span}(|v_{j,1}\rangle\,, |w_{j,1}\rangle)$. If $s_j < 1$, then $\mathcal{S}_j$ is two-dimensional. The $\mathcal{S}_j$ correspond to the Jordan subspaces of $(\Pi_{\mathsf{A}}, \Pi_{\mathsf{B}})$. As such, we also have $|v_{j,0}\rangle\,, |w_{j,0}\rangle \in \mathcal{S}_j$. A straightforward calculation shows that these are left and right singular vectors of $(\mathbf{I} - \Pi_{\mathsf{A}})(\mathbf{I} - \Pi_{\mathsf{B}})$ with singular value s_j. The Jordan subspace values p_j are the squares of the corresponding singular values. In our setting it is more natural to use the squares (since they correspond to probabilities), and so the guarantees in this section are stated with respect to the squared singular values.

6.1 Fixed-Runtime Algorithms

In this section we recall a selection of algorithms for manipulating singular vectors of $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$. All of these algorithms make black-box use of $U_{\mathsf{A}}, U_{\mathsf{B}}$; we consider their complexity as circuits with $U_{\mathsf{A}}, U_{\mathsf{B}}$ gates. All of these algorithms take as input some threshold $a \in (0, 1]$, such that their correctness guarantee will hold for singular vectors of value at least a, and their running time is linear in $1/a$.

The first algorithm $\mathsf{Transform}$ implements a fixed-runtime singular vector transformation, taking left singular vectors to their corresponding right singular vectors.

Theorem 6.1 (Singular vector transformation [GSLW19])**.** *There is a uniform family of circuits $\{\mathsf{Transform}_{a,\delta}\}_{a,\delta\in(0,1]}$ with $U_{\mathsf{A}}, U_{\mathsf{B}}$ gates, of size $O(\log(1/\delta)/\sqrt{a})$, such that the following holds. Let $|v_{j,1}\rangle$ be a left singular vector of $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$ with singular value s_j. If $a \leq s_j^2$, $\mathsf{Transform}_{a,\delta}[\mathsf{A} \to \mathsf{B}](|v_{j,1}\rangle)$ outputs the state $|w_{j,1}\rangle$ with probability at least $1 - \delta$. Moreover, for all a, $\mathcal{S}_j$ is invariant under $\mathsf{Transform}_{a,\delta}$.*

The second algorithm $\mathsf{Threshold}$ implements a measurement determining, given a threshold a and a singular vector with singular value s_j, whether $a \leq s_j$ or $a > 2s_j$ (and otherwise has no guarantee).

Theorem 6.2 (Singular value threshold [GSLW19])**.** *There is an algorithm $\mathsf{Threshold}$ which, for all binary projective measurements A, B, given black-box access to operators $U_{\mathsf{A}}, U_{\mathsf{B}}$, achieves the following guarantee. Given $\delta > 0, b \geq \varepsilon > 0$ and a state $|v_{j,1}\rangle$ which is a left singular vector of $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$ with singular value s_j:*

- *if $s_j^2 \geq b$, then $\Pr\Big[\mathsf{Threshold}^{\mathsf{A},\mathsf{B}}_{p,\varepsilon,\delta}(|v_{j,1}\rangle) \to 1\Big] \geq 1 - \delta$, and*
- *if $s_j^2 \leq b - \varepsilon$, then $\Pr\Big[\mathsf{Threshold}^{\mathsf{A},\mathsf{B}}_{p,\varepsilon,\delta}(|v_{j,1}\rangle) \to 1\Big] \leq \delta$.*

Moreover, $\mathcal{S}_j$ is invariant under $\mathsf{Threshold}$, and if the outcome is 1 the post-measurement state is $|v_{j,1}\rangle$. $\mathsf{Threshold}$ runs in time $O(\log(1/\delta)\sqrt{b}/\varepsilon)$.

Next, we describe an algorithm which, with access to $U_{\mathsf{A}}, U_{\mathsf{B}}$, can "flip" a singular vector state from $\mathrm{image}(\mathbf{I} - \Pi_{\mathsf{A}})$ to $\mathrm{image}(\Pi_{\mathsf{A}})$ using Π_{B}, provided that the singular value is sufficiently far from both 0 and 1.

Lemma 6.3. *Let $\Pi_{\mathsf{A}}, \Pi_{\mathsf{B}}$ be projectors. There is an algorithm $\mathsf{Flip}_{\varepsilon}[\Pi_{\mathsf{A}}, \Pi_{\mathsf{B}}]$ which, on input a state $|v_{j,0}\rangle$ that is a left singular vector of $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$ with $\varepsilon \leq s_j^2 \leq 3/4$, outputs the state $|v_{j,1}\rangle$ with probability $1-\delta$ in time $O(\log(1/\delta)/\sqrt{\varepsilon})$. Flip is invariant on the subspace spanned by $\{\,|v_{j,1}\rangle\,,\,|v_{j,0}\rangle\}$.*

Proof. The algorithm operates as follows:

1. Apply A, B in an alternating fashion until either $\mathsf{A} \to 1$, $\mathsf{B} \to 1$ or $3\log(1/\delta)$ measurements have been applied.
2. If $\mathsf{A} \to 1$, stop.
3. If $\mathsf{B} \to 1$, apply $\mathsf{Transform}_{\varepsilon,\delta}[\Pi_{\mathsf{B}}, \Pi_{\mathsf{A}}]$.

The lemma follows since the probability that Step 1 takes more than k steps is $(3/4)^k$, and then by the guarantee of $\mathsf{Transform}$. □

6.2 Variable-Runtime Singular Vector Transformation (vrSVT)

In this section we describe our variable-runtime SVT algorithm. In fact, for technical reasons our algorithm consists of two parts: a variable-runtime *singular value estimation* procedure which *preserves* singular vectors, and a *singular vector transformation* procedure which transforms left singular vectors to right singular vectors, whose running time is fixed given a classical input from the estimation procedure.

Below we give a proof of Theorem 6.4 that makes use of the singular value discrimination and singular vector transformation algorithms of [GSLW19]. We note that it is possible to prove Theorem 6.4 via more "elementary" means using high-probability phase estimation [NWZ09] and amplitude amplification. Indeed, phase estimation for $(2\mathbf{I}-\Pi_{\mathsf{A}})(2\mathbf{I}-\Pi_{\mathsf{B}})$ is equivalent to singular value estimation for $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$ and amplitude amplification can be viewed as a (non-coherent) singular vector transformation.

Theorem 6.4 (Two-stage variable-runtime singular vector transformation)**.** *Let $\mathsf{A} = (\Pi_{\mathsf{A}}, \mathbf{I}-\Pi_{\mathsf{A}})$, $\mathsf{B} = (\Pi_{\mathsf{B}}, \mathbf{I}-\Pi_{\mathsf{B}})$ be projective measurements. There is a pair of algorithms $\mathsf{VarEstimate}[\Pi_{\mathsf{A}} \rightleftarrows \Pi_{\mathsf{B}}]$ and $\mathsf{Transform}[\Pi_{\mathsf{B}} \to \Pi_{\mathsf{A}}]$ with U_{A} and U_{B} gates with the following properties. Let $|w_{j,1}\rangle$ be a left singular vector of $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$ with singular value $s_j > 0$, and let $|v_{j,1}\rangle$ be the corresponding right singular vector. Then*

1. *The subspace $\mathcal{S}_j$ is invariant under both $\mathsf{VarEstimate}$ and $\mathsf{Transform}$.*
2. *The running time of $\mathsf{VarEstimate}(\,|v_{j,1}\rangle)$ is $O(\log(1/\delta)/s_j)$ with probability $1-\delta$ and $O(\log(1/\delta)/\delta)$ with probability 1.*
3. *The output $(q,\,|\psi\rangle) \leftarrow \mathsf{VarEstimate}(\,|v_{j,1}\rangle)$ is such that $|\psi\rangle = |w_{j,1}\rangle$ with probability $1-\delta$.*
4. *The running time of $\mathsf{Transform}(q,\,|\psi'\rangle)$, where $(\gamma,\,|\psi\rangle) \leftarrow \mathsf{VarEstimate}(\,|v_{j,1}\rangle)$ and $|\psi'\rangle$ is any state, is $O(\log(1/\delta)/s_j)$ with probability $1-\delta$ and at most $1/\delta$ with probability 1.*
5. *The output state of $\mathsf{Transform}(\mathsf{VarEstimate}(\,|v_{j,1}\rangle))$ is $|w_{j,1}\rangle$ with probability $1-\delta$.*

The $\mathsf{Transform}$ procedure above can be instantiated directly via the singular vector transformation algorithm of [GSLW19], see Theorem 6.1.

We describe an implementation of $\mathsf{VarEstimate}$ using the singular value discrimination algorithm (Theorem 6.2). For a binary projective measurement A, let $\bar{\mathsf{A}}$ denote the same measurement with the outcome labels reversed. For k in the procedure below, define $b := 2^{-k}$ and $\varepsilon := 2^{-k-1}$.

1. Set $b := 0$, $k := 0$. Repeat the following two steps until $b = 1$ or $k \geq \lceil \log(1/\delta) \rceil$:
   (a) Set $k \leftarrow k + 1$.
   (b) Apply B, obtaining outcome c.
   (c) If $c = 1$, apply $\mathsf{Threshold}^{\mathsf{A},\mathsf{B}}(\gamma, \varepsilon, \delta / \log(1/\delta))$ obtaining outcome $b \in \{0, 1\}$.
   (d) If $c = 0$, apply $\mathsf{Threshold}^{\bar{\mathsf{A}},\bar{\mathsf{B}}}(\gamma, \varepsilon, \delta / \log(1/\delta))$ obtaining outcome $b \in \{0, 1\}$.
2. Apply B, obtaining outcome c. If $c = 0$, apply $\mathsf{Flip}_{2^{-k-1}}[\mathsf{A}, \mathsf{B}]$.
3. Output 2^{-k-1}.

Lemma 6.5 (Variable-runtime singular value estimation)**.** *Let $|v_{j,1}\rangle$ be a left singular vector with singular value s_j. Let $\delta > 0$. $\mathsf{VarEstimate}_\delta[\mathsf{A} \rightleftarrows \mathsf{B}](|v_{j,1}\rangle, \delta)$ runs in time $O(\log(1/\delta)/s_j)$ with probability $1 - \delta$ and $O(\log(1/\delta)/\delta)$ with probability 1. Moreover, $\mathsf{VarEstimate}$ outputs a in the range $\max(\delta, s_j^2)/4 \leq a \leq \max(\delta, s_j^2)$ with probability $1 - \delta$.*

Proof. First, observe that k iterations of Step 1 take time $O(\log(1/\delta) \cdot 2^k)$. Since $\mathsf{VarEstimate}$ terminates within $\lceil \log(1/\delta) \rceil$ iterations of Step 1 with probability 1, $\mathsf{VarEstimate}$ runs in time $O(\log(1/\delta)/\delta)$ with probability 1.

The probability that the singular value discrimination algorithm outputs 1 when $2^{-k} > 2s_j$ is at most $\delta/(\log(1/\delta))$. Similarly, the probability that it outputs 1 when $2^{-k} \leq s_j$ is at least $1 - \delta/(\log(1/\delta))$. By a union bound, with probability at least $1 - \delta$ the algorithm either stops in the first iteration where $2^{-k} \leq 2s_j$ (so $s_j < 2^{-k} \leq 2s_j$) or in the following iteration ($s_j/2 < 2^{-k} \leq s_j$). Thus $2^{-k} \in [s_j/2, 2s_j]$, so $2^{-k-1} \in [s_j/4, s_j]$ as required. The running time in this case is $O(\log(1/\delta)/s_j)$.

If $s_j \geq 1/2$ then the algorithm stops after one iteration in state $|w_{j,1}\rangle$ with probability $1 - \delta$. Otherwise the probability that $\log(1/\delta)$ alternating measurements A, B are applied with only 0 outcomes is at most δ. If Step 2 terminates with $\mathsf{B} \to 1$, then the resulting state is $|w_{j,1}\rangle$. Otherwise, the resulting state is $|v_{j,1}\rangle$. In this case the $\mathsf{Transform}$ algorithm rotates the state to $|w_{j,1}\rangle$ with probability $1 - \delta$. □

The next two claims follow directly from the correctness and subspace invariance guarantees of $\mathsf{Threshold}$ and $\mathsf{VarEstimate}$.

Corollary 6.6. *For any state $\boldsymbol{\rho}$, $\delta > 0$, $\varepsilon \colon [0, 1] \to [\delta, 1]$:*

$$\Pr\Big[\mathsf{Threshold}_{p,\varepsilon(p),\delta}(\mathsf{VarEstimate}(\boldsymbol{\rho})) = 1\Big] \geq 1 - 2\delta,$$

where p is the classical output from $\mathsf{VarEstimate}$.

Corollary 6.7. *For any state $\boldsymbol{\rho}$, $\delta > 0$, $\varepsilon \in [\delta, 1]$:*

$$\Pr\left[b_1 = 1 \wedge b_2 = 0 \;\middle|\; \begin{array}{r} (b_1, \boldsymbol{\rho}_1) \leftarrow \mathsf{Threshold}_{p,\varepsilon,\delta}(\boldsymbol{\rho}) \\ (b_2, \boldsymbol{\rho}_2) \leftarrow \mathsf{Threshold}_{p-\varepsilon,\varepsilon,\delta}(\boldsymbol{\rho}_1) \end{array} \right] \leq 2\delta \ .$$

Moreover,

$$\Pr\left[b_1 = 1 \wedge p_j < p - \varepsilon \;\middle|\; \begin{array}{r} (b_1, \boldsymbol{\rho}_1) \leftarrow \mathsf{Threshold}_{p,\varepsilon,\delta}(\boldsymbol{\rho}) \\ j \leftarrow \mathsf{M}_{\mathsf{Jor}}[\mathsf{A}, \mathsf{B}](\boldsymbol{\rho}_1) \end{array} \right] \leq \delta \ .$$

7 Pseudoinverse Lemma

In this section we show that for binary projective measurements A,B any state $|\psi_{\mathsf{A}}\rangle$ in the image of Π_{A}, there is a state $|\psi_{\mathsf{B}}\rangle$ in the image of B such that $|\psi_{\mathsf{A}}\rangle$ is (approximately) obtained by applying A to $|\psi_{\mathsf{B}}\rangle$ and conditioning on obtaining a 1. Moreover, if $|\psi_{\mathsf{A}}\rangle$ has Jordan spectrum that is concentrated around eigenvalue p, then $|\psi_{\mathsf{A}}\rangle$ has the same property. We refer to this as the "pseudoinverse lemma" because $|\psi_{\mathsf{B}}\rangle$ is obtained from $|\psi_{\mathsf{A}}\rangle$ by applying the pseudoinverse of the matrix $\Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$.

Lemma 7.1 (Pseudoinverse Lemma)**.** *Let A,B be binary projective measurements, and let $\{\mathcal{S}_j\}_j$ be the induced Jordan decomposition. Let Π_j^{Jor} be the projection on to $\mathcal{S}_j$ and let p_j be the eigenvalue of $\mathcal{S}_j$. Let $\boldsymbol{\rho}$ be a state such that $\mathrm{Tr}(\Pi_{\mathsf{A}}\boldsymbol{\rho}) = 1$ and let $\Pi_0 \coloneqq \sum_{j,p_j=0}\Pi_j^{\mathsf{Jor}}$. Let $E \coloneqq \sum_{j,p_j>0}\frac{1}{p_j}\Pi_j^{\mathsf{Jor}}$. There exists a "pseudoinverse" state $\boldsymbol{\rho}'$ with $\mathrm{Tr}(\Pi_{\mathsf{B}}\boldsymbol{\rho}') = 1$ such that all of the following are true:*

1. $\mathrm{Tr}(\Pi_{\mathsf{A}}\boldsymbol{\rho}') = \frac{1-\mathrm{Tr}(\Pi_0\boldsymbol{\rho})}{\mathrm{Tr}(E\boldsymbol{\rho})}$,
2. $d\left(\boldsymbol{\rho}, \frac{\Pi_{\mathsf{A}}\boldsymbol{\rho}'\Pi_{\mathsf{A}}}{\mathrm{Tr}(\Pi_{\mathsf{A}}\boldsymbol{\rho}')}\right) \leq 2\sqrt{\mathrm{Tr}(\Pi_0\boldsymbol{\rho})}$,
3. *for all j such that $p_j > 0$ it holds that* $\mathrm{Tr}\left(\Pi_j^{\mathsf{Jor}}\boldsymbol{\rho}'\right) = \frac{\mathrm{Tr}\left(\Pi_j^{\mathsf{Jor}}\boldsymbol{\rho}\right)}{p_j\cdot\mathrm{Tr}(E\boldsymbol{\rho})}$, *and*
4. *for all j such that $p_j = 0$ it holds that* $\mathrm{Tr}\left(\Pi_j^{\mathsf{Jor}}\boldsymbol{\rho}'\right) = 0$.

An important consequence of (3) and (4) is that for all j, if $\mathrm{Tr}\left(\Pi_j^{\mathsf{Jor}}\boldsymbol{\rho}\right) = 0$ then $\mathrm{Tr}\left(\Pi_j^{\mathsf{Jor}}\boldsymbol{\rho}'\right) = 0$.

Proof. Let $C \coloneqq \Pi_{\mathsf{A}}\Pi_{\mathsf{B}}$, and note that $|v_{j,1}\rangle, |w_{j,1}\rangle$ are corresponding left and right singular vectors of C with singular value $\sqrt{p_j}$. Hence $C = \sum_{p_j>0}\sqrt{p_j}\,|v_{j,1}\rangle\langle w_{j,1}|$. Let C^+ be the pseudoinverse of C, i.e., $C^+ = \sum_{p_j>0}\frac{1}{\sqrt{p_j}}|w_{j,1}\rangle\langle v_{j,1}|$. Define

$$\boldsymbol{\rho}' \coloneqq \frac{C^+\boldsymbol{\rho}(C^+)^\dagger}{\mathrm{Tr}(C^+\boldsymbol{\rho}(C^+)^\dagger)}.$$

Since $\mathrm{Tr}(\Pi_{\mathsf{A}}\boldsymbol{\rho}) = 1$, we have $\mathrm{Tr}(\Pi_{\mathsf{B}}\boldsymbol{\rho}') = 1$. We also have

$$\mathrm{Tr}\left(C^+\boldsymbol{\rho}(C^+)^\dagger\right) = \mathrm{Tr}\left((CC^\dagger)^+\boldsymbol{\rho}\right) = \sum_j \frac{1}{p_j}\langle v_{j,1}|\boldsymbol{\rho}|v_{j,1}\rangle = \mathrm{Tr}(E\boldsymbol{\rho}). \tag{1}$$

Next, observe that since $CC^+ = \sum_{p_j>0}|v_{j,1}\rangle\langle v_{j,1}| = \mathbf{I} - \Pi_0$, we have

$$\begin{aligned}
\Pi_{\mathsf{A}}\boldsymbol{\rho}'\Pi_{\mathsf{A}} &= \Pi_{\mathsf{A}}\left(\frac{C^+\boldsymbol{\rho}(C^+)^\dagger}{\mathrm{Tr}(C^+\boldsymbol{\rho}(C^+)^\dagger)}\right)\Pi_{\mathsf{A}} \\
&= \Pi_{\mathsf{A}}\left(\frac{C^+\boldsymbol{\rho}(C^+)^\dagger}{\mathrm{Tr}(E\boldsymbol{\rho})}\right)\Pi_{\mathsf{A}} \\
&= \Pi_{\mathsf{A}}\Pi_{\mathsf{B}}\left(\frac{C^+\boldsymbol{\rho}(C^+)^\dagger}{\mathrm{Tr}(E\boldsymbol{\rho})}\right)\Pi_{\mathsf{B}}\Pi_{\mathsf{A}} \\
&= \frac{1}{\mathrm{Tr}(E\boldsymbol{\rho})}CC^+\boldsymbol{\rho}(CC^+)^\dagger \\
&= \frac{1}{\mathrm{Tr}(E\boldsymbol{\rho})}(\mathbf{I}-\Pi_0)\boldsymbol{\rho}(\mathbf{I}-\Pi_0).
\end{aligned} \tag{2}$$

Given these calculations, we can prove the claimed properties (1-3) in the lemma statement:

- **Proof of (1).** Taking the trace of both sides of Eq. (2), we see that

$$\mathrm{Tr}(\Pi_{\mathsf{A}}\boldsymbol{\rho}') = \mathrm{Tr}(\Pi_{\mathsf{A}}\boldsymbol{\rho}'\Pi_{\mathsf{A}}) = \frac{1}{\mathrm{Tr}(E\boldsymbol{\rho})}\mathrm{Tr}\left((\mathbf{I}-\Pi_0)\boldsymbol{\rho}(\mathbf{I}-\Pi_0)\right) = \frac{\mathrm{Tr}\left((\mathbf{I}-\Pi_0)\boldsymbol{\rho}\right)}{\mathrm{Tr}(E\boldsymbol{\rho})} = \frac{1-\mathrm{Tr}\left(\Pi_0\boldsymbol{\rho}\right)}{\mathrm{Tr}(E\boldsymbol{\rho})}.$$

- **Proof of (2).** Given Eq. (2) and the trace calculation above, we have that

$$\frac{\Pi_{\mathsf{A}}\boldsymbol{\rho}'\Pi_{\mathsf{A}}}{\mathrm{Tr}(\Pi_{\mathsf{A}}\boldsymbol{\rho}')} = \frac{1}{1-\mathrm{Tr}(\Pi_0\boldsymbol{\rho})}(\mathbf{I}-\Pi_0)\boldsymbol{\rho}(\mathbf{I}-\Pi_0)$$

The inequality $d\left(\boldsymbol{\rho}, \frac{\Pi_{\mathsf{A}}\boldsymbol{\rho}'\Pi_{\mathsf{A}}}{\mathrm{Tr}(\Pi_{\mathsf{A}}\boldsymbol{\rho}')}\right) \leq 2\sqrt{\mathrm{Tr}(\Pi_0\boldsymbol{\rho})}$ now follows from Lemma 3.1 (gentle measurement).

- **Proof of (3).** For all j such that $p_j > 0$, making use of the same calculation as Eq. (1), we have

$$\mathrm{Tr}\left(\Pi_j^{\mathsf{Jor}}\boldsymbol{\rho}'\right) = \frac{\mathrm{Tr}\left(\Pi_j^{\mathsf{Jor}}C^{+}\boldsymbol{\rho}(C^{+})^{\dagger}\right)}{\mathrm{Tr}(C^{+}\boldsymbol{\rho}(C^{+})^{\dagger})} = \frac{\mathrm{Tr}\left(C^{+}\Pi_j^{\mathsf{Jor}}\boldsymbol{\rho}(C^{+})^{\dagger}\right)}{\mathrm{Tr}(E\boldsymbol{\rho})} = \frac{\mathrm{Tr}\left(E\,\Pi_j^{\mathsf{Jor}}\boldsymbol{\rho}\right)}{\mathrm{Tr}(E\boldsymbol{\rho})} = \frac{\mathrm{Tr}\left(\frac{1}{p_j}\Pi_j^{\mathsf{Jor}}\boldsymbol{\rho}\right)}{\mathrm{Tr}(E\boldsymbol{\rho})}.$$

- **Proof of (4).** This follows immediately from the fact that $\Pi_0 C^{+} = C^{+}\Pi_0 = 0$.

This completes the proof of Lemma 7.1. □

We conclude this section by showing that under a mild condition, any state $\boldsymbol{\rho}$ that is close to $\mathrm{image}(\Pi_{\mathsf{A}})$ has a nearby state in $\mathrm{image}(\Pi_{\mathsf{A}})$ with the same Jordan decomposition.

Claim 7.2. *Let $\boldsymbol{\rho}$ be any state. Let $\Pi_{\mathsf{stuck}}^{\mathsf{Jor}}$ project on to one-dimensional subspaces $\mathcal{S}_j$ in the image of $\mathbf{I}-\Pi_{\mathsf{A}}$. There exists a state $\boldsymbol{\sigma}$ such that for all j, $\mathrm{Tr}\left(\Pi_j^{\mathsf{Jor}}\boldsymbol{\sigma}\right) = \mathrm{Tr}\left(\Pi_j^{\mathsf{Jor}}\boldsymbol{\rho}\right)$, $\mathrm{Tr}(\Pi_{\mathsf{A}}\boldsymbol{\sigma}) = 1-\mathrm{Tr}\left(\Pi_{\mathsf{stuck}}^{\mathsf{Jor}}\cdot\boldsymbol{\rho}\right)$, and $d(\boldsymbol{\rho},\boldsymbol{\sigma}) \leq \sqrt{1-\mathrm{Tr}(\Pi_{\mathsf{A}}\boldsymbol{\rho})}$.*

Proof. Define a unitary U which is invariant on the $\mathcal{S}_j$ and, in each two-dimensional $\mathcal{S}_j$, rotates $|v_{j,0}\rangle$ to $|v_{j,1}\rangle$. Formally,

$$U := \sum_{j, p_j \notin \{0,1\}} \left(|v_{j,1}\rangle\langle v_{j,0}| + |v_{j,0}\rangle\langle v_{j,1}|\right) + \mathbf{I}_{\mathcal{S}^{(1)}},$$

where $\mathcal{S}^{(1)}$ is the direct sum of the 1D subspaces. Set

$$\boldsymbol{\sigma} := \Pi_{\mathsf{A}}\boldsymbol{\rho}\Pi_{\mathsf{A}} + U(I-\Pi_{\mathsf{A}})\boldsymbol{\rho}(I-\Pi_{\mathsf{A}})U^{\dagger}.$$ □

8 Post-Quantum Guaranteed Extraction

In this section, we give a post-quantum extraction procedure for various 3- and 4-message public-coin interactive protocols. In particular, we will consider interactive protocols satisfying *partial collapsing* (Definition 3.5) with respect to some class of efficiently computable functions $F = \{f : T \times R \times Z \to \{0,1\}^*\}$. Our goal is to establish *guaranteed extraction*, defined below (essentially matching Definition 2.2).

Definition 8.1. (P_Σ, V_Σ) is a post-quantum proof of knowledge with *guaranteed extraction* if it has an extractor $\mathsf{Extract}^{P^*}$ of the following form.

1. $\mathsf{Extract}^{P^*}$ first runs the cheating prover P^* to generate a (classical) first message a along with an instance x (in a 4-message protocol, this requires first sampling a random vk and running $P^*(\mathsf{vk})$ to obtain x, a).
2. $\mathsf{Extract}^{P^*}$ runs P^* coherently on the superposition $\sum_{r \in R} |r\rangle$ of all challenges to obtain a superposition $\sum_{r,z} \alpha_{r,z} |r, z\rangle$ over challenge-response pairs.[27]
3. $\mathsf{Extract}^{P^*}$ then computes (in superposition) the verifier's decision $V(x, a, r, z)$ and measures it. If the measurement outcome is 0, the extractor gives up.
4. If the measurement outcome is 1, run some quantum procedure $\mathsf{FindWitness}^{P^*}$ that outputs a string w.

We require that the following two properties hold.

- **Correctness (guaranteed extraction):** The probability that the initial measurement returns 1 but the output witness w is not a valid witness for x is $\mathrm{negl}(\lambda)$.
- **Efficiency:** For any QPT P^*, the procedure $\mathsf{Extract}^{P^*}$ is in EQPT_m.

We remark that this definition is written to capture (first-message) *adaptive soundness*, where the prover P^* is allowed to choose the instance x when it sends its first message. One could alternatively define a non-adaptive variant of this definition in which the instance x is fixed in advance (and this section's results would hold in this setting as well). Definition 8.1 suffices for our purposes since none of the 4-message protocols we consider have the first verifier message vk depend on x (in all cases we consider, vk is just a commitment key or hash function key), and the protocols all satisfy adaptive soundness.

8.0.1 Notation

Let $\mathcal{R}$ denote a register with the basis $\{|r\rangle\}_{r \in R}$ and let $|+_R\rangle_{\mathcal{R}} := \frac{1}{\sqrt{|R|}} \sum_{r \in R} |r\rangle$. Let $\mathcal{H}$ denote the prover's state (including its workspace), and let U_r denote the unitary on $\mathcal{H}$ that the prover applies on challenge r. Let $\mathcal{Z}$ denote the subregister of $\mathcal{H}$ that the prover measures to obtain its response z after applying U_r.

Define the projector

$$\Pi_{V,r} = U_r^\dagger \left(\sum_{z : V(r,z)=1} |z\rangle\langle z|_{\mathcal{Z}} \otimes \mathbf{I} \right) U_r$$

which intuitively projects onto subspace of $\mathcal{H}$ where the prover gives an accepting response on challenge r.

Define the binary projective measurement $\mathsf{C} = (\Pi_{\mathsf{C}}, \mathbf{I} - \Pi_{\mathsf{C}})$ where

$$\Pi_{\mathsf{C}} = \sum_r |r\rangle\langle r|_{\mathcal{R}} \otimes \Pi_{V,r},$$

and $\mathsf{U} = (\Pi_{\mathsf{U}}, \mathbf{I} - \Pi_{\mathsf{U}})$ where

$$\Pi_{\mathsf{U}} = |+_R\rangle\langle +_R|_{\mathcal{R}} \otimes \mathbf{I}_{\mathcal{H}}.$$

[27] In general, the response z will be entangled with the prover's state; here we suppress this dependence.

8.1 Description of the Extractor

We first give a full description of an extraction procedure $\mathsf{Extract}$, defined for any partially collapsing protocol.

The threshold unitary. Consider the following measurement procedure $\mathsf{T}_{p,\varepsilon,\delta}$ on $\mathcal{H}$, parameterized by threshold p, accuracy ε and error δ.

- Initialize a fresh register $\mathcal{R}$ to $|+_R\rangle_{\mathcal{R}}$.
- Run $\mathsf{Threshold}^{\mathsf{U},\mathsf{C}}_{p,\varepsilon,\delta}$ on $\mathcal{H} \otimes \mathcal{R}$, obtaining outcome b.
- Trace out $\mathcal{R}$ and output b.

We define $U_{p,\varepsilon,\delta}$ to be a *coherent* implementation of $\mathsf{T}_{p,\varepsilon,\delta}$. $U_{p,\varepsilon,\delta}$ acts on $\mathcal{H}\otimes\mathcal{W}\otimes\mathcal{B}$ where $\mathcal{W}\otimes\mathcal{B}$ is an ancilla register: $\mathcal{W}$ contains the algorithm's workspace and $\mathcal{B}$ is a single qubit containing the measurement outcome. In particular, applying $U_{\varepsilon,\delta}$ to $|\psi\rangle_{\mathcal{H}}\,|0\rangle_{\mathcal{W},\mathcal{B}}$, measuring $\mathcal{B}$, and then tracing out $\mathcal{W}$ implements the above measurement.

The repair measurements. We define the two projective measurements $\mathsf{D}_r = (\Pi_r, \mathbf{I} - \Pi_r)\,, \mathsf{G}_{p,\varepsilon,\delta} = (\Pi_{p,\varepsilon,\delta}, \mathbf{I} - \Pi_{p,\varepsilon,\delta})$ for our repair step.

For any $p, \varepsilon, \delta > 0$, define the projector $\Pi_{p,\varepsilon,\delta}$ on $\mathcal{H} \otimes \mathcal{W} \otimes \mathcal{B}$ as follows:

$$\Pi_{p,\varepsilon,\delta} \coloneqq U^{\dagger}_{p,\varepsilon,\delta}(\mathbf{I}_{\mathcal{H},\mathcal{W}} \otimes |1\rangle\!\langle 1|_{\mathcal{B}})U_{p,\varepsilon,\delta}.$$

For any $r \in R$, we define the projector Π_r on $\mathcal{H} \otimes \mathcal{W}$ as

$$\Pi_r \coloneqq (\Pi_{V,r})_{\mathcal{H}} \otimes |0\rangle\!\langle 0|_{\mathcal{W}}\,.$$

We describe the extraction procedure $\mathsf{Extract}^{P^*}_V(x)$. The procedure is defined with respect to k efficiently computable functions $f_1, \ldots, f_k : T \times R \times Z \to \{0,1\}^*$.

1. **Initial Execution.** Use P^* to generate (vk, a), and let $|\psi\rangle$ denote the residual prover state. Apply $\mathsf{C} = (\Pi_{\mathsf{C}}, \mathbf{I} - \Pi_{\mathsf{C}})$ to $|\psi\rangle_{\mathcal{H}} \otimes |+_R\rangle_{\mathcal{R}}$. If 0, terminate (note that we do not consider this an "abort".) Otherwise:

2. **Estimate success probability.** Run $\mathsf{VarEstimate}[\mathsf{C} \rightleftarrows \mathsf{U}]$ (as defined in Section 6.2) with $\frac{1}{2}$-multiplicative error and failure probability $\delta = 1/2^{\lambda}$, outputting a value p. Note that since the input state is in Π_{C}, the algorithm produces an output state in Π_{C} with probability $1 - \delta$.

   Abort if $p < \lambda k\sqrt{\delta}$. Define $\varepsilon = \frac{p}{4k}$.

3. **Main Loop.** Repeat the following "main loop" for i from 1 to k:

   (a) **Lower bound success probability.** Run $\mathsf{Threshold}^{\mathsf{C},\mathsf{U}}_{p,\varepsilon,\delta}$ on $\mathcal{H} \otimes \mathcal{R}$, obtaining outcome b. Abort if $b = 0$. Update $p \coloneqq p - \varepsilon$.

   (b) **Measure the challenge.** Measure the $\mathcal{R}$ register, obtaining a particular challenge $r_i \in R$. Discard the $\mathcal{R}$ register.

   (c) **Estimate the running time of Transform.** Initialize the $\mathcal{W}$ register to $|0\rangle_{\mathcal{W}}$ and run $\mathsf{VarEstimate}[\mathsf{D}_r \rightleftarrows \mathsf{G}_{p,\varepsilon,\delta}]$ with $\frac{1}{2}$-multiplicative error and failure probability

$\delta = 2^{-\lambda}$, obtaining classical output q. Since the input state is in Π_r, the algorithm produces an output state in Π_r with probability $1-\delta$.

(d) **Record part of the accepting response.** Make a partial measurement of the prover response z_i; specifically, measure $y_i = f_i(z_i)$.[a] **If $i = k$, go to Step 4.**

(e) **Transform onto good states.** Apply $\mathsf{Transform}_q[\mathsf{D}_r \to \mathsf{G}_{p,\varepsilon,\delta}]$ with failure probability $\delta = 2^{-\lambda}$.

(f) Next, apply $U_{p,\varepsilon,\delta}$ and then discard the $\mathcal{W}$ register. Update $p := p - \varepsilon$.

(g) **Transform onto accepting executions.** Re-initialize $\mathcal{R}$ to $|+_R\rangle$ and then apply $\mathsf{Transform}_p[\mathsf{U} \to \mathsf{C}]$; abort if this procedure fails.

4. Output $(\mathsf{vk}, a, r_1, y_1, \cdots, r_k, y_k)$.

The above procedure deterministically terminates and aborts if it has not already stopped after $O(k)/\sqrt{\delta}$ steps, for $\delta := 2^{-\lambda}$.

[a] Formally, we (1) apply the prover unitary U_{r^*} to $\mathcal{H}$, (2) apply the projective measurement $\left(\Pi_y\right)_y$ for $\Pi_y = \sum_{z:f_i(z)=y} |z\rangle\langle z|_{\mathcal{Z}} \otimes \mathbf{I}_{\mathcal{H}'}$ (where $\mathcal{H} = \mathcal{Z} \otimes \mathcal{H}'$), and (3) apply $U_{r^*}^\dagger$ to $\mathcal{H}$.

8.2 Partial Transcript Extraction Theorem

Our most general extraction theorem is stated for *any* partially collapsing protocol, but is only guaranteed to output partial transcripts (rather than a witness). In Section 8.4, we show how this theorem can be used to establish guaranteed extraction of a witness.

Theorem 8.2. *For any 4-message public-coin interactive argument satisfying partial collapsing (Definition 3.4) with respect to the functions $f_1, \ldots, f_{k-1}$ (but* not *necessarily f_k), the procedure $\mathsf{Extract}_V$ has the following properties for any instance x.*

1. ***Efficiency:*** *For any QPT prover P^*, $\mathsf{Extract}_V^{P^*}$ runs in expected polynomial time (EQPT_m). More formally, the number of calls that $\mathsf{Extract}_V^{P^*}$ makes to P^* is a classical random variable whose expectation is a fixed polynomial in k, λ.*

2. ***Correctness:*** *$\mathsf{Extract}$ aborts with negligible probability.*

3. ***Distribution of outputs****: For every choice of (vk, a), let $\gamma = \gamma_{\mathsf{vk},a}$ denote the success probability of P^* conditioned on first two messages (vk, a). Then, if $\gamma > \delta^{1/3}$, the distribution of $(r_1, \ldots, r_k)$ (conditioned on (vk, a) and a successful first execution) is $O(1/\gamma)$-admissible (Definition 5.5).*

8.3 Proof of Theorem 8.2

8.3.1 Intermediate State Notation

Our extraction procedure and analysis make use of four relevant registers:

- A challenge randomness register $\mathcal{R}$,
- A prover state register $\mathcal{H}$, and
- A phase estimation workspace register $\mathcal{W}$.

- A one qubit register $\mathcal{B}$ that contains a bit b where $b = 1$ indicates that the computation has not aborted during a sub-computation.

We now establish some conventions:

- states written using the letter $\boldsymbol{\rho}$ satisfy $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{B} \otimes \mathcal{H} \otimes \mathcal{R})$ or $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H} \otimes \mathcal{R})$, where we use $\mathbf{S}(\mathcal{H})$ to denote the space of Hermitian operators on $\mathcal{H}$;
- states using the letter $\boldsymbol{\sigma}$ satisfy $\boldsymbol{\sigma} \in \mathbf{S}(\mathcal{B} \otimes \mathcal{H} \otimes \mathcal{W})$ or $\boldsymbol{\sigma} \in \mathbf{S}(\mathcal{H} \otimes \mathcal{W})$;
- states using the letter $\boldsymbol{\phi}$ satisfy $\boldsymbol{\phi} \in \mathbf{S}(\mathcal{B} \otimes \mathcal{H})$ or $\boldsymbol{\phi} \in \mathbf{S}(\mathcal{H})$;
- states using the letter $\boldsymbol{\tau}$ satisfy $\boldsymbol{\tau} \in \mathbf{S}(\mathcal{H} \otimes \mathcal{W} \otimes \mathcal{R})$

With these conventions in mind, we define some intermediate states related to the extraction procedure:

- Let ψ denote the prover state after (vk, a) is generated.
- Let $\boldsymbol{\rho}^{(2)}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ denote the state obtained at the end of Step 2.
- For each iteration of the Step 3 loop, we define the following states:
  - Let $\boldsymbol{\rho}^{(\mathrm{init})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ denote the state at the beginning of Step 3. The register $\mathcal{B}$ is initialized to $|1\rangle\langle 1|$. For the rest of the loop iteration, $\mathcal{B}$ is set to $|0\rangle\langle 0|$ if the computation aborts.
  - Let $\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ denote the state at the end of Step 3a.
  - Let $\boldsymbol{\phi}^{(3b)}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ denote the state at the end of Step 3b.
  - Let $\boldsymbol{\sigma}^{(3c)}_{\mathcal{B},\mathcal{H},\mathcal{W}}$ denote the state at the end of Step 3c.
  - Let $\boldsymbol{\sigma}^{(3e)}_{\mathcal{B},\mathcal{H},\mathcal{W}}$ denote the state immediately before the $\mathcal{W}$ register is traced out during Step 3e.
  - Let $\boldsymbol{\phi}^{(3f)}_{\mathcal{B},\mathcal{H}}$ denote the state at the end of Step 3f.
  - Let $\boldsymbol{\rho}^{(3g)}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ denote the state at the end of Step 3g.

As in Lemma 7.1, let the Jordan decomposition of $\mathcal{H} \otimes \mathcal{R}$ corresponding to $\Pi_{\mathsf{C}}, \Pi_{\mathsf{U}}$ be $\{\mathcal{S}_j\}_j$ where subspace $\mathcal{S}_j$ is associated with the eigenvalue/success probability p_j. Let Π^{Jor}_j the projection onto $\mathcal{S}_j$, i.e., $\mathrm{image}(\Pi^{\mathsf{Jor}}_j) = \mathcal{S}_j$. Define the following projections on $\mathcal{H} \otimes \mathcal{R}$:

- $\Pi^{\mathsf{Jor}}_0 := \sum_{j,p_j=0} \Pi^{\mathsf{Jor}}_j$
- $\Pi^{\mathsf{Jor}}_{\geq p} = \sum_{j:p_j \geq p} \Pi^{\mathsf{Jor}}_j$
- $\Pi^{\mathsf{Jor}}_{<p} = \sum_{j:p_j < p} \Pi^{\mathsf{Jor}}_j$

We additionally define the following projectors on $\mathcal{B} \otimes \mathcal{H} \otimes \mathcal{R}$.

$$\Pi^{\mathsf{Jor}}_{\mathsf{Bad}} = |1\rangle\langle 1|_{\mathcal{B}} \otimes \Pi^{\mathsf{Jor}}_{<p} \quad \text{and} \quad \Pi^{\mathsf{Jor}}_{\mathsf{Good}} = \mathbf{I}_{\mathcal{B},\mathcal{H},\mathcal{R}} - \Pi^{\mathsf{Jor}}_{\mathsf{Bad}}.$$

Claim 8.3. *For any estimate p and any state $\boldsymbol{\rho}^{(\mathrm{init})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ such that $\mathrm{Tr}\Big((\mathbf{I}_{\mathcal{B}} \otimes \Pi_{\mathsf{C}})\boldsymbol{\rho}^{(\mathrm{init})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\Big) = 1$, the state $\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ obtained by running $b \leftarrow \mathsf{Threshold}^{\mathsf{C},\mathsf{U}}_{p,\varepsilon,\delta}$ (and then redefining $p := p - \varepsilon$) and setting $\mathcal{B} = |b\rangle\langle b|$ satisfies*

$$\mathrm{Tr}\Big(\Pi^{\mathsf{Jor}}_{\mathsf{Bad}}\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\Big) \leq \delta.$$

Proof. When $\mathsf{Threshold}^{\mathsf{C},\mathsf{U}}_{p,\varepsilon,\delta}$ returns 0, the computation aborts. Therefore, the lemma follows immediately from the almost-projectivity of $\mathsf{Threshold}$ (Corollary 6.7). □

8.3.2 Analysis of Steps 1 and 2

We first show that Steps 1 and 2 run in expected polynomial time, and bound the statistic $\mathbb{E}[1/p \cdot X_1]$, where p is the output of Step 2 and X_1 is the indicator for the event "Step 1 does not abort".

Lemma 8.4. *The expected runtime of Steps 1 and 2 is $O(1)$. Moreover,*

$$\mathbb{E}[1/p \cdot X_1] = O(1).$$

Proof. Let $|\psi\rangle$ denote the state of P^* after (ck, a) are generated. Then, consider the (U, C)-Jordan decomposition

$$|\psi\rangle \otimes |+_R\rangle = \sum_j \alpha_j |v_{j,1}\rangle ,$$

where each $|v_{j,1}\rangle \in \mathcal{S}_j \cap \text{image}(\Pi_{\mathsf{U}})$. Let $\gamma = \sum_j |\alpha_j|^2 p_j$ denote the initial success probability of $|\psi\rangle$.

Step 1 runs in a fixed polynomial time and aborts with probability $1 - \gamma$. Otherwise, Step 2 is run on the residual state

$$\frac{1}{\sqrt{\gamma}} \sum_j \alpha_j \sqrt{p_j} |w_{j,1}\rangle ,$$

where $|w_{j,1}\rangle$ is a basis vector in $\mathcal{S}_j \cap \text{image}(\Pi_{\mathsf{C}})$. Lemma 6.5 tells us that *both* the runtime of $\mathsf{VarEstimate}^{\mathsf{C},\mathsf{U}}$ on this state (making oracle use of C, U) and the expectation of $1/p$ (where p is the output of Step 2) are at most a constant times

$$\frac{1}{\gamma} \sum_j \alpha_j^2 p_j \cdot \frac{1}{p_j} + \delta \cdot 1/\delta \le \frac{1}{\gamma} \sum_j \alpha_j^2 + 1 = \frac{1}{\gamma} + 1,$$

so since $\Pr[\text{Step 1 does not abort}] = \gamma$, the overall expected value bounds are as claimed. □

8.3.3 The Pseudoinverse State

As defined earlier, let $\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ denote the state at the end of Step 3a (for some arbitrary iteration of Step 3). We prove some important properties of the subsequent states in the execution of Step 3.

We begin with Step 3b, which measures the $\mathcal{R}$ register, obtaining a challenge r and resulting state $\boldsymbol{\phi}^{(3b)}_{\mathcal{B},\mathcal{H}}$. Let

$$\boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}} = \frac{\Pi^{\mathsf{Jor}}_{\mathsf{Good}} \boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}} \Pi^{\mathsf{Jor}}_{\mathsf{Good}}}{\mathrm{Tr}\left(\Pi^{\mathsf{Jor}}_{\mathsf{Good}} \boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\right)}$$

denote the residual state; we write $\boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}} = \alpha_0 |0\rangle\langle 0|_{\mathcal{B}} \otimes \boldsymbol{\rho}'^{(\mathsf{C},0)}_{\mathcal{H},\mathcal{R}} + \alpha_1 |1\rangle\langle 1|_{\mathcal{B}} \otimes \boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$. By Claim 8.3, we have that:

Claim 8.5. $\mathrm{Tr}\left(\Pi^{\mathsf{Jor}}_{\mathsf{Good}} \boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\right) \ge 1 - \delta$

By gentle measurement, it then follows that

$$d(\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}, \boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}) \le 2\sqrt{\delta}.$$

Let $\boldsymbol{\rho}^{(\mathsf{U})}_{\mathcal{B},\mathcal{H},\mathcal{R}} = \alpha_0 |0\rangle\langle 0|_{\mathcal{B}} \otimes \boldsymbol{\rho}^{(\mathsf{U},0)}_{\mathcal{H},\mathcal{R}} + \alpha_1 |1\rangle\langle 1|_{\mathcal{B}} \otimes \boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}$ where $\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}$ denotes the state guaranteed to exist by applying Lemma 7.1 with $\mathsf{B} = \mathsf{U}$ and $\mathsf{A} = \mathsf{C}$ on $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$. Recall from Lemma 7.1 that $\mathrm{Tr}\left(\Pi_{\mathsf{U}} \boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right) = 1$, since $\mathrm{Tr}\left(\Pi^{\mathsf{Jor}}_0 \boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}\right) = 0$. Moreover, we also have:

Claim 8.6.

$$\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}} = \frac{\Pi_{\mathsf{C}}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\Pi_{\mathsf{C}}}{\operatorname{Tr}\left(\Pi_{\mathsf{C}}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right)}.$$

Proof. $\boldsymbol{\rho}^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$ is a state satisfying $\operatorname{Tr}\left(\Pi_{\mathsf{C}}\boldsymbol{\rho}^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}\right) = 1$, and $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$ is then a (re-normalized) projection of $\boldsymbol{\rho}^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$ onto (U,C)-Jordan subspaces with bounded Jordan p_j-value. Therefore, $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$ also satisfies $\operatorname{Tr}\left(\Pi_{\mathsf{C}}\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}\right) = 1$.

From Lemma 7.1 (Property 2) we then have

$$d\Big(\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}, \frac{\Pi_{\mathsf{C}}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\Pi_{\mathsf{C}}}{\operatorname{Tr}\left(\Pi_{\mathsf{C}}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right)}\Big) \leq 2\sqrt{\operatorname{Tr}\left(\Pi_0\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}\right)} = 0,$$

which implies $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}} = \Pi_{\mathsf{C}}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\Pi_{\mathsf{C}}/\operatorname{Tr}\left(\Pi_{\mathsf{C}}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right)$. □

Finally, because $\operatorname{Tr}\left(\Pi^{\mathsf{Jor}}_{\mathsf{Bad}}\boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\right) = 0$, Lemma 7.1 (Property 3) tells us that $\operatorname{Tr}\left(\Pi^{\mathsf{Jor}}_{\mathsf{Bad}}\boldsymbol{\rho}^{(\mathsf{U})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\right) = 0$ as well.

Define $p_{\mathsf{U}} = \operatorname{Tr}\left(\Pi_{\mathsf{C}}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right)$ to be the normalization factor above, which is equal to the $(\mathsf{C}$-)success probability of $\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}$.

Claim 8.7. $p_{\mathsf{U}} \geq p$.

Proof. Since $\operatorname{Tr}\left(\Pi^{\mathsf{Jor}}_{\geq p}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right) = 1$ and Π_{C} commutes with each Π^{Jor}_j, we have:

$$\begin{aligned}
\operatorname{Tr}\left(\Pi_{\mathsf{C}}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right) &= \operatorname{Tr}\left(\Pi_{\mathsf{C}}\Pi^{\mathsf{Jor}}_{\geq p}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right) \\
&= \operatorname{Tr}\left(\sum_{j:p_j\geq p}\Pi_{\mathsf{C}}\Pi^{\mathsf{Jor}}_j\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right) \\
&\geq p\operatorname{Tr}\left(\sum_{j:p_j\geq p}\Pi^{\mathsf{Jor}}_j\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right) \\
&= p.
\end{aligned}$$

□

Since $\operatorname{Tr}\left(\Pi_{\mathsf{U}}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right) = 1$, it can be written in the form $\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|$. For each r, we define $\zeta_r := \operatorname{Tr}\left(\Pi_{V,r}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\right)$ to be the success probability of $\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}$ on r. Finally, define $\zeta_R = \sum_r \zeta_r$.

We now proceed to analyze the state $\boldsymbol{\phi}^{(3b)}_{\mathcal{B},\mathcal{H}}$. To do so, we first define $\boldsymbol{\phi}'^{(3b)}_{\mathcal{B},\mathcal{H}}$ to be the state at the end of Step 3b when $\boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ is used in place of $\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$. We know that $d(\boldsymbol{\phi}^{(3b)}_{\mathcal{B},\mathcal{H}}, \boldsymbol{\phi}'^{(3b)}_{\mathcal{B},\mathcal{H}}) \leq 2\sqrt{\delta}$, so this characterization will suffice.

Claim 8.8. *The state $\boldsymbol{\phi}'^{(3b)}_{\mathcal{B},\mathcal{H}}$ is a mixed state with the following form: with probability α_0, it is in the abort state. Otherwise, with conditional probability ζ_r/ζ_R, Step 3g measures challenge r and the resulting state is $|1\rangle\langle 1|_{\mathcal{B}} \otimes \frac{\Pi_{V,r}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\Pi_{V,r}}{\zeta_r}$.*

Proof. By definition of the pseudoinverse state $\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}} = \boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|$, we can write $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$ as

$$\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}} = \frac{\Pi_{\mathsf{C}}(\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|)\Pi_{\mathsf{C}}}{\mathrm{Tr}\Big(\Pi_{\mathsf{C}}(\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|)\Big)}.$$

Since $\Pi_{\mathsf{C}} = \sum_{r \in R} \Pi_{V,r} \otimes |r\rangle\langle r|$, we can write

$$\begin{aligned}\Pi_{\mathsf{C}}(\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|)\Pi_{\mathsf{C}} &= (\sum_{r\in R} \Pi_{V,r} \otimes |r\rangle\langle r|) \Big(\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|\Big) (\sum_{r\in R} \Pi_{V,r} \otimes |r\rangle\langle r|) \\ &= \frac{1}{|R|} \sum_{r\in R} \Pi_{V,r}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\Pi_{V,r} \otimes |r\rangle\langle r|\,.\end{aligned}$$

Thus, we can rewrite $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$ as

$$\begin{aligned}\frac{\Pi_{\mathsf{C}}(\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|)\Pi_{\mathsf{C}}}{\mathrm{Tr}\Big(\Pi_{\mathsf{C}}(\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|)\Big)} &= \frac{\frac{1}{|R|}\sum_{r\in R}\Pi_{V,r}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\Pi_{V,r} \otimes |r\rangle\langle r|}{\mathrm{Tr}\Big(\frac{1}{|R|}\sum_{r\in R}\Pi_{V,r}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\Pi_{V,r} \otimes |r\rangle\langle r|\Big)} \\ &= \frac{\frac{1}{|R|}\sum_{r\in R}\Pi_{V,r}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\Pi_{V,r} \otimes |r\rangle\langle r|}{\frac{1}{|R|}\sum_{r\in R}\zeta_r} \\ &= \frac{\sum_{r\in R}\Pi_{V,r}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\Pi_{V,r} \otimes |r\rangle\langle r|}{\zeta_R}.\end{aligned}$$

Therefore, the probability of obtaining r after measuring $\mathcal{R}$ of $\boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ is

$$\begin{aligned}\frac{\mathrm{Tr}\Big((\mathbf{I} \otimes |r\rangle\langle r|)\sum_{r'\in R}\Pi_{V,r'}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\Pi_{V,r'} \otimes |r'\rangle\langle r'|\Big)}{\zeta_R} &= \frac{\mathrm{Tr}\Big(\Pi_{V,r}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\Pi_{V,r} \otimes |r\rangle\langle r|\Big)}{\zeta_R} \\ &= \frac{\zeta_r}{\zeta_R},\end{aligned}$$

and the post-measurement state is $\Pi_{V,r}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\Pi_{V,r}$. Thus, the state $\boldsymbol{\phi}'^{(3b)}_{\mathcal{B},\mathcal{H}}$ is as claimed. □

In particular, Claim 8.8 tells us that the ratio $\frac{\zeta_R}{|R|}$ is exactly $\mathrm{Tr}(\Pi_{\mathsf{C}}\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}) = p_{\mathsf{U}}$.

We begin our analysis of the repair step by defining the following states:

1. $\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}} := \boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |0\rangle\langle 0|_{\mathcal{W}}$. Here, $\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}$ is the state satisfying $\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}} = \boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |0\rangle\langle 0|_{\mathcal{R}}$.
2. $\boldsymbol{\sigma}^{(r,1)}_{\mathcal{H},\mathcal{W}} := \Pi_r\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}\Pi_r/\zeta_r$.

By Claim 8.8 we can view our variant of Steps 3b to 3e as follows:

- With probability α_0, abort. Otherwise:
- A challenge is sampled so that each string r occurs with probability $\frac{\zeta_r}{\zeta_R}$
- If the string r is sampled, initialize the state to $|1\rangle\langle 1|_{\mathcal{B}} \otimes \boldsymbol{\sigma}^{(r,1)}_{\mathcal{H},\mathcal{W}}$.

Unfortunately, the state $\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}$ only satisfies $\mathrm{Tr}\Big(\Pi_{p,\varepsilon}\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}\Big) \geq 1-\delta$ (it is not *quite* fully in the image of $\Pi_{p,\varepsilon}$). With this in mind, we define two additional states:

3. $\widetilde{\boldsymbol{\sigma}}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}} := \frac{\Pi_{p,\varepsilon}\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}\Pi_{p,\varepsilon}}{\mathrm{Tr}\Big(\Pi_{p,\varepsilon}\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}\Big)}$. Since $\mathrm{Tr}\Big(\Pi_{p,\varepsilon}\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}\Big) = 1-\delta$, we have $d(\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}, \widetilde{\boldsymbol{\sigma}}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}) \leq 2\sqrt{\delta}$ by Lemma 3.1.
4. $\widetilde{\boldsymbol{\sigma}}^{(r,1)}_{\mathcal{H},\mathcal{W}} := \Pi_r\widetilde{\boldsymbol{\sigma}}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}\Pi_r/\,\mathrm{Tr}\Big(\Pi_r\widetilde{\boldsymbol{\sigma}}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}\Big)$.

Let $\widetilde{\zeta}_r := \mathrm{Tr}\Big(\Pi_r\widetilde{\boldsymbol{\sigma}}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}\Big)$, and observe that $\widetilde{\zeta}_r \in [\zeta_r \pm 2\sqrt{\delta}]$. Define $\widetilde{\zeta}_R := \sum_{r\in R}\widetilde{\zeta}_r$ and $\widetilde{p}_{\mathsf{U}} := \widetilde{\zeta}_R/|R|$.

Claim 8.9. $|\widetilde{p}_{\mathsf{U}} - p_{\mathsf{U}}| \leq 2\sqrt{\delta}$

Proof. For every string r, we have that

$$|\zeta_r - \widetilde{\zeta}_r| = \Big|\mathrm{Tr}(\Pi_r(\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}} - \widetilde{\boldsymbol{\sigma}}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}})\Big| \leq 2\sqrt{\delta}$$

since $||\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}} - \widetilde{\boldsymbol{\sigma}}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}|| \leq 2\sqrt{\delta}$. Therefore, we have that

$$\left|\frac{\zeta_R}{|R|} - \frac{\widetilde{\zeta}_R}{|R|}\right| \leq 2\sqrt{\delta}$$

by subadditivity. □

Consider the following two mixed states

$$\boldsymbol{\tau}_{\mathcal{H},\mathcal{R},\mathcal{W}} := \sum_r \frac{\zeta_r}{\zeta_R}\,|r\rangle\langle r| \otimes \frac{\Pi_r\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}\Pi_r}{\zeta_r}, \text{ and}$$

$$\widetilde{\boldsymbol{\tau}}_{\mathcal{H},\mathcal{R},\mathcal{W}} := \sum_r \frac{\widetilde{\zeta}_r}{\widetilde{\zeta}_R}\,|r\rangle\langle r| \otimes \frac{\Pi_r\widetilde{\boldsymbol{\sigma}}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}\Pi_r}{\widetilde{\zeta}_r}.$$

We claim that these two mixed states are close in trace distance.

Claim 8.10. $||\boldsymbol{\tau} - \widetilde{\boldsymbol{\tau}}||_1 \leq \frac{4\sqrt{\delta}}{p_{\mathsf{U}}}$.

To see this, we first note that

$$\begin{aligned}\left|\left|\widetilde{\boldsymbol{\tau}} - \frac{\widetilde{\zeta}_R}{\zeta_R}\widetilde{\boldsymbol{\tau}}\right|\right|_1 &= |1 - \frac{\widetilde{\zeta}_R}{\zeta_R}| \cdot ||\widetilde{\boldsymbol{\tau}}||_1 \\ &= \left|1 - \frac{\widetilde{\zeta}_R}{\zeta_R}\right| \\ &= \left|1 - \frac{\widetilde{p}_{\mathsf{U}}}{p_{\mathsf{U}}}\right| \\ &\leq \frac{2\sqrt{\delta}}{p_{\mathsf{U}}}\end{aligned}$$

Moreover, we have that

$$\begin{aligned}\left|\left|\boldsymbol{\tau} - \frac{\widetilde{\zeta}_R}{\zeta_R}\widetilde{\boldsymbol{\tau}}\right|\right|_1 &= \frac{1}{\zeta_R}\left|\left|\sum_r |r\rangle\langle r| \otimes \Pi_r(\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}} - \widetilde{\boldsymbol{\sigma}}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}})\Pi_r\right|\right|_1 \\ &\leq \frac{|R|}{\zeta_R} \cdot ||\boldsymbol{\sigma}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}} - \widetilde{\boldsymbol{\sigma}}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{W}}||_1 \\ &\leq \frac{|R|}{\zeta_R} \cdot 2\sqrt{\delta} \\ &= \frac{2\sqrt{\delta}}{p_{\mathsf{U}}}.\end{aligned}$$

Thus, we conclude that $||\boldsymbol{\tau} - \widetilde{\boldsymbol{\tau}}||_1 \leq \frac{2\sqrt{\delta}}{p_{\mathsf{U}}} + \frac{2\sqrt{\delta}}{p_{\mathsf{U}}} \leq \frac{4\sqrt{\delta}}{p_{\mathsf{U}}}$ by the triangle inequality.

This trace bound will allow us to analyze correctness and bound the expected runtime of the extractor by appealing to properties of the state $\widetilde{\boldsymbol{\tau}}$.

8.3.4 Runtime Analysis

In this section, we bound the expected running time of Ext (proving property (1) of Theorem 8.2).

Theorem 8.11. *For any QPT P^*, Ext^{P^*} runs in EQPT_m.*

We note that Lemma 8.4 already showed that the expected running time of Steps 1 and 2 is $O(1)$ calls to (U, C).

Next, we show that the expected runtime of the main loop (Step 3) is also $\text{poly}(\lambda)$. To prove this, we make use of the syntactic property (enforced by the definition of Step 3g) that for every $i \in \{0, 1, \ldots, k\}$, the state $\boldsymbol{\rho}^{(\mathsf{C},\text{init})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ at the beginning of the ith iteration of Step 3 is in $\text{image}(\Pi_{\mathsf{C}})$ (provided that the computation has not aborted). We then show

Lemma 8.12. *Let p be an arbitrary real number output by Step 2, and let $\boldsymbol{\rho}^{(\mathsf{C},\text{init})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ be an* arbitrary *non-aborted state (i.e. $\mathcal{B}$ is initialized to $|1\rangle\langle 1|_{\mathcal{B}}$) that is in the image of Π_{C}.*

Then, the expected runtime of one iteration of Step 3 on $\boldsymbol{\rho}^{(\mathsf{C},\text{init})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ is $\text{poly}(\lambda)/p$.

Proof. We analyze the running time assuming that the collapsing measurement of Step 3d is not performed. This is without loss of generality; the steps following the (partial) collapsing measurement have a fixed runtime (as a function of previously computed parameters in the execution), so the collapsing measurement cannot affect the overall expected running time.[28]

First, note that Steps 3a and 3g run in a fixed $\text{poly}(\lambda)/\sqrt{p}$ time by Theorem 6.1. Thus, we focus on Steps 3c and 3e.

We bound the expected runtime of Steps 3c and 3e via the following hybrid argument.

- Hyb_0: This is the real procedure, assuming that Step 3d is not performed.
- Hyb_1: In this hybrid, the $\mathcal{R}$-measurement outcome and residual state $\boldsymbol{\phi}^{(3b)}_{\mathcal{B},\mathcal{H}}$ is prepared differently:

[28] We only remove the collapsing measurement of the *current* loop iteration; previous collapsing measurements are baked into the (arbitrary) state $\boldsymbol{\rho}^{(\mathsf{C},\text{init})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$.

  - With probability α_0, abort. Otherwise:
  - The challenge r is sampled with probability equal to ζ_r/ζ_R.
  - If the string r is sampled, initialize the state to $|1\rangle\langle 1|_{\mathcal{B}} \otimes \frac{\Pi_{V,r}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\Pi_{V,r}}{\zeta_r}$.

  This is an alternate description of the state $\boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$.

- Hyb_2: In this hybrid, the state at the beginning of Step 3c (which is usually $\boldsymbol{\phi}^{(3b)}_{\mathcal{B},\mathcal{H}} \otimes |0\rangle\langle 0|_{\mathcal{W}}$) is prepared differently:
  - With probability α_0, abort. Otherwise:
  - A challenge is sampled so that each string r occurs with probability $\widetilde{\zeta}_r/\widetilde{\zeta}_R$
  - If the string r is sampled, initialize the state to $\widetilde{\boldsymbol{\sigma}}^{(r,1)}_{\mathcal{H},\mathcal{W}}$.

Claim 8.13. *The expected running times of* Hyb_0, Hyb_1 *differ by at most* $O(k)$*, and the expected running times of* Hyb_1 *and* Hyb_2 *differ by at most* $O(k/p)$*.*

Proof. The *worst-case* running time of Ext is bounded to be $k/\sqrt{\delta}$ by definition. We will combine this with trace distance bounds to prove the claim.

For Hyb_0 and Hyb_1, we note that the running time of Steps 3c and 3e can be viewed as a classical distribution over integers obtained via applying a CPTP map to the input state, which is either $\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ or $\boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$. Since trace distance is contractive under CPTP maps, we conclude that these integer distributions are $2\sqrt{\delta}$-close in statistical distance. Since (as integers) they are $(k/\sqrt{\delta})$-bounded, we conclude that their expectations differ by $O(k)$.

The argument is similar for Hyb_1 and Hyb_2, except that the running time of Steps 3c and 3e can instead be viewed as a classical distribution obtained via a CPTP map from either $\boldsymbol{\tau}$ or $\widetilde{\boldsymbol{\tau}}$, which have trace distance at most $\frac{4\sqrt{\delta}}{p_{\mathsf{U}}} \leq \frac{4\sqrt{\delta}}{p}$. □

Thus, it suffices to bound the expected runtime in the procedure Hyb_2.

Without Step 3d, we can view Steps 3c and 3e as a variable-runtime $\mathsf{Transform}^{\mathsf{D}_r \to \mathsf{G}_{p,\varepsilon,\delta}}$ with respect to the projectors $(\Pi_r, \Pi_{p,\varepsilon})$, where r is sampled from the above distribution. We first analyze the runtime of this procedure for a fixed value of r.

Let $\mathsf{Jor}_r = (\Pi^{\mathsf{Jor}_r}_j)_j$ denote the Jordan measurement corresponding to projections $(\Pi_r, \Pi_{p,\varepsilon})$, and let q_j denote the eigenvalue associated with $\Pi^{\mathsf{Jor}_r}_j$. Define the *Jordan weights* of $\widetilde{\boldsymbol{\sigma}}^{(\mathsf{U})}_{\mathcal{B},\mathcal{H},\mathcal{W}}$ as the vector $(y^{\mathsf{Jor}_r}_j)_j$ where

$$y^{\mathsf{Jor}_r}_j := \mathrm{Tr}\Big(\Pi^{\mathsf{Jor}_r}_j \widetilde{\boldsymbol{\sigma}}^{(\mathsf{U})}_{\mathcal{B},\mathcal{H},\mathcal{W}}\Big).$$

Then, the Jordan weights of $\widetilde{\boldsymbol{\sigma}}^{(r)}_{\mathcal{B},\mathcal{H},\mathcal{W}}$ are $(z^{\mathsf{Jor}_r}_j)_j$ where

$$z^{\mathsf{Jor}_r}_j := q_j y^{\mathsf{Jor}_r}_j / \widetilde{\zeta}_r.$$

Claim 8.14. *Given the string* r *and state* $\widetilde{\boldsymbol{\sigma}}^{(r)}_{\mathcal{B},\mathcal{H},\mathcal{W}}$ *as input, Steps 3c to 3e make an expected* $\mathrm{poly}(\lambda) \cdot 1/\sqrt{\widetilde{\zeta}_r}$ *calls to* $\Pi_{p,\varepsilon}$ *and* Π_r*.*

Proof. By Theorem 6.4, the expected running time (in number of calls to $\Pi_r, \Pi_{p,\varepsilon}$) of Steps 3c to 3e on a state with Jordan weights $(q_j y_j^{\mathsf{Jor}_r}/\widetilde{\zeta}_r)_j$ is

$$\sum_j \frac{q_j y_j^{\mathsf{Jor}_r}}{\widetilde{\zeta}_r} \cdot \frac{\mathrm{poly}(\lambda)}{\sqrt{q_j}} \le \mathrm{poly}(\lambda)\sqrt{\sum_j \frac{q_j y_j^{\mathsf{Jor}_r}}{\widetilde{\zeta}_r q_j}}$$
$$= \mathrm{poly}(\lambda)\sqrt{\frac{1}{\widetilde{\zeta}_r}}.$$

This completes the proof of Claim 8.14. □

By Claim 8.14, along with the fact that $\Pi_{p,\varepsilon}$ is implemented in a fixed $\mathrm{poly}(\lambda)/\sqrt{p}$ time, the expected running time of Steps 3c to 3e in Hyb_2 is:

$$\begin{aligned}
\sum_{r \in R} \frac{\widetilde{\zeta}_r}{\widetilde{\zeta}_R} \cdot \frac{\mathrm{poly}(\lambda)}{\sqrt{\widetilde{\zeta}_r p}} &= \frac{\mathrm{poly}(\lambda)}{\sqrt{p}} \sum_{r \in R} \frac{\widetilde{\zeta}_r}{\widetilde{\zeta}_R} \cdot \frac{1}{\sqrt{\widetilde{\zeta}_r}} \\
&\le \frac{\mathrm{poly}(\lambda)}{\sqrt{p}} \sqrt{\sum_{r \in R} \frac{\widetilde{\zeta}_r}{\widetilde{\zeta}_R} \cdot \frac{1}{\widetilde{\zeta}_r}} \\
&= \frac{\mathrm{poly}(\lambda)}{\sqrt{p}} \sqrt{\frac{|R|}{\widetilde{\zeta}_R}} \\
&= \frac{\mathrm{poly}(\lambda)}{\sqrt{p}} \sqrt{\frac{1}{\widetilde{p}_{\mathsf{U}}}} \\
&\le \frac{\mathrm{poly}(\lambda)}{\sqrt{p(p - 2\sqrt{\delta})}} \\
&\le \frac{\mathrm{poly}(\lambda)}{p} \qquad (3)
\end{aligned}$$

where the first inequality is an application of Jensen's inequality, the second inequality holds by Claim 8.9, and the last inequality holds by the abort condition in Step 2 (p drops by a factor of at most 2 in the entire process).

This completes the proof of Lemma 8.12.

□

Finally, combining Lemma 8.12 with Lemma 8.4 along with the fact that throughout the extraction procedure, the updated value of p is at most a factor of 2 smaller than the initial output of Step 2, we conclude that the overall expected running time of Step 3 is at most $\mathbb{E}[\mathrm{poly}(\lambda)/p] \le \mathrm{poly}(\lambda)$, completing the proof of Theorem 8.11.

8.3.5 Correctness of the repair step

In this section, we prove that $\mathsf{Extract}$ aborts with negligible probability (property (2) of Theorem 8.2).

Lemma 8.15. *The probability that the procedure aborts is negligible.*

Proof. By Theorem 8.11, the probability that the procedure aborts because it ran for too long is $O(\mathrm{poly}(\lambda)/\sqrt{\delta}) = \mathrm{negl}(\lambda)$.

By Lemma 8.4, $\mathbb{E}[1/p \cdot X] = O(1)$, where X is the indicator for whether Step 1 outputs 1. Hence by Claim 8.16 (proven below) and Corollary 6.6 (which implies that the first iteration of Step 3a only aborts with negligible probability), the probability that any iteration of the loop aborts when we remove Step 3d is at most

$$k \cdot O(\sqrt{\delta})\mathbb{E}[1/p \cdot X] = O(\sqrt{\delta}).$$

Then by the collapsing guarantee (applied to the measurements of $y_1, \ldots, y_{k-1}$; it is not necessary for y_k), and by Theorem 8.11, the probability that any iteration of the loop aborts is negligible. □

Claim 8.16. *Let $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H} \otimes \mathcal{R})$ be a state such that $\mathrm{Tr}(\Pi_{\mathsf{C}} \boldsymbol{\rho}_{\mathcal{H},\mathcal{R}}) = 1$, and consider running two iterations of Step 3 in sequence on $\boldsymbol{\rho}_{\mathcal{H},\mathcal{R}}$, with the following modifications:*

- *Step 3d is not applied, and*
- *The $O(k)/\sqrt{\delta}$ runtime cutoff has been removed.*

Then, for any choice of $p \in [0,1]$, the probability that the first iteration (with initial value p) does not abort in Step 3a and the second iteration aborts in Step 3a is at most $O(\sqrt{\delta}/p)$.

Also, the probability that an iteration of Step 3 (where Step 3d is not applied) does not abort in Step 3a but does abort in Step 3g is at most $O(\sqrt{\delta}/p)$.

Proof. For a projector Π, we write $\Pi^{\mathcal{B}}$ to denote the projection

$$|0\rangle\langle 0|_{\mathcal{B}} \otimes \mathbf{I} + |1\rangle\langle 1|_{\mathcal{B}} \otimes \Pi.$$

Let $\mathcal{B}$ store the output of $\mathsf{Threshold}$ in the first application of Step 3a. Recall that in Section 8.3.3 we have defined the following states:

- $\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ denotes the state after applying Step 3a (i.e., coherently applying $\mathsf{Threshold}$ to $\boldsymbol{\rho}_{\mathcal{H},\mathcal{R}}$ where the output is stored on $\mathcal{B}$). The extraction procedure now re-defines/updates $p := p - \varepsilon$. Note that $\mathrm{Tr}\Big((\Pi^{\mathsf{Jor}}_{\geq p})^{\mathcal{B}} \boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\Big) \geq 1 - \delta$ (Claim 8.5).
- $\boldsymbol{\rho}^{(\mathsf{C},b)}_{\mathcal{H},\mathcal{R}} := \langle b|_{\mathcal{B}}\, \boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\, |b\rangle_{\mathcal{B}} / q$ where $q := \mathrm{Tr}\Big(\langle b|_{\mathcal{B}}\, \boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\, |b\rangle_{\mathcal{B}}\Big)$. We may assume $q > 0$ or else the claim holds trivially.
- $\boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}} := \frac{\left(\Pi^{\mathsf{Jor}}_{\geq p}\right)^{\mathcal{B}} \boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}} \left(\Pi^{\mathsf{Jor}}_{\geq p}\right)^{\mathcal{B}}}{\mathrm{Tr}\left(\left(\Pi^{\mathsf{Jor}}_{\geq p}\right)^{\mathcal{B}} \boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\right)}$ is the result of projecting $\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ onto eigenvalues $\geq p$ (for the Jordan decomposition corresponding to $\Pi_{\mathsf{C}}, \Pi_{\mathsf{U}}$) when $\mathcal{B} = 1$.
- $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}} := \langle 1|_{\mathcal{B}}\, \boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\, |1\rangle_{\mathcal{B}} / q'$ where $q' := \mathrm{Tr}\Big(\langle 1|_{\mathcal{B}}\, \boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{B},\mathcal{H},\mathcal{R}}\, |1\rangle_{\mathcal{B}}\Big)$. Note that $\mathrm{Tr}\Big(\Pi_{\mathsf{C}} \boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}\Big) = 1$.
- $\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}$ denotes the pseudoinverse of $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$ with respect to (U, C) as guaranteed by the pseudoinverse lemma (Lemma 7.1); by definition, $\mathrm{Tr}\Big(\Pi_{\mathsf{U}} \boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\Big) = 1$. Moreover $\mathrm{Tr}\Big(\Pi_{\mathsf{C}} \boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\Big) \geq p$ (Claim 8.7) since all the $(\Pi_{\mathsf{C}}, \Pi_{\mathsf{U}})$-Jordan-eigenvalues of $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$ are at least p, which implies the same property holds for the pseudoinverse state.

- $\boldsymbol{\phi}_{\mathcal{H}}^{(\mathsf{U},1)} \coloneqq \mathrm{Tr}_{\mathcal{R}}(\boldsymbol{\rho}_{\mathcal{H},\mathcal{R}}^{(\mathsf{U},1)})$. Note that since $\mathrm{Tr}\Big(\Pi_{\mathsf{U}}\boldsymbol{\rho}_{\mathcal{H},\mathcal{R}}^{(\mathsf{U},1)}\Big) = 1$, we have $\boldsymbol{\rho}_{\mathcal{H},\mathcal{R}}^{(\mathsf{U},1)} = \boldsymbol{\phi}_{\mathcal{H}}^{(\mathsf{U},1)} \otimes |+\rangle\langle+|_{\mathcal{R}}$.
- $\boldsymbol{\rho}_{\mathcal{H},\mathcal{R}}^{\prime(3b,1)} \coloneqq \frac{\sum_r \Pi_{V,r}\boldsymbol{\phi}_{\mathcal{H}}^{(\mathsf{U},1)}\Pi_{V,r}\otimes|r\rangle\langle r|}{|R|\cdot\mathrm{Tr}\Big(\Pi_{\mathsf{C}}\boldsymbol{\rho}_{\mathcal{H},\mathcal{R}}^{(\mathsf{U},1)}\Big)}$ is within trace distance $2\sqrt{\delta}$ of the state after measuring the $\mathcal{R}$ register in Step 3b but *before* discarding $\mathcal{R}$ (Claim 8.8).
- $\boldsymbol{\sigma}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)} \coloneqq \boldsymbol{\phi}_{\mathcal{H}}^{(\mathsf{U},1)} \otimes |0\rangle\langle 0|_{\mathcal{W}}$. We have that $\mathrm{Tr}\Big(\Pi_{p,\varepsilon,\delta}\boldsymbol{\sigma}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)}\Big) \geq 1-\delta$ because $\mathsf{Threshold}_{p,\varepsilon,\delta}$ outputs 1 on $\boldsymbol{\rho}_{\mathcal{H},\mathcal{R}}^{(\mathsf{U},1)}$ with probability $1-\delta$ by the Jordan spectrum guarantee of Lemma 7.1.
- $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)} \coloneqq \frac{\Pi_{p,\varepsilon,\delta}\boldsymbol{\sigma}^{(\mathsf{U},1)}\Pi_{p,\varepsilon,\delta}}{\mathrm{Tr}\big(\Pi_{p,\varepsilon,\delta}\boldsymbol{\sigma}^{(\mathsf{U},1)}\big)}$. By the gentle measurement lemma (Lemma 3.1), we have $d(\boldsymbol{\phi}_{\mathcal{H}}^{(\mathsf{U},1)} \otimes |0\rangle\langle 0|_{\mathcal{W}}, \widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)}) \leq 2\sqrt{\delta}$.
- $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(r,1)} \coloneqq \frac{\Pi_r\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)}\Pi_r}{\widetilde{\zeta}_r}$ where $\widetilde{\zeta}_r \coloneqq \mathrm{Tr}\Big(\Pi_r\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)}\Big)$ for all $r \in R$.
- $\widetilde{\boldsymbol{\tau}}_{\mathcal{H},\mathcal{W},\mathcal{R}} \coloneqq \sum_r \frac{\widetilde{\zeta}_r}{\widetilde{\zeta}_R}\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(r,1)} \otimes |r\rangle\langle r|$ for $\widetilde{\zeta}_R = \sum_r \widetilde{\zeta}_r$. By Claim 8.10, we have that $\widetilde{\boldsymbol{\tau}}_{\mathcal{H},\mathcal{W},\mathcal{R}}$ is within trace distance $\frac{4\sqrt{\delta}}{p}$ of the state $\boldsymbol{\tau} = \boldsymbol{\rho}_{\mathcal{H},\mathcal{R}}^{\prime(3b,1)} \otimes |0\rangle\langle 0|_{\mathcal{W}}$.

We now consider the application of the variable-runtime singular vector transform performed across Steps 3c and 3e (recall that we omit Step 3d for this analysis). We consider applying these steps to the state $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(r,1)}$. Note that Steps 3c and 3e commute with $\mathsf{M}_{\mathsf{Jor}}[\mathsf{D}_r, \mathsf{G}_{p,\varepsilon,\delta}]$. Hence writing $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(3e,r,1)}$ for the state after applying Steps 3c and 3e to $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(r,1)}$, we have

$$\mathrm{Tr}\Big(\Pi_j^{\mathsf{Jor}_r}\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(3e,r,1)}\Big) = \mathrm{Tr}\Big(\Pi_j^{\mathsf{Jor}_r}\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(r,1)}\Big) = \frac{q_j\,\mathrm{Tr}\Big(\Pi_j^{\mathsf{Jor}_r}\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)}\Big)}{\widetilde{\zeta}_r}$$

where $\Pi_j^{\mathsf{Jor}_r}$ is the j-th element of $\mathsf{M}_{\mathsf{Jor}}[\mathsf{D}_r, \mathsf{G}_{p,\varepsilon,\delta}]$. Let $\Pi_{\mathsf{G}_{p,\varepsilon,\delta},\mathsf{stuck}}^{\mathsf{Jor}_r}$ be defined (analogous to $\Pi_{\mathsf{stuck}}^{\mathsf{Jor}}$ in Claim 7.2) as $\Pi_{\mathsf{stuck}}^{\mathsf{Jor}} \coloneqq \sum_{j\in S}\Pi_j^{\mathsf{Jor}_r}$ where S is the set of all j where $\mathcal{S}_j$ is a one-dimensional Jordan subspace $\mathcal{S}_j \in \mathrm{image}(\mathbf{I}-\Pi_{p,\varepsilon,\delta})$. We now invoke Claim 7.2 to "rotate" the state $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(3e,r,1)}$ into $\mathrm{image}(\Pi_{p,\varepsilon,\delta})$ while preserving the Jordan spectrum, which is possible as long as the component of the state in $\Pi_{\mathsf{G}_{p,\varepsilon,\delta},\mathsf{stuck}}^{\mathsf{Jor}_r}$ is 0. This is satisfied here because $\mathrm{Tr}\Big(\Pi_{\mathsf{stuck}}^{\mathsf{Jor}_r}\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(3e,r,1)}\Big) = \mathrm{Tr}\Big(\Pi_{\mathsf{stuck}}^{\mathsf{Jor}_r}\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)}\Big) = 0$ since $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)}$ was defined so that $\mathrm{Tr}\Big(\Pi_{p,\varepsilon,\delta}\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)}\Big) = 1$.

Additionally, by the guarantee of Theorem 6.4, $\mathrm{Tr}\Big(\Pi_{p,\varepsilon,\delta}\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(3e,r,1)}\Big) \geq 1-\delta$. Hence by Claim 7.2, there exists a state $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{G},r,1)}$ with the same Jordan spectrum as $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(3e,r,1)}$, $\mathrm{Tr}\Big(\Pi_{p,\varepsilon,\delta}\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{G},r,1)}\Big) = 1$ and $d(\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{G},r,1)}, \widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(3e,r,1)}) \leq \sqrt{\delta}$. Note that $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{G},r,1)}$ also has the same Jordan spectrum of $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(r,1)}$

Consider the pseudoinverse state $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{D},r,1)}$ under $(\mathsf{D}_r, \mathsf{G}_{p,\varepsilon,\delta})$ of $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{G},r,1)}$. Since the Jordan spectrum of $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{G},r,1)} \in \mathrm{image}(\Pi_{p,\varepsilon,\delta})$ is identical to the Jordan spectrum of $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(r,1)} \in \mathrm{image}(\Pi_r)$, and $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)}$ is the pseudoinverse under $(\mathsf{G}_{p,\varepsilon,\delta}, \mathsf{D}_r)$ of $\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(r,1)}$, it follows that

$$\mathrm{Tr}\Big(\Pi_{p,\varepsilon,\delta}\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{D},r,1)}\Big) = \mathrm{Tr}\Big(\Pi_r\widetilde{\boldsymbol{\sigma}}_{\mathcal{H},\mathcal{W}}^{(\mathsf{U},1)}\Big) = \widetilde{\zeta}_r,$$

and moreover $\widetilde{\boldsymbol{\sigma}}^{(\mathsf{G},r,1)}_{\mathcal{H},\mathcal{W}} = \Pi_{p,\varepsilon,\delta}\widetilde{\boldsymbol{\sigma}}^{(\mathsf{D},r,1)}_{\mathcal{H},\mathcal{W}}\Pi_{p,\varepsilon,\delta}/\widetilde{\zeta}_r$.

Hence, if the state before Step 3c is $\sum_r \frac{\widetilde{\zeta}_r}{\widetilde{\zeta}_R}\widetilde{\boldsymbol{\sigma}}^{(r,1)}_{\mathcal{H},\mathcal{W}}$ (which is $\frac{4\sqrt{\delta}}{p}$ close to the actual state before Step 3c) then the state after Step 3e is $O(\sqrt{\delta})$-close to the following state:

$$\widetilde{\boldsymbol{\sigma}}^{(3e,1)}_{\mathcal{H},\mathcal{W}} \coloneqq \frac{1}{\widetilde{\zeta}_R}\sum_r \Pi_{p,\varepsilon,\delta}\left(\widetilde{\boldsymbol{\sigma}}^{(\mathsf{D},r,1)}_{\mathcal{H},\mathcal{W}}\right)\Pi_{p,\varepsilon,\delta}.$$

Therefore, writing $\widetilde{\boldsymbol{\sigma}}^{(\mathsf{D},r,1)}_{\mathcal{H},\mathcal{W}} = \widetilde{\boldsymbol{\phi}}^{(\mathsf{D},r,1)}_{\mathcal{H}} \otimes |0\rangle\langle 0|_{\mathcal{W}}$ ($\widetilde{\boldsymbol{\sigma}}^{(\mathsf{D},r,1)}_{\mathcal{H},\mathcal{W}}$ has this form since $\widetilde{\boldsymbol{\sigma}}^{(\mathsf{D},r,1)}_{\mathcal{H},\mathcal{W}} \in \mathrm{image}(\Pi_r)$), the state at the end of Step 3f is $O(\sqrt{\delta})$-close to

$$\widetilde{\boldsymbol{\phi}}^{(3f,1)}_{\mathcal{H}} \coloneqq \frac{1}{\widetilde{\zeta}_R}\sum_{r\in R} M_{p,\varepsilon,\delta}\left(\widetilde{\boldsymbol{\phi}}^{(\mathsf{D},r,1)}_{\mathcal{H}}\right)M^{\dagger}_{p,\varepsilon,\delta},$$

where $M_{p,\varepsilon,\delta}$ is the measurement element of $\mathsf{T}_{p,\varepsilon,\delta}$ that corresponds to a 1 outcome.

By the guarantee of $\mathsf{Threshold}$ (Theorem 6.2), it holds that for all states $\boldsymbol{\phi} \in \mathbf{S}(\mathcal{H})$,

$$\mathrm{Tr}\left(\Pi^{\mathsf{Jor}}_{<p-\varepsilon} M_{p,\varepsilon,\delta}\boldsymbol{\phi}M^{\dagger}_{p,\varepsilon,\delta}\right) \leq \delta,$$

and so by linearity,

$$\begin{aligned}
\mathrm{Tr}\left(\Pi^{\mathsf{Jor}}_{<p-\varepsilon}\widetilde{\boldsymbol{\phi}}^{(3f,1)}_{\mathcal{H}}\right) &= \frac{1}{\widetilde{\zeta}_R}\sum_{r\in R}\mathrm{Tr}\left(\Pi^{\mathsf{Jor}}_{<p-\varepsilon}M_{p,\varepsilon,\delta}\left(\widetilde{\boldsymbol{\phi}}^{(\mathsf{D},r,1)}_{\mathcal{H}}\right)M^{\dagger}_{p,\varepsilon,\delta}\right)\\
&\leq \frac{1}{\widetilde{\zeta}_R}\cdot|R|\cdot\delta\\
&\leq \frac{\delta}{p-2\sqrt{\delta}} \qquad (4)\\
&= O(\delta/p),
\end{aligned}$$

where Eq. (4) holds by Claim 8.9. It follows from the guarantees of the fixed-runtime singular vector transform (Theorem 6.1) that the state at the end of Step 3g is $O(\delta)$-close to the state $\widetilde{\boldsymbol{\rho}}^{(3g,1)}_{\mathcal{H},\mathcal{R}} = \mathsf{Transform}_{p-\varepsilon}[\mathsf{U} \to \mathsf{C}](\widetilde{\boldsymbol{\phi}}^{(3f,1)}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|_{\mathcal{R}})$, which has the property that

$$\mathrm{Tr}\left((\mathbf{I}-\Pi_{\mathsf{C}})\Pi^{\mathsf{Jor}}_{\geq p-\varepsilon}(\widetilde{\boldsymbol{\phi}}^{(3f,1)}_{\mathcal{H}} \otimes |+_R\rangle\langle +_R|_{\mathcal{R}})\right) \leq \delta.$$

Combining this with $\mathrm{Tr}\left(\Pi^{\mathsf{Jor}}_{<p-\varepsilon}\widetilde{\boldsymbol{\phi}}^{(3f,1)}_{\mathcal{H}}\right) = O(\delta/p)$, we conclude that if the state before Step 3c is $\sum_r \frac{\widetilde{\zeta}_r}{\widetilde{\zeta}_R}\widetilde{\boldsymbol{\sigma}}^{(r,1)}_{\mathcal{H},\mathcal{W}}$, then the probability that Step 3g aborts is at most $O(\delta/p)$. Additionally, the guarantee of $\mathsf{Threshold}$ (Theorem 6.2) implies that in the next iteration of Step 3, the probability that Step 3a aborts on $\widetilde{\boldsymbol{\rho}}^{(3g,1)}_{\mathcal{H},\mathcal{R}}$ is also at most $O(\delta/p)$. By a trace distance argument, the probability that Step 3g or the subsequent Step 3a aborts in a *real* execution of Step 3 (with the modifications as in the statement of Claim 8.16) when the first Step 3a did not abort is at most $O(\sqrt{\delta}/p)$. This completes the proof of Claim 8.16. □

8.3.6 Correctness of Transcript Generation

Finally, we prove property (3) of Theorem 8.2.

Lemma 8.17. *For every $\tau_{\mathrm{pre}} = (\mathsf{vk}, a)$, let $\gamma = \gamma_{\mathsf{vk},a}$ denote the initial success probability of P^* conditioned on τ_{pre}. Then, if $\gamma > \delta^{1/3}$, the distribution D_k on $(r_1, \ldots, r_k)$ (conditioned on (vk, a) and a successful first execution) is $O(1/\gamma)$-admissible (Definition 5.5).*

This follows by appealing to the following claim in each round, making use of the fact that the expectation of $1/p$ conditioned on an accepting initial execution is equal[29] to $1/\gamma$; the $O(\sqrt{\delta})$-closeness from the claim also degrades to $O(\sqrt{\delta}/\gamma)$ when conditioning on an accepting initial execution.

Claim 8.18. *Consider the distribution D supported on $R \cup \{\bot\}$ obtained running a single iteration of Step 3 with parameter p on an arbitrary state $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H} \otimes \mathcal{R})$ with $\mathrm{Tr}(\Pi_{\mathsf{C}}\boldsymbol{\rho}) = 1$ (where $r := \bot$ if* Extract *aborts). There exists a procedure* Samp *that makes expected $O(1/p)$ queries to uniform sampling oracle O_R (but can otherwise behave arbitrarily and inefficiently) that outputs a distribution $O(\sqrt{\delta})$-close to D, and if the output of* Samp *is not $\bot$ then is one of the responses to its oracle queries.*

Proof. Samp initially behaves similarly to Extract: apply $\mathsf{Threshold}_{p,\varepsilon,\delta}$ to $\boldsymbol{\rho}$; if Threshold outputs 0 then output $\bot$. Let $\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$ be the state after applying Threshold, and (as in Extract) re-set $p := p - \varepsilon$.

As before, let $\boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}} := \frac{\left(\Pi^{\mathsf{Jor}}_{\geq p}\right)^{\mathcal{B}} \boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}} \left(\Pi^{\mathsf{Jor}}_{\geq p}\right)^{\mathcal{B}}}{\mathrm{Tr}\left(\left(\Pi^{\mathsf{Jor}}_{\geq p}\right)^{\mathcal{B}} \boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\right)}$. Let $\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}$ be the pseudoinverse of $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}} = \langle 1|_{\mathcal{B}}\, \boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}\, |1\rangle_{\mathcal{B}} \,/\, \mathrm{Tr}\left(\langle 1|_{\mathcal{B}}\, \boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{B},\mathcal{H},\mathcal{R}}\, |1\rangle_{\mathcal{B}}\right)$ as guaranteed by the pseudoinverse lemma (Lemma 7.1).

We have by Lemmas 3.1 and 7.1 that $d(\boldsymbol{\rho}^{\mathsf{C}}_{\mathcal{B},\mathcal{H},\mathcal{R}}, \boldsymbol{\rho}'^{\mathsf{C}}_{\mathcal{B},\mathcal{H},\mathcal{R}}) \leq 2\sqrt{\delta}$ and $\boldsymbol{\rho}'^{\mathsf{C},1}_{\mathcal{H},\mathcal{R}} = \frac{\Pi_C \boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}} \Pi_C}{\mathrm{Tr}\left(\Pi_{\mathsf{C}} \boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\right)}$. Finally, write $\boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}} = \boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |+\rangle\langle +|_{\mathcal{R}}$.

Samp now behaves differently than Extract. Samp "clones" $\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}$ (recall that Samp can be an arbitrary function) and repeats the following until $b = 1$: query O_R, obtaining $r \in R$; on a fresh copy of $\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}$, measure whether the verifier accepts on challenge r (i.e., $(\Pi_{V,r}, \mathbf{I} - \Pi_{V,r})$), obtaining bit b. Output r if $b = 1$.

Let $p_{\mathsf{U}} := \mathrm{Tr}\left(\Pi_{\mathsf{C}} \boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\right)$; we have that $p_{\mathsf{U}} \geq p$ by Claim 8.7. Hence, the expected number of queries Samp makes is $1/p$. Observe that p_{U} is the probability that a uniform r is accepted. Let $\zeta_r := \mathrm{Tr}\left(\Pi_{V,r} \boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\right)$; ζ_r is the probability that r is accepted. Then, for every $r^* \in R$,

$$\Pr_{\mathsf{Samp}}[r = r^*] = \sum_{n=0}^{\infty} \Pr[r_1, \ldots, r_n \text{ rejected}] \Pr[r_{n+1} = r^*, r^* \text{ accepted}] = \sum_{n=0}^{\infty} (1 - p_{\mathsf{U}})^n \cdot \frac{\zeta_{r^*}}{|R|} = \frac{\zeta_{r^*}}{p_{\mathsf{U}} \cdot |R|} \,.$$

Consider now the distribution on r obtained by measuring $\mathcal{R}$ on state $\boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}$: for every r^*,

$$\Pr[r = r^*] = \mathrm{Tr}\left(|r^*\rangle\langle r^*|\, \boldsymbol{\rho}'^{(\mathsf{C},1)}_{\mathcal{H},\mathcal{R}}\right) = \frac{\mathrm{Tr}\left(|r^*\rangle\langle r^*|\, \Pi_{\mathsf{C}} \boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}} \Pi_{\mathsf{C}}\right)}{p_{\mathsf{U}}} = \frac{\mathrm{Tr}\left(\Pi_{V,r^*} \boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}}\right)}{p_{\mathsf{U}} \cdot |R|} = \frac{\zeta_{r^*}}{p_{\mathsf{U}} \cdot |R|}$$

[29] Here (and elsewhere) we informally make use of the fact that the "current" value of p in any iteration of Step 3 is always at least $p_0/2$, where p_0 is the *initial* estimated p.

since $(\mathbf{I}_{\mathcal{H}} \otimes |r^*\rangle\langle r^*|)\Pi_{\mathsf{C}} = \Pi_{V,r^*} \otimes |r^*\rangle\langle r^*|$ and $|r^*\rangle\langle r^*|\, \boldsymbol{\rho}^{(\mathsf{U},1)}_{\mathcal{H},\mathcal{R}}\, |r^*\rangle\langle r^*| = \frac{1}{|R|}\boldsymbol{\phi}^{(\mathsf{U},1)}_{\mathcal{H}} \otimes |r^*\rangle\langle r^*|_{\mathcal{R}}$.

Overall, D is obtained by measuring $(\mathcal{B},\mathcal{R})$ on the state $\boldsymbol{\rho}^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$, which is $O(\sqrt{\delta})$-close to $\boldsymbol{\rho}'^{(\mathsf{C})}_{\mathcal{B},\mathcal{H},\mathcal{R}}$; the claim follows by contractivity of trace distance. □

Having established properties (1), (2), and (3), we have proved Theorem 8.2!

8.4 Obtaining Guaranteed Extraction

In this section, we combine the guarantees of Theorem 8.2 with additional analysis to prove that all of the example protocols from Section 5.3 have guaranteed extractors (additionally assuming partial collapsing where necessary). We remark that handling the graph isomorphism subroutine requires a slight modification of the Theorem 8.2, which we detail below.

We begin with a general-purpose corollary of Theorem 8.2 for the case of protocols satisfying (k,g)-PSS (Definition 5.6) in addition to $f_1,\ldots,f_{k-1}$-partial collapsing (which was assumed in Theorem 8.2).

Corollary 8.19. *Let (P_Σ, V_Σ) be a 3- or 4- message public coin interactive argument with a consistency function $g : T \times (R \times \{0,1\}^*)^* \to \{0,1\}$, and let $f_1,\ldots,f_k$ be functions. Suppose that:*

- *The protocol is partially collapsing with respect to $f_1,\ldots,f_{k-1}$, and*
- *The protocol is (k,g)-PSS for some $k = \mathrm{poly}(\lambda)$.*

Then, one of the two following conclusions holds:

1. *The extractor from Theorem 8.2 composed with the PSS extractor* PSSExtract *satisfies* guaranteed extraction*, OR*
2. *The extractor from Theorem 8.2 outputs a k-tuple of partial transcripts $(r_1,y_1,\ldots,r_k,y_k)$ such that $g(\tau_{\mathrm{pre}},r_1,y_1,\ldots,r_k,y_k) = 0$ (the transcripts are* inconsistent*) with non-negligible probability.*

Proof. Suppose that conclusion (1) is false, meaning that there exist infinitely many λ and a constant c such that the extractor from Theorem 8.2 has an accepting initial execution but the call to PSSExtract fails to produce a witness with probability at least $1/\lambda^c$. We know that the Theorem 8.2 extractor aborts with negligible probability, so we also assume that the extractor does not abort here. Then, by an averaging argument, with probability at least $\frac{1}{2\lambda^c}$ over the distribution of (vk,a), the above event conditioned on (vk,a) holds with probability at least $\frac{1}{2\lambda^c}$. This in particular implies that $\gamma_{\mathsf{vk},a}$ (as defined in Theorem 8.2) is at least $\frac{1}{2\lambda^c}$ for these choices of (vk,a). Then, property (3) of Theorem 8.2 implies that the distribution of $(r_1,\ldots,r_k)$ is *admissible* for these choices of (vk,a) (and choices of λ). Thus, the (k,g)-PSS property of (P_Σ,V_Σ) implies that for every such (vk,a), the k-tuple of partial transcripts must be inconsistent with probability at least $\frac{1}{2\lambda^c}$ (as otherwise PSSExtract would succeed with $1-\mathrm{negl}$ probability). Therefore, assuming that conclusion (1) is false, the probability that the k-tuple of transcripts output by the Theorem 8.2 extractor are inconsistent is at least $\frac{1}{4\lambda^{2c}}$ for infinitely many λ, implying conclusion (2). □

Finally, we apply Corollary 8.19 to obtain guaranteed extractors for all of the Section 5.3 example protocols (along with a general result for k-special sound protocols).

Corollary 8.20. *If (P_Σ, V_Σ) is (fully) collapsing and k-special sound, and $|R| = 2^{\omega(\log \lambda)}$, then the protocol has guaranteed extraction.*

Proof. Since (P_Σ, V_Σ) is k-special sound and $|R| = 2^{\omega(\log \lambda)}$, we know that the protocol is (k, g)-PSS for the "trivial" transcript consistency predicate g. Therefore, Corollary 8.19 applies to this protocol (where the extractor sets $f_1 = \ldots = f_k = \mathsf{Id}$). However, conclusion (2) of Corollary 8.19 cannot happen because the consistency predicate of $\mathsf{PSSExtract}$ in this case simply checks that the transcripts are accepting, which is guaranteed by the fact that $(r_i, y_i = z_i)$ was a measurement outcome of a state in Π_C. □

Corollary 8.21. *If (P_Σ, V_Σ) is a commit-and-open protocol (Definition 5.11) satisfying commit-and-open k-special soundness and $R = 2^{\omega(\log \lambda)}$ (either natively or enforced by parallel repetition), and the commitment scheme is instantiated using a collapse-binding commitment [Unr16b], then the protocol has a guaranteed extractor.*

Proof. Under the hypotheses of the corollary (along with Claims 5.3 and 5.7), the protocol satisfies either (k, g)-PSS (if it has a natively superpolynomial challenge space) or $(k^2 \log^2(\lambda), g)$-PSS (if parallel repeated; see Lemma 5.8), where g is a predicate that enforces the constraint that all opened messages are consistent with each other. We set $f_1 = \ldots = f_k = f$ where $f(z)$ outputs the substring of z corresponding to the opened messages (and not the openings). Then, the Theorem 8.2 extraction procedure does not violate g-consistency by the unique-message binding of the commitment scheme (shown in Lemma 4.2). Thus, Corollary 8.19 implies that (P_Σ, V_Σ) has a guaranteed extraction procedure. □

Corollary 8.22. *Kilian's succinct argument system [Kil92], when instantiated using a collapsing hash function and a PCP of knowledge, has a guaranteed extraction procedure.*

Proof. We know from Claim 5.14 the [Kil92] succinct argument system is (1) (fully) collapsing, and (2) (k, g)-PSS for $k = \mathrm{poly}(n, \lambda)$ and g defined so that when z_i and z_j contain overlapping leaves of the Merkle tree, the leaf values are equal. We set $f_1 = \ldots = f_k = \mathsf{Id}$, and observe that the Theorem 8.2 extractor does not violate g-consistency, because if it output two transcripts $(r_1, z_1), (r_2, z_2)$ with inconsistent leaf values, since the transcripts are accepting (they were obtained by measuring a state in Π_C), this would violate the collision-resistance (implied by collapsing) of the hash family. Thus, by Corollary 8.19, the protocol has a guaranteed extractor. □

Corollary 8.23. *The one-out-of-two graph isomorphism subroutine has a guaranteed extraction procedure that extracts the bit b (when G_0 and G_1 are not isomorphic).*

Proof. By Claim 5.10, this protocol is $(2, g')$-PSS where g' is the following asymmetric function:

- For the first partial transcript $(\tau_{\mathrm{pre}}, r^{(1)}, c^{(1)})$, g' checks that for all i such that $r_i = 0$, $(H_{0,i}, H_{1,i})$ are isomorphic to $(G_{c_i^{(1)}}, G_{1-c_i^{(1)}})$.
- For the second partial transcript $(\tau_{\mathrm{pre}}, r^{(2)}, c^{(2)})$, g' *additionally* checks that for all i such that $r_i = 1$, $H_{c_i^{(2)},i}$ is isomorphic to H.

We define the following pair of functions f_1, f_2:

- $f_1(\tau_{\text{pre}}, r, z)$ outputs the following substring of z. For every i such that $r_i = 0$, the substring includes the bit c_i (where $z_i = (c_i, \sigma_{0,i}, \sigma_{1,i})$).
- $f_2(\tau_{\text{pre}}, r, z)$ outputs the substring c (the distinguished single bit of each z_i).

We note that the graph isomorphism subprotocol is f_1-collapsing; this follows from the fact that for any *accepting* transcript $(\tau_{\text{pre}}, r, z)$, the bits c_i (for $r_i = 0$) are information-theoretically determined as a function of $(G_0, G_1, H_{0,i}, H_{1,i})$.

Thus, if we instantiate the Theorem 8.2 extractor using (f_1, f_2) (note that we require no properties of f_2) we have that Corollary 8.19 applies. Moreover, g'-consistency of the transcripts output by the extractor is not violated, because it is formally implied by the fact that they were obtained by partially measuring a state in Π_C (any accepting partial transcript (r_i, c_i) satisfies the condition checked by g'). Thus, we conclude that the protocol has a guaranteed extractor by Corollary 8.19. □

9 Expected Polynomial Time for Quantum Simulators

We introduce a notion of efficient computation we call coherent-runtime expected quantum polynomial time (EQPT_c). We then formalize a new definition of post-quantum zero-knowledge with EQPT_c simulation.

9.1 Quantum Turing Machines

We recall the definition of a quantum Turing machine (QTM) of Deutsch [Deu85]. A QTM is a tuple $(\Sigma, Q, \delta, q_0, q_f)$ where Σ is a finite set of symbols, Q is a finite set of states, $\delta \colon Q \times \Sigma \to \mathbb{C}^{Q \times \Sigma \times \{-1,0,1\}}$ is a transition function, and q_0, q_f are the initial and final (halting) states respectively.

We fix registers $\mathcal{Q}$ containing the state, $\mathcal{I}$ containing the position of the tape head, and $\mathcal{T}$ containing the tape. A configuration state of a Turing machine is a vector $|q, i, \mathbb{T}\rangle \in \mathcal{Q} \otimes \mathcal{I} \otimes \mathcal{T}$ where $q \in Q$ is the current state, $i \in \mathbb{N}$ is the location of the tape head, and $\mathbb{T} \in \Sigma^*$ is the (finite) contents of the tape.

A transition is given by the map U_δ, which acts on basis states as follows:

$$|q, i, T\rangle \mapsto \sum_{q' \in Q} \sum_{a \in \Sigma} \sum_{d \in \{-1,0,1\}} \alpha_{q',a,d,b} \, |q', i + d, \mathbb{T}_{i \to a}\rangle$$

where $\delta(q, \mathbb{T}_i) = \sum_{q',a,d} \alpha_{q',a,d} \, |q', a, d\rangle$. δ is a valid transition function if and only if U_δ is unitary. The definition of QTMs generalises to multiple tapes in the natural way. We will consider QTMs having a separate input/output tape on register $\mathcal{A}$ (with head position in $\mathcal{I}_{\mathsf{in}}$).

The execution of a T-bounded QTM proceeds as follows.

1. Initialize register $\mathcal{Q}$ to $|q_0\rangle$, $\mathcal{I}, \mathcal{I}_{\mathsf{in}}$ to $|0\rangle$, and $\mathcal{T}$ to the empty tape state $|\varnothing\rangle$.
2. Repeat the following for at most T steps:
   (a) Apply the measurement $\Pi_f = (\, |q_f\rangle\langle q_f| \, , \mathbf{I} - |q_f\rangle\langle q_f|)$ to $\mathcal{Q}$. If the outcome is 1, halt and discard all registers except $\mathcal{A}$.
   (b) Apply U_δ.

The **output** $M(\boldsymbol{\rho})$ of a QTM M on input $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{A})$ is the state on $\mathcal{A}$ when the machine halts. The **running time** $t_M(\boldsymbol{\rho})$ of M on input $\boldsymbol{\rho}$ is the number of iterations of Step 2. Note that both of these quantities are random variables.

Definition 9.1. The expected running time $E_M(n)$ of a QTM M is the maximum over all n-qubit states $\boldsymbol{\rho}$ of $\mathbb{E}[t_M(\boldsymbol{\rho})]$. A T-bounded QTM M (for some $T \leq \exp(n)$) is EQPT_m and there exists a polynomial p such that $E_M(n) \leq p(n)$ for all n.

9.2 Coherent-Runtime EQPT

Definition 9.2. A D-circuit is a quantum circuit C with special gates $\{G_i, G_i^{-1}\})_{i=1}^k$ with the following restriction: for each i, there is a single G_i gate and a single G_i^{-1} gate acting on a designated register $\mathcal{X}_i$, where G_i acts before G_i^{-1}. All other gates may act arbitrarily on $\mathcal{Y} \otimes \bigotimes_{i=1}^k \mathcal{X}_i$, for some register $\mathcal{Y}$. For any CPTP maps $\Phi_i \colon \mathbf{S}(\mathcal{X}_i) \to \mathbf{S}(\mathcal{X}_i)$, $C[\Phi_1, \ldots, \Phi_k] \colon \mathbf{S}(\mathcal{Y} \otimes \bigotimes_{i=1}^k \mathcal{X}_i) \to \mathbf{S}(\mathcal{Y} \otimes \bigotimes_{i=1}^k \mathcal{X}_i)$ is the superoperator defined as follows:

1. For each i, U_i be a unitary dilation of Φ_i. That is, let $\mathcal{Z}_i$ be an ancilla Hilbert space and U_Φ unitary on $\mathcal{X}_i \otimes \mathcal{Z}_i$ such that $\Phi(\boldsymbol{\sigma}) = \mathrm{Tr}_{\mathcal{Z}_i}(U_i(\boldsymbol{\sigma} \otimes |0\rangle\langle 0|_{\mathcal{Z}_i})U_i^\dagger)$ for all $\boldsymbol{\sigma} \in \mathbf{S}(\mathcal{X}_i)$.
2. Construct a circuit C' on $\mathcal{Y} \otimes \bigotimes_{i=1}^k (\mathcal{X}_i \otimes \mathcal{Z}_i)$ from C by replacing G_i with U_i and G_i^{-1} with $U_i^\dagger$ for each i.
3. Let C be the superoperator $\boldsymbol{\rho} \mapsto \mathrm{Tr}_{\mathcal{Z}}(C'(\boldsymbol{\rho} \otimes \bigotimes_{i=1}^k |0\rangle\langle 0|_{\mathcal{Z}_i}))$.

Since all choices of U_i are equivalent up to a local isometry on $\mathcal{Z}_i$, the map $C[\Phi_1, \ldots, \Phi_k]$ is well-defined.

We are now ready to define our notion of *coherent-runtime expected quantum polynomial time.*

Definition 9.3. A sequence of CPTP maps $\{\Phi_n\}_{n \in \mathbb{N}}$ is a EQPT_c **computation** if there exist a uniform family of D-circuits $\{C_n\}_{n \in \mathbb{N}}$ and EQPT_m computations $M_1, \ldots, M_k$ such that $C_n[M_1, \ldots, M_k] = \Phi_n$ for all n.

We show that any EQPT_c computation can be approximated to any desired precision by a polynomial-size quantum circuit. We first show the following claim. Let $|\mathsf{init}\rangle := |q_0\rangle_{\mathcal{Q}} \, |0,0\rangle_{\mathcal{I},\mathcal{I}_{\mathsf{in}}} \, |\varnothing\rangle_{\mathcal{T}}$.

Claim 9.4. *Let M be a T-bounded QTM running in expected time t, and let U be the unitary dilation of M as in Fig. 2. For all $\gamma \colon \mathbb{N} \to (0,1]$, there is a uniform sequence of unitary circuits $\{V_n\}_n$ of size $\mathrm{poly}(n)/\gamma(n)^2$ such that for every unitary A on $\mathcal{A}$ and state $|\psi\rangle \in \mathcal{A}$:*

$$\left\| (U^\dagger(\mathbf{I} \otimes A)U - V_n^\dagger(\mathbf{I} \otimes A)V_n) \, |\psi\rangle \, |\mathsf{init}\rangle \, |0^T\rangle_{\mathcal{B}} \right\| \leq \gamma(n).$$

Proof. Let V be the unitary given by truncating U to just after the τ-th iteration of U_δ, where $\tau := \lceil t/4\gamma^2 \rceil$. Let $\Pi := |1^{T-\tau}\rangle\langle 1^{T-\tau}|_{\mathcal{B}_{\tau+1},\ldots,\mathcal{B}_T}$. Observe that for every state $|\psi\rangle$,

$$\Pi U_i \, |\psi\rangle \, |\mathsf{init}\rangle \, |0^T\rangle_{\mathcal{B}} = \Pi_f V_i \, |\psi\rangle \, |\mathsf{init}\rangle \, |0^\tau 1^{T-\tau}\rangle_{\mathcal{B}} \,,$$

because Π projects on to computations that finish in at most τ steps, and once the computation finishes, the remaining CNOT_{Π_f} gates flip the corresponding $\mathcal{B}_i$ from 0 to 1.

Moreover, for every state $|\phi\rangle \in \mathcal{A} \otimes \mathcal{Q} \otimes \mathcal{W}$ and $z \in \{0,1\}^\tau$,

$$U_i^\dagger \, |\phi\rangle \, |z 1^{T-\tau}\rangle = V_i^\dagger \, |\phi\rangle \, |z 0^{T-\tau}\rangle \,,$$

since the applications of U_δ controlled on $\mathcal{B}_{\tau+1}, \ldots, \mathcal{B}_T$ act as the identity and the CNOT_{Π_f} gates acting on $\mathcal{B}_{\tau+1}, \ldots, \mathcal{B}_T$ flip the corresponding register from 1 to 0. Hence

$$U^\dagger (I \otimes A) \Pi U \left|\psi\right\rangle \left|\mathsf{init}\right\rangle \left|0^T\right\rangle_{\mathcal{B}} = V^\dagger (I \otimes A) \Pi_f V \left|\psi\right\rangle \left|\mathsf{init}\right\rangle \left|0^T\right\rangle_{\mathcal{B}} .$$

The claim follows since, by Markov's inequality,

$$\left\| \Pi U_i \left|\psi\right\rangle \left|\mathsf{init}\right\rangle \left|0^T\right\rangle_{\mathcal{B}} \right\| \leq \sqrt{t/\tau} \leq \varepsilon(n)/2. \qquad \square$$

Lemma 9.5. *For any* EQPT_c *computation* $\{\Phi_n\}_n$ *and* $\varepsilon \colon \mathbb{N} \to (0,1]$*, there is a uniform sequence of (standard) quantum circuits* $\{C_n\}_n$ *of size* $\mathrm{poly}(n, 1/\varepsilon(n))$ *such that* $d(\Phi_n(\boldsymbol{\rho}), C_n(\boldsymbol{\rho})) \leq \varepsilon(n)$ *for all* $\boldsymbol{\rho}$.

Proof. Let D_n be a D-circuit and $M_1, \ldots, M_k$ such that $\Phi_n = D_n[M_1, \ldots, M_k]$, and let U_n be the unitary circuit obtained by replacing each G_i, G_i^{-1} with the corresponding coherent implementation of M_i as in Fig. 2. Let U_n' be as U_n, but where the G_i-gates are replaced with unitaries V_i as guaranteed by Claim 9.4, with $\gamma(n) \coloneqq \varepsilon(n)/k$. The circuit C_n is obtained by initializing the ancillas to $\left|\mathsf{init}\right\rangle \left|0^\tau\right\rangle_{\mathcal{B}}$, applying U_n', and then tracing out the ancillas.

We make use of the fidelity distance d_F, defined in [Wat06] to be

$$d_F(\boldsymbol{\rho}, \boldsymbol{\sigma}) \coloneqq \inf\{\| \left|\psi\right\rangle - \left|\phi\right\rangle \| : \left|\psi\right\rangle, \left|\phi\right\rangle \text{ purify } \boldsymbol{\rho}, \boldsymbol{\sigma}, \text{ respectively}\}.$$

[Wat06] shows that $d_F(\boldsymbol{\rho}, \boldsymbol{\sigma}) \geq d(\boldsymbol{\rho}, \boldsymbol{\sigma})$. We can choose the purifications $U_n \left|\psi\right\rangle \left|\mathsf{init}\right\rangle \left|0^T\right\rangle$ of Φ_n and $U_n' \left|\psi\right\rangle \left|0\right\rangle$ of C_n. By Claim 9.4, and the triangle inequality, the distance between these states is at most $\varepsilon(n)$. □

9.3 Zero Knowledge with EQPT_c Simulation

Given our definition of EQPT_c above, we now formally define zero-knowledge with EQPT_c simulation for interactive protocols.

For an interactive protocol (P, V), let $\mathsf{out}_{V^*}\langle P, V^* \rangle$ denote the output of V^* after interacting with P.

Definition 9.6. An interactive argument is *black-box* statistical (resp. computational) post-quantum zero knowledge if there exists an EQPT_c simulator Sim such that for all polynomial-size quantum malicious verifiers V^* and all $(x, w) \in R_L$, the distributions

$$\mathsf{out}_{V^*}\langle P(x,w), V^* \rangle \quad \text{and} \quad \mathsf{Sim}^{V^*}(x)$$

are statistically (resp. quantum computationally) indistinguishable.

10 State-Preserving Extraction

So far, we have constructed EQPT_m *guaranteed extractors* for various protocols of interest (Section 8) and established the EQPT_c model that allows for *state-preserving* extraction (Section 9). In this section, we prove a generalization of Lemma 2.3, showing how to convert a EQPT_m guaranteed extractor into a state-preserving EQPT_c extractor.

In Section 10.1, we write down an explicit reduction from state-preserving extraction to guaranteed extraction and prove Lemma 10.3, which gives a condition (Definition 10.2) under which the reduction is valid (intuitively capturing "computational uniqueness" of the witness given the first message of the protocol). Then, in Section 10.2, we show examples to which Lemma 10.3 applies; namely, protocols for languages with unique (partial) witnesses and general commit-and-prove protocols. Finally, in Section 10.3, we conclude Theorems 1.6 and 1.7.

10.1 From Guaranteed Extraction to State-Preserving Extraction

We first recall our definition of state-preserving proofs of knowledge (Definition 1.5).

Definition 10.1. An interactive protocol Π is defined to be a $\mathsf{state\text{-}preserving\ argument}$ (resp. proof) of $\mathsf{knowledge}$ if there exists an extractor $\mathsf{Ext}^{(\cdot)}$ with the following properties:

- **Syntax**: For any quantum algorithm P^* and auxiliary state $|\psi\rangle$, $\mathsf{Ext}^{P^*,|\psi\rangle}$ outputs a protocol transcript τ, prover state $|\psi'\rangle$, and witness w.
- **Extraction Efficiency**: If P^* is a QPT algorithm, $E^{P^*,|\psi\rangle}$ runs in expected quantum polynomial time (EQPT_c).
- **Extraction Correctness**: the probability that τ is an accepting transcript but w is an invalid NP witness is negligible.
- **State-Preserving**: the pair $(\tau, |\psi'\rangle)$ is computationally (resp. statistically) indistinguishable from a transcript-state pair $(\tau^*, |\psi^*\rangle)$ obtained through an honest one-time interaction with $P^*(\cdot, |\psi\rangle)$ (where $|\psi^*\rangle$ is the prover's residual state).

We now introduce the notion of "witness-binding" protocols, i.e., protocols that are collapse-binding to functions of the witness w. For an adversary Adv and an interactive protocols (P, V) we define a witness-binding experiment $\mathsf{Exp}_{\mathsf{wb}}^{\mathsf{Adv}}(b, \mathsf{Pred}, f, \lambda)$ parameterized by a challenge bit b, a predicate Pred and a function f.

1. The challenger generates the first verifier message vk and sends it to Adv; skip this step if the protocol is a 3-message protocol.
2. Adv replies with a classical instance x, classical first prover message a, and a quantum state on registers $\mathcal{W}_{\text{witness}} \otimes \mathcal{Y}_{\text{aux}}$.
3. The challenger performs a binary-outcome projective measurement to learn the output of $\mathsf{Pred}(x, \mathsf{vk}, a, \cdot, \cdot)$ on $\mathcal{W}_{\text{witness}} \otimes \mathcal{Y}_{\text{aux}}$. If the output is 0, the experiment aborts.
4. If $b = 0$, the challenger does nothing. If $b = 1$, the challenger initializes a fresh ancilla $\mathcal{K}$ to $|0\rangle_{\mathcal{K}}$, applies the unitary U_f (acting on $\mathcal{W}_{\text{witness}} \otimes \mathcal{K}$) that computes $f(\cdot)$ on $\mathcal{W}_{\text{witness}}$ and XORs the output onto $\mathcal{K}$, measures $\mathcal{K}$, and then applies $U_f^\dagger$.
5. The challenger returns the $\mathcal{W}_{\text{witness}} \otimes \mathcal{Y}_{\text{aux}}$ registers to Adv. Finally, Adv outputs a bit b', which is the output of the experiment (if the experiment has not aborted).

Definition 10.2 ((Pred, f)-binding to the witness)**.** A 3 or 4-message protocol is witness binding with respect to predicate Pred and function f if for any computationally bounded quantum adversary Adv,

$$\left| \Pr\left[\mathsf{Exp}_{\mathsf{wb}}^{\mathsf{Adv}}(0, \mathsf{Pred}, f, \lambda) = 1\right] - \Pr\left[\mathsf{Exp}_{\mathsf{wb}}^{\mathsf{Adv}}(1, \mathsf{Pred}, f, \lambda) = 1\right] \right| \leq \operatorname{negl}(\lambda).$$

Next, we write down a general-purpose reduction from state-preserving extraction to guaranteed extraction and show (Lemma 10.3) that the reduction is valid under an appropriate witness-binding assumption.

Lemma 10.3. *Suppose that (P_Σ, V_Σ) is a post-quantum proof/argument of knowledge with guaranteed extraction. We optionally assume that the extractor $\mathsf{Extract}^{P^*}$ outputs some auxiliary information y in addition to the witness w. We then make the following additional assumptions with respect to a predicate Pred:*

- *The protocol (P_Σ, V_Σ) is $(\mathsf{Pred}, f = \mathsf{Id})$-witness binding, and*
- *The tuple (w, y) output by the guaranteed extractor $\mathsf{Extract}^{P^*}$ satisfies $\mathsf{Pred}(\mathsf{vk}, x, a, w, y) = 1$ with $1 - \operatorname{negl}$ probability.*

Then, (P_Σ, V_Σ) is a state-preserving proof/argument of knowledge with EQPT_c extraction.

Remark 10.4. *This lemma is stated with respect to $f = \mathsf{Id}$ to match the state-preserving proof of knowledge abstraction; however, we also consider (Corollary 10.7) versions of this reduction where $f \neq \mathsf{Id}$.*

Proof. We want to show that (P_Σ, V_Σ) is a state-preserving proof/argument of knowledge. We begin by describing our candidate state-preserving extractor $\overline{\mathsf{Extract}}^{P^*}$.

Construction 10.5. Let $\mathsf{Extract}^{P^*}$ be a post-quantum guaranteed extractor (Definition 8.1). We present an EQPT_c extractor $\overline{\mathsf{Extract}}^{P^*}$ that has the form of an EQPT_c computation (see Fig. 1) where the unitary U is a coherent implementation of the following EQPT_m computation on input register $\mathcal{H} \otimes \mathcal{R} \otimes \mathcal{S}$:

1. Measure $\mathcal{R} \otimes \mathcal{S}$ with the projective measurement

   $$\left(|+_R\rangle\langle +_R|_{\mathcal{R}} \otimes |0\rangle\langle 0|_{\mathcal{S}}, \mathbf{I} - |+_R\rangle\langle +_R|_{\mathcal{R}} \otimes |0\rangle\langle 0|_{\mathcal{S}} \right).$$

   If the output is 0, abort.

2. If the output is 1, we are guaranteed that $\mathcal{R} \otimes \mathcal{S}$ is $|+_R\rangle_{\mathcal{R}} \otimes |0\rangle_{\mathcal{S}}$. Run $\mathsf{Extract}^{P^*}$ on prover state $\mathcal{H}$ using $\mathcal{R}$ as the superposition of challenges (in Step 2 of Definition 8.1). We assume that the randomness $\mathsf{Extract}^{P^*}$ uses to sample a classical random vk is generated by applying a Hadamard to a subregister of $\mathcal{S}$.

   Write everything that is measured/obtained during the execution of $\mathsf{Extract}^{P^*}$ onto subregisters of $\mathcal{S}$. This includes the instance x, the first two messages of the 4-message protocol (vk, a), the bit b indicating the verifier's decision (i.e., whether the prover succeeds when run on the uniform superposition of challenges), and the extracted output (w, y) (if $b = 1$, w is a valid witness for x and $\mathsf{Pred}(x, \mathsf{vk}, a, w, y) = 1$ with $1 - \operatorname{negl}(\lambda)$ probability).

The fact that the above computation is in EQPT_m follows from the fact that $\mathsf{Extract}^{P^*}$ is EQPT_m. Let U denote its coherent implementation (as in Section 9, U is a unitary on $\mathcal{H} \otimes \mathcal{R} \otimes \mathcal{S}$ and an exponential-size ancilla register).

Our state-preserving EQPT_c extractor $\overline{\mathsf{Extract}}^{P^*}$ takes as input a prover state on $\mathcal{H}$ and does the following.

$\overline{\mathsf{Extract}}^{P^*}$:

1. Initialize additional registers $\mathcal{R} \otimes \mathcal{S}$ to $|+_R\rangle_{\mathcal{R}} \, |0\rangle_{\mathcal{S}}$.
2. Apply U.
3. Measure the subregister of $\mathcal{S}$ containing $(x, \mathsf{vk}, a, b, w)$ where $w = 0$ is interpreted as $\bot$. Note that $\mathcal{S}$ contains a subregister corresponding to y, but y is not measured here.
4. Apply $U^\dagger$.
5. Run the prover P^* on first message vk to obtain x, a (again). Then run P^* on challenge $\mathcal{R}$. Measure $\mathcal{R}$ to obtain r, and measure the register of $\mathcal{H}$ corresponding to its output to obtain z. Output $(x, \mathsf{vk}, a, r, z, w)$ and $\mathcal{H}$.

First, we note that the above procedure is EQPT_c by construction. To prove the extraction correctness guarantee, it suffices to show that when $b = 1$, the witness w is valid with $1 - \mathrm{negl}(\lambda)$ probability, and that when $b = 0$, the extractor outputs a rejecting transcript. The former statement follows immediately from the assumption that $\mathsf{Extract}^{P^*}$ is a guaranteed extractor. For the latter, observe (using the definition of $\mathsf{Extract}^{P^*}$ and the fact that U is a coherent implementation of $\mathsf{Extract}^{P^*}$) that when $b = 0$, the state on $\mathcal{H} \otimes \mathcal{R}$ after running P^* to obtain a in Step 5 corresponds to a rejecting execution, so the transcript measured in Step 5 will be rejecting.

It remains to argue that the state-preserving extractor satisfies the indistinguishability property. Observe that $\overline{\mathsf{Extract}}^{P^*}$ can be rewritten so that vk, x, a, b are no longer obtained by running $\mathsf{Extract}^{P^*}$ coherently as U and then measuring those values afterwards, but instead by running those steps accroding to the standard EQPT_m implementation of $\mathsf{Extract}^{P^*}$. Thus the only part of $\mathsf{Extract}^{P^*}$ that is written as a coherent implementation of a variable runtime procedure is the $\mathsf{FindWitness}^{P^*}$ subroutine; let U_{FW} denote the coherent implementation of $\mathsf{FindWitness}^{P^*}$. Note that while $\mathsf{FindWitness}^{P^*}$ is technically not EQPT_m on its own (i.e., there exist inputs that could make it run for too long), the fact that $\mathsf{Extract}^{P^*}$ is EQPT_m ensures that U_{FW} is only applied on inputs where it runs for expected polynomial time.

Given the above definitions, the output of $\overline{\mathsf{Extract}}^{P^*}$ is perfectly equivalent to the following:

1. Sample a random vk, and run the prover P^* to obtain x, a.
2. Initialize $\mathcal{R}$ to $|+_R\rangle_{\mathcal{R}}$ and measure C (this is the binary projective measurement on $\mathcal{H} \otimes \mathcal{R}$ defined in Section 8.0.1 that measures whether the verifier accepts when the prover with state $\mathcal{H}$ is run on the challenge $\mathcal{R}$).
3. If $\mathsf{C} = 1$, apply U_{FW}. Otherwise if $\mathsf{C} = 0$, set $w = \bot$ and skip to Step 6.
4. Measure the subregister corresponding to the part of the output of U_{FW} containing w. Note that there is also a subregister corresponding to y, but y is not measured.
5. Apply $U_{\mathsf{FW}}^\dagger$.
6. Measure $\mathcal{R}$ to obtain r and run the prover P^* on r to obtain its response z.
7. Output $(x, \mathsf{vk}, a, r, z, w)$ and $\mathcal{H}$.

Let Hybrid_0 be identical to $\overline{\mathsf{Extract}}^{P^*}$ except that Step 7 is modified to output $(x, \mathsf{vk}, a, r, z)$ and $\mathcal{H}$ (i.e., omitting w). To show computational indistinguishability, it suffices to show that the output of Hybrid_0 is computational indistinguishable from Hybrid_1 defined as follows:

1. Sample a random vk, and run the prover P^* to obtain x, a.
2. Initialize $\mathcal{R}$ to $|+_R\rangle_{\mathcal{R}}$ and measure C (this is the binary projective measurement on $\mathcal{H} \otimes \mathcal{R}$ defined in Section 8.0.1 that measures whether the verifier accepts when the prover with state $\mathcal{H}$ is run on the challenge $\mathcal{R}$).
3. Measure $\mathcal{R}$ to obtain r and run the prover P^* on r to obtain its response z.
4. Output $(x, \mathsf{vk}, a, r, z)$ and $\mathcal{H}$.

Hybrid_1 corresponds to an honest execution of P^* since the measurement of C commutes with the measurement of $\mathcal{R}$.

By assumption, in Hybrid_0, the reduced density $\boldsymbol{\rho}_{\mathcal{S}}$ of $\mathcal{S}$ satisfies $\mathrm{Tr}(\Pi_{\mathrm{Valid}}\boldsymbol{\rho}_{\mathcal{S}}) = 1 - \mathrm{negl}(\lambda)$, where Π_{Valid} checks that either $b = 0$ or (1) w is a valid witness for x and (2) $\mathsf{Pred}(\mathsf{vk}, x, a, w, y) = 1$. Therefore, the indistinguishability of Hybrid_0 and Hybrid_1 should intuitively follow from the witness-binding property, since if the measurement of w is skipped, then U_{FW} cancels out with $U_{\mathsf{FW}}^\dagger$. However, to appeal to the guarantee that measuring w is undetectable, we need to ensure that U_{FW} corresponds to an efficient operation.

We handle this by considering a fixed polynomial-time truncation of U_{FW}. Suppose that a distinguisher can distinguish Hybrid_0 from Hybrid_1 with non-negligible advantage $\varepsilon(\lambda)$. Then we can modify Hybrid_0 to use $U_{\mathsf{FW},\varepsilon}$, a coherent implementation of a strict $\mathrm{poly}(\lambda, 1/\varepsilon)$-runtime algorithm that approximates $\mathsf{FindWitness}^{P^*}$ to precision $\varepsilon/2$. Now the same distinguisher must distinguish between $\mathsf{Hybrid}_{0,\varepsilon}$ and Hybrid_1 with advantage $\varepsilon/2$, where $\mathsf{Hybrid}_{0,\varepsilon}$ is the following:

1. Sample a random vk, and run the prover P^* to obtain x, a.
2. Initialize $\mathcal{R}$ to $|+_R\rangle_{\mathcal{R}}$ and measure C.
3. If $\mathsf{C} = 1$, apply $U_{\mathsf{FW},\varepsilon}$. Otherwise if $\mathsf{C} = 0$, set $w = \perp$ and skip to Step 6.
4. Measure a subregister of the output register of $U_{\mathsf{FW},\varepsilon}$ to obtain w.
5. Apply $U_{\mathsf{FW},\varepsilon}^\dagger$.
6. Measure $\mathcal{R}$ to obtain r and run the prover P^* on r to obtain its response z.
7. Output $(x, \mathsf{vk}, a, r, z)$ and $\mathcal{H}$.

Since $\varepsilon(\lambda)$ is at least $1/\lambda^c$ for some constant c for infinitely many λ, it follows that $U_{\mathsf{FW},\varepsilon}$ and $U_{\mathsf{FW},\varepsilon}^\dagger$ are $\mathrm{poly}(\lambda)$-runtime algorithms for infinitely many λ. Then a distinguisher that distinguishes between $\mathsf{Hybrid}_{0,\varepsilon}$ and Hybrid_1 contradicts the witness-binding property of (P, V). □

10.2 Applying Lemma 10.3

We now show that the witness-binding hypotheses in Lemma 10.3 are satisfied in two cases of interest: protocols for unique-witness (or partial witness) languages (Corollary 10.6), and commit-and-prove protocols (Corollary 10.8).

Corollary 10.6. *Let $L \in \mathsf{UP}$ be a language with unique NP witnesses. Then, if L has a post-quantum proof of knowledge with guaranteed extraction, it also has a post-quantum state-preserving proof of knowledge.*

Proof. This follows immediately from the fact that any protocol for a UP language is (Pred, f)-witness binding for $\mathsf{Pred} = 1$ (the trivial predicate) and $f = \mathsf{Id}$ (because there is a unique valid witness). Since $\mathsf{Pred} = 1$, any guaranteed extractor also satisfies the Pred-hypothesis of Lemma 10.3, so we are done. □

We briefly state how Corollary 10.6 can be extended to languages L with unique *partial* witnesses, provided that the extractor only measures a function $f(w)$ that is a deterministic function of the instance x.

Corollary 10.7. *Let $L \in \mathsf{NP}$, and let f be an efficient function such that for all instances $x \in L$ and all witnesses $w \in R_x$, $f(x,w) = g(x)$ is equal to some fixed (possibly inefficient) function of x.*

Suppose that L has a proof/argument of knowledge (P_Σ, V_Σ) with guaranteed extraction. Then, a modified variant of $\overline{\mathsf{Extract}}$ (Construction 10.5), in which only $f(x,w)$ is measured instead of w, is a state-preserving proof/argument of knowledge extractor for (P_Σ, V_Σ) that outputs $g(x)$.

This holds by the same reasoning as Corollary 10.6: the hypothesis of Corollary 10.7 implies that any protocol for L is $(\mathsf{Pred} = 1, f)$-witness binding, and so the reduction from Lemma 10.3 applies (when $f(x,w)$ is measured rather than w).

10.2.1 Commit-and-Prove Protocols

Let (P_Σ, V_Σ) denote a post-quantum proof/argument of knowledge with guaranteed extraction (Definition 8.1). Recall that Definition 8.1 has been designed to capture (first-message) adaptive soundness, in which the prover P^* can adaptively choose the instance x as it sends its first message.

Then, we consider a *commit-and-prove* compiled protocol $(P_{\mathsf{Com}}, V_{\mathsf{Com}})$ using (P_Σ, V_Σ) and a commitment scheme Com. $(P_{\mathsf{Com}}, V_{\mathsf{Com}})$ is executed as follows:

- V_{Com} sends a first message for (P_Σ, V_Σ) (if the protocol has four messages). Moreover, if Com is a two-message commitment scheme, V_{Com} sends a commitment key ck.
- P_{Com} then sends:
  - A commitment $\mathsf{com} = \mathsf{Com}(\mathsf{ck}, w)$ to a witness w for the underlying language L, and
  - A first prover message for an execution of (P_Σ, V_Σ) for the statement "$\exists w, r$ such that $\mathsf{com} = \mathsf{Com}(\mathsf{ck}, w; r)$ and w is an NP-witness for $x \in L$.
- P_{Com} and V_{Com} then complete the execution of (P_Σ, V_Σ).

Corollary 10.8. *If (P_Σ, V_Σ) is a post-quantum proof/argument of knowledge with guaranteed extraction for all NP languages and Com is a collapse-binding commitment scheme, then the commit-and-prove compiled protocol is a state-preserving proof/argument of knowledge.*

Proof. We first remark that since (P_Σ, V_Σ) is a post-quantum proof/argument of knowledge with guaranteed extraction, the commit-and-prove composed protocol is also immediately a post-quantum proof/argument of knowledge with guaranteed extraction. Namely, $\mathsf{Extract}^{P^*}$ interprets the cheating prover as an adaptive-input cheating prover for (P_Σ, V_Σ) with respect to the language

$$L_{\mathsf{ck},\mathsf{com}} = \{(w, \omega) : w \in R_x \text{ and } \mathsf{Com}(\mathsf{ck}, w; \omega) = \mathsf{com}\}$$

and runs the guaranteed extractor for (P_Σ, V_Σ). Morevoer, this extraction procedure outputs *both* an NP-witness w *and* commitment randomness ω such that $\mathsf{com} = \mathsf{Com}(\mathsf{ck}, w; \omega)$; we treat ω as auxiliary information y.

We then define $\mathsf{Pred}(x, (\mathsf{ck}, \mathsf{vk}), (\mathsf{com}, a), w, \omega)$ to output 1 if and only if $\mathsf{Com}(\mathsf{ck}, w; \omega) = \mathsf{com}$. Then, we observe that the commit-and-prove protocol is $(\mathsf{Pred}, \mathsf{Id})$-witness binding (for the language

L) by the collapse-binding of the commitment scheme Com. Moreover, the correctness property of $\mathsf{Extract}^{P^*}$ further guarantees that $\mathsf{Pred}(x, (\mathsf{ck}, \mathsf{vk}), (\mathsf{com}, a), w, \omega) = 1$ with probability $1 - \mathrm{negl}(\lambda)$.

Thus, we conclude that Lemma 10.3 applies, and so the commit-and-prove protocol has a state-preserving extractor. □

10.3 Concluding Theorems 1.6 and 1.7

Finally, we describe how to conclude the results of Theorems 1.6 and 1.7. We begin with Theorem 1.6, re-stated below.

Theorem 10.9 (Theorem 1.6)**.** *Assuming collapsing hash functions exist, there exists a 4-message public-coin state-preserving succinct argument of knowledge for* NP*.*

Proof. Given a collapsing hash function family H, we construct a state-preserving succinct argument of knowledge for NP as follows:

- First, we define Kilian's succinct argument system (see Section 5.3.3) with respect to H. By Corollary 8.22, this argument system is a post-quantum argument of knowledge with guaranteed extraction.
- Next, we apply the commit-and-prove compiler (Corollary 10.8) using the collapse-binding commitment scheme $\mathsf{Com}(\mathsf{ck} = h, m) = h(m)$. This commitment scheme does not formally satisfy any hiding property, but it is *succinct*, which is what is relevant for Theorem 1.6.

Corollary 10.8 tells us that the resulting composed protocol is a state-preserving argument of knowledge for NP. Moreover, it satisfies all of the properties (4-message, public-coin, succinct) claimed in the theorem statement. □

Next, we prove Theorem 1.7, re-stated below.

Theorem 10.10. *Assuming collapsing hash functions or* super-polynomially secure *one-way functions, there exists a 4-message public-coin state-preserving witness-indistinguishable argument (in the case of collapsing) or proof (in the case of OWFs) of knowledge. Assuming* super-polynomially secure *non-interactive commitments, there exists a 3-message PoK achieving the same properties.*

Proof. All three variants of this theorem are proved via the same approach: combining commit-and-prove with a (strong) witness-indistinguishable Σ-protocol.

Formally, let Com denote a (possibly keyed) non-interactive commitment scheme. We use Com to instantiate a commit-and-open Σ-protocol (Definition 5.11) such as the [GMW87] protocol for graph 3-coloring or the (potentially modified) [Blu86] protocol for Hamiltonicity. We do a sufficient parallel repetition of the commit-and-open protocol so that its challenge space satisfies $|R| = 2^t$ for $t \leq \mathrm{poly}(\lambda)$[30] and it achieves $\mathrm{negl}(\lambda)$ soundness error. Then, Corollary 8.21 tells us that this protocol is a post-quantum proof/argument of knowledge (depending on whether Com is statistically or collapse-binding) with guaranteed extraction.

Next, we additionally assume (as is the case for [GMW87, Blu86]) that the Σ-protocol satisfies special honest-verifier zero knowledge (Definition 3.6). In fact, we assume that it satisfies SHVZK

[30] Using [Blu86], one can set $t = \mathrm{poly}(\log \lambda)$.

against quantum adversaries that run in time $2^t \cdot \mathrm{poly}(\lambda)$, which holds (for these examples) provided that Com is computationally hiding against $2^t \cdot \mathrm{poly}(\lambda)$-time adversaries.

Under this assumption, Watrous' rewinding lemma [Wat06] implies that the Σ-protocol has a time $2^t \cdot \mathrm{poly}(\lambda)$ malicious verifier post-quantum simulator.

We now plug this Σ-protocol into the commit-and-prove compiler (Corollary 10.8), again making use of the commitment scheme Com (for simplicity of the proof, we assume here that a different commitment key is used, although this is not necessary). Corollary 10.8 tells us that the resulting protocol is a state-preserving proof/argument of knowledge (again depending on whether Com is statistically binding).

It remains to show WI of the commit-and-prove protocol. That is, we want to show that for every malicious verifier V^* (and maliciously chosen commitment key ck), a commitment $\mathsf{com} = \mathsf{Com}(\mathsf{ck}, w_1)$ and the view of V^* in an execution of the Σ-protocol is computationally indistinguishable from the analogous state when a second witness w_2 is instead used. This is argued via the usual hybrid argument:

- Define $\mathsf{Hybrid}_{0,b}$ to be $\mathsf{Com}(\mathsf{ck}, w_b)$ along with the actual Σ-protocol view of V^*.
- Define $\mathsf{Hybrid}_{1,b}$ to consist of $\mathsf{com} = \mathsf{Com}(\mathsf{ck}, w_b)$ along with a $2^t \cdot \mathrm{poly}(\lambda)$-time *simulated* view of V^* on input $(\mathsf{ck}, \mathsf{com})$. We have that $\mathsf{Hybrid}_{1,b} \approx_c \mathsf{Hybrid}_{0,b}$ by the super-polynomial time simulatability of the Σ-protocol (as discussed above).
- Finally, we have that $\mathsf{Hybrid}_{1,0} \approx_c \mathsf{Hybrid}_{1,1}$ by the (already assumed) $2^t \cdot \mathrm{poly}(\lambda)$-hiding of Com.

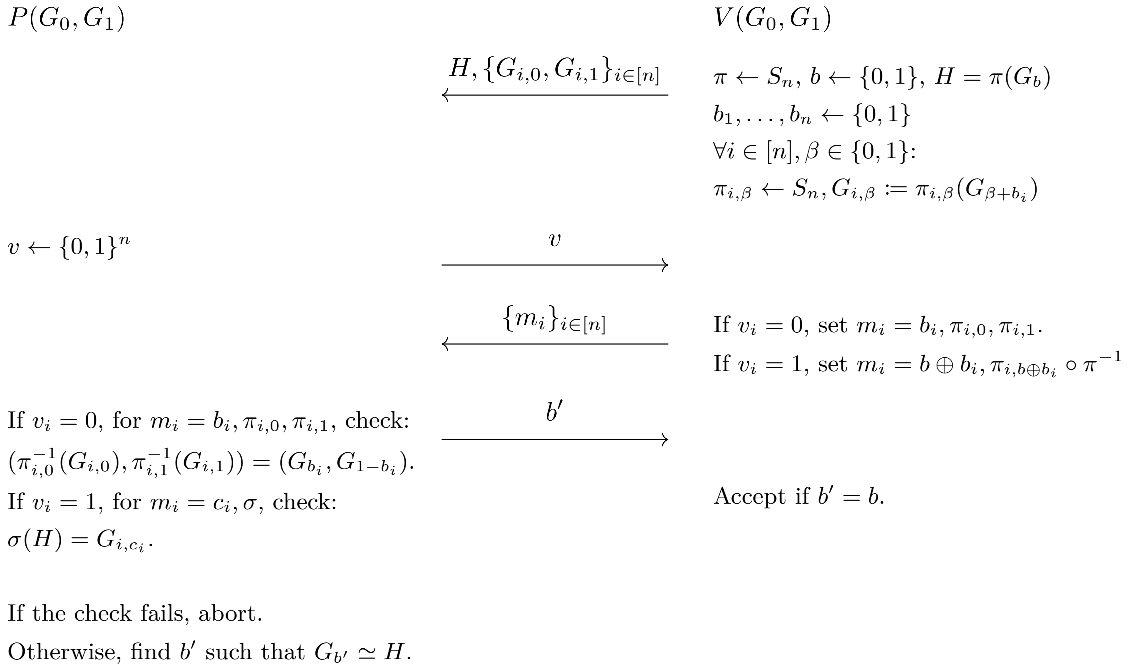
To conclude the theorem statement, it suffices to instantiate Com in three ways:

- Assuming $2^t \cdot \mathrm{poly}(\lambda)$-secure non-interactive commitments (e.g. [BOV03, GHKW17, LS19]), one obtains the claimed 3-message protocol.
- Assuming $2^t \cdot \mathrm{poly}(\lambda)$-secure one-way functions, one obtains the OWF-based 4-message protocol.
- Assuming polynomially-secure collapsing hash functions, one obtains the collapsing-based 4-message protocol by defining $\mathsf{Com}(h, m; r, s) = (h(r), s, \langle r, s\rangle \oplus m)$. This commitment scheme is *statistically* hiding (i.e. hiding against unbounded adversaries), and so WI of the commit-and-prove protocol holds unconditionally, while the AoK property relies on collapsing.

This completes the proof of Theorem 1.7. □

11 The [GMW86] GNI Protocol is Post-Quantum Zero Knowledge

In this section, we show that our state-preserving extraction results imply the post-quantum ZK of the graph non-isomorphism protocol, proving Theorem 1.2. We begin by giving a description of the GNI protocol in Fig. 8. Our description achieves soundness error 1/2 (as does the original [GMW86]), but can be extended to the negligible soundness case (without increasing the number of rounds) with essentially the same proof of (post-quantum) ZK.



$P(G_0, G_1)$

$V(G_0, G_1)$

$\xleftarrow{H, \{G_{i,0}, G_{i,1}\}_{i \in [n]}}$

$\pi \leftarrow S_n$, $b \leftarrow \{0,1\}$, $H = \pi(G_b)$
$b_1, \ldots, b_n \leftarrow \{0,1\}$
$\forall i \in [n], \beta \in \{0,1\}$:
$\pi_{i,\beta} \leftarrow S_n, G_{i,\beta} := \pi_{i,\beta}(G_{\beta + b_i})$

$v \leftarrow \{0,1\}^n$ $\xrightarrow{v}$

$\xleftarrow{\{m_i\}_{i \in [n]}}$

If $v_i = 0$, set $m_i = b_i, \pi_{i,0}, \pi_{i,1}$.
If $v_i = 1$, set $m_i = b \oplus b_i, \pi_{i,b \oplus b_i} \circ \pi^{-1}$

If $v_i = 0$, for $m_i = b_i, \pi_{i,0}, \pi_{i,1}$, check:
$(\pi_{i,0}^{-1}(G_{i,0}), \pi_{i,1}^{-1}(G_{i,1})) = (G_{b_i}, G_{1-b_i})$.
If $v_i = 1$, for $m_i = c_i, \sigma$, check:
$\sigma(H) = G_{i,c_i}$.

$\xrightarrow{b'}$

Accept if $b' = b$.

If the check fails, abort.
Otherwise, find b' such that $G_{b'} \simeq H$.

Figure 8: The Zero Knowledge Proof System for Graph Non-Isomorphism.

Next, we give a slightly more abstract description of the protocol using instance-dependent commitments [BMO90, IOS97, MV03].

Construction 11.1. Fix a language L, let IDC be a non-interactive instance-dependent commitment[31] [BMO90, IOS97, MV03] for L, and let PoK be a statistically witness-indistinguishable proof of knowledge of the committed bit for IDC. Then, we define the following interactive proof system for the complement language $\overline{L}$.

1. The verifier commits to a bit $b \in \{0,1\}$ using IDC and sends it to the prover.

2. The prover and verifier engage in PoK where the verifier proves knowledge of b.

3. If the prover accepts in PoK, then it sends b' as determined by the verifier's commitment.

4. The verifier accepts if $b' = b$.

[GMW86] instantiates this framework for the language L consisting of pairs of isomorphic graphs (and so $\overline{L}$ consists of pairs of non-isomorphic graphs, up to well-formedness of the string x).

Let GIComm be the following instance-dependent commitment scheme: $\mathsf{GIComm}((G_0, G_1), b; \pi) = \pi(G_b) := H$. Observe that if G_0, G_1 are isomorphic then this commitment is perfectly hiding, and if they are not then it is perfectly binding. Moreover, this commitment scheme admits a proof of knowledge of the committed bit as follows.

1. The prover chooses $b_1, \ldots, b_\lambda \in \{0,1\}$ uniformly at random and sends commitments $C_{i,0} := \mathsf{GIComm}((G_0, G_1), b_i; \sigma_{i,0})$ and $C_{i,1} := \mathsf{GIComm}((G_0, G_1), b_i \oplus 1; \sigma_{i,1})$.

[31] That is, when $x \in L$, a commitment $\mathsf{Com}(x, m)$ statistically hides the message m. When $x \notin L$, a commitment $\mathsf{Com}(x, m)$ statistically binds the committer to m.

2. The verifier sends a random string $v \in \{0,1\}^\lambda$.

3. The prover sends b_i and opens $C_{i,0}, C_{i,1}$ for all i such that $v_i = 0$. The prover sends $c_i := b \oplus b_i$, $\tau_i := \sigma_{i,c_i} \circ \pi^{-1}$ for all i such that $v_i = 1$.

4. The verifier accepts if the received openings are valid when $v_i = 0$, and $C_{i,c_i} = \tau_i(H)$ when $v_i = 1$, where H is the commitment graph.

Classically, we obtain b by rewinding to find two accepting transcripts with $v_i \neq v_i'$; then $b = c_i \oplus b_i$.

Lemma 11.2. *If (G_0, G_1) are non-isomorphic then the above protocol is a statistically state-preserving proof of knowledge of the committed message for* GIComm.

Proof. We have already shown (Corollary 8.23) that this protocol has a guaranteed extractor, because when G_0 and G_1 are not isomorphic, this protocol is collapsing onto the b_i part of 0-challenge responses (as b_i is fixed by the commitments $C_{i,0}, C_{i,1}$) and the protocol is $(2, g)$-probabilistically special sound (where g checks (for the first challenge-partial response pair) the correctness of the 0 challenge response bits b_i for $v_i = 0$ and (for the second challenge-partial response pair) the correctness of all b_i $(v_i = 0)$ and c_i $(v_i = 1)$).

Moreover, the language $L_{G_0,G_1} = \{H : \exists(b, \pi) \text{ such that } \pi G_b \simeq H\}$ has partial unique witnesses: for any $H \in L_{G_0,G_1}$, the bit b is uniquely determined (given that G_0 and G_1 are not isomorphic). Thus, the state-preserving reduction of Lemma 10.3 applies (see Corollary 10.7), so this protocol has a state-preserving extractor. □

Finally, we note that Lemma 11.2 immediately implies that the GNI protocol is post-quantum (statistical) zero knowledge. We assume without loss of generality that the cheating verifier V^* has a "classical" first message by replacing V^* (with auxiliary state $|\psi\rangle$) with $(V^*, \boldsymbol{\rho})$ for the mixed state $\boldsymbol{\rho}$ obtained by running U_{V^*} on $|\psi\rangle$ to generate a first message, measuring it, and running $U_{V^*}^\dagger$.

The simulator is then described as follows:

- Given cheating verifier V^* with classical first message $(\mathsf{com}, \mathsf{pok}_1)$, run the state-preserving PoK extractor on V^* (which now acts as a PoK cheating prover).

- If the transcript generated by the state-preserving extractor is accepting, then output the bit b in the "partial witness" slot of the extractor's output. Otherwise, send an aborting message.

The (statistical) zero knowledge property of this simulator follows immediately from the state-preserving property of the extractor. Moreover, the simulator inherits the EQPT_c structure directly from the extractor (with additional fixed polynomial-time pre- and post-processing). This completes the proof of Theorem 1.2.

12 The [FS90] Protocol is Post-Quantum Zero Knowledge

We recall the Feige-Shamir 4-message zero knowledge argument system for NP. This protocol uses three primitives as building blocks:

- A non-interactive commitment scheme Com.
- The 3-message WI argument of knowledge AoK constructed in Section 10.3. We note that AoK is public-coin.
- A 3-message delayed-witness WI argument of knowledge dAoK.

We will argue security using the particular instantiations of $\mathsf{AoK}, \mathsf{dAoK}$ due to subtleties arising from the concurrent composition. Unlike AoK, we do *not* require that dAoK is state-preserving. The protocol is executed as follows.

- The verifier sends the following strings as its first message:
  - Two commitments c_0, c_1 generated as $c_i = \mathsf{Com}(0; r_i)$ for i.i.d. random strings r_i. For the post-quantum variant, following [Unr12, Unr16b], we additionally include commitments $c'_i = \mathsf{Com}(r_i; \rho_i)$ to the two random strings r_0, r_1.
  - A first (prover) message of AoK corresponding to the statement "$\exists i, r_i, \rho_i$ such that $c_i = \mathsf{Com}(0; r_i)$ and $c'_i = \mathsf{Com}(r_i; \rho_i)$." By default, the verifier uses (b, r_b, ρ_b) as its witness for a randomly chosen bit b.
- The prover sends two strings as its first message:
  - A second (verifier) message of AoK (which is a uniformly random string).
  - A first (prover) message of dAoK corresponding to the statement "$x \in L$ or $\exists i, r_i, \rho_i$ such that $c_i = \mathsf{Com}(0; r_i)$ and $c'_i = \mathsf{Com}(r_i; \rho_i)$." No witness is required.
- The verifier sends two strings as its second message:
  - A third (prover) message of AoK, computed using (b, r_b, ρ_b).
  - A second (verifier) message of dAoK (which is a uniformly random string).
- Finally, the prover sends the third message of dAoK. The prover uses a witness w for $x \in L$ to generate this message.

12.1 Building Block: Delayed-Witness Proofs of Knowledge

In order to instantiate the Feige-Shamir protocol, we need a post-quantum instantiation of dAoK. In particular, we need:

Lemma 12.1. *Assume that post-quantum non-interactive commitments exist. Then, there exists a delayed-witness Σ-protocol for* NP *that is witness indistinguishable against quantum verifiers and is a post-quantum proof of knowledge with negligible knowledge error.*

Lemma 12.1 does not immediately follow from extraction techniques such as [Unr12, Lemma 7] or [CMSZ21] because the canonical delayed-witness Σ-protocol [LS91] is not collapsing, and these works only give results for collapsing protocols. Nonetheless, we show that (similar to the one-out-of-two graph isomorphism subprotocol of [GMW86]) making use of a variant $(2, g)$-PSS (Definition 5.6), a simple modification of Unruh's rewinding technique [Unr12] suffices to prove Lemma 12.1.

12.1.1 The [LS91] Protocol

We begin by recalling the [LS91] Σ-protocol for graph Hamiltonicity. The protocol uses a non-interactive commitment scheme Com as a building block, and is executed as follows.

- The prover, given as input the security parameter 1^λ and an input length 1^n,[32] sends λ commitments com_i to adjacency matrices of i.i.d. random cycle graphs on n vertices (i.e., graphs $H_i = \sigma_i C_n$ that are random permutations of a fixed cycle graph on n vertices).
- The verifier sends a uniformly random string $r \leftarrow \{0,1\}^\lambda$.
- For the third round, the prover is given a graph G and a fixed n-cycle represented by a permutation π mapping C_n to G. The prover then sends the following messages.
  - For each i such that $r_i = 0$, the prover sends a full opening of the ith commitment com_i.
  - For each i such that $r_i = 1$, the prover sends $\sigma_i \pi^{-1}$ and opens the substring of com_i consisting of commitments to each non-edge of $\sigma_i \pi^{-1}(G)$.
- For each i such that $r_i = 0$, the verifier checks that com_i was correctly opened to the adjacency matrix of a cycle graph. For each i such that $r_i = 1$, the verifier checks that every matrix entry opened is a valid decommitment to 0.

By the perfect binding of Com, we know that this protocol satisfies 2-special soundness. In fact, it is the parallel repetition of a protocol satisfying 2-special soundness: for any index i, a commitment string a_i along with a valid response z_0 to $r_i = 0$ and a valid response z_1 to $r_i = 1$ can be used to compute a Hamiltonian cycle in G. Indeed, it satisfies a variant special soundness (implicitly related to $(2, g')$-PSS) described here:

Claim 12.2. *There exists an extractor* $\mathsf{SSExtract}(a, r_1, z_{1,i}^{(1)}, r_2, z_{2,i})$ *for the [LS91] protocol such that* $\mathsf{SSExtract}$ *outputs a valid* NP *witness under the following conditions:*

- $r_{1,i} = 0, r_{2,i} = 1$.
- $(a_i, r_{2,i}, z_{2,i})$ *is an accepting transcript.*
- *There* exists *a response* $z_{1,i}$ *xwith prefix* $z_{1,i}^{(1)}$ *such that* $(a_i, r_{1,i}, z_{1,i})$ *is an accepting transcript.*

Here, $z^{(1)}$ *denotes the part of a response* z *consisting of the* messages *opened (but not the commitment randomness).*

Moreover, we note that the protocol is partially collapsing on 0-challenges: given a tuple (x, a, r) and a state $|\phi\rangle = \sum_z \alpha_z |z\rangle$, any accepting response z_i such that $r_i = 0$ can be *partially* measured — namely, the committed bits (but not the openings) can be measured — without disturbing $|\phi\rangle$. This is sufficient to prove Lemma 12.1.

[32] Note that the prover does not even need to know the instance x to compute this message; however, we consider an a priori fixed statement x to make sense of the proof-of-knowledge property.

12.1.2 Proof of Lemma 12.1

The fact that this protocol is witness indistinguishable follows from the fact that it is a parallel repetition of a post-quantum ZK protocol [Wat06]. What remains is to establish the proof-of-knowledge property.

We consider the following variant of Unruh's approach to knowledge extraction [Unr12]:

1. Given a cheating prover P^*, first generate a (classical) first message a from P^*. Let $|\psi\rangle$ denote the internal state of P^* at this point.
2. Sample a uniformly random challenge r, compute the P^* unitary $U_r |\psi\rangle$, which writes its response onto some register $\mathcal{Z}$. Apply the one-bit measurement $(\Pi_{V,r}, \mathbf{I} - \Pi_{V,r})$ that checks whether $V(x, a, r, z) = 1$.
3. If the measurement returns 1, additionally measure every register $\mathcal{Z}_i^{(1)}$ (the opened messages, but not the commitment randomness) corresponding to $r_i = 0$.
4. Apply $U_r^\dagger$ to the prover state.
5. Sample an independent random challenge r' and apply $U_{r'}$. Apply the one-bit measurement $(\Pi_{V,r'}, \mathbf{I} - \Pi_{V,r'})$.
6. If the measurement returns 1, additionally measure the *entire* response $\mathcal{Z}$.
7. If both measurements returned 1, and there exists an index i such that $r_i = 0$ and $r'_i = 1$, compute $\mathsf{SSExtract}(x, \mathsf{com}_i, 0, z_i^{(1)}, 1, z'_i)$ where $z_i^{(1)}$ is the first partially measured response in location i and z'_i is the second measured response in location i. Otherwise, abort.

To show that this extraction procedure works, we first consider the variant in which no response measurements are applied (Step 3 and Step 6 are omitted). Then, by Unruh's rewinding lemma [Unr12, Lemma 7], if $U_r |\psi\rangle$ produces an accepting response with probability at least ε (over the randomness of r), then the two binary measurements applied above will *both* return 1 with probability at least ε^3. Then, by the fact that the protocol is partially collapsing on 0-challenges, this continues to hold even if the measurement in Step 3 is applied.

Finally, since the probability that i.i.d. uniform strings r, r' do not have an index i such that $r_i = 0$ and $r'_i = 1$ is $(3/4)^\lambda = \mathrm{negl}(\lambda)$, we conclude that with probability $\varepsilon^3 - \mathrm{negl}(\lambda)$, the above extractor produces partial accepting response $z_i^{(1)}$ and accepting response z'_i for some i such that $r_i = 0$ and $r'_i = 1$, and so $\mathsf{SSExtract}$ successfully outputs a witness. If P^* is convincing with initial non-negligible probability γ, then with probability at least $\frac{\gamma}{2}$, $|\psi\rangle$ is at least $\frac{\gamma}{2}$-convincing, and so $\mathsf{SSExtract}$ outputs a valid witness with probability at least $\Omega(\gamma^3)$. This completes the proof of Lemma 12.1.

12.2 Proof of Security for the [FS90] protocol

We now prove the security of the Feige-Shamir protocol using suitable building blocks ($\mathsf{Com}, \mathsf{AoK}, \mathsf{dAoK}$).

Theorem 12.3. *Suppose that:*

- Com *is a post-quantum non-interactive commitment scheme,*

- AoK *is the* 3*-message state-preserving WI proof of knowledge for* NP *(with* EQPT_c *extraction) from Section 10.3.*

- dAoK *is the argument system from Lemma 12.1.*

Then, the Feige-Shamir protocol is both sound and zero-knowledge against QPT adversaries. The zero-knowledge simulator is EQPT_c*.*

Combining Theorem 12.3 with the results of Section 10 implies Theorem 1.3.

We remark that the theorem is non-generic with respect to $\mathsf{AoK}, \mathsf{dAoK}$ due to complications in the security proof coming from the fact that AoK and dAoK are executed simultaneously.

Proof. We first prove soundness, followed by ZK.

Proof of Soundness. Suppose that $x \notin L$ and P^* is a QPT prover that convinces V with non-negligible probability. Given such a P^*, we define a cheating prover P^*_{dAoK} for the underlying dAoK that is given as additional auxiliary input strings $(c_0, c'_0, c_1, c'_1, b, r_b, \rho_b)$ such that $c_b = \mathsf{Com}(0; r_b)$ and $c'_b = \mathsf{Com}(r_b; \rho_b)$. P^*_{dAoK} simply emulates P^* while generating AoK messages using its auxiliary input. That is:

- P^*_{dAoK} generates a message aok_1 using its auxiliary input and calls P^* on $(c_0, c'_0, c_1, c'_1, \mathsf{aok}_1)$. This results in a P^*-message $(\mathsf{aok}_2, \mathsf{daok}_1)$. P^*_{dAoK} returns daok_1.

- Upon receiving a verifier challenge r, P^*_{dAoK} computes an honestly generated message aok_3 (deterministic[33] and independent of r) using its auxiliary input and calls P^* on (aok_3, r). This results in a P^*-message daok_3, which P^*_{dAoK} outputs.

If the auxiliary input $(c_0, c'_0, c_1, c'_1, b, r_b, \rho_b)$ is sampled from the correct distribution, P^*_{dAoK} perfectly emulates the interaction of P^* and the honest Feige-Shamir verifier, so P^*_{dAoK} is convincing with non-negligible probability ε by assumption. Thus, the dAoK knowledge extractor from Lemma 12.1 outputs a valid witness for the statement "$\exists i, r_i, \rho_i$" such that $c_i = \mathsf{Com}(0; r_i)$ and $c'_i = \mathsf{Com}(r_i; \rho_i)$" with probability at least $\Omega(\varepsilon^3)$.

Claim 12.4. *The probability that the* dAoK *extractor succeeds and* $i \neq b$ *is also* $\Omega(\varepsilon^3)$*.*

Proof. If this is not the case, then we obtain an algorithm breaking the WI property of AoK. For a fixed statement (c_0, c'_0, c_1, c'_1), the algorithm V^*_{dAoK}, given an honestly generated message aok_1, calls $(\mathsf{aok}_2, \mathsf{dAoK}_1) \leftarrow P^*(c_0, c'_0, c_1, c'_1, \mathsf{aok}_1)$ and returns the message aok_2. Given a *fixed* response aok_3, V^*_{dAoK} emulates the dAoK extractor from Lemma 12.1 by sampling i.i.d. strings r, r' for dAoK and re-using the message aok_3. Then, if the extractor returns a valid witness (i, r_i, ρ_i), V^*_{dAoK} returns the bit i. If not, V^*_{dAoK} guesses at random.

Since this faithfully emulates the execution of the dAoK extractor on P^*_{dAoK} and we assumed that it succeeds with probability $\Omega(\varepsilon^3)$, we conclude that the WI property of AoK with respect to V^*_{dAoK} implies the claim. □

[33] If randomness is required to generate this message, let it be fixed in advance in P^*_{dAoK}'s internal state.

- Construct a first message daok_1 using the honest dAoK prover algorithm.
- For fixed classical strings $(c_0, c_0', c_1, c_1', \mathsf{aok}_1, \mathsf{daok}_1)$, define an AoK cheating prover P^*_{AoK} with the following description:
  - Send aok_1
  - On challenge s, call V^* on (s, daok_1). Upon receiving (aok_3, r), return aok_3.
- Run the state-preserving extractor $\mathsf{Extract}^{P^*_{\mathsf{AoK}}, \mathsf{daok}_1, \boldsymbol{\rho}}$, outputting the (unique possible) witness td along with a P^*_{AoK}-view (which includes a V^*-view in it).
- If the output witness is $\perp$, send an aborting final message. Otherwise, compute daok_3 using td.

Figure 9: The Feige-Shamir protocol simulator

However, this implies that the dAoK extractor breaks the computational hiding property of Com. This is because if c_{1-b} were instead sampled as $\mathsf{Com}(1; r_{1-b})$ and c'_{1-b} sampled as $\mathsf{Com}(r_{1-b}; \rho_{1-b})$, it is information theoretically impossible for the dAoK extractor to output a witness such that $i \neq b$. This concludes the proof of soundness.

Proof of ZK. We assume without loss of generality that the cheating verifier V^* has a "classical" first message $(c_0, c_0', c_1, c_1', \mathsf{aok}_1)$ by replacing V^* (with auxiliary state $|\psi\rangle$) with $(V^*, \boldsymbol{\rho})$ for the mixed state $\boldsymbol{\rho}$ obtained by running U_{V^*} on $|\psi\rangle$ to generate a first message, measuring it, and running $U^\dagger_{V^*}$.

By the construction of AoK (see Corollary 10.8 and Theorem 10.10) we know that the tuple $(c_0, c_0', c_1, c_1', \mathsf{aok}_1)$ *uniquely* determines a witness $\mathsf{td} = (b, r_b, \rho_b)$ that the AoK extractor can ever output (if such a witness exists; otherwise, we define td to be $\perp$). We non-uniformly include td in the description of the V^* state $\boldsymbol{\rho}$ without loss of generality (this does not affect the simulator, only the analysis).

Our black-box zero-knowledge simulator is defined in Fig. 9:

We claim that this achieves negligible simulation accuracy. We prove this via a hybrid argument:

- Hyb_0: This is the simulated view of V^*.
- Hyb_1: This is the same as Hyb_0, *except* that daok_3 is computed using an NP-witness w for x.
- Hyb_2: This is the real view of V^*.

The indistinguishability of Hyb_2 and Hyb_1 follows immediately from the state-preserving property of AoK, as the view of P^*_{AoK} contains an entire correctly emulated view of V^*.

The indistinguishability of Hyb_1 and Hyb_0 follows from the witness indistinguishability of dAoK. To prove this, we assume for the sake of contradiction that Hyb_1 and Hyb_0 are distinguishable by a polynomial-time distinguisher D with non-negligible advantage ε. Then, we construct the following two additional hybrids:

- Hyb'_0: This is simulated view of V^*, *except* that $\mathsf{Extract}$ is replaced by a $\mathrm{poly}(\lambda, 1/\varepsilon)$-size oracle algorithm that achieves accuracy $\frac{\varepsilon}{4}$.
- Hyb'_1: This is the same as Hyb'_0 *except* that daok_3 is computed using an NP-witness w for x.

By a hybrid argument, we conclude that D also distinguishes Hyb'_0 and Hyb'_1 with advantage $\varepsilon/2$. We claim that this breaks the witness indistinguishability of dAoK. Define a dAoK verifier V^*_{dAoK} operating as follows

- V^*_{dAoK} has the state $\boldsymbol{\rho}$ as auxiliary input (including $c_0, c'_0, c_1, c'_1, \mathsf{aok}_1, \mathsf{td}$). V^*_{dAoK} wants to distinguish between proofs using witness w and proofs using witness td.
- V^*_{dAoK} receives daok_1 from the prover. It then calls (the $\varepsilon/4$-truncated) $\mathsf{Extract}^{P^*_{\mathsf{AoK}}, \mathsf{daok}_1, \boldsymbol{\rho}}$, which returns a P^*_{AoK}-view. V^*_{dAoK} sends the challenge r from the P^*_{AoK}-view to the prover.
- Finally, upon receiving daok_3 from the prover, V^*_{dAoK} outputs the emulated V^* view.

V^*_{dAoK} has been constructed to be (aux-input) QPT, and (along with the distinguisher D) violates the WI property of dAoK, giving the claimed contradiction.

We conclude that the Feige-Shamir protocol is ZK, as desired. We note that the zero-knowledge simulator inherits the EQPT_c structure of the AoK state-preserving extractor (with some additional fixed poly-time pre- and post-processing). □

13 The [GK96] Protocol is Post-Quantum Zero Knowledge

In this section we show that the Goldreich–Kahan constant-round proof system for NP is post-quantum zero knowledge by giving an EQPT_c simulator. In Section 13.1 we give a technical lemma about the distinguishability of certain purifications that will be of central importance in the proof. In Section 13.2 we describe our quantum simulator.

13.1 Indistinguishability of Projections onto Indistinguishable States

Consider the states $|\tau_b\rangle := \sum_x |x\rangle_{\mathcal{X}} |D_b(x)\rangle_{\mathcal{Y}}$ where D_0, D_1 are computationally indistinguishable (w.r.t. quantum adversaries) efficiently sampleable *classical* distributions with random coins x (in a slight abuse of notation, D_b denotes both the distribution and the sample). If we are only given access to $\mathcal{Y}$, then distinguishing $|\tau_0\rangle$ from $|\tau_1\rangle$ is clearly hard since $\mathrm{Tr}_{\mathcal{X}}(|\tau_b\rangle\langle\tau_b|)$ is equivalent to a random classical sample from D_b.

In this subsection, we show that this indistinguishability generically extends to the setting where *we additionally give the distinguisher access to the projection* $|D_b\rangle\langle D_b|$ *on* $\mathcal{X} \otimes \mathcal{Y}$. This is formalized by giving the distinguisher an additional one-qubit register $\mathcal{O}$ and black-box access (see Section 3.2) to the unitary U_b and its inverse acting on $\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{O}$ defined as

$$U_b := |D_b\rangle\langle D_b|_{\mathcal{X},\mathcal{Y}} \otimes \mathbf{X}_{\mathcal{B}} + (\mathbf{I}_{\mathcal{X},\mathcal{Y}} - |D_b\rangle\langle D_b|_{\mathcal{X},\mathcal{Y}}) \otimes \mathbf{I}_{\mathcal{B}},$$

where $\mathbf{X}_{\mathcal{B}}$ denotes the bit-flip operator on $\mathcal{B}$. In particular, it is no longer the case that access to $|\tau_b\rangle$ is equivalent to a random classical sample from D_b, since the distinguisher's access to U_b means the $\mathcal{X}$ is no longer independent of its view. Nevertheless, we prove the following.

Lemma 13.1. *If there exists a polynomial-time quantum oracle distinguisher S^{U_b} without direct access to $\mathcal{X}$ achieving*

$$\left|\Pr\left[S^{U_0}(|D_0\rangle_{\mathcal{X},\mathcal{Y}}) = 1\right] - \Pr\left[S^{U_1}(|D_1\rangle_{\mathcal{X},\mathcal{Y}}) = 1\right]\right| \geq 1/\mathrm{poly}(\lambda),$$

then there exists a polynomial-time quantum algorithm S that distinguishes classical samples from the distributions D_0 and D_1

Our proof will make use of two results by Zhandry [Zha12, Zha15], which we restate here for convenience. In the following, quantum oracle access to a function $f : X \to Y$ refers to black-box access to the unitary that maps $|x\rangle\,|y\rangle \to |x\rangle\,|f(x) \oplus y\rangle$ for all x, y.

Theorem 13.2 (Theorem 1.1 of [Zha12])**.** *Let D_0 and D_1 be efficiently sampleable distributions on a set Y, and let X be some other set. Let O_0 and O_1 be the distributions of functions from X to Y where for each $x \in X$, $O_b(x)$ is chosen independently according to D_b. Then if A is an efficient quantum algorithm that can distinguish between quantum access to the oracle O_0 from quantum access to the oracle O_1, we can construct an efficient quantum algorithm B that distinguishes classical samples from D_0 and D_1.*

Theorem 13.3 ([Zha15])**.** *An efficient quantum algorithm cannot distinguish between quantum access to an oracle f implementing a random function $X \to X$ and an oracle π implementing a random permutation $X \to X$.*

Proof. By [Zha12, Theorem 1.1], it suffices for us to show that if there exists a distinguisher S^{U_b} that distinguishes $|D_0\rangle_{\mathcal{X},\mathcal{Y}}$ from $|D_1\rangle_{\mathcal{X},\mathcal{Y}}$ can without directly accessing the $\mathcal{X}$ register, then there is an algorithm to distinguish between quantum oracle access to $D_0 \circ f$ and $D_1 \circ f$ (where $D_b \circ f$ is the composed function $D_b(f(\cdot))$ where $f : X \to X$ is a random function.

By [Zha15], we observe that it suffices to show that S^{U_b} implies an algorithm to distinguish between quantum oracle access to $D_0 \circ \pi$ and $D_1 \circ \pi$ for a random *permutation* $\pi : X \to X$.

Given quantum oracle access to $D_b \circ \pi$, we can implement a unitary $V_{b,\pi}$ that maps $|0\rangle_{\mathcal{X},\mathcal{Y}}$ to the state $|D_{b,\pi}\rangle \coloneqq \sum_x |x\rangle\,|D_b(\pi(x))\rangle$ as follows: apply a Hadamard to $\mathcal{X}$, then apply the $D_b \circ \pi$ oracle to $\mathcal{X} \otimes \mathcal{Y}$.

We can hence use S to distinguish $b = 0$ from $b = 1$ as follows. We prepare the state $|D_{b,\pi}\rangle_{\mathcal{X},\mathcal{Y}}$ using $V_{b,\pi}$. Using $V_{b,\pi}$ we can also implement the operation

$$U_{b,\pi} \coloneqq |D_{b,\pi}\rangle\langle D_{b,\pi}| \otimes \mathbf{X}_{\mathcal{B}} + (\mathbf{I} - |D_{b,\pi}\rangle\langle D_{b,\pi}|) \otimes \mathbf{I}_{\mathcal{B}}$$

as follows: apply $V_{b,\pi}^{\dagger}$ to $\mathcal{X} \otimes \mathcal{Y}$, apply $|0\rangle\langle 0|_{\mathcal{X},\mathcal{Y}} \otimes \mathbf{X}_{\mathcal{B}} + (\mathbf{I} - |0\rangle\langle 0|_{\mathcal{X},\mathcal{Y}}) \otimes \mathbf{I}_{\mathcal{B}}$, then apply $U_{b,\pi}$. We can therefore run $S^{U_{b,\pi}}\,|D_{b,\pi}\rangle$.

Since S^{U_b} does not act on $\mathcal{X}$ except via its oracle, and $|D_b\rangle$ is related to $|D_{b,\pi}\rangle$ by a unitary acting on $\mathcal{X}$ only, it holds that

$$\mathrm{Tr}_{\mathcal{X}}(S^{U_{b,\pi}}(|D_{b,\pi}\rangle\langle D_{b,\pi}|)) = \mathrm{Tr}_{\mathcal{X}}(S^{U_b}(|D_b\rangle\langle D_b|)),$$

which completes the proof. □

13.2 Quantum Simulator

We begin by describing a variable-runtime EQPT_m estimation procedure that will be a useful subroutine in our quantum zero-knowledge simulator for [GK96]. Following Theorem 6.4, let $\mathsf{VarEstimate}$ and $\mathsf{Transform}$ be the first and second stages of the variable-runtime singular vector transform (vrSVT). For binary projective measurements $\mathsf{A}, \mathsf{B}, \mathsf{C}$ on $\mathcal{A}$, we define a "estimate-disturb-transform" procedure $\mathsf{EDT}[\mathsf{A}, \mathsf{B}, \mathsf{C}]$. Intuitively, this procedure first uses $\mathsf{VarEstimate}$ to compute an upper bound on the running time of $\mathsf{Transform}[\mathsf{A} \to \mathsf{B}]$, but then disturbs the state with the measurement C before running $\mathsf{Transform}[\mathsf{A} \to \mathsf{B}]$. However, to ensure that $\mathsf{VarEstimate}$ does not run for unbounded time, the input is first "conditioned" by applying B followed by A, and only proceeding if both measurements return 1.

Formally, the procedure takes an input state on $\mathcal{A}$ and does the following:

$\mathsf{EDT}[\mathsf{A}, \mathsf{B}, \mathsf{C}]$:

1. Apply B to $\mathcal{A}$, obtaining outcome b_1.
2. Apply A to $\mathcal{A}$, obtaining outcome b_2.
3. If $b_1 = 0$ or $b_2 = 0$, stop and output $(0, \bot)$.
4. Otherwise, run $\mathsf{VarEstimate}_\delta[\mathsf{A} \rightleftarrows \mathsf{B}]$ on $\mathcal{A}$, obtaining classical output y.
5. Apply C to $\mathcal{A}$, obtaining outcome c.
6. Run $\mathsf{Transform}_y[\mathsf{A} \to \mathsf{B}]$ on $\mathcal{A}$.
7. Output $\mathcal{A}$ and $(1, c)$.

Let $\widehat{\mathsf{EDT}}[\mathsf{A}, \mathsf{B}, \mathsf{C}]$ denote a coherent implementation of this procedure.

Claim 13.4. *For any efficient measurements* $\mathsf{A}, \mathsf{B}, \mathsf{C}$*,* $\mathsf{EDT}[\mathsf{A}, \mathsf{B}, \mathsf{C}]$ *is* EQPT_m*.*

Proof. Since $\mathsf{EDT}[\mathsf{A}, \mathsf{B}, \mathsf{C}]$ commutes with $\mathsf{M}_{\mathsf{Jor}}[\mathsf{A}, \mathsf{B}]$, it suffices to analyze its running time for states contained within a single Jordan subspace. Let $|\psi_j\rangle \coloneqq \alpha |w_{j,1}\rangle + \beta |w_{j,0}\rangle$. Then

$$\Pr[b_1 = b_2 = 1] = |\alpha|^2 \Pr[\mathsf{A}(|w_{j,1}\rangle) \to 1] \leq p_j.$$

Note that C does not affect the running time of $\mathsf{Transform}_y$. Hence the expected running time of this procedure on $|\psi_j\rangle$ is

$$O((p_j \cdot \log(1/\delta)/\sqrt{p_j} + 1) \cdot (t_{\mathsf{A}} + t_{\mathsf{B}})) = O(\log(1/\delta) \cdot (t_{\mathsf{A}} + t_{\mathsf{B}})).$$

It follows that this procedure is EQPT_m. □

We define the states and measurements used in the simulator.

- For $r \in R$, let $|\mathsf{Sim}_r\rangle \coloneqq \frac{1}{\sqrt{2^\lambda}} \sum_\mu |\mu\rangle \, |\mathsf{SHVZK.Sim}(r; \mu)\rangle$.
- Let $\mathsf{M}_{\mathsf{Sim}} \coloneqq (\Pi_{\mathsf{Sim}}, \mathbf{I} - \Pi_{\mathsf{Sim}})$, where $\Pi_{\mathsf{Sim}} \coloneqq \sum_r |r\rangle\langle r| \otimes |\mathsf{Sim}_r\rangle\langle \mathsf{Sim}_r| \otimes \mathbf{I}$.
- Let $\mathsf{M}_{\mathcal{R}} \coloneqq (\Pi_r)_{r \in R}$, where $\Pi_r \coloneqq U_{V^*}^\dagger |r\rangle\langle r|_{\mathcal{R}} U_{V^*}$.
- Let $\mathsf{M}_{\mathsf{com}} \coloneqq (\Pi_{\mathsf{com}}, \mathbf{I} - \Pi_{\mathsf{com}})$, where

$$\Pi_{\mathsf{com}} \coloneqq \sum_{\substack{r,\omega \\ \mathsf{Commit}(\mathsf{ck}, r, \omega) = \mathsf{com}}} |r, \omega\rangle\langle r, \omega| \;.$$

Sim^{V^*}:

1. Run $V^*(\mathsf{ck})$ for $\mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda)$ to obtain a commitment com.
2. Generate the state $|0\rangle_{\mathcal{R}'} |\mathsf{Sim}_0\rangle_{\mathcal{M},\mathcal{A},\mathcal{Z}}$.
3. Apply $\widehat{\mathsf{EDT}}[\mathsf{M}_{\mathsf{com}}, \mathsf{M}_{\mathsf{Sim}}, \mathsf{M}_{\mathcal{R}}]$, obtaining outcome (b, r) (in superposition). Measure b.
4. If $b = 1$, measure r and replace the state on $\mathcal{R}', \mathcal{M}, \mathcal{A}, \mathcal{Z}$ with $|r\rangle_{\mathcal{R}'} |\mathsf{Sim}_r\rangle_{\mathcal{M},\mathcal{A},\mathcal{Z}}$.
5. Apply $\widehat{\mathsf{EDT}}[\mathsf{M}_{\mathsf{com}}, \mathsf{M}_{\mathsf{Sim}}, \mathsf{M}_{\mathcal{R}}]^\dagger$.
6. Measure register $\mathcal{A}$, obtaining outcome a. Apply U_{V^*} and measure $\mathcal{R}, \mathcal{W}$ to obtain (r', ω); if $\mathsf{Commit}(r', \omega) \neq \mathsf{com}$, stop and output the view of V^*. Otherwise, measure $\mathcal{Z}$, obtaining outcome z. Send z to V^* and output the view of V^*.

Lemma 13.5. *If* Com *is a collapse-binding commitment then* $\mathsf{Sim}^{V^*}(\boldsymbol{\rho})$ *is computationally indistinguishable from* $\mathsf{out}_{V^*}\langle P, V^*\rangle$. Sim^{V^*} *is an* EQPT_c *algorithm.*

Proof. By Claim 13.4, $P[\mathsf{M}_{\mathsf{com}}, \mathsf{M}_{\mathsf{Sim}}, \mathsf{M}_{\mathcal{R}}]$ is EQPT_m, and so Sim^{V^*} is EQPT_c.

We consider three hybrid simulators H_1, H_2, H_3, as follows. All three are provided with some witness w such that $(x, w) \in \mathfrak{R}$. We first define H_1.

$H_1^{V^*}(x, w)$:

1. Run $V^*(\mathsf{ck})$ for $\mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda)$ to obtain a commitment com.
2. Generate the state $|P\rangle := \sum_\mu |\mu\rangle_{\mathcal{M}} |P_\Sigma(x, w; \mu)\rangle_{\mathcal{A}}$.
3. Let $\mathsf{M}_P := (|P\rangle\langle P|, \mathbf{I} - |P\rangle\langle P|)$. Apply $\widehat{\mathsf{EDT}}[\mathsf{M}_{\mathsf{com}}, \mathsf{M}_P, \mathsf{M}_{\mathcal{R}}]$, obtaining outcome (b, r) (in superposition). Measure b.

4-6. As in Sim.

H_1 is indistinguishable from Sim by Lemma 13.1: H_1 is obtained from Sim by replacing $|\mathsf{Sim}_0\rangle$ and $\mathsf{M}_{\mathsf{Sim}}$ with $|P\rangle$ and M_P and interacts only with the $\mathcal{A}$ register, and the distributions on a induced by $(a, z) \leftarrow \mathsf{SHVZK.Sim}(0; \mu)$ and $a \leftarrow P_\Sigma(x, w; \mu')$ are computationally indistinguishable.

$H_2^{V^*}(x, w)$:

1-3. As in H_1.

4. If $b = 1$, measure r and replace the state on $\mathcal{M}, \mathcal{A}, \mathcal{Z}$ with

$$|P_r\rangle = \sum_\mu |\mu\rangle_{\mathcal{M}} |P_\Sigma(x, w; \mu)\rangle_{\mathcal{A}} |P_\Sigma(x, w, r; \mu)\rangle_{\mathcal{Z}}.$$

5. Let $\mathsf{M}_{P,r} := (|P_r\rangle\langle P_r|, \mathbf{I} - |P_r\rangle\langle P_r|)$. Apply $\widehat{\mathsf{EDT}}[\mathsf{M}_{\mathsf{com}}, \mathsf{M}_{P,r}, \mathsf{M}_{\mathcal{R}}]^\dagger$.
6. As in Sim.

By the SHVZK guarantee, the distributions on (a, z) given by $a \leftarrow P_\Sigma(x, w; \mu)$, $z \leftarrow P_\Sigma(x, w, r; \mu)$ and $(a, z) \leftarrow \mathsf{SHVZK.Sim}(r; \mu')$ are computationally indistinguishable. Hence by Lemma 13.1, H_1 and H_2 are computationally indistinguishable.

By the correctness guarantee of Q, if $b = 1$ then the state at the beginning of Step 4 has $\mathrm{Tr}(|P\rangle\langle P| \boldsymbol{\rho}) \geq 1 - \delta$. Note that $|P\rangle$ and $|P_r\rangle$ are related by an efficient local isometry $T_r : \mathcal{M} \to \mathcal{M} \otimes \mathcal{Z}$. Hence Step 4 is $\sqrt{\delta}$-close in trace distance to an application of this isometry. Switching to this state, we can commute the isometry through $\widehat{\mathsf{EDT}}[\mathsf{M}_{\mathsf{com}}, \mathsf{M}_{P,r}, \mathsf{M}_{\mathcal{R}}]^\dagger$, which conjugates it to $\widehat{\mathsf{EDT}}[\mathsf{M}_{\mathsf{com}}, \mathsf{M}_P, \mathsf{M}_{\mathcal{R}}]^\dagger$. This leads to the third hybrid, below.

$H_3^{V^*}(x, w)$:

1-3. As in H_2.

4. If $b = 1$, measure r.
5. Apply $\widehat{\mathsf{EDT}}[\mathsf{M}_{\mathsf{com}}, \mathsf{M}_P, \mathsf{M}_{\mathcal{R}}]^\dagger$.
6. Apply U_{V^*} and measure $\mathcal{R}, \mathcal{W}$ to obtain (r', ω). If $\mathsf{Commit}(r', \omega) \neq \mathsf{com}$, stop and output the view of V^*. Otherwise, apply $T_{r'}$ to $\mathcal{M}$ and measure $\mathcal{Z}$, obtaining outcome z. Send z to V^* and output the view of V^*.

H_3 is statistically close to H_2 provided that $\Pr[r = r'] = 1 - \mathrm{negl}(\lambda)$. Moreover, the collapsing property of the commitment implies that Step 4 is computationally undetectable. If this step is removed then the effect of Steps 3 and 5 is simply to apply $\mathsf{M}_{\mathsf{com}}$; the output is then precisely the view of V^* in a real execution.

Finally, we have that by $r = r'$ with all but negligible probability by the unique message-binding of the commitment scheme (Lemma 4.2). □

Acknowledgments

We thank Zvika Brakerski, Ran Canetti, Yael Kalai, Vinod Vaikuntanathan, and Mark Zhandry for helpful discussions.

References

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando Brandao, David Buell, Brian Burkett, Yu Chen, Jimmy Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Michael Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew Harrigan, Michael Hartmann, Alan Ho, Markus Rudolf Hoffmann, Trent Huang, Travis Humble, Sergei Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, Dave Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod Ryan McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin Jeffery Sung, Matt Trevithick, Amit Vainsencher, Benjamin Villalonga, Ted White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574:505–510, 2019.

[ACL21] Prabhanjan Ananth, Kai-Min Chung, and Rolando L. La Placa. On the concurrent composition of quantum zero-knowledge. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 346–374, Virtual Event, August 2021. Springer, Heidelberg.

[AL20] Prabhanjan Ananth and Rolando L. La Placa. Secure quantum extraction protocols. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 123–152. Springer, Heidelberg, November 2020.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th FOCS*, pages 474–483. IEEE Computer Society Press, October 2014.

[BBBV97] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

[BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 467–496, Virtual Event, August 2021. Springer, Heidelberg.

[BG93] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 390–420. Springer, Heidelberg, August 1993.

[BG02] Boaz Barak and Oded Goldreich. Universal arguments and their applications. In *Proceedings 17th IEEE Annual Conference on Computational Complexity*, pages 194–203. IEEE, 2002.

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

[BL02] Boaz Barak and Yehuda Lindell. Strict polynomial-time in simulation and extraction. In *34th ACM STOC*, pages 484–493. ACM Press, May 2002.

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. Citeseer, 1986.

[BMO90] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. Perfect zero-knowledge in constant rounds. In *Proceedings of the twenty-second annual ACM symposium on Theory of Computing*, pages 482–493, 1990.

[BOV03] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 299–315. Springer, Heidelberg, August 2003.

[BS20] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 269–279. ACM Press, June 2020.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on computing*, 26(5):1411–1473, 1997.

[CCLY21] Nai-Hui Chia, Kai-Min Chung, Qipeng Liu, and Takashi Yamakawa. On the impossibility of post-quantum black-box zero-knowledge in constant rounds. FOCS '21, 2021.

[CCY21] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. A black-box approach to post-quantum zero-knowledge in constant rounds. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 315–345, Virtual Event, August 2021. Springer, Heidelberg.

[CMSZ21] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. Post-quantum succinct arguments: breaking the quantum rewinding barrier. FOCS '21, 2021.

[Deu85] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat–Shamir transformation in the quantum random-oracle model. In *Proceedings of the 39th Annual International Cryptology Conference*, CRYPTO '19, pages 356–383, 2019.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *22nd ACM STOC*, pages 416–426. ACM Press, May 1990.

[GHKW17] Rishab Goyal, Susan Hohenberger, Venkata Koppula, and Brent Waters. A generic approach to constructing and proving verifiable random functions. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 537–566. Springer, Heidelberg, November 2017.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, June 1996.

[GLSV21] Alex B. Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in MiniQCrypt. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 531–561. Springer, Heidelberg, October 2021.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In *27th FOCS*, pages 174–187. IEEE Computer Society Press, October 1986.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 171–185. Springer, Heidelberg, August 1987.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 193–204. ACM Press, June 2019.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007.

[IOS97] Toshiya Itoh, Yuji Ohta, and Hiroki Shizuya. A language-dependent cryptographic primitive. *Journal of Cryptology*, 10(1):37–50, December 1997.

[Jor75] Camille Jordan. Essai sur la géométrie à n dimensions. *Bulletin de la Société mathématique de France*, 3:103–174, 1875.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992.

[Klo20] Michael Klooß. On (expected polynomial) runtime in cryptography. Cryptology ePrint Archive, Report 2020/809, 2020. https://eprint.iacr.org/2020/809.

[Lev86] Leonid A Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

[LP98] Noah Linden and Sandu Popescu. The halting problem for quantum computers. *arXiv preprint quant-ph/9806054*, 1998.

[LS91] Dror Lapidot and Adi Shamir. Publicly verifiable non-interactive zero-knowledge proofs. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 353–365. Springer, Heidelberg, August 1991.

[LS19] Alex Lombardi and Luke Schaeffer. A note on key agreement and non-interactive commitments. Cryptology ePrint Archive, Report 2019/279, 2019. https://eprint.iacr.org/2019/279.

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat–Shamir. In *Proceedings of the 39th Annual International Cryptology Conference*, CRYPTO '19, pages 326–355, 2019.

[MV03] Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 282–298. Springer, Heidelberg, August 2003.

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

[Mye97] John M Myers. Can a universal quantum computer be fully quantum? *Physical Review Letters*, 78(9):1823, 1997.

[NWZ09] Daniel Nagaj, Pawel Wocjan, and Yong Zhang. Fast amplification of QMA. *Quantum Information & Computation*, 9(11&12):1053–1068, 2009.

[Oza98a] Masanao Ozawa. Quantum nondemolition monitoring of universal quantum computers. *Physical Review Letters*, 80(3):631, 1998.

[Oza98b] Masanao Ozawa. Quantum turing machines: local transition, preparation, measurement, and halting. *arXiv preprint quant-ph/9809038*, 1998.

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *43rd FOCS*, pages 366–375. IEEE Computer Society Press, November 2002.

[PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 403–418. Springer, Heidelberg, March 2009.

[Reg06] Oded Regev. Fast amplification of QMA (lecture notes), Spring 2006.

[RK99] Ransom Richardson and Joe Kilian. On the concurrent composition of zero-knowledge proofs. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 415–431. Springer, Heidelberg, May 1999.

[Ros04] Alon Rosen. A note on constant-round zero-knowledge proofs for NP. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 191–202. Springer, Heidelberg, February 2004.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, Heidelberg, April 2012.

[Unr16a] Dominique Unruh. Collapse-binding quantum commitments without random oracles. In *Proceedings of the 22nd International Conference on the Theory and Applications of Cryptology and Information Security*, ASIACRYPT '16, pages 166–195, 2016.

[Unr16b] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 497–527. Springer, Heidelberg, May 2016.

[vdG97] Jeroen van de Graaf. *Towards a formal definition of security for quantum protocols.* PhD thesis, University of Montreal, 1997.

[Wat06] John Watrous. Zero-knowledge against quantum attacks. In Jon M. Kleinberg, editor, *38th ACM STOC*, pages 296–305. ACM Press, May 2006.

[Win99] Andreas J. Winter. Coding theorem and strong converse for quantum channels. *IEEE Transactions on Information Theory*, 45(7):2481–2485, 1999.

[Zha12] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Heidelberg, August 2012.

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Inf. Comput.*, 15(7&8):557–567, 2015.