Electronic Colloquium on Computational Complexity, Report No. 105 (2018)

Quantum proof systems for iterated exponential time, and beyond

Joseph Fitzsimons[*] Zhengfeng Ji[†] Thomas Vidick[‡] Henry Yuen[§]

Abstract

We show that any language in nondeterministic time $\exp(\exp(\cdots\exp(n)))$, where the number of iterated exponentials is an arbitrary function $R(n)$, can be decided by a multiprover interactive proof system with a classical polynomial-time verifier and a constant number of quantum entangled provers, with completeness 1 and soundness $1-\exp(-C\exp(\cdots\exp(n)))$, where the number of iterated exponentials is $R(n)-1$ and $C>0$ is a universal constant. The result was previously known for $R=1$ and $R=2$; we obtain it for any time-constructible function R.

The result is based on a compression technique for interactive proof systems with entangled provers that significantly simplifies and strengthens a protocol compression result of Ji (STOC'17). As a separate consequence of this technique we obtain a different proof of Slofstra's recent result (unpublished) on the uncomputability of the entangled value of multiprover games.

Finally, we show that even minor improvements to our compression result would yield remarkable consequences in computational complexity theory and the foundations of quantum mechanics: first, it would imply that the class MIP* contains all computable languages; second, it would provide a negative resolution to a multipartite version of Tsirelson's problem on the relation between the commuting operator and tensor product models for quantum correlations.

[*] Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372 and Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore 117543.

[†] Centre for Quantum Software and Information, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia

[‡] California Institute of Technology, USA.

[§] UC Berkeley, USA and University of Toronto, Canada.

ISSN 1433-8092

1 Introduction

The combined study of interactive proof systems and quantum entanglement has led to multiple discoveries at the intersection of theoretical computer science and quantum physics. On the one hand, the study has revealed that quantum entanglement, a fundamental physical phenomenon, can be harnessed in interactive protocols to accomplish an array of novel computing and cryptographic tasks, ranging from the certified generation of random numbers to improved protocols for multiparty cryptography and classically-verifiable quantum computation. On the other hand, interactive proof systems, a cornerstone of modern complexity theory and cryptography, have provided a powerful lens through which to examine the counter-intuitive properties of quantum entanglement. This lens has enabled researchers to develop sophisticated ways of exploring phenomena such as the monogamy of entanglement, embezzlement of quantum states, and more.

We investigate a central question in this area: what is the *computational complexity* of interactive proof systems with multiple quantum entangled provers? The starting point for this question dates back to the seminal result of Babai, Fortnow and Lund, who showed that the set of languages that can be decided by a (classical) multiprover interactive proof system, denoted by MIP, equals the set of languages that can be decided in nondeterministic exponential time (denoted by NEXP) [BFL91]. It is not difficult to show that MIP $\subseteq$ NEXP, but the reverse containment is nontrivial and the work of [BFL91] was an influential stepping stone towards the PCP Theorem [AS98, ALM$^+$98].

A long line of work, starting with that of Cleve et al. [CHTW04], has explored the setting of interactive proof systems where a classical polynomial-time verifier interacts with provers that are *quantum* and may share *entanglement*. This gives rise to the complexity class MIP*, which is the set of all languages decidable by such proof systems.[1] Quantum entanglement is a resource that allows isolated parties to generate correlations that cannot be reproduced by (classical) shared randomness alone; however, entanglement does not allow for instantaneous communication. A central question raised by [CHTW04] is whether MIP* = MIP, or equivalently, whether MIP* = NEXP.

A richer set of correlations gives additional power to provers in an interactive proof system, making the relationship between MIP* and MIP non-obvious. On the one hand, a multiprover interactive proof system that is sound against "cheating" classical provers may no longer be sound against "cheating" entangled provers; this prevents one from automatically concluding that MIP $\subseteq$ MIP*. On the other hand, a proof system may require "honest provers" to use quantum entanglement in order to satisfy the completeness property. Entanglement thus allows one to consider a broader set of protocols, putting in question the inclusion MIP* $\subseteq$ MIP.

The quest to pin down the computational power of proof systems with entangled provers has led to a number of surprising discoveries. The best lower bound that is currently known is that NEXP = MIP $\subseteq$ MIP*, a nontrivial result that follows from a more general technique of "immunization" of classical proof systems against malicious entangled provers [IV12, NV17b]. Surprisingly, there are no meaningful upper bounds known for MIP*. In a striking result, Slofstra gave evidence that the complexity of MIP* might be very different from its classical counterpart: he proved that it is *undecidable* to determine whether an interactive proof system with two provers has an entangled strategy that is accepted with probability 1 (in other words, whether there is a *perfect* entangled strategy) [Slo16, Slo17]. In contrast, the complexity of determining whether such a proof system has a perfect *classical* strategy is exactly equal to NEXP. Another recent result of Ji [Ji17] points in the same direction: Ji showed that any language in non-deterministic doubly-exponential time can be decided by a classical polynomial-time verifier interacting with $k = 11$ provers, with

[1] The * in MIP* refers to the entanglement.

completeness 1 and soundness that is exponentially close to 1.[2]

In this work we explore the expanse of complexity-space that entangled-prover interactive proof systems can reach. We focus on the "small gap" regime: we consider the problem of distinguishing between the cases when a multiprover proof system has a perfect entangled strategy, or when all entangled provers are rejected with probability at least ε, where ε is a quantity that may go to 0 quickly with the size of the verifier in the proof system. Our results smoothly interpolate between the hardness result of [IV12, NV17b, Ji17] and Slofstra's undecidability result. For clarity we restrict our attention to *hyper-exponential* time functions, i.e. time-constructible functions of the form $t(n) = \Lambda_R(n)$, where $\Lambda_0(n) = n$ and for any integer-valued function $R = R(n) \geq 0$, $\Lambda_{R+1}(n) = 2^{\Lambda_R(n)}$. For a multiprover game $\mathcal{G}$, the *entangled value* $\omega^*(\mathcal{G})$ is the maximum success probability of quantum provers sharing entanglement in the game.

Theorem 1.1. *Let $k \geq 15$ be an integer. Let $t : \mathbb{N} \to \mathbb{N}$ be a hyper-exponential function. There are universal constants $C, c > 0$ such that given the description of polynomial-size circuits for the verifier in a k-prover game $\mathcal{G}$, the problem of distinguishing between*

$$\omega^*(\mathcal{G}) = 1 \qquad \text{or} \qquad \omega^*(\mathcal{G}) \leq 1 - \frac{C}{(t(n))^c}$$

is hard for nondeterministic $2^{t(n)}$ time.

The "base case" for Theorem 1.1, corresponding to $R = 0$ and $t(n) = n$, is the result that NEXP $\subseteq$ MIP* [IV12, NV17b], where MIP* is the class of languages that can be decided using an entangled-prover interactive proof system, with completeness $\frac{2}{3}$ and soundness $\frac{1}{3}$ (the completeness-soundness gap can be amplified from inverse polynomial to constant using hardness amplification techniques [BVY17]). The first step, $R = 1$ and $t(n) = 2^n$, follows from Ji's result [Ji17] mentioned earlier, albeit using a game with $k = 11$ provers.

A corollary of both our and Ji's earlier result is that the "honest strategy" for the provers (i.e. those satisfying the completeness property) in the games constructed through the reduction from Theorem 1.1 provably require the provers to share entanglement. Moreover, it is often possible to obtain lower bounds on the dimension of entanglement required to achieve close to optimal success probability; this is the case for our result, as described below.

The proof of Theorem 1.1 is based on a compression technique that significantly simplifies and extends the approach pioneered in [Ji17]. Our generalized compression result can be recursively composed with itself in order to obtain the statement of Theorem 1.1 for any integer-valued $R(n) \geq 1$.

The starting point of the compression approach of [Ji17] is to extend the notion of a *history state*. The concept of a history state was first introduced by Kitaev in order to efficiently encode any polynomial-time quantum computation as the ground state of a local Hamiltonian, in a way that is also efficiently verifiable [KSV02]. The compression result of [Ji17] as well as the one in this paper constructs a game to verify history states that encode the execution of a (different) multiprover game, including the actions of the provers (which in general are not efficiently computable). The verification is performed by executing a "games" version of the traditional verification procedure for history states, that consists in randomly sampling a local Hamiltonian term and measuring its energy.

There are two key ideas behind our generalized compression technique. The first is to ensure that the game $\mathcal{G}$ that verifies the history state of a multiprover game $\mathcal{G}'$ can be executed using a circuit that is logarithmic in the size of $\mathcal{G}'$, provided that $\mathcal{G}'$ is specified in a sufficiently uniform and

[2] Due to the vanishing gaps neither Slofstra's nor Ji's result directly separates MIP* from MIP, though they do separate the zero-error and exponentially-small error variants respectively.

succinct manner. The second idea is to compose the first idea with itself, i.e. consider the history state for the computation performed by the history state verification procedure. At this point there are a number of delicate issues to consider, including identifying the right model for specifying verifiers, verifiers of verifiers, etc.; we give more details in Section 1.1.

On a more informal note, we observe that the kind of compression achieved here may be thought of as a "bootstrapping" of Kitaev's history state technique, in a similar sense to the composition technique from the PCP literature that "bootstraps" an efficient PCP into a super-efficient one.[3] The fact that history states are ground states of local Hamiltonians is a statement about the local verifiability of arbitrary quantum computation. Our result goes further by making the following observations. First, not only is the verification procedure local, it is also exceedingly efficient — it can be executed in time logarithmic in the size of the original computation. Second, it is possible to consider a history state for the verification procedure itself. Third, and most strikingly, the latter history state can be verified with the same complexity as the verification procedure, without reference to the size of the original computation. This last step crucially relies on *rigidity* properties of entanglement which acts as a "leash" on quantum systems. It is sufficient to only control the leash-holder: if the leash-holder manages to hold the dog tightly enough, then there is no longer any reason to worry about the (hyper-exponential-size) dog itself.

It is worth noting that such "PCP composition on steroids" has no classical analogue. A classical PCP verifier runs in polynomial time and uses polynomially many random bits to verify an exponentially long proof. Encoding the computation performed by such a verifier in a way that can be verified using, say, a classical multiprover interactive proof system, again requires a polynomial-sized verifier flipping polynomially many bits. This is because the only way to "verify the verification procedure" is to, at least with some probability, access some of the original proof bits. In the quantum case, it is possible to leverage entanglement between provers to avoid the need for the "inner" verifier (to borrow some terminology from the PCP literature) to make any query at all to the original proof qubits.

Before proceeding we formulate another consequence of compression that highlights the versatility of our approach. As already mentioned, it was recently shown by Slofstra that the problem of determining whether a given multiprover game has a perfect entangled strategy is undecidable. Slofstra's result proceeds by an ingenious (and intricate) reduction to the word problem in finitely presented groups, which is known to be undecidable. The proof of the latter itself involves a sophisticated embedding of the computation of an arbitrary Turing Machine (in fact, a Minsky machine) in an instance of the word problem in a suitable finitely presented group [Nov55, Boo58, Kar82].

We give a different proof of Slofstra's undecidability result, by directly constructing an interactive proof system from a Turing machine. Arguably, our result provides an intuitive reason for *why* the problem is undecidable, showing in a precise sense how smaller and smaller gaps can be leveraged to verify that the provers are performing an increasingly complex computation. More precisely, the main idea for our proof is to design a family of games $\{\mathcal{G}_n\}_{n\geq 1}$ such that for any $n \geq 1$ the verifier in the game $\mathcal{G}_n$ verifies if a Turing machine provided as input halts within n steps, and if it does not, executes a game with the provers that verifies that, either the provers hold a quantum proof that the Turing machine halts within 2^n steps, or they hold a history state for the verification of a quantum proof that either the Turing machine halts within 2^{2^n} steps, or... Somewhat more formally, we obtain the following (see Theorem 7.6 for a more complete statement).

[3] The analogy only goes so far: composition in PCPs reduces the answer size; here, we reduce the query size.

Theorem 1.2. *For all deterministic Turing machines M, there exists a multiprover game $\mathcal{G}_M$ (that can be computed from the description of M) such that if M halts in finite time then $\omega^*(\mathcal{G}_M) < 1$, whereas if M does not halt then $\omega^*(\mathcal{G}_M) = 1$. Furthermore, there exists a universal constant $\eta > 0$ such that for any non-halting M, any strategy for the provers that succeeds with probability at least $1 - \varepsilon$ in $\mathcal{G}_M$, for some $\varepsilon \geq 0$, requires the use of an entangled state of local dimension at least $2^{\Omega(\varepsilon^{-\eta})}$.*

The game $\mathcal{G}_M$ in Theorem 1.2 is a game with 15 provers that can be efficiently computed from M; the undecidability result follows immediately. In addition, as stated in the theorem our game can be used as a form of dimension test for the strategies of the provers. Up to the value of the constant η the bound $2^{\Omega(\varepsilon^{-\eta})}$ matches the best bound known, for a three-prover game considered in [JLV18].

1.1 Proof overview

We provide a detailed overview for the proof of Theorem 1.1. In Section 1.1.1 we sketch our main "compression" result and expand on the compression technique from [Ji17]. The following sections sketch the proof of the compression theorem. We start by describing a method to succinctly describe the actions of a verifier in a multiprover game in Section 1.1.2. In Section 1.1.3 we describe the main steps of the proof: (1) design a history state associated with the execution of a multiprover game, (2) design a game that verifies the history state with the help of an additional trusted prover, and finally (3) design a game in which the honest prover has been merged into existing provers. This last step, prover merging, is described in more detail in Section 1.1.4. In Section 1.1.5 we sketch how the compression theorem can be applied recursively to show Theorem 1.1 and Theorem 1.2.

1.1.1 Protocol compression

The main workhorse of this paper is a compression theorem for quantum multiprover interactive protocols that simplifies and strengthens the compression result of [Ji17]. To state the result, we first review the notion of k-prover "extended nonlocal (ENL) game", which is a type of quantum multiprover game introduced in [JMVW16]. A k-prover ENL game is a three-turn interaction between a quantum verifier and k quantum provers sharing entanglement. The game (or "protocol") proceeds in three stages. First, the provers send a quantum register C to the verifier. Second, the verifier measures the register C to obtain an outcome t. [4] The verifier then computes a classical query $Q = (q_1, \ldots, q_k)$ that it distributes to the provers. Third, the provers respond with classical answers $a = (a_1, \ldots, a_k)$ to their respective questions. In general, each prover's answer is determined by performing a measurement on the prover's share of a quantum state that may be entangled with C. Finally, the verifier makes an accept/reject decision based on the outcome t, its internal randomness, and the provers' answers. The maximum acceptance probability of an ENL game $\mathcal{G}$ is denoted $\omega^*(\mathcal{G})$, and is also called the (entangled) *value* of $\mathcal{G}$.

The whole interaction between verifier and provers in an ENL game can be represented as a quantum circuit of a special form that we call a *protocol circuit*, as depicted in Figure 1. A protocol circuit starts with the application of a quantum circuit C_Q on registers C (which holds the provers' first message), V (the verifier's private workspace), and M (which holds the messages exchanged between the verifier and provers). The circuit C_Q implements the verifier's measurement on register C, and the verifier's choice of questions to the provers. The circuit C_Q is followed by an arbitrary unitary transformation for each prover i, applied on the component M_i of the message register that the prover has access to, as well as its private workspace P_i (that contains the prover's part of

[4] Our definition of ENL game is slightly more general than that in [JMVW16], where the sampling of questions is classical and does not depend on C.

shared entangled state). Finally, the last step in the protocol circuit is the application of a circuit C_A that acts on C, V and M and computes the verifier's decision in the game, that is written on a specially designated "output qubit".

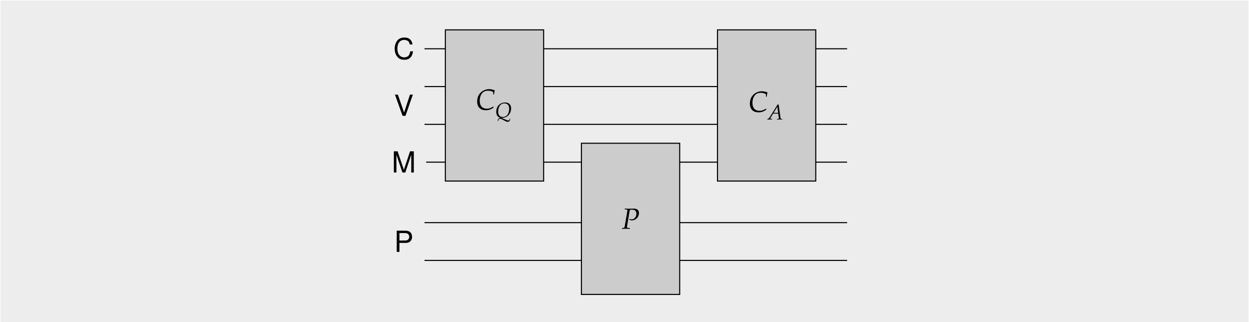

Figure 1: The protocol circuit of an extended nonlocal game.

The compression theorem applies to families of ENL games $\{\mathcal{G}_N\}$ that have *succinct descriptions*. By this we mean, not only that the protocol circuit associated with $\mathcal{G}_N$ has size polynomial in N, but moreover there exists a deterministic Turing machine G (called a *Gate Turing Machine* (GTM)) that on input (N,t), where N and t are two integers written in binary, runs in polynomial time and returns the description of the t-th gate of the protocol circuit associated with $\mathcal{G}_N$ (and a special symbol if t is larger than the circuit size). If the t-th gate is an action of the prover, the GTM returns another special symbol.

Theorem 1.3 (Compression Theorem)**.** *Let $k \geq 7$ be an integer and let $\{\mathcal{G}_N\}$ be a succinctly described family of k-prover ENL games with GTM G. Then there exists a family of k-prover ENL games $\{\mathcal{G}_n^\sharp\}$ such that for all integer $n \geq 1$ and $N = 2^n$, it holds that*

$$\omega^*(\mathcal{G}_n^\sharp) \leq 1 - \frac{(1-\omega^*(\mathcal{G}_N))^\alpha}{\mathrm{poly}(N)} , \tag{1}$$

where $\alpha \geq 1$ is a universal constant, and if $\omega^*(\mathcal{G}_N) = 1$ then we have $\omega^*(\mathcal{G}_n^\sharp) = 1$. Moreover, there exists a Turing machine $A^\sharp$ that on input $(1^n, G)$ returns the description of $\mathcal{G}_n^\sharp$ in polynomial time.

The strength of the theorem lies in the exponential reduction in the size of the verifiers of the ENL game, from poly(N) (the size of $\mathcal{G}_N$) to poly(n) = poly($\log N$) (the size of $\mathcal{G}_n^\sharp$). The cost of this exponential compression of game size is that the value of the game gets "compressed" towards 1; nevertheless, games with value 1 (resp. < 1) are compressed to games with value 1 (resp. < 1). Theorem 1.3 differs from the results of [Ji17] in two significant ways. First, the compression result in [Ji17] does not yield a family $\{\mathcal{G}_n^\sharp\}$ that is as efficiently described as the games returned by our reduction.[5] The recourse to succinct descriptions via Gate Turing Machines is an essential ingredient for the recursive application of Theorem 1.3. Second, the compression result in [Ji17] increases the number of provers, from k to $k + 8$. Our result does not require the use of additional provers; this is again essential in allowing a large (or even infinite) number of recursive applications of the theorem.

In the following subsections we sketch the proof of Theorem 1.3. The first step is to make the notion of "succinctly described" more concrete.

[5] Although the question lengths of the "compressed" game in [Ji17] are $O(\log N)$, the verifier itself has size poly(N). The verifier for the game $\mathcal{G}_n^\sharp$, in contrast, has size poly($\log N$).

1.1.2 Succinct descriptions of verifiers

In the study of quantum interactive proof systems, families of games $\{\mathcal{G}_N\}$ are usually presented as a uniformly generated family of circuits for the verifier: there exists a polynomial-time deterministic Turing machine A that on input 1^N returns a circuit description of the verifier in $\mathcal{G}_N$. However, such uniform descriptions of verifier circuits are insufficient for our compression result: from a game $\mathcal{G}_N$ we aim to design a "compressed game" $\mathcal{G}_n^\sharp$ that has size $\mathrm{poly}(n)$, exponentially smaller than the size of $\mathcal{G}_N$. In particular, $\mathcal{G}_n^\sharp$ does not have nearly enough time to run A to get a circuit description of the verifier of $\mathcal{G}_N$. What we need is that the verifier of $\mathcal{G}_n^\sharp$ be granted some form of *implicit* description of the verifier of $\mathcal{G}_N$.

We achieve this via the notion of a *Gate Turing Machine* (GTM) for a family of ENL games $\{\mathcal{G}_N\}$. As mentioned before, it is a Turing machine G that on input (N, t) outputs in $\mathrm{poly}\log(N)$ time the description of the t-th gate of the protocol circuit of $\mathcal{G}_N$ (which has size $\mathrm{poly}(N)$).

Thus, our notion of "succinct description" for a family of ENL games $\{\mathcal{G}_N\}$ is that there is a GTM G for the family. With this notion in place, it remains to show the compression theorem: any succinctly described family of games $\{\mathcal{G}_N\}$ can be "compressed" to another family of ENL games $\{\mathcal{G}_n^\sharp\}$ with the properties described in Theorem 1.3. We sketch how this is done in the next sections.

1.1.3 Testing history states of protocol circuits

With the appropriate notion of succinct description in place, we describe the three main steps that go into the proof of Theorem 1.3.

The first step consists in considering the history state $|\Psi_{\mathcal{G}}(N)\rangle$ of the protocol circuit (Figure 1) associated with an execution of $\mathcal{G} = \mathcal{G}_N$, where $N = 2^n$. This state is defined on the registers CVMP, and may be extremely large, depending on the size of the provers' registers. In addition, the state has a component on a clock register C_{outer} of the same dimension as the total number of gates τ_N in the protocol circuit, which is polynomial in N; thus the register C_{outer} is over $O(n)$ qubits. Concretely, the state $|\Psi_{\mathcal{G}}(N)\rangle$ has the form

$$|\Psi_{\mathcal{G}}(N)\rangle \;=\; \frac{1}{\sqrt{\tau_N + 1}} \sum_{t=0}^{\tau_N} |t\rangle_{\mathsf{C}_{outer}} \otimes U_t \cdots U_1 |\psi_{\mathcal{G}}(0)\rangle_{\mathsf{CVMP}} \,. \tag{2}$$

Here $|\psi_{\mathcal{G}}(0)\rangle$ is the initial state of the verifier and the provers' registers in G, with C denoting the initial register received from the provers, V the private workspace for the verifier, $\mathsf{M} = \mathsf{M}_1, \ldots, \mathsf{M}_k$ the message registers, and $\mathsf{P} = \mathsf{P}_1, \ldots, \mathsf{P}_k$ the private spaces for the provers.

Note that in (2), almost all unitaries are gates applied by the verifier, except k of them, one for each prover, that can be considered "wild cards". The important property is that, if $\omega^*(\mathcal{G}) = 1$ then there exists a state of the form (2), for some choice of $|\psi_{\mathcal{G}}(0)\rangle$, and some choice of unitaries to apply in the "wildcard" locations, that is a ground state (energy 0) of the local Hamiltonian $H_{\mathcal{G}}(N)$ that verifies the history state (this is entirely analogous to Kitaev's circuit-to-Hamiltonian construction, but for the use of the prover gates which may induce large non-local Hamiltonian terms to verify their propagation). Conversely, if $\omega^*(\mathcal{G}_N) = 1$ then no such state exists, irrespective of the choice of the "wildcard" unitaries.

The next step is to design an intermediate ENL game $\mathcal{G}_H$ that has one additional prover, called the "Pauli Prover" PV. We call the verifier in $\mathcal{G}_H$ the *outer verifier*. The goal of the outer verifier is to verify that the provers share the state $|\Psi_{\mathcal{G}}(N)\rangle$, where registers associated with the verifier in $\mathcal{G}$ (that we call the *inner verifier*), i.e. C, V and M, are given to PV, while the clock register C_{outer} is the prover's first message in the ENL game $\mathcal{G}$. As already mentioned, this initial message has length $O(n)$ qubits.

Informally, to achieve this verification task the outer verifier and the Pauli Prover collaborate to implement a family of tests that are game-like versions of the tests implemented by the local Hamiltonian $H_{\mathcal{G}}(N)$. This includes an "input check" (the state $|\psi_{\mathcal{G}}(0)\rangle$ is well-formatted), a "gate check" (each time step corresponds to the application of a unitary, and unitaries associated with the inner verifier are the right ones, as specified in the circuits C_Q and C_A), and an "output check" (the final decision made by the inner verifier is to accept). Each of these checks involves not only the verifier and PV, but also the other provers, that are required to apply their prover gate when the corresponding propagation check is performed.

In designing $\mathcal{G}_H$, we take advantage of the fact that the Pauli Prover is considered "honest": it always implements the observable that it is asked by the outer verifier. However, for reasons that will soon become clear the Pauli Prover can only be asked to implement single- or two-qubit Pauli observables.[6] This means that all tests performed by the outer verifier can only require such observables on the registers CVM.

The crucial point here is that the complexity of the verifier in the game $\mathcal{G}_H$ is exponentially smaller than the complexity of the verifier in $\mathcal{G}$. The reason this is possible is that in order for the verifier in $\mathcal{G}_H$ to check that the entangled state shared by the provers is a valid history state for the protocol circuit associated with $\mathcal{G}$ it is enough to select a random time step in that circuit, and implement the associated check. Both of these can be performed in time poly $\log(N)$; the first trivially so, and the second thanks to our assumption that $\mathcal{G}$ is specified through a "succinct description", provided by the verifier $\mathcal{V}$ and GTM G associated with $\{\mathcal{G}_N\}$, as described in Section 1.1.2.

In the last step we convert the Single Pauli Prover game $\mathcal{G}_H$ into a new ENL game $\mathcal{G}^\sharp = \mathcal{G}^\sharp_n$, with the same number of provers as in the original ENL $\mathcal{G}$, but with drastically reduced question length — it is now $O(n)$, when questions in $\mathcal{G}$ might have been poly(N) bits long. For this we need to remove the "honest" assumption on PV, and moreover we need to "merge" PV with existing provers. This step of prover merging is explained in the next subsection.

1.1.4 Prover merging

Prover merging is performed in two steps. The first step uses somewhat standard techniques, similar to those employed in [Ji17], that originate in the self-testing literature. The main idea is to require the honest Pauli prover PV in $\mathcal{P}$ to implement the observable it is asked to measure transversally, on an error-encoded version of his share of the state (this is the main motivation for restricting the prover to Pauli observables), and then to split PV into as many provers as the error-correcting code requires. It is then possible, using self-testing technique, to test the "split" PV so as to ensure that any deviation from the honest actions is detected by the verifier.

The second step is the actual merging step. This step is somewhat delicate: we take the split provers, and merge them into existing provers from $\mathcal{G}$. Since each prover P now simultaneously receives two questions — its question in $\mathcal{G}$, as well as the share of the question to PV that would have been sent to the split prover that got merged into P — soundness is non-obvious.

To show that this step does not compromise soundness, we leverage the fact that, by construction, the prover that is to be merged only has to perform very simple operations: Pauli σ_X and σ_Z observables, on a constant number of qubits at a time. These kinds of operations can be tested, indeed "commanded", in a very rigid way by using self-testing results. Therefore, we can embed these actions into any prover. It is then straightforward to enforce that a prover performs the right

[6] In fact, triples of commuting two-qubit observables; we gloss over this for purposes of this overview.

action on a Pauli observable. However, its action on the real question may depend on the Pauli question. To get around this we once again leverage the structure of the Pauli Prover game as well as the quantum error-correcting code. More details on this part are given in Section 5.

1.1.5 Recursive compression

Ultimately, we use our compression theorem (Theorem 1.3) in a recursive fashion to prove Theorem 1.1. To illustrate the essential idea behind the recursive compression approach, we give an informal overview of the proof of the statement that any language computable in deterministic time $t(n)$ has a quantum interactive proof system with completeness-soundness gap that scales as an inverse polynomial in $t(n)$.

Let L be such a language. Then there exists a deterministic Turing machine M that on input $x \in \{0,1\}^n$ decides whether $x \in L$ in time $t(n)$. For every $x \in \{0,1\}^n$ and integer $N \geq n$, we construct a verifier $\mathcal{V}_{x,N}$ for a 7-prover ENL game $\mathcal{G}_{x,N}$ that does the following. The verifier first runs M for N steps on input x. If M accepts in this time, then $\mathcal{V}_{x,N}$ accepts. If M rejects in this time, then $\mathcal{V}_{x,N}$ rejects. Otherwise, M has not halted. In this case $\mathcal{V}_{x,N}$ executes a *compressed* version of the protocol corresponding to $\mathcal{V}_{x,2^N}$, which is an exponentially larger version of itself. This compressed protocol is provided by Theorem 1.3. The recursion continues until at some point, M is run for a large enough "tower of exponential" number of steps that exceeds $t(n)$, in which case M either accepts or rejects input x. The following can then be shown by induction on R such that $t(n) \leq \Lambda_R(n)$. If $x \in L$ then the value of the game $\mathcal{G}_{x,t(n)}$ is 1, and therefore for all $N \leq t(n)$ the value of $\mathcal{G}_{x,N}$ is 1, which implies that $\mathcal{G}_{x,n}$ has value 1. Otherwise, if $x \notin L$, then using Theorem 1.3 we obtain that the value of $\mathcal{G}_{x,n}$ is at most $1 - \Omega(1/\operatorname{poly}(t(n)))$.

This nearly shows the desired conclusion, except that Theorem 1.3 requires that the family of games to be compressed have a succinct description in the manner described in Section 1.1.2. We thus need to argue that the family of games $\left\{\mathcal{G}_{x,n}\right\}$ has a GTM G associated with it. *A priori* it is unclear whether the verifiers $\left\{\mathcal{V}_{x,n}\right\}$ are structured enough so that any particular gate of the verifier circuits can be specified in polylogarithmic time. However, we show that as long as the verifiers $\left\{\mathcal{V}_{x,n}\right\}$ are *uniformly generated* (meaning that there is some polynomial time Turing machine A that on input $(1^n, x)$ returns the description of the verifier circuits of $\mathcal{V}_{x,n}$), there is an *equivalent* family of verifiers $\left\{\mathcal{V}'_{x,n}\right\}$ that has a *succinct description*. We prove this fact in Section 3.4; the proof relies on a concept from classical complexity theory known as *oblivious simulation* of Turing machines. Since the family of verifiers $\left\{\mathcal{V}_{x,N}\right\}$ is uniformly generated, we obtain that the verifiers have a succinct description via a GTM, which in turn allows us to apply the compression theorem as outlined above.

Adapting this sketch to handle languages that are decided by *nondeterministic* Turing machines (as needed in Theorem 1.1), as well as reproving Slofstra's undecidability result (Theorem 1.2), requires additional care. We give details in Section 7.

1.2 Improving the compression theorem?

Theorem 1.3 offers the following tradeoff between "compression in size" and "compression of the gap": the former is scaled by an exponential factor, from polynomial in $N = 2^n$ to polynomial in n, while the latter is divided by a quantity that is polynomial in N, or equivalently, exponential in n.

Surprisingly, we show that *any* better tradeoff, i.e. one in which the gap gets reduced by a subexponential factor in n, would have far-reaching consequences in complexity theory and mathematics. The result provides a possible explanation for the absence of meaningful upper bounds on MIP^* (provided an improved compression result does hold): not only would every computable language be decided by an MIP^* proof system, there would even be *undecidable* languages in MIP^*.

Theorem 1.4 (Consequences of an improved compression theorem)**.** *Suppose an analogue of Theorem 1.3 holds, such that the factor* $\mathrm{poly}(N)$ *in the denominator on the right-hand side of* (1) *is replaced by a subexponential function of* $n = \log N$*. Then*

1. MIP^* *with constant gap contains all computable languages.*
2. MIP^* *with constant gap contains undecidable languages.*
3. *The commuting operator model of multipartite correlations is strictly more powerful than the tensor product model.*

We precisely define what we mean by "improved compression theorem" in Section 8 (see Conjecture 8.1). The idea behind the proof of Theorem 1.4 is that the tradeoff between a subexponential compression in gap and an exponential reduction in size can be "boosted" to a tradeoff where the gap does not get compressed at all, but the game size still gets compressed by a nontrivial amount. This uses *hardness amplification* techniques for multiprover entangled games [BVY17], which employs a variant of parallel repetition to achieve this boosting.

We briefly explain what we mean by the third item in Theorem 1.4, and refer to the end of Section 8.2 for an expanded discussion. In this paper, we define the entangled value of a nonlocal game as the supremum of the success probabilities over all "tensor product" strategies for the provers, which consist of a finite-dimensional Hilbert space for each prover, an entangled state in the tensor product of those Hilbert spaces, and a collection of measurement operators on each prover's space.

There is an alternate definition of the entangled value, which considers the supremum over so-called "commuting operator" strategies, for which there is a single (possibly infinite-dimensional) Hilbert space shared by all players, and the only restriction is that measurement operators applied by distinct provers commute with each other. Since tensor product strategies are also commuting operator strategies, the entangled value in the tensor product model is at most the entangled value in the commuting operator model. It is known that in the finite dimensional case, the two models are equivalent. Whether they coincide in general is a famous problem in quantum information known as "Tsirelson's problem" (see e.g. [Fri12]).

As we explain in Section 8 (and is well known to experts, though we could not find an explicit reference), a positive resolution to Tsirelson's problem implies the existence of an algorithm to approximate the value of any nonlocal game. However, the second item of Theorem 1.4 shows that an improved compression theorem would refute the existence of such an algorithm, and thus would give a negative answer to (the multipartite version of) Tsirelson's problem.

It is known that Tsirelson's problem for two-prover games is essentially equivalent to Connes' Embedding Conjecture [Con76], a longstanding open problem in functional analysis (see [JNP+11, Fri12, Oza13]). In particular, a separation between the definitions of entangled value for games with *two* provers would refute Connes' Embedding Conjecture. We do not know if a separation for games with more than two provers (e.g., 15) would still refute Connes' Embedding Conjecture.

1.3 Related work

We were informed of a forthcoming paper [CS18] by Coudron and Slofstra that establishes a result similar (though strictly incomparable) to Theorem 1.1, using completely different techniques. In particular, the authors show that distinguishing between entangled value 1 or $1 - 1/\operatorname{poly}(t(n))$ for games with *two* provers in the commuting operator model is hard for nondeterministic $t(n)$ time (whereas our result shows hardness for nondeterministic $2^{t(n)}$ time for games with 15 provers in the tensor product model). This result relies on the group-theoretic framework that was pioneered in [Slo16, Slo17].

1.4 Outlook

The most important structural properties of classical multiprover interactive proof systems have been established since the 90s. It is known that any multiprover interactive proof system can be parallelized to a single round of interaction, with two provers only; that completeness 1 can be achieved without loss of generality; that soundness can be amplified in parallel; finally, and most importantly, that the class MIP of languages that can be recognized by any multiprover interactive proof system, for any nontrivial choice of completeness and soundness parameters, is exactly NEXP. Here, by nontrivial we mean any (c, s) such that $\exp(-\operatorname{poly}(n)) \leq s < c \leq 1$, where $c - s$ is at least $\exp(-\operatorname{poly}(n))$. We use $\mathrm{MIP}_{c,s}(k, r)$ to denote the class of languages that can be decided by a polynomial-time verifier interacting with k provers through an r-round interaction, with completeness c and soundness s. Thus, $\mathrm{MIP}_{c,s}(2, 1) = \mathrm{NEXP}$ for all nontrivial values of (c, s). When we write MIP we mean the union of all $\mathrm{MIP}_{c,s}(k, r)$ for polynomially bounded functions k, r, and c, s such that $0 < s < c \leq 1$ and $(c - s)^{-1}$ is polynomially bounded.

In contrast, complexity-theoretic aspects of entangled-prover interactive proof systems remain, to put it mildly, an untamed wilderness. Prior to our work it was known that $\mathrm{NEXP} \subseteq \mathrm{MIP}^*$ [IV12, Vid13, NV17b] with completeness 1 and soundness $\frac{1}{2}$, and that if one allows the completeness-soundness gap to close exponentially fast with n, then the inclusion can be strengthened to NEEXP, or, in our notation, $\mathrm{NTIME}(\Lambda_2(n))$ [Ji17]. Interestingly, a similar phenomenon had previously been observed for single-prover interactive proof systems, for which it is known that QIP = PSPACE with constant gap [JJUW10], but QIP contains EXP if one allows a doubly exponentially small gap [IKW12]. Unlike MIP^*, however, the power of QIP does not grow arbitrarily when the gap goes to zero; for any positive gap the class is contained in EXPSPACE [IKW12].

For the case of multiprover interactive proof systems with entangled provers, there is no compelling reason that a shrinking gap would be necessary for the verification of languages beyond NEXP. Indeed, no upper bounds are known on MIP^* with constant gap — it is not even known to be contained in the set of decidable languages. In fact, recent works provide indication that the class may be larger than NEXP: it is known that $\mathrm{QMA}_{\mathrm{EXP}}$, the "exponential-size proof" analogue of QMA, is such that $\mathrm{QMA}_{\mathrm{EXP}} \subseteq \mathrm{MIP}^*_{1,1-2^{-n}}(5, 1)$ [FV15, Ji16], and inclusion with a constant gap holds under randomized reductions [NV18]. It is therefore an interesting question to determine to what extent the exponentially small completeness-soundness gap that our technique requires is necessary. As mentioned earlier, significant consequences in complexity theory and mathematics would follow from even a small improvement in our compression theorem, Theorem 1.3.

Another major open question on entangled-prover interactive proof systems is the role of the number of provers. Currently, it is not known if e.g. 3 provers allow to determine more languages than 2 (for any setting of the completeness-soundness gap). Our proof of the compression theorem involves a "prover merging" step that reduces the number of provers, albeit for a very restricted type of interactive proof systems. We also note that our techniques restrict us to games with at least 7 provers. This could potentially be decreased to 5, or even 3, by replacing the use of the 7-qubit Steane code with, say, a qutrit error-detecting code. Achieving a result with two provers seems more challenging. Yet, the undecidability results in [Slo17] apply to two-prover games; it would be interesting to investigate whether some improvements on our techniques could take us all the way to hardness results for two-prover games as well.

A number of problems in quantum information theory are known to be undecidable. One that bears superficial similarity with the problem considered in this paper, in the statement as well as in the techniques, is the undecidability of the spectral gap of an infinite translation-invariant Hamiltonian, shown in [CPGW15]. It would be interesting to determine whether there could be a direct reduction from a multiprover game to that problem.

Acknowledgments. Joseph Fitszsimons acknowledges support from Singapore's Ministry of Education and National Research Foundation, and the US Air Force Office of Scientific Research under AOARD grant FA2386-15-1-4082. This material is based on research funded in part by the Singapore National Research Foundation under NRF Award NRF-NRFF2013-01. Thomas Vidick is supported by NSF CAREER Grant CCF-1553477, AFOSR YIP award number FA9550-16-1-0495, a CIFAR Azrieli Global Scholar award, and the IQIM, an NSF Physics Frontiers Center (NSF Grant PHY-1125565) with support of the Gordon and Betty Moore Foundation (GBMF-12500028). Henry Yuen is supported by ARO Grant W911NF-12-1-0541 and NSF Grant CCF-1410022.

Outline. The rest of the paper is organized as follows. We cover preliminaries and definitions in Section 2. In Section 3 we formally define the model of extended nonlocal games and strategies, as well as Gate Turing Machines. In Sections 4, 5, and 6 we prove our compression theorem. In Section 7 we prove Theorem 1.1 and Theorem 1.2. In Section 8 we show that quantitative improvements to our compression theorem would lead to interesting consequences in computational complexity theory and in foundations of quantum mechanics.

2 Preliminaries

Let $\mathbb{Z}$ and $\mathbb{N}$ be the set of integers and the set of natural numbers respectively. We write $\mathrm{poly}(N)$ for any function $f : \mathbb{N} \mapsto \mathbb{R}_+$ such that there is an $\alpha > 0$ and an $N_0 \in \mathbb{N}$ such that $f(N) \leq N^\alpha$ for all $N \geq N_0$. We write $\mathrm{poly}(N; \varepsilon)$ for any function $f : \mathbb{N} \times \mathbb{R}_+ \to \mathbb{R}_+$ such that there exists $\alpha, \beta > 0$ and $N_0 \in \mathbb{N}$, $\varepsilon_0 > 0$ such that, for all $N \geq N_0$ and all $\varepsilon \leq \varepsilon_0$, $f(N, \varepsilon) \leq N^\alpha \varepsilon^\beta$.

2.1 Quantum information theory

All Hilbert spaces considered in the paper are finite dimensional. We use the terminology "quantum register" to name specific quantum systems with finite dimensional Hilbert spaces. We use sans-serif font to denote registers, such as A, B. For example, "register A", to which is implicitly associated the Hilbert space $\mathcal{H}_\mathsf{A}$.

$\mathrm{D}(\mathsf{A})$ denotes the set of density matrices on A, and $\mathrm{L}(\mathsf{A})$ the set of linear operators on A. For a density matrix ρ and an operator M, we use $\mathrm{Tr}_\rho(M)$ to denote $\mathrm{Tr}(\rho M)$. A unitary matrix U is a reflection if it has eigenvalues in $\{\pm 1\}$.

Universal gate set. The quantum circuits we discuss in this paper use single-qubit Hadamard and three-qubit Toffoli gates, a universal gate set for quantum computation [Shi02].

Pauli observables. Let $\sigma_I, \sigma_X, \sigma_Y, \sigma_Z$ denote the four single-qubit Pauli observables

$$\sigma_I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} , \qquad \sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} , \qquad \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} , \qquad \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} .$$

We use two ways of specifying a Pauli observable acting on a specific qubit.

1. Let $W \in \{I, X, Y, Z\}$ be a label and let R be a single-qubit register. We write $\sigma_W(\mathsf{R})$ to denote the observable σ_W acting on R.
2. Let R be an n-qubit register, and let $i \in \{1, \ldots, n\}$. Let $W = X_i$ (resp. $W = Z_i$). We write σ_W to denote the σ_X (resp. σ_Z) operator acting on the i-th qubit in R (the register R is implicit).

We also use W to label Pauli operators that have higher "weight". For example, for $W = X_i Z_j$ the operator σ_W denotes the tensor product $\sigma_{X_i} \otimes \sigma_{Z_j}$. For a vector $u \in \{0,1\}^n$ and $W \in \{X, Z\}$ we write $\sigma_W(u)$ for $\bigotimes_{i:u_i=1} \sigma_{W_i}$.

Lemma 2.1. *Let A, R be registers. Let H be a positive semidefinite matrix acting on A with smallest eigenvalue 0 and second smallest eigenvalue $\Delta > 0$. If $|\psi\rangle$ is a state on AR such that $\langle \psi | H_{\mathsf{A}} \otimes \mathbb{1}_{\mathsf{R}} | \psi \rangle \leq \varepsilon$, then there exists a state $|\theta\rangle$ on AR such that $H|\theta\rangle = 0$ and*

$$\big\| |\psi\rangle\langle\psi| - |\theta\rangle\langle\theta| \big\|_1 \leq 4\sqrt{\varepsilon/\Delta} .$$

Proof. Let P denote the projector onto the kernel of H. Let $Q = \mathbb{1} - P$. Then since $\Delta Q \leq H$ in the positive semidefinite ordering we have $\langle \psi | Q | \psi \rangle \leq \varepsilon/\Delta$. The Gentle Measurement Lemma [ON02] states that for all density matrices ρ and for all positive semidefinite X satisfying $0 \leq X \leq \mathbb{1}$, we have

$$\left\| \rho - \sqrt{X} \rho \sqrt{X} \right\|_1 \leq 2\sqrt{\mathrm{Tr}(\rho(\mathbb{1} - X))} . \tag{3}$$

Setting $\rho = |\psi\rangle\langle\psi|$ and $X = P$ in (3) we obtain the desired conclusion with

$$|\theta\rangle = \frac{P|\psi\rangle}{\sqrt{\langle\psi|P|\psi\rangle}} .$$

□

3 Nonlocal games

In this paper we consider interactive protocols between a quantum verifier V and k quantum provers. We mostly work with a restricted type of three-turn interactive protocols of the following form. First, the provers send a quantum message to the verifier; second, the verifier sends classical questions to the provers; third, the provers reply with classical answers. Following the terminology introduced in [JMVW16] we call such protocols "extended nonlocal games", or ENL. We also consider *nonlocal games*, which are extended nonlocal games in which the first message is trivial (i.e. there is a single round of classical communication, from verifier to provers and back).

This section formally introduces extended nonlocal games, as well as a convenient representation of the verifier for such games as a special kind of Turing machine, called a "gate Turing machine", or GTM.

We start by defining extended nonlocal games (and the special case of nonlocal games) in Section 3.1. In Section 3.2 we recall the definition of the class MIP*. In Section 3.3 we introduce the formalism for representing strategies for the provers in an ENL. In Section 3.4 we introduce a representation of a verifier in an ENL as a Turing machine.

3.1 Extended nonlocal games

Extended nonlocal games are a special kind of three-turn interactive protocol between a quantum verifier and k quantum provers. For simplicity we first introduce notation for the case when there is a single prover P. There are four registers involved: $\mathsf{C}, \mathsf{V}, \mathsf{M}, \mathsf{P}$. The verifier $\mathcal{V}$ acts on registers C (the register containing the prover's initial message), V (the verifier's private space) and M (the message register). The prover P acts on M and P (the prover's private space). The registers V and M are initialized in the $|0\rangle$ state. The registers C and P are initialized in an arbitrary state, chosen by the prover. The verifier applies a circuit C_Q to the three registers CVM (Q stands for "questions"). The prover then applies an arbitrary unitary transformation P to the registers MP. Finally, the verifier applies a circuit C_A to the three registers CVM (A stands for "answers"). The first qubit of V is designated as the "output qubit", and measured in the standard basis to determine whether the verifier accepts or rejects. See Figure 1 for a representation.

We can (and often do) assume without loss of generality that every operation in this protocol, including the prover's, is a reflection, i.e. a Hermitian operator that squares to identity. Indeed, the verifier circuits C_Q, C_A consist of Hadamard gates (H) and Toffoli gate (T), which are reflections. The prover's unitary P can be embedded into a reflection by introducing an ancilla qubit initialized to $|0\rangle$ and considering the reflection $\widetilde{P} = |1\rangle\langle 0| \otimes P + |0\rangle\langle 1| \otimes P^\dagger$.

The extension to k provers is straightforward. The registers M and P are divided into k parts: $\mathsf{M}_1, \dots, \mathsf{M}_k$ and $\mathsf{P}_1, \dots, \mathsf{P}_k$, such that the i-th prover's unitary P_i acts on $\mathsf{M}_i\mathsf{P}_i$.

We say that a verifier $\mathcal{V} = (C_Q, C_A)$ for a k-prover three-turn protocol is a classical-message verifier if there are question and answer alphabets $\mathcal{Q} = \mathcal{Q}_1 \times \cdots \times \mathcal{Q}_k$ and $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_k$ such that

- The only gates of circuit C_Q acting on the message registers M are CNOT gates, controlled on qubits in V. In other words, C_Q copies messages of length $\log |\mathcal{Q}_i|$ from the register V to the register M_i for all i.
- Similarly, the circuit C_A is restricted to classically copying messages of length $\log |\mathcal{A}_i|$ from the register M_i into the register V for all i. (After this, an arbitrary quantum computation can be performed on V only.)

We call such protocols with classical-message verifiers *extended nonlocal (ENL) games*. Note that while the verifier sends and receives classical messages in the register M, it may receive a quantum message in the register C in the first turn. A k-prover *nonlocal game* is a restricted type of ENL game where the verifier ignores the register C.

3.2 The class MIP*

Given a certain class of games, or more generally interactive protocols, it is possible to define an associated class of languages. The most common such class is the class MIP* of languages that can be decided by the verifier in a multiprover interactive proof system in which the verifier is classical and communicates with the provers in a polynomial number of rounds of interaction, using classical messages only. Although we have only formally defined nonlocal games with a single round of interaction, the extension to multiple rounds is straightforward. For more background and definitions of complexity classes associated with quantum interactive proof systems, we refer to the introductory text [Wat09].

Definition 3.1 (MIP*). *Let k, r be polynomially bounded functions of n, and $0 \le s < c \le 1$ computable functions of n. We say that a language L is in $\mathrm{MIP}^*_{c,s}(k, r)$ if there is an efficient classical procedure that on input 1^n returns a family of circuits for a verifier that interacts with k provers in r rounds and is such that*

1. (Completeness:) *If $x \in L$, then there is a strategy for the provers that is accepted with probability at least c;*

2. (Soundness:) *If $x \notin L$, no strategy for the provers has an acceptance probability that is larger than s.*

We write

$$\mathrm{MIP}^*(k,r) = \bigcup_{c\in(0,1],\, g\in\mathrm{poly}} \mathrm{MIP}^*_{c,c-1/g}(k,r) .$$

The following problem is complete, under polynomial time Karp reductions, for the class $\mathrm{MIP}^*_{c,s}(k,1)$: given the description of a verifier $\mathcal{V}$ for a k-prover nonlocal game $\mathcal{G}$, decide whether $\omega^*(\mathcal{G}) \geq c$ or $\omega^*(\mathcal{G}) \leq s$.

3.3 Strategies

The definition of an ENL in Section 3.1 models the action of each prover as a single reflection acting jointly on its message and private registers. We refer to the collection of the provers' shared state $|\psi\rangle_{\mathsf{CPR}}$, where R is a reference register, and each prover's reflection P_i, $i \in \{1,\ldots,k\}$, as a *reflection strategy* $\mathcal{S} = (|\psi\rangle, \{P_i\})$.

Since the message register only contains classical information, it is always possible to represent a prover's reflection as a sequence of three operations: copy the message to the prover's private register; apply an arbitrary reflection on the private register; copy the answer from the private register onto the message register. We call a strategy for the provers that are decomposed in this form a *normal form strategy*. The structure of normal form strategies will be crucial for our compression result later on.

We use the following notation to refer to normal form strategies. Let $\mathcal{V} = (C_Q, C_A)$ be the circuits for the verifier in a k-prover ENL game $\mathcal{G}$. Assume without loss of generality that all question and answer sets $\mathcal{Q}_i$ and $\mathcal{A}_i$ have the same cardinality For $i \in \{1,\ldots,k\}$ and $j \in \{1,\ldots,\log|\mathcal{Q}_k|\}$, let M_{ij} denote the j-th qubit of M_i.

Definition 3.2. *A* normal form ENL game strategy *is a tuple $\mathcal{S} = (\rho, \{Q_{ij}\}, \{P_i\}, \{A_{ij}\})$, where $\{Q_{ij}\}$ is a set of reflections indexed by $i \in \{1,\ldots,k\}$ and $j \in \{1,\ldots,\log|\mathcal{Q}_i|\}$, $\{P_i\}$ is a set of reflections indexed by $i \in \{1,\ldots,k\}$, and $\{A_{ij}\}$ is a set of reflections indexed by $i \in \{1,\ldots,k\}$ and $j \in \{1,\ldots,\log|\mathsf{A}_i|\}$. For all (i,j), the reflections Q_{ij}, P_i, A_{ij} act on P_i.*

The execution of a normal form ENL game strategy $\mathcal{S}$ in the game $\mathcal{G}$ proceeds as follows:

1. The circuit C_A is executed on the registers $\mathsf{C}, \mathsf{V}, \mathsf{M}$.

2. For each $i \in \{1,\ldots,k\}$, the i-th prover applies the sequence of gates $\{\text{CTL-}Q_{ij}\}$ for $j \in \{1,\ldots,\log|\mathcal{Q}_i|\}$, where
$$\text{CTL-}Q_{ij} = |0\rangle\langle 0|_{\mathsf{M}_{ij}} \otimes \mathbb{1}_{\mathsf{P}} + |1\rangle\langle 1|_{\mathsf{M}_{ij}} \otimes Q_{ij} .$$

3. The i-th prover applies a reflection P_i on P_i.

4. For each $i \in \{1,\ldots,k\}$, the i-th prover applies the sequence of gates $\{\text{TGT-}A_{ij}\}$ for $j \in \{1,\ldots,\log|\mathcal{A}_i|\}$, where
$$\text{TGT-}A_{ij} = \mathbb{1}_{\mathsf{M}} \otimes \frac{\mathbb{1} + A_{ij}}{2} + \sigma_X(\mathsf{M}_{ij}) \otimes \frac{\mathbb{1} - A_{ij}}{2} .$$

5. The circuit C_A is executed on the registers $\mathsf{C}, \mathsf{V}, \mathsf{M}$.

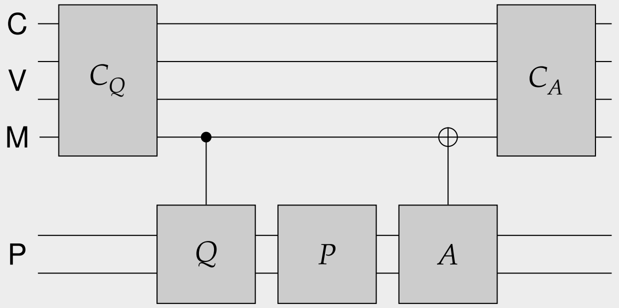

Figure 2: An extended nonlocal game in normal form.

Figure 2 gives a representation for the circuit associated with this protocol. Gates of the form CTL-Q_{ij} and TGT-A_{ij} are referred to as *communication gates*. Gates of the form P_i are referred to as *prover reflection gates*.

It is clear that any strategy for the players in an ENL game can be converted to the normal form: the provers use the gates CTL-Q_{ij} to classically read the message register M one bit at a time, apply an arbitrary measurement, controlled on the copied message, on their private register P_i, and finally use TGT-A_{ij} to classically write their answers into M one bit at a time.

In addition we consider a second type of strategy, called *measurement strategies*, which is the standard type of strategies in the study of nonlocal games. Reflection strategies and measurement strategies in ENL games are easily converted from one to another.

Definition 3.3. *A* measurement strategy $\mathcal{S}$ *for the provers in a k-prover ENL game* $\mathcal{G}$ *with question set* $\mathcal{Q}_1 \times \cdots \times \mathcal{Q}_k$ *and answer set* $\mathcal{A}_1 \times \cdots \times \mathcal{A}_k$ *consists of a pair* $(\rho, \{M_i\})$, *where*

1. ρ *is a state on* $(k+1)$ *registers denoted* $\mathsf{C}, \mathsf{P}_1, \ldots, \mathsf{P}_k$.
2. *For each* $i \in \{1, \ldots, k\}$, M_i *is a map from* $\mathcal{Q}_i \times \mathcal{A}_i$ *to the set of positive semidefinite operators acting on* P_i, *satisfying the constraint that for all* $q \in \mathcal{Q}_i$,

$$\sum_{a \in \mathcal{A}_i} M_i(q,a) = \mathbb{1}_{\mathsf{P}_i} \ .$$

For each $q \in \mathcal{Q}_i$, *we write* $M_i(q) = \{M_i(q,a)\}_a$ *to denote the associated POVM on* P_i.

Next we define the value of a game.

Definition 3.4. *The* value *of a strategy* $\mathcal{S}$ *(either measurement or reflection) in a game* $\mathcal{G}$ *is denoted by* $\omega^*_{\mathcal{S}}(\mathcal{G})$ *and is defined as the probability that players implementing strategy* $\mathcal{S}$ *are accepted by the verifier in* $\mathcal{G}$, *i.e. the probability that a measurement of the verifier's output qubit at the end of the interaction returns the outcome* 1. *The* value *of a game* $\mathcal{G}$ *is denoted by* $\omega^*(\mathcal{G})$ *and is defined as*

$$\omega^*(\mathcal{G}) \ = \ \sup_{\mathcal{S}} \omega^*_{\mathcal{S}}(\mathcal{G}) \ ,$$

where the supremum is over all (finite dimensional) strategies $\mathcal{S}$ *for* $\mathcal{G}$.

Distance between measurement strategies. We define notions of closeness of measurement strategies. (There are analogous notions of closeness of reflection strategies; however we will not need them in this paper).

Definition 3.5 (State-dependent closeness of POVMs). *Let ρ be a density matrix and let $M = \{M^a\}_a, N = \{N^a\}_a$ be two POVMs that have the same set of possible outcomes. Then define*

$$d_\rho(M, N) := \Big[\sum_a \mathrm{Tr}\Big((M^a - N^a)^2 \rho\Big)\Big]^{1/2}. \tag{4}$$

Definition 3.6 (Closeness of strategies). *Let $\mathcal{S} = (\rho, \{M_i\}), \mathcal{S}' = (\rho', \{M_i'\})$ be strategies for an k-prover ENL game $\mathcal{G}$. Then $\mathcal{S}$ is ε-**close** to $\mathcal{S}'$ if and only if*

1. *$\|\rho - \rho'\|_{tr} \le \varepsilon$*
2. *For all $i \in \{1, \ldots, k\}$, $\mathbb{E}_q\, d_\rho(M_i(q), M_i'(q)) \le \varepsilon$, where the expectation is over q drawn from the marginal distribution of the ith prover's questions in the game $\mathcal{G}$.*

Definition 3.7 (Isometric strategies). *Let $\mathcal{S} = (\rho, \{M_i\})$ and $\mathcal{S}' = (\rho', \{M_i'\})$ be strategies for an k-prover ENL game G, where $\rho \in \mathrm{D}(\mathsf{CP}_1 \cdots \mathsf{P}_k)$ and $\rho' \in \mathrm{D}(\mathsf{CP}_1' \cdots \mathsf{P}_k')$. Then $\mathcal{S}$ is ε-**isometric** to $\mathcal{S}'$ if and only if there exist isometries: $V_i : \mathsf{P}_i \to \mathsf{P}_i'$ for each $i \in \{1, \ldots, k\}$ such that the strategy $\widetilde{\mathcal{S}} = (\widetilde{\rho}, \{\widetilde{M}_i\})$ is ε-close to $\mathcal{S}'$, where $\widetilde{\mathcal{S}}$ is defined by*

1. *$\widetilde{\rho} = (V_1 \otimes \cdots \otimes V_k)\rho(V_1 \otimes \cdots \otimes V_k)^\dagger$*
2. *For all i, for all $(q, a) \in \mathcal{Q}_i \times \mathcal{A}_i$, $\widetilde{M}_i(q, a) = V_i\, M_i(q, a)\, V_i^\dagger$.*

The following lemma shows that if strategy $\mathcal{S}_1$ in a k-prover ENL game G is ε-isometric to $\mathcal{S}_2$, then their success probabilities differ by at most $O(k\varepsilon)$.

Lemma 3.8. *Let $\mathcal{V} = (C_Q, C_A)$ be a verifier in an ENL game $\mathcal{G}$, and let $\mathcal{S}' = (\rho_1, \{P_i^{(1)}\}), \mathcal{S} = (\rho_2, \{P_i^{(2)}\})$ be strategies for $\mathcal{G}$ such that $\mathcal{S}$ is ε-isometric to $\mathcal{S}'$. Then*

$$\left|\omega^*_{\mathcal{S}}(\mathcal{G}) - \omega^*_{\mathcal{S}'}(\mathcal{G})\right| \le O(k\varepsilon)\ .$$

Proof. Observe that $\omega^*_{\mathcal{S}}(G) = \omega^*_{\widetilde{\mathcal{S}}}(G)$ where $\widetilde{\mathcal{S}}$ is the strategy that is ε-close to $\mathcal{S}'$ as given by the definition of isometric strategies. Let $\mathcal{S}''$ denote the strategy that is the same as $\widetilde{\mathcal{S}}$ except the shared state ρ'' is taken to be the shared state ρ' of $\mathcal{S}'$. We have that $\left|\omega^*_{\mathcal{S}''}(G) - \omega^*_{\mathcal{S}}(G)\right| \le \varepsilon$.

Consider a sequence of $(k+1)$ hybrid strategies $\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_k$ where $\mathcal{S}_0 = \mathcal{S}''$ and $\mathcal{S}_k = \mathcal{S}'$, and strategies $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$ differ in that the i-th prover's measurement operators are switched from those of $\mathcal{S}''$ to those of $\mathcal{S}'$. Lemma 7 of [Ji17] implies that $\left|\omega^*_{\mathcal{S}_i}(G) - \omega^*_{\mathcal{S}_{i+1}}(G)\right| \le \varepsilon$. We thus obtain the statement of the lemma. □

Protocol circuits. A protocol circuit is a quantum circuit description of a normal form strategy in an extended nonlocal game (see Figure 2 for an example). Formally, a k-prover protocol circuit C is specified by a set of s *verifier wires*, a set of k *prover wires*, and a finite sequence of gates $g_1, g_2, \ldots, g_\tau$. Every gate g has a *type*, denoted by $\mathsf{type}(g)$:

1. H, which stands for a *double Hadamard* gate[7]

[7] A double Hadamard gate is simply a two-qubit gate that applies two Hadamard gates in parallel. We use this gate for technical reasons.

2. T, which stands for a Toffoli gate
3. Q, which stands for a gate of the form CTL-Q_{ij}, for an arbitrary reflection Q_{ij} acting on P_i.
4. A, which stands for a gate of the form TGT-A_{ij}, for an arbitrary reflection A_{ij} acting on P_i.
5. P, which stands for a prover reflection P_i acting on P_i.

The *wires* of a gate g, denoted by $\mathsf{wire}(g)$, is the set of wires it acts on. Each gate acts on up to 3 wires. The size of a k-prover protocol circuit with τ gates and $s + k$ wires is defined to be $(\tau + s + k)$.

It is easy to see that, from the protocol circuit C of a game, we can extract the circuits C_Q and C_A defining the verifier $\mathcal{V}$ of the game. We may use protocol circuits C and the corresponding verifier $\mathcal{V}$ interchangeably.

3.4 Turing machine descriptions of verifier circuits

In this section, we discuss Turing machine descriptions of verifier circuits.

Definition 3.9. *Let Λ denote a countable set. A family of verifier circuits $\{\mathcal{V}_{n,\lambda}\}_{n\in\mathbb{N},\lambda\in\Lambda}$ is* uniformly generated *if there is a deterministic Turing machine M that on input $(1^n, \lambda)$ runs in polynomial time and returns a description of $\mathcal{V}_{n,\lambda}$.*

Remark. In the usual definition of uniformly generated circuits, the circuits are only parameterized by an integer n that denotes the size. In our definition, the verifier circuits are parameterized by both a size parameter n as well as an auxiliary symbol λ; this generalization will be useful in our proof of the compression theorem. Alternatively, one can think of a family of verifier circuits $\{\mathcal{V}_{n,\lambda}\}_{n,\lambda}$ as specifying, for each fixed $\lambda \in \Lambda$, a family of uniformly generated verifiers circuits $\{\mathcal{V}_{n,\lambda}\}_{n\in\mathbb{N}}$ (in the standard sense). Furthermore, there is a *single* Turing machine M, that by fixing the input λ, generates each family $\{\mathcal{V}_{n,\lambda}\}_{n\in\mathbb{N}}$.

For our compression result it is not enough for verifiers to have uniform Turing machine descriptions — it is crucial that they also have highly *succinct* descriptions, defined as follows.

Definition 3.10. *A family of verifier circuits $\{\mathcal{V}_{n,\lambda}\}$ has a* succinct description *if there exists a deterministic Turing machine G, called the* Gate Turing Machine (GTM) for the protocol circuits $\{C_{n,\lambda}\}$ specified by $\{\mathcal{V}_{n,\lambda}\}$ *if on input (n, t, λ) the Turing machine G runs in polynomial time and returns the description of the t-th gate g of $C_{n,\lambda}$ (and a special error symbol in case t is larger than the size of $C_{n,\lambda}$). In addition, we assume that a GTM always returns the size $p_\lambda(n)$ of the protocol circuit $C_{n,\lambda}$ it specifies when provided the input* $(1^n, -1, \lambda)$.

In the definition, by "description" of a gate we mean the pair $(\mathsf{type}(g), \mathsf{wire}(g))$.

We use $\mathrm{CKT}(G, n)$ to denote the protocol circuit whose gates are specified by G on input (n, t) for $1 \le t \le p(n)$. We call the circuit $\mathrm{CKT}(G, n)$ the n-th protocol circuit specified by G, and the game $\mathcal{G}_n$ corresponding to $\mathrm{CKT}(G, n)$ the n-th game specified by G. We say that G is a GTM for a family of ENL games $\{\mathcal{G}_n\}$ if $\mathcal{G}_n$ is the n-th game specified by G.

The following lemma shows that if a verifier family $\{\mathcal{V}_n\}$ is uniformly generated, then there is an *equivalent* verifier family $\left\{\mathcal{V}'_n\right\}$ that has a succinct description. Here, we use a strong notion of equivalence: the question and answer alphabets of $\mathcal{V}'_n$ are the same as $\mathcal{V}_n$, and furthermore, the value of any strategy $\mathcal{S}$ is the same in $\mathcal{G}'_n$ and $\mathcal{G}_n$.

Lemma 3.11. *Let $k \geq 0$ be an integer. Let $\{\mathcal{V}_{n,\lambda}\} = \{(C_{Q,n,\lambda}, C_{A,n,\lambda})\}$ be a family of verifier circuits for a k-prover ENL game that is uniformly generated by a Turing machine M. Here λ denotes an auxiliary string that is part of the input $(1^n, \lambda)$ to M. Let $\mathcal{G}_{n,\lambda}$ denote the ENL game associated with $\mathcal{V}_{n,\lambda}$. Then there exists a GTM G_M, that is computable from M, such that the n-th game specified by G_M is $\mathcal{G}'_{n,\lambda}$ such that:*

1. *The question and answer alphabets of the verifier of $\mathcal{G}'_{n,\lambda}$ are the same as in $\mathcal{G}_{n,\lambda}$;*
2. *For all n and for all ENL game strategies $\mathcal{S}$, $\omega^*_{\mathcal{S}}(\mathcal{G}'_{n,\lambda}) = \omega^*_{\mathcal{S}}(\mathcal{G}_{n,\lambda})$.*

Proof. From the Turing machine M it is possible to design two Turing machines M_Q and M_A that specify the families of circuits $\{C_{Q,n,\lambda}\}$ and $\{C_{A,n,\lambda}\}$. As shown in Lemma A.5 in Appendix 3.4, any uniformly generated family of circuits has a succinct representation of the form described in Definition 3.10. Let G_Q and G_A be the associated GTMs. The GTM G_M is a straightforward combination of G_Q and G_A. On input (n, t, λ), the GTM first determines if the time t corresponds to a gate in C_Q, or is among the CTL-Q_{ij}, P_i or TGT-A_{ij} gates, or a gate in C_A (recall the notation for normal form verifiers introduced in Section 3.3). This can be determined in polynomial time as each part has an easily computable size. If t belongs to the first or last part, G_M determines the appropriate gate by executing G_Q or G_A respectively. In the remaining cases, the correct communication gate or prover reflection gate can easily be computed in polynomial time. □

4 Honest Pauli Prover games

As mentioned in the introduction, we prove Theorem 1.3 in two parts: first we show how to compress a family of k-prover ENL games $\{\mathcal{G}_N\}$ specified by a GTM G to a family of $(k+1)$-prover *Honest Pauli Prover* games $\left\{\mathcal{G}^\sharp_{H,n}\right\}$, in which one of the provers is a specially designated "Honest Pauli Prover" who is "commanded" to measure multi-qubit Pauli observables. We describe Honest Pauli Prover games in this section. In Section 5 we show how to simulate an Honest Pauli Prover game $\mathcal{G}^\sharp_{H,n}$ with a k-prover ENL game $\left\{\mathcal{G}^\sharp_n\right\}$. In Section 6 we put the two parts together to prove Theorem 1.3.

Throughout this section, we fix a GTM G for a family of k-prover ENL games $\{\mathcal{G}_n\}$. We write $\mathrm{CKT}(G, n)$ for the n-th protocol circuit specified by G, and let $p(n)$ denote the size of $\mathrm{CKT}(G, n)$. When n is fixed we let $N = 2^n$ and write $\widehat{\mathsf{C}}, \widehat{\mathsf{V}}, \widehat{\mathsf{M}}$ for the registers that the verifier $\mathcal{V}_N$ in $\mathcal{G}_N$ acts on, and $\widehat{\mathsf{X}} = \widehat{\mathsf{CVM}}$ for the union of these registers. We interpret $\widehat{\mathsf{X}}$ as an ordered sequence of single-qubit registers $\{\widehat{\mathsf{C}}_i\}$, $\{\widehat{\mathsf{V}}_i\}$, and $\{\widehat{\mathsf{M}}_i\}$. For any register R_i of this form, we write $\mathrm{ind}(\mathsf{R}_i) \in \{1, \ldots, |\widehat{\mathsf{X}}|\}$ for the qubit of $\widehat{\mathsf{X}}$ that R_i corresponds to.

In this section we introduce a family of games $\left\{\mathcal{G}^\sharp_{H,n}\right\}$ that is designed to force the provers to hold a history state of the protocol circuit $\mathrm{CKT}(G, N)$. (The $\sharp$ superscript in $\mathcal{G}^\sharp_{H,n}$ indicates that the game is a compression of $\mathcal{G}_N$.) These games fall in a category of *Honest Pauli Prover* games, defined as follows.

Definition 4.1 (Honest Pauli Prover game)**.** *Let $k, S \geq 1$ be integer. An extended nonlocal game $\mathcal{G}$ is an $(k+1)$-prover S-qubit* Honest Pauli Prover game *if the following holds. The game has $(k+1)$ provers, the first of which is labelled PV and called the "Pauli prover", and the remaining k are labelled $PP_1, \ldots, PP_k$. In the game, queries take the form $Q = (q_V, q_{P,1}, \ldots, q_{P,k})$, where the question q_V to the Pauli prover is a set of labels $\{\mathrm{W}^{(j)}\}$ for up to three commuting S-qubit Pauli observables, each of which acts nontrivially on at most two qubits. Answers in the game are labeled $a_V, a_{P,1}, \ldots, a_{P,k}$, respectively.*

We introduce a class of strategies for Honest Pauli Prover games in which the Pauli prover performs Pauli operations honestly.

Definition 4.2 (Honest Pauli Prover strategy)**.** *For $k \geq 0$ we say that a $(k+1)$-prover measurement strategy $(|\psi\rangle, \{M_i\})$ for an Honest Pauli Prover Game $\mathcal{G}_H$ is an S-qubit* honest Pauli Prover strategy *(or honest Pauli strategy for short) if the following holds. The state $|\psi\rangle$ is on $(k+3)$ registers: C (held by the the verifier), P_V (held by the prover PV), $\mathsf{P}_{P,1}, \ldots, \mathsf{P}_{P,k}$ (held by provers $PP_1, \ldots, PP_k$ respectively), and R (a reference register). We use P to denote the $(k+1)$ prover registers collectively. Furthermore, the register P_V consists of S qubits, and on any question q_V the answer bits a_V returned by the Pauli prover are obtained by measuring the set of commuting Pauli observables that is specified by its question (the prover reports one answer bit for each observable).*

The verifier $\mathcal{V}^{\sharp}_{H,n}$ for the game $\mathcal{G}^{\sharp}_{H,n}$ is summarized in Figure 3. The verifier randomly executes one of three possible routines. We give the description of each subprotocol in Section 4.1, Section 4.2 and Section 4.3 respectively. We conclude with the analysis of $\mathcal{V}^{\sharp}_{H,n}$ in Section 4.4.

Verifier name: $\mathcal{V}^{\sharp}_{H,n}$:

- Execute each of the following subprotocols with probability 1/3: Gate Check(n), Input Check(n), and Output Check(n).

Figure 3: The verifier $\mathcal{V}^{\sharp}_{H,n}$.

4.1 Gate Check

The goal of the Gate Check subprotocol is to check that the provers (already assumed to be using an honest Pauli strategy) share a state close to a history state corresponding to the execution of the protocol circuit CKT(G, N). More precisely, their strategy must be close to one of the following form.

Definition 4.3 (Honest Gate Check strategy)**.** *An honest Pauli strategy $\mathcal{S} = (|\psi\rangle, \{M_i\})$ is an* honest Gate Check strategy *for the game $\mathcal{G}^{\sharp}_{H,n}$ derived from the GTM G if the shared state $|\psi\rangle_{\mathsf{CPR}}$ is a history state of the circuit* CKT(G, N)*,*

$$|\psi\rangle_{\mathsf{CPR}} = \frac{1}{\sqrt{p(N)+1}} \sum_{t=0}^{p(N)} |t\rangle_{\mathsf{C}} \otimes |\psi_t\rangle_{\mathsf{PR}} , \tag{5}$$

where the state $|\psi_0\rangle_{\mathsf{PR}}$ is arbitrary and for all $t \geq 1$, the state $|\psi_t\rangle_{\mathsf{PR}}$ is defined as $U_{g_t}|\psi_{t-1}\rangle_{\mathsf{PR}}$ where $g_t = G(N,t)$ and U_{g_t} is the unitary specified in (6), acting on the registers specified by $\mathsf{wire}(g_t)$*. In particular, the register P_V is isomorphic to $\widehat{\mathsf{X}} = \widehat{\mathsf{CVM}}$, and $S = |\widehat{\mathsf{X}}|$.*

We proceed to describe the Gate Check, and then state its properties. In the check, the verifier samples a random time $t \in \{1, \ldots, p(N)\}$, and computes the t-th gate $g = G(N,t)$ (the verifier can compute this gate by simulating the Turing machine G for poly $\log(N)$ steps). Depending on the type of g, a double Hadamard gate, a Toffoli gate, a communication channel gate (see Section 3.3), or a prover reflection gate, the verifier executes a specially tailored subprotocol to check the propagation of that particular gate.

Subprotocol name: Gate Check(n):

1. Select a uniformly random integer $t \in \{1, \ldots, p(N)\}$, and measure the clock register C using the POVM
$$\left\{\Pi^0 = |+_t\rangle\langle +_t|, \Pi^1 = |-_t\rangle\langle -_t|, \Pi^2 = \mathbb{1} - \Pi^0 - \Pi^1\right\},$$
where $|\pm_t\rangle = \frac{1}{\sqrt{2}}\big(|t-1\rangle \pm |t\rangle\big)$. Let $s \in \{0,1,2\}$ denote the result of the measurement. If $s = 2$, accept.
2. Simulate the execution of the the GTM G on input (N,t) to obtain $g = G(N,t)$.
3. If $\mathsf{type}(g) = T$, run Toffoli Check(n, s, g).
4. If $\mathsf{type}(g) = H$, run Hadamard Check(n, s, g).
5. If $\mathsf{type}(g) \in \{Q, A\}$, run Communication Channel Check(n, s, g).
6. If $\mathsf{type}(g) = P$, run Prover Reflection Check(n, s, g).

Figure 4: Gate Check

Figure 5 details the subprotocols invoked by Gate Check. The subprotocols Toffoli Check and Hadamard Check are taken from [Ji17]. A Toffoli or doubled Hadamard gate g returned by the GTM G always comes together with labels for a set of qubits on which the gate acts on. In the subprotocols Hadamard Check, Communication Channel Check, and Prover Reflection Check, the verifier artificially accepts with probability 1/2 without testing anything; this is to adjust the normalization of the rejection probabilities of these subprotocols.

The next lemma establishes an expression for the rejection probability for Gate Check conditioned on a choice of random $t \in \{1, \ldots, p(N)\}$.

Lemma 4.4. *Let $\mathcal{S} = (|\psi\rangle, \{M_i\})$ be an honest Pauli strategy for the* Gate Check *subprotocol. For all $i \in \{1, \ldots, k\}$ let Q_{ij}, A_{ij}, P_i be prover PP_i's observables on questions $Q_{ij}, A_{ij}, \star$ respectively. Let CTL-Q_{ij} and TGT-A_{ij} denote the associated controlled operators defined in Section 3.3.*

Fix $t \in \{1, \ldots, p(N)\}$. Let $g = G(N,t)$ denote the t-th gate of the protocol circuit CKT(G, N). *Let*

$$U_g = \begin{cases} H^{\otimes 2} & \text{if } \mathsf{type}(g) = H \\ T & \text{if } \mathsf{type}(g) = T \\ \mathit{CTL}\text{-}Q_{ij} & \text{if } \mathsf{type}(g) = Q, \mathsf{wire}(g) = (i,j) \\ \mathit{TGT}\text{-}A_{ij} & \text{if } \mathsf{type}(g) = A, \mathsf{wire}(g) = (i,j) \\ P_i & \text{if } \mathsf{type}(g) = P, \mathsf{wire}(g) = i. \end{cases} \tag{6}$$

Then the rejection probability of Gate Check, *conditioned on the verifier selecting time $t \in \{0, 1, \ldots, p(N)\}$ in Step 1 of Figure 4, is*
$$\frac{1}{4}\operatorname{Tr}_\rho\left(K_t(\mathbb{1} - J_t \otimes U_g)K_t\right),$$
where $\rho = |\psi\rangle\langle\psi|$, K_t denotes the projector $|+_t\rangle\langle +_t| + |-_t\rangle\langle -_t|$ acting on C and J_t denotes the unitary operator $\mathbb{1} - 2|-_t\rangle\langle -_t|$ acting on C.

Subprotocol name: Toffoli Check(n, s, g):
Description of input: g is a Toffoli gate acting on qubits u_1, u_2, u_3, and $s \in \{0, 1\}$.

1. Sample $\alpha \in \{0, 1\}$ uniformly at random, and accept if $\alpha = 1$. Otherwise, continue.
2. Set $q_V = (Z_{u_1}, Z_{u_2}, X_{u_3})$. Let $a_V = (a_1, a_2, a_3)$ be the three answer bits from P_V. Reject if $a_1 = a_2 = 1 \wedge s \oplus a_3 = 1$, or $a_1 a_2 = 0 \wedge s = 1$. Accept otherwise.

Subprotocol name: Hadamard Check(n, s, g):
Description of input: g is a double Hadamard gate acting on qubits u_1, u_2, and $s \in \{0, 1\}$.

1. Sample $\alpha \in \{0, 1\}$ uniformly at random.
2. If $\alpha = 0$, set $q_V = (X_{u_1} X_{u_2}, Z_{u_1} Z_{u_2})$. Let a_1, a_2 be the two answer bits from P_V. Reject if $s \oplus a_1 = s \oplus a_2 = 1$, accept otherwise.
3. If $\alpha = 1$, set $q_V = (X_{u_1} Z_{u_2}, Z_{u_1} X_{u_2})$. Let a_1, a_2 be the two answer bits from P_V. Reject if $s \oplus a_1 = s \oplus a_2 = 1$ and accept otherwise.

Subprotocol name: Communication Channel Check(n, s, g):
Description of input: g is a communication gate CTL-Q_{ij} or TGT-A_{ij}, and $s \in \{0, 1\}$.

1. Sample $\alpha \in \{0, 1\}$ uniformly at random, and accept if $\alpha = 1$. Otherwise, continue.
2. Let $(i, j) = \mathsf{wire}(g)$. Let $u = \mathrm{ind}(\widehat{\mathsf{M}}_{ij})$.
3. If $\mathsf{type}(g) = Q$: Set $q_V = Z_u$. Set $q_{P,i} = Q_{ij}$. Reject if $a_V = 1 \wedge s \oplus a_{P,i} = 1$, or $a_V = 0 \wedge s = 1$. Accept otherwise.
4. If $\mathsf{type}(g) = A$: Set $q_V = X_u$. Set $q_{P,i} = A_{ij}$. Reject if $a_{P,i} = 1 \wedge s \oplus a_V = 1$, or $a_{P,i} = 0 \wedge s = 1$. Accept otherwise.

Subprotocol name: Prover Reflection Check(n, s, g):
Description of input: g is a prover reflection gate, and $s \in \{0, 1\}$.

1. Sample $\alpha \in \{0, 1\}$ uniformly at random, and accept if $\alpha = 1$. Otherwise, continue.
2. Let $i = \mathsf{wire}(g)$. Set $q_{P,i} = \star$.
3. Reject if $a_{P,i} \neq s$. Accept otherwise.

Figure 5: Toffoli, Hadamard, Communication Channel, and Prover Reflection Checks.

Proof. The rejection probability for the double Hadamard and Toffoli gates was established in [Ji17]. In the case of $\mathsf{type}(g) = Q$, the rejection probability is

$$\frac{1}{2}\,\mathrm{Tr}_\rho\left(K_t\left[|-_t\rangle\langle-_t| \otimes \frac{\mathbb{1} + \sigma_{Z_u}}{2} + \frac{\mathbb{1} - J_t \otimes Q_{ij}}{2} \otimes \frac{\mathbb{1} - \sigma_{Z_u}}{2}\right]K_t\right)$$

which can be verified to be equal to $\frac{1}{4}\operatorname{Tr}_\rho(K_t(\mathbb{1}-J_t\otimes U_g)K_t)$. In the case that $\mathsf{type}(g)=A$, the rejection probability is

$$\frac{1}{2}\operatorname{Tr}_\rho\left(K_t\left[|-_t\rangle\langle-_t|\otimes\frac{\mathbb{1}+A_{ij}}{2}+\frac{\mathbb{1}-J_t\otimes\sigma_{X_u}}{2}\otimes\frac{\mathbb{1}-A_{ij}}{2}\right]K_t\right)$$

which again can be verified to be equal to $\frac{1}{4}\operatorname{Tr}_\rho(K_t(\mathbb{1}-J_t\otimes U_g)K_t)$. In the case of $\mathsf{type}(g)=P$, the rejection probability is by definition

$$\frac{1}{4}\operatorname{Tr}_\rho(K_t(\mathbb{1}-J_t\otimes U_g)K_t).$$

□

Lemma 4.5. *The following hold for the* Gate check *subprotocol described in Figure 5:*

1. *(Completeness) An honest Gate Check strategy passes the* Gate check *subprotocol with probability* 1.
2. *(Soundness) Any honest Pauli strategy that passes the* Gate check *subprotocol with probability at least* $1-\varepsilon$ *is* δ*-close (see Definition 3.6) to an honest Gate Check strategy, for* $\delta=O(p(N)^{3/2}\sqrt{\varepsilon})$

Proof. Completeness is straightforward. We show soundness. The analysis largely follows [Ji17]. Let $\mathcal{S}$ be an honest Pauli strategy that succeeds with probability at least $1-\varepsilon$ in the Gate check subprotocol. Let $|\psi\rangle_{\mathsf{CPR}}$ denote the provers' shared state in $\mathcal{S}$, and let $\rho=|\psi\rangle\langle\psi|$.

We calculate the rejection probability of Gate Check. At step 1. in Gate Check the verifier selects a time t uniformly at random from $\{1,\ldots,p(N)\}$. Let $g_t=G(N,t)$ denote the t-th gate of CKT(G,N). Let r_t denote the rejection probability of Gate Check conditioned on time t having been selected. By Lemma 4.4, the rejection probability is $r_t=\frac{1}{4}\operatorname{Tr}_\rho\left(K_t(\mathbb{1}-J_t\otimes U_{g_t})K_t\right)$. Thus the overall rejection probability satisfies

$$\begin{aligned}\varepsilon&\geq\mathop{\mathbb{E}}_t r_t\\&\geq\frac{1}{4}\mathop{\mathbb{E}}_t\operatorname{Tr}_\rho\left(K_t(\mathbb{1}-J_t\otimes U_{g_t})K_t\right)\\&=\frac{1}{4}\mathop{\mathbb{E}}_t\operatorname{Tr}_\rho\left(|t-1\rangle\langle t-1|\otimes\mathbb{1}+|t\rangle\langle t|\otimes\mathbb{1}-|t-1\rangle\langle t|\otimes U_{g_t}^\dagger-|t\rangle\langle t-1|\otimes U_{g_t}\right)\end{aligned}\tag{7}$$

where in the last equality we used the fact that $U_{g_t}^\dagger=U_{g_t}$. Define $Q=\sum_t|t\rangle\langle t|_{\mathsf{C}}\otimes U_{g_t}\cdots U_{g_1}$. It is straightforward to verify that (7) implies

$$\operatorname{Tr}_\rho\mathop{\mathbb{E}}_t\left(Q|-_t\rangle\langle-_t|Q^\dagger\right)\leq2\varepsilon\ .$$

Let H_{prop} denote the operator $\sum_t\left(Q|-_t\rangle\langle-_t|Q^\dagger\right)$. Notice that H_{prop} is a positive semidefinite operator that is exactly the same as the propagation term of the Feynman-Kitaev clock Hamiltonian [KSV02]. It has been shown that this propagation term has a spectral gap of at least $\Omega(1/p(N)^2)$ [AVDK+08], and therefore the scaled operator $\mathbb{E}_t\left(Q|-_t\rangle\langle-_t|Q^\dagger\right)$ has spectral gap of at least $\Omega(1/p(N)^3)$. Using Lemma 2.1, we have that ρ is δ-close to a pure state $|\theta\rangle\langle\theta|$ satisfying $H_{prop}|\theta\rangle=0$ for $\delta=O(p(N)^{3/2}\sqrt{\varepsilon})$. Since the ground space of the propagation term of is spanned by history states of the form $|\theta\rangle_{\mathsf{CPR}}=\frac{1}{\sqrt{p(N)+1}}\sum_t|t\rangle_{\mathsf{C}}\otimes|\theta_t\rangle_{\mathsf{PR}}$ where $|\theta_t\rangle=U_{g_t}|\theta_{t-1}\rangle$, this establishes the lemma.

□

4.2 Input check

Assume that the provers' strategy is an honest Gate Check strategy (Definition 4.3). The Input Check subprotocol is designed to check that the component $|\psi_0\rangle_{\mathsf{PR}}$ of the history state (5) at time $t = 0$ is a valid initial state for the protocol circuit.

Definition 4.6 (Honest Input Check strategy)**.** *An honest Gate Check strategy $\mathcal{S} = (|\psi\rangle, \{M_i\})$ is an* honest Input Check strategy *if the initial state $|\psi_0\rangle_{\mathsf{PR}}$ is such that the registers $\widehat{\mathsf{VM}}$ of P_V are initialized to the all zero state.*

Subprotocol name: Input Check(n):

1. Measure the clock register C in the computational basis. Let $t \in \{0, \ldots, p(N)\}$ be the outcome. If $t \neq 0$, accept.
2. Pick a random qubit index $j \in \mathrm{supp}(\widehat{\mathsf{VM}})$.
3. Set $q_V = Z_j$. Accept if $a_V = 0$. Otherwise, reject.

Figure 6: Input Check.

Lemma 4.7. *The following hold for the* Input check *subprotocol described in Figure 6:*

1. *(Completeness) An honest Input Check strategy passes the* Input Check *subprotocol with probability* 1.
2. *(Soundness) Any honest Gate Check strategy that passes the* Input Check *subprotocol with probability at least $1 - \varepsilon$ is δ-close to an Honest Input Check strategy for $\delta = O(p(N)\sqrt{\varepsilon})$.*

Proof. Completeness is straightforward. We show soundness. Let $\mathcal{S}$ be a strategy that passes the Input Check subprotocol with probability at least $1 - \varepsilon$. Let $|\psi\rangle_{\mathsf{CPR}}$ denote the shared state in $\mathcal{S}$. Since the strategy $\mathcal{S}$ is an honest Gate Check strategy (and therefore an honest Pauli Check strategy), we have that

$$|\psi\rangle_{\mathsf{CPR}} = \frac{1}{\sqrt{p(N)+1}} \sum_{t=0}^{p(N)} |t\rangle_{\mathsf{C}} \otimes |\psi_t\rangle_{\mathsf{PR}} \,.$$

Let $\Pi = |0\rangle\langle 0|_{\mathsf{C}}$, and let $\rho = |\psi\rangle\langle\psi|$. We have that $\mathrm{Tr}_\rho(\Pi) \geq (p(N)+1)^{-1}$. Let

$$\rho_0 = \frac{\Pi\rho\Pi}{\mathrm{Tr}_\rho(\Pi)} = |0\rangle\langle 0|_{\mathsf{C}} \otimes |\psi_0\rangle\langle\psi_0|_{\mathsf{PR}} \,.$$

The probability that Input Check rejects when the shared state is ρ_0 instead of ρ is at most $\varepsilon' = (p(N)+1)\varepsilon$.

Suppose now that the shared state in Input Check is ρ_0. The probability of rejection is then

$$\mathrm{Tr}_{\rho_0}(H_{init}) \leq \varepsilon' \,, \tag{8}$$

where

$$H_{init} = \frac{1}{|\widehat{\mathsf{VM}}|} \sum_{i \in \mathrm{supp}(\widehat{\mathsf{VM}})} |1\rangle\langle 1|_i \,,$$

with $|\widetilde{\mathsf{VM}}| \le p(N)$ the number of qubits in register $\widetilde{\mathsf{VM}}$.

Observe that the operator H_{init} is positive semidefinite, has smallest eigenvalue 0, and has spectral gap of at least $1/p(N)$. Furthermore, the kernel of H_{init} is spanned by states of the form $|\theta\rangle_{\mathsf{PR}}$ where the register $\widetilde{\mathsf{VM}}$ is in the all zeroes state. Using Lemma 2.1, we conclude that $|\psi_0\rangle$ is δ-close to such a state $|\theta\rangle_{\mathsf{PR}}$ for $\delta = O(p(N)\sqrt{\varepsilon})$. This concludes the proof.

□

4.3 Output check

As for the Input check, assume that the provers share a valid history state of the protocol circuit $\mathrm{CKT}(G, N)$. The Output Check subprotocol is designed to check that the state held by the provers is a history state of an accepting computation. In other words, the Output Check subprotocol enforces that the output qubit of the last time step of the history state is in the state $|1\rangle$.

Subprotocol name: Output Check(n):

1. Measure the clock register C in the computational basis. Let $t \in \{0, \ldots, p(N)\}$ be the outcome. If $t \neq p(N)$, accept.
2. Let u denote the index of the decision bit in $\widehat{\mathsf{V}}$.
3. Set $q_V = Z_u$. If $a_V = 0$, reject. Otherwise, accept.

Figure 7: Output Check

Lemma 4.8. *The following hold for the* Output check *subprotocol described in Figure 7:*

1. *(Completeness) For all $\gamma > 0$ there exists an honest Input Check strategy that passes the* Output Check *subprotocol with probability*
$$1 - \frac{1 - \omega^*(\mathcal{G}_N) + \gamma}{p(N) + 1}.$$
2. *(Soundness) Any honest Input Check strategy passes the* Output Check *subprotocol with probability at most*
$$1 - \frac{1 - \omega^*(\mathcal{G}_N)}{p(N) + 1}\ .$$

Proof. We show the Completeness part. Consider a normal form k-prover strategy $\mathcal{T}$ for $\mathcal{G}_N$ that achieves the value at least $\omega^*(\mathcal{G}_N) - \gamma$ (there isn't necessarily a strategy that achieves the optimal value $\omega^*(\mathcal{G}_N)$). The strategy $\mathcal{T}$ is comprised of a shared state $|\varphi\rangle$ on register $\widetilde{\mathsf{CP}}$ and reflections $\{A_{ij}\}$, $\{Q_{ij}\}$, and $\{P_i\}$ as described in Section 3.3.

Consider the following Honest Input Check strategy $\mathcal{S}$: the shared state $|\psi\rangle$ is the history state of the protocol circuit $\mathrm{CKT}(G, N)$ where the provers' reflections $\{A_{ij}\}$, $\{Q_{ij}\}$, and $\{P_i\}$ are given by the strategy $\mathcal{T}$. Since the strategy $\mathcal{T}$ succeeds in $\mathcal{G}_N$ with probability at least $\omega^*(\mathcal{G}_N) - \gamma$, strategy $\mathcal{S}$ succeeds in Output Check with the claimed probability.

We now show soundness. Let $\mathcal{S}$ be an Honest Input Check strategy that passes the Output Check subprotocol with probability at least $1-\varepsilon$. Let $|\psi\rangle_{\mathsf{CPR}}$ denote the shared state. Since the strategy is an Honest Input Check strategy, the shared state is a history state of the protocol circuit C

$$|\psi\rangle_{\mathsf{CPR}} = \frac{1}{\sqrt{p(N)+1}} \sum_{t=0}^{p(N)} |t\rangle_{\mathsf{C}} \otimes |\psi_t\rangle_{\mathsf{PR}} ,$$

with the initial snapshot state $|\psi_0\rangle$ representing the state of the verifier and provers at the start of an execution of the game $\mathcal{G}_N$. Let $\rho = |\psi\rangle\langle\psi|$. Let $\Pi = |N\rangle\langle N|_{\mathsf{C}}$. We have that $\mathrm{Tr}_\rho(\Pi) = 1/(p(N)+1)$. Let

$$\rho_f = \frac{\Pi\sigma\Pi}{\mathrm{Tr}_\rho(\Pi)} = |\psi_N\rangle\langle\psi_N|.$$

The probability that Output Check rejects when the shared state ρ_f is at most $\varepsilon' = (p(N)+1)\varepsilon$.

Note that $|\psi_N\rangle$ final snapshot of a history state of the protocol circuit $CKT(G,N)$, which specifies a reflection strategy $\mathcal{T}$ for the game $\mathcal{G}_N$. Therefore the rejection probability of Output Check when the shared state is ρ_f is $\mathrm{Tr}\left(|0\rangle\langle 0|_{out}\, |\psi_N\rangle\langle\psi_N|\right)$, which is at least $1-\omega^*(\mathcal{G}_N)$. This concludes the proof of the lemma. □

4.4 Analysis of $\mathcal{V}^\sharp_{H,n}$

The following lemma states the important properties of the verifier $\mathcal{V}^\sharp_{H,n}$ specified in Figure 3.

Lemma 4.9. *Let G be a GTM for a family of k-prover ENL games $\{\mathcal{G}_n\}$, and let $\mathcal{V}^\sharp_{H,n}$ be the verifier described in Figure 3. Let $n \geq 1$ be an integer, $N = 2^n$, $S = p(N)$, and $\mathcal{G}^\sharp_{H,n}$ be the S-qubit Honest Pauli Prover game whose verifier is specified by $\mathcal{V}^\sharp_{H,n}$. Then the following hold:*

1. *(Completeness) For all $\gamma > 0$ there exists an honest Pauli strategy $\mathcal{S}$ that has value*

$$\omega^*_{\mathcal{S}}(\mathcal{G}^\sharp_{H,n}) = 1 - \frac{1-\omega^*(\mathcal{G}_N)+\gamma}{p(N)+1} .$$

2. *(Soundness) There exists universal constants $\alpha \geq 1, \beta > 0$ such that for all Honest Pauli strategies $\mathcal{S}$,*

$$\omega^*_{\mathcal{S}}(\mathcal{G}^\sharp_{H,n}) \leq 1 - \left(\frac{1-\omega^*(\mathcal{G}_N)}{\beta\, p(N)}\right)^\alpha .$$

Proof. Completeness follows from combining the completeness statements of the Gate Check, Input Check, and Output Check.

We prove soundness. Let $\mathcal{S}$ be an Honest Pauli Prover strategy that succeeds with probability $1-\varepsilon$ in the game $\mathcal{G}^\sharp_{H,n}$. Then it succeeds with probability at least $1-3\varepsilon$ in each of the Gate Check, Input Check, and Output Check subprotocols.

Let $\delta = O(p(N)^{3/2}\sqrt{\varepsilon})$. By Lemma 4.5, there exists an honest Gate Check strategy $\mathcal{S}_1$ that is δ-close to $\mathcal{S}$. Using Lemma 3.8, this implies that $\mathcal{S}_1$ succeeds in the Input and Output Check subprotocols with probability at least $1-3\delta$.

Let $\delta' = O(p(N)\sqrt{\delta})$. Applying Lemma 4.7, there exists an honest Input Check strategy $\mathcal{S}_2$ that is δ'-close to $\mathcal{S}_1$. The strategy $\mathcal{S}_2$ succeeds in the Output Check subprotocol with probability at least $1-3\delta'$ (using Lemma 3.8 again).

Finally, applying Lemma 4.8, the success probability of $\mathcal{S}_2$ in Output Check is at most

$$1 - \frac{1 - \omega^*(\mathcal{G}_N)}{p(N) + 1} \, .$$

This implies that there exist universal constants $\beta, \mu, \nu > 0$ we have

$$\omega^*_{\mathcal{S}_2}(\mathcal{G}^\sharp_{H,n}) = 1 - p(N)^\mu \varepsilon^\nu \leq 1 - \frac{1 - \omega^*(\mathcal{G}_N)}{\beta \, p(N)} \, ,$$

which implies

$$\omega^*_{\mathcal{S}}(\mathcal{G}^\sharp_{H,n}) = 1 - \varepsilon \leq 1 - \left(\frac{1 - \omega^*(\mathcal{G}_N)}{\beta \, p(N)} \right)^\alpha \, ,$$

for some universal constant α. This concludes the proof.

□

We point out some properties of the games specified by $\mathcal{V}^\sharp_{H,n}$ that will be relevant for the next stage of the argument. In all the subprotocols above, the honest Pauli prover PV gets a question that specifies up to three commuting Pauli observables. (Furthermore, the honest Pauli prover's question can be embedded in what we call an *MS-compatible triple*; see Section 5.2.) All other provers PP_i get questions from the set $\{\star\} \cup \{Q_{ij}\} \cup \{A_{ij}\} \cup \{\perp\}$, where the $\perp$ symbol is used to denote the absence of a question. Furthermore, note that at any one time, at most one PP_i prover gets sent a message that is not $\perp$.

5 Simulating Honest Pauli Prover games

Let $\mathcal{G}_H$ be any $(k+1)$-prover S-qubit Honest Pauli Prover game (Definition 4.1) such that $k \geq 7$. In this section we introduce a k-prover *Simulated Pauli Prover game* $\mathcal{G}_S$ that uses 7 out of the k provers to simulate the Pauli prover in $\mathcal{G}_H$ (provided that $\mathcal{G}_H$ satisfies some mild conditions) using a technique similar to the "code-check" test in [Ji17, NV18].

In Section 5.1 we introduce a class of error-correcting codes that will be used in the game. In Section 5.2 we present a multi-qubit test for constant-weight Pauli observables. In Section 5.3 we define the simulated Pauli Prover game and state its properties.

5.1 Stabilizer codes

We consider weakly self-dual *Calderbank-Shor-Steane (CSS) codes* [CS96, Ste96b]. Let C be a classical $[m, d]$ linear error-correcting code over $\mathbb{F}_2$: C is specified by a generator matrix $H \in \mathbb{F}_2^{m \times d}$ and a parity check matrix $K \in \mathbb{F}_2^{(m-d) \times d}$ such that $C = \mathrm{Im}(H) = \ker(K)$. We say that C is weakly self-dual if the dual code $C^\perp$, with generator matrix K^T, is such that $C \subseteq C^\perp$; equivalently, $H^T H = 0$. To any such code C we associate a subspace $\mathcal{C}$ of $(\mathbb{C}^2)^{\otimes m}$ that is the simultaneous $+1$ eigenspace of a set of stabilizers $\{S_{W,j}\}_{W \in \{X,Z\}, j \in \{1,\dots,k'\}}$ such that $S_{W,j}$ is a tensor product of Pauli σ_W observables over $\mathbb{F}_2$ in the locations indicated by the j-th column of the generator matrix H, i.e.

$$S_{W,j} = \sigma_W(H_{1j}) \otimes \sigma_W(H_{2j}) \otimes \cdots \otimes \sigma_W(H_{mj}),$$

where H_{ij} is the (i,j)-th entry of H. The condition that $H^T H = 0$ implies that all the $S_{W,j}$ commute, so that $\mathcal{C}$ is well-defined.

The 7-qubit Steane code. We make use of the *Steane* code, a CSS code that encodes 1 qubit into 7 physical qubits [Ste96a]. In Figure 8, we list the stabilizer generators of the code as well as several logical X and logical Z operators (that are equal up to multiplication by a stabilizer). The logical generators satisfy the useful property that for every $i \in \{1,\ldots,7\}$, there exists a logical X (resp. logical Z) operator that acts trivially on the i-th qubit.

Stabilizer Generators							
S_1	X	X	X	X	I	I	I
S_2	X	X	I	I	X	X	I
S_3	X	I	X	I	X	I	X
S_4	Z	Z	Z	Z	I	I	I
S_5	Z	Z	I	I	Z	Z	I
S_6	Z	I	Z	I	Z	I	Z
Logical Operators							
$\overline{X}$	I	I	I	I	X	X	X
	X	X	I	I	I	I	X
	X	I	X	I	I	X	I
$\overline{Z}$	I	I	I	I	Z	Z	Z
	Z	Z	I	I	I	I	Z
	Z	I	Z	I	I	Z	I

Figure 8: The 7-qubit Steane code.

The next lemma establishes some basic properties of the Steane code (shared by any CSS code that can correct single-qubit errors).

Lemma 5.1. *Consider the 7-qubit Steane code (Figure 8). Let* $\mathsf{E}_1,\ldots,\mathsf{E}_7,\mathsf{F}_1,\mathsf{F}_1'$ *be qubit registers. Let* $\mathsf{E} = \mathsf{E}_1\cdots\mathsf{E}_7$*. Let* R *be a register of arbitrary dimension.*

1. *There exists a unitary* U *acting on registers* $\mathsf{E}_2\cdots\mathsf{E}_7\mathsf{F}_1\mathsf{F}_1'\mathsf{X}$ *and a state* $|\tau\rangle$ *such that for all states* $|\psi\rangle_{\mathsf{E}_1\cdots\mathsf{E}_7\mathsf{R}}$ *such that* $\mathrm{Tr}_{\mathsf{R}}(|\psi\rangle\langle\psi|)$ *is in the code space,*

$$U\Big(|\psi\rangle_{\mathsf{E}_1\cdots\mathsf{E}_7\mathsf{R}} \otimes |0\rangle_{\mathsf{F}_1\mathsf{F}_1'\mathsf{X}}\Big) \;=\; |\psi\rangle_{\mathsf{F}_1\mathsf{E}_2\cdots\mathsf{E}_7\mathsf{R}} \otimes |\tau\rangle_{\mathsf{E}_1\mathsf{F}_1'\mathsf{X}}\;.$$

   Moreover, the reduced density matrix of $|\tau\rangle$ *on* E_1 *is the maximally mixed state on one qubit.*

2. *For* $W \in \{X, Z\}$ *let* $\mathcal{L}_W$ *denote a logical* W *operator that does not act on* E_1*. For all states* $|\psi\rangle$ *on* $\mathsf{E}_1\cdots\mathsf{E}_7$ *that lie in the code space,*

$$\begin{aligned} U\Big(\mathcal{L}_W\,|\psi\rangle_{\mathsf{E}_1\cdots\mathsf{E}_7} \otimes |0\rangle_{\mathsf{F}_1\mathsf{F}_1'\mathsf{X}}\Big) &= \mathcal{L}_W U\left(|\psi\rangle_{\mathsf{E}_1\cdots\mathsf{E}_7} \otimes |0\rangle_{\mathsf{F}_1\mathsf{F}_1'\mathsf{X}}\right) \\ &= \left(\mathcal{L}_W|\psi\rangle_{\mathsf{F}_1\mathsf{E}_2\cdots\mathsf{E}_7\mathsf{R}}\right) \otimes |\tau\rangle_{\mathsf{E}_1\mathsf{F}_1'\mathsf{X}}\;. \end{aligned}$$

Proof. We first establish item 1. Since the Steane code is a quantum error-correcting code that can correct any one qubit error, there exists a unitary U that acts on registers $\mathsf{E}_2\cdots\mathsf{E}_7$ and ancilla registers $\mathsf{F}_1\mathsf{F}_1'\mathsf{X}$ and can correct an erasure error in the register E_1. Since the 7-qubit code can correct any single qubit erasure, the resulting state on registers $\mathsf{F}_1\mathsf{E}_2\cdots\mathsf{E}_7$ is the original state $\mathrm{Tr}_{\mathsf{R}}(|\psi\rangle\langle\psi|)$. Formally, let $|\overline{0}\rangle$ and $|\overline{1}\rangle$ denote the 7-qubit encodings of $|0\rangle$ and $|1\rangle$, respectively. Since the code corrects any single-qubit erasure, for any $b \in \{0,1\}$, applying U to the state $|\bar{b}\rangle_{\mathsf{E}} \otimes |0\rangle_{\mathsf{F}_1\mathsf{F}'_1\mathsf{X}}$ yields a pure state $|\theta\rangle_{\mathsf{E}\mathsf{F}_1\mathsf{F}'_1\mathsf{X}}$ such that

$$\mathrm{Tr}_{\mathsf{E}_1\mathsf{F}'_1\mathsf{X}}\,(|\theta\rangle\!\langle\theta|) = |\bar{b}\rangle\!\langle\bar{b}| \,.$$

Since $|\theta\rangle$ is pure, after rearranging registers we obtain that

$$U|\bar{b}\rangle_{\mathsf{E}} \otimes |0\rangle_{\mathsf{F}_1\mathsf{F}'_1\mathsf{X}} \;=\; |\theta\rangle_{\mathsf{F}_1\mathsf{E}_2\cdots\mathsf{E}_7\mathsf{E}_1\mathsf{F}'_1\mathsf{X}} \;=\; |\bar{b}\rangle_{\mathsf{F}_1\mathsf{E}_2\cdots\mathsf{E}_7} \otimes |\tau_b\rangle_{\mathsf{E}_1\mathsf{F}'_1\mathsf{X}} \,. \tag{9}$$

Now we establish two claims: (1) $\mathrm{Tr}_{\mathsf{F}'_1\mathsf{X}}(|\tau_b\rangle\!\langle\tau_b|)$ is the maximally mixed state on one qubit, and (2) $|\tau_0\rangle = |\tau_1\rangle$. The first claim follows from the fact that the reduced density matrix on one qubit of any code state of a CSS code that corrects single-qubit errors is maximally mixed. The second claim follows from the fact that if $|\tau_0\rangle \neq |\tau_1\rangle$, then U would fail to correct an erasure error on the superposition $\frac{1}{\sqrt{2}}(|\bar{0}\rangle + |\bar{1}\rangle)$. Now write

$$|\psi\rangle_{\mathsf{ER}} = \alpha_0|\bar{0}\rangle_{\mathsf{E}} \otimes |\psi_0\rangle_{\mathsf{R}} + \alpha_1|\bar{1}\rangle_{\mathsf{E}} \otimes |\psi_1\rangle_{\mathsf{R}} \,.$$

Applying (9),

$$U|\psi\rangle_{\mathsf{ER}} \otimes |0\rangle_{\mathsf{F}_1\mathsf{F}'_1\mathsf{X}} = \sum_b \alpha_b|\bar{b}\rangle_{\mathsf{F}_1\mathsf{E}_2\cdots\mathsf{E}_7} \otimes |\psi_b\rangle_{\mathsf{R}} \otimes |\tau\rangle_{\mathsf{E}_1\mathsf{F}'_1\mathsf{X}} = |\psi\rangle_{\mathsf{F}_1\mathsf{E}_2\cdots\mathsf{E}_7\mathsf{R}} \otimes |\tau\rangle_{\mathsf{E}_1\mathsf{F}'_1\mathsf{X}} \,.$$

This establishes item 1. of the lemma.

To show item 2., we note that applying a logical operator $\mathcal{L}_W$ to a code state $|\psi\rangle$, erasing the first qubit, and then performing error correction, yields the state $\mathcal{L}_W|\psi\rangle$, except on a different set of registers. □

5.2 Multi-qubit entanglement tests

In this subsection we present the S-qubit EPR test, which is an elementary test that aims to verify that two provers A and B share S EPR pairs, on which they measure several commuting single- or two-qubit Pauli operators when asked to do so. This test uses as a primitive the Magic Square game, which is a two-prover nonlocal game that is a *self-test* for two EPR pairs. We present the Magic Square game next.

The Magic Square game. The 3×3 matrix presented in Figure 9 is called the *operator solution* for the Magic Square game. Each entry consists of the label for a two-qubit Pauli observable; the observables all commute within a row or a column. The product of the observables along every row and column is equal to I, except for the last column, which multiplies to $-I$.

$$\begin{bmatrix} XI & IX & XX \\ IZ & ZI & ZZ \\ XZ & ZX & YY \end{bmatrix}$$

Figure 9: Operator solution for the Magic Square game

The Magic Square game is played as follows: the verifier randomly chooses one of the provers to be prover A, and the other to be prover B. The verifier then chooses a random row r and column c from the operator solution for the Magic Square game. Let W denote the two-qubit Pauli observable in the intersection of r and c. The verifier then chooses random Pauli observables W_r, W_c from the row r and column c, respectively. The pairs (W, W_r) and (W, W_c), both formatted in lexicographic order, are sent to prover A and prover B, respectively. For example, the verifier could select the first column and second row, and send observables (IZ, XZ) to prover A and (IZ, ZZ) to prover B.

The provers are required to respond with two-bit answers $a, b \in \{0, 1\}^2$, respectively. The verifier checks that the bits in a and b that correspond to the common observable W sent to both provers are equal.

Definition 5.2 (Honest Magic Square strategy)**.** *The* honest Magic Square strategy $\mathcal{S}$ *is such that the shared state $|\psi\rangle$ is two EPR pairs (i.e. $|\psi\rangle = \frac{1}{2}(|00\rangle + |11\rangle)^{\otimes 2}$), and when a prover receives a pair of labels for commuting two-qubit Pauli observables, they measure the observables on their half of the EPR pairs and respond with the two bit outcome.*

Theorem 5.3 (Magic Square test, Theorem 5.9 in [CS17])**.** *The Magic Square game satisfies the following properties:*

1. *(Completeness) The honest Magic Square strategy succeeds in the Magic Square (MS) game with probability* 1*.*
2. *(Soundness) For any $\varepsilon \geq 0$ there is a $\delta = O(\sqrt{\varepsilon})$ such that any strategy with success probability at least $1 - \varepsilon$ in the game is δ-isometric to the honest Magic Square strategy.*

The EPR test. The S-qubit EPR test is described in Figure 10. The test and its analysis are adapted from [CRSV16]. The provers in the test are denoted prover A and prover B. Furthermore, the provers each receive a triple of commuting two-qubit Pauli observables $(W^{(1)}, W^{(2)}, W^{(3)})$. (This is purposefully formatted as questions to the Honest Pauli Prover in Section 4.)

The EPR test consists of two subtests, which check that the provers' measurements satisfy the Pauli commutation and anticommutation relations, respectively. The Magic Square game is used to test the anticommutation relations. In order for the EPR test — as well as the other protocols presented in this section — to be sound, we need to ensure that the provers cannot distinguish between the subtests. Thus we require a definition of a triple $(W^{(1)}, W^{(2)}, W^{(3)})$ that is *compatible* with the Magic Square game.

Definition 5.4. *A triple of commuting two-qubit Pauli observables $(W^{(1)}, W^{(2)}, W^{(3)})$ is* MS-compatible *if at least two of the observables act on the same pair of qubits, and furthermore those two observables can occur together in a row or column in Figure 9.*

In the EPR test (and the other protocols in this section) we require that any question to the provers is embedded in a uniformly random MS-compatible triple that is consistent with the question. For example, suppose the verifier samples the question (X_1, Z_2, Z_4) to send to prover A where the subscripts indicate which qubits the observables are supposed to act on. This question can be embedded in, say, the MS-compatible triple $(X_1 I_2, I_1 Z_2, I_3 Z_4)$, which is then sent to prover A. Note that any commuting pair of two-qubit Pauli observables, where each single-qubit observable is taken from $\{I, X, Z\}$, can be embedded in an MS-compatible triple in several ways; it does not matter which MS-compatible triple is chosen for any particular question.

The verifier performs each of the following with equal probability:

1. (Commutation test)
   (a) Select distinct $i, j \in \{1, \ldots, S\}$ and let $W, W' \in \{X, Z\}$ uniformly at random.
   (b) Send the pair of single-qubit observables (W_i, W'_j), embedded in an MS-compatible triple to prover A.
   (c) Send the two-qubit observable $W_i W'_j$, embedded in an MS-compatible triple to prover B.
   (d) Receive bits (a, a', a'') from prover A and (b, b', b'') from prover B. Let a, a' denote the answer bits corresponding to W_i and W'_j respectively, and let b denote the answer bit corresponding to $W_i W'_j$. Accept if and only if $a \oplus a' = b$.
2. (Anticommutation test)
   (a) Select distinct $i, j, \in \{1, \ldots, S\}$ and a pair of questions (q, q') in the Magic Square game. Note that q, q' both consist of a pair of commuting two-qubit Pauli observables.
   (b) Send q and q', embedded in MS-compatible triples, to prover A and prover B, respectively.
   (c) Accept if and only if the provers' answers associated with the query (q, q') would be accepted in the Magic Square game.

Figure 10: S-qubit EPR test [CRSV16].

Definition 5.5 (Honest EPR strategy). *An* honest S-qubit EPR strategy $\mathcal{S}$ *is a two-prover strategy that satisfies the following conditions. In the strategy the provers share the S-qubit maximally entangled state $|\Phi\rangle_{\mathsf{AB}}$, where prover A has register A and prover B has register B. When sent an MS-compatible triple $(W^{(1)}, W^{(2)}, W^{(3)})$ of mutually commuting two-qubit Pauli observables, the prover returns the three bits obtained by simultaneously measuring the three Pauli observables $\sigma_{W^{(1)}}$, $\sigma_{W^{(2)}}$ and $\sigma_{W^{(3)}}$ on its share of $|\Phi\rangle$.*

The following is a consequence of the results in [CRSV16].

Theorem 5.6. *The S-qubit EPR test (Figure 10) has the following guarantees.*

- *(Complexity) Questions in the test are $O(\log S)$-bit long. Answers are $O(1)$-bit long.*
- *(Completeness) Any honest S-qubit EPR strategy succeeds with probability 1 in the test.*
- *(Soundness) For any $\varepsilon \geq 0$ there is a $\delta = \mathrm{poly}(S; \varepsilon)$ such that any strategy that succeeds with probability at least $1 - \varepsilon$ in the test is δ-isometric to a honest S-qubit EPR strategy.*

5.3 Simulated Pauli Prover game

Let $\mathcal{V}_H$ be a verifier for a $(k+1)$-prover S-qubit Honest Pauli Prover game $\mathcal{G}_H$ satisfying some special properties that will be specified later (these properties are satisfied by the verifier $\mathcal{V}_{H,n}$ introduced in Section 4). Assume $k \geq 7$.

We define a k-prover ENL game $\mathcal{G}_S$ that simulates $\mathcal{G}_H$. Label the provers in $\mathcal{G}_S$ as $P_1, \dots, P_k$. Of the k provers, the first seven, $P_1, \dots, P_7$, are chosen to be the "simulated Pauli provers." The idea is that the provers $P_1, \dots, P_7$ are supposed to share the state of PV where each qubit is encoded using the 7-qubit Steane code, and prover P_i holds the i-th share of each encoded qubit. When in game $\mathcal{G}_H$, PV is asked to measure a certain Pauli observable, in game $\mathcal{G}_S$ the simulated Pauli provers are asked to implement a logical observable on their share of the encoding. In addition, the prover is sent its own question, as in $\mathcal{G}_H$, and asked to provide an answer. Since, in contrast to PV, none of the provers $\{P_i\}$ in $\mathcal{G}_S$ are trusted, the verifier in $\mathcal{G}_S$ executes a sub-test (called *Stabilizer Check*) to ensure that the simulated Pauli provers do indeed share an encoding of some state (on some sub-registers), and measure a Pauli observable when asked to do so.

The game $\mathcal{G}_S$ is described in Figure 11. In the game questions are of the form (W, g) where W is called an "EPR question" (i.e. is an MS-compatible triple that could arise in the EPR test) and g is a "$\mathcal{G}_H$ question" (i.e. a question that is asked in the game $\mathcal{G}_H$). The provers reply with answers (A, a) where A is the answer to the EPR question and a is the answer to g. We use q_i to denote the i-th prover's question in $\mathcal{G}_S$.

Let g_P be a $\mathcal{G}_H$ question. For an answer $A = (A^{(1)}, A^{(2)}, A^{(3)})$ to an (MS-compatible) EPR question W that contains g_P (which we denote by $g_P \subseteq W$), let $A|_{g_P}$ denote the projection of A's three bits to those that correspond to g_P. If $g_P = \perp$, then $A|_{g_P}$ is defined to be 0.

The description of $\mathcal{G}_S$ in Figure 11 involves notions of "composite query" and "composite answer" that are defined as follows. Let H be the generator matrix corresponding to the Steane code described in Figure 8.

Definition 5.7 (Composite queries and answers)**.** *Let W be an S-qubit Pauli observable.*

1. *The composite query associated with W, denoted $\overline{W}$, is obtained by sending each prover forming the composite prover the question W.*

2. *Given answers $(A_i)_{i \in \{1,\dots,k\} \setminus \{t\}}$ from the 6 provers forming the composite prover, the composite answer $\overline{A}$ is obtained by selecting a uniformly random vector v in the column span of H such that $v_t = 1$, and computing the sum $\overline{A} = \sum_{i \in \{1,\dots,7\} \setminus \{t\}} v_i A_i$.*

Let $\mathcal{G}_H$ denote a $(k+1)$-prover Honest Pauli Prover game such that $k \geq 7$.
The first 7 of the k provers are designated the "simulated Pauli prover".
The verifier in $\mathcal{G}_S$ perform one of the following tests, each chosen with equal probability:

1. *(Stabilizer Check)*

   (a) Pick $t \in \{1, \ldots, 7\}$ uniformly at random. Prover P_t is designated the "special prover". The other provers $\{P_1, \ldots, P_7\} \setminus \{P_t\}$ are jointly referred to as the "composite prover". A prover is not told whether it is the special prover, or a composite prover.

   (b) Generate a query (W, W') in the S-qubit EPR test, and for $i \in \{1, \ldots, k\}$ independently sample a question g_i according to the marginal distribution of the i-th prover's question in $\mathcal{G}_H$.

   (c) Set $q_t = (W, g_t)$ and $q_i = (W', g_i)$ for each $i \in \{1, \ldots, 7\} \setminus \{t\}$. For $i > 7$, set $q_i = (W'', g_i)$ where W'' is a random EPR question.

   (d) Let (A_i, a_i) denote the i-th prover's answer. Accept if and only if $(A_t, \overline{A})$ would be accepted in the EPR test, where $\overline{A}$ is the composite answer associated with $\{A_i\}_{i \neq t}$. (Answers to $\mathcal{G}_H$ questions are ignored.)

2. *($\mathcal{G}_H$ Simulation)*

   (a) Generate a query $Q = (g_P, g_1, \ldots, g_k)$ as in $\mathcal{G}_H$. Let $i^* \in \{1, \ldots, k\}$ denote the index such that $g_{i^*} \neq \perp$ if it exists. If it doesn't, set $i^* = 1$.

   (b) Let W be a uniformly random MS-compatible triple that contains g_P.

   (c) For all $i \in \{1, \ldots, k\}$, if $g_i = \perp$ set $q_i = (W, \widetilde{g}_i)$, where $\widetilde{g}_i$ is uniformly random question sampled from the marginal distribution of the i-th prover's question in $\mathcal{G}_H$. If $g_i \neq \perp$ set $q_i = (W_i, g_i)$, where W_i is a uniformly random EPR question.

   (d) Let $v \in \{0,1\}^7$ be such that $\sigma_X(v)$ and $\sigma_Z(v)$ are logical operators for the 7-qubit code, and moreover $v_{i^*} = 0$.

   (e) Let (A_i, a_i) denote the i-th prover's answer. Let $A = \sum_{i \in \{1,\ldots,7\}} v_i A_i$. Accept if and only if $(A|_{g_P}, a_1, \ldots, a_k)$ would be accepted in $\mathcal{G}_H$.

Figure 11: k-prover ENL game $\mathcal{G}_S$.

For a label $W \in \{X, Z\}$, an integer $i \in [S]$, and bit $A \in \{0,1\}$, let $\sigma_{W_i}^A$ denote the projector $\frac{1}{2}(\mathbb{1} + (-1)^A \sigma_{W_i})$. We first analyze the Stabilizer Check of the game $\mathcal{G}_S$. We show that succeeding in the Stabilizer Check with high probability enforces that the provers hold a state that is encoded using the Steane code, and furthermore they apply honest Pauli measurements. This type of rigidity statement is common to the works of [Ji16, Ji17, NV17a, NV18].

Definition 5.8 (Honest Stabilizer Check strategy)**.** *A strategy* $\mathcal{S} = (|\psi\rangle, \{M_i\})$ *is an* honest Stabilizer Check strategy *(implicitly, for code* C*) if the following holds.*

- *The state* $|\psi\rangle$ *is on registers* $\mathsf{C}, \mathsf{P}_1, \ldots, \mathsf{P}_k$*, and a reference register* R*, where for each* $i \in \{1, \ldots, k\}$*,* $\mathsf{P}_i = \mathsf{E}_i \mathsf{A}_i$ *with* E_i *a register of* S *qubits labeled* $\mathsf{E}_{i1}, \ldots, \mathsf{E}_{iS}$*.*

- *For $j \in \{1,\dots,S\}$, the reduced density matrix $\rho_{\mathsf{E}_{1j}\cdots\mathsf{E}_{7j}}$ of $|\psi\rangle$ is in the code space of $\mathcal{C}$. We refer to E_i as the S "code qubits" of prover P_i.*

- *Let $\{M_i((W,g),(A,a))\}$ denote the i-th prover's POVM for the question (W,g), where $W = (W^{(1)}, W^{(2)}, W^{(3)})$ is an EPR question and g is a $\mathcal{G}_H$ question. Then*

$$\mathop{\mathbb{E}}_{g} \sum_{a} M_i((W,g),(A,a)) = \sigma_W^A \,, \tag{10}$$

where the expectation is taken with respect to the marginal distribution of questions g to the i-th prover in $\mathcal{G}_H$ and $\sigma_W^A = \sigma_{W^{(1)}}^{A_1}\sigma_{W^{(2)}}^{A_2}\sigma_{W^{(3)}}^{A_3}$ is the product of the three commuting projectors corresponding to the Pauli observables W acting on E_i.

Lemma 5.9 (Rigidity for Stabilizer Check)**.** *The following properties hold for the Stabilizer Check (item 1. in Figure 11).*

1. *(Completeness) An honest Stabilizer Check strategy $\mathcal{S}$ passes the Stabilizer Check with probability* 1.
2. *(Soundness) For any $\varepsilon \geq 0$ there is a $\delta = \mathrm{poly}(S;\varepsilon)$ such that any strategy $\mathcal{S}$ that pass the Stabilizer Check with probability at least $1-\varepsilon$ is δ-isometric to an honest Stabilizer Check strategy.*

Proof. We first show completeness. Let $\mathcal{S}$ be an honest Stabilizer Check strategy. Suppose without loss of generality that prover 1 is selected to be the special prover, and provers $\{2,\dots,7\}$ are chosen to form the composite prover. In the Stabilizer Check, the EPR test is executed between the special prover and the composite prover; thus $\mathcal{S}$ can then be viewed as a two-prover strategy in the EPR test, where the special prover measures the Pauli observables corresponding to its EPR question on its share of the shared state $|\psi\rangle$, generating a triple of bits $A \in \{0,1\}^3$ as its answer. The composite prover performs the Pauli measurements of provers $P_2,\dots,P_7$ on registers $\mathsf{E}_2,\dots,\mathsf{E}_7$, generating 6 strings $A_2,\dots,A_7 \in \{0,1\}^3$. Assume without loss of generality that the composite answer is the sum $\overline{A} = A_2 + A_3 + A_4$ modulo 2 (this corresponds to selecting the vector $v = 1111000$ in the column span of the generator matrix H corresponding to the Steane code).

It is straightforward to verify that this two-prover strategy passes the EPR test with probability 1. Suppose first that the commutation subtest of the EPR test is chosen by the verifier, and let i, j, W_i, W_j' be as in Figure 10. Then the special prover measures $\sigma_{W_i}(i)$ and $\sigma_{W_j'}(j)$ on registers E_{1i} and E_{1j} of $|\psi\rangle$ to obtain answer bits a and a', respectively. The composite prover independently measures $\sigma_{W_i}(i) \otimes \sigma_{W_j'}(j)$ on registers $\mathsf{E}_{2i}\mathsf{E}_{2j}$, $\mathsf{E}_{3i}\mathsf{E}_{3j}$, and $\mathsf{E}_{4i}\mathsf{E}_{4j}$ to obtain answer bits a_2, a_3, a_4 which then form the composite answer $\overline{a} = a_2 + a_3 + a_4$. Since Pauli observables $\sigma_{W_i}(i)^{\otimes 4}$ acting on $\mathsf{E}_{1i}\mathsf{E}_{2i}\mathsf{E}_{3i}\mathsf{E}_{4i}$ and $\sigma_{W_j'}(j)^{\otimes 4}$ acting on $\mathsf{E}_{1j}\mathsf{E}_{2j}\mathsf{E}_{3j}\mathsf{E}_{4j}$ are stabilizers of the Steane code, this implies that $a + a' + \overline{a} = 0$, which is the condition checked in the EPR test. A similar argument holds for the anticommutation test.

Next we show soundness of the Stabilizer Check. Fix a $t \in \{1,\dots,7\}$, and condition on prover P_t being selected as the special prover. The provers' strategy $\mathcal{S}$ is accepted in the Stabilizer Check with probability at least $1-7\varepsilon$. From $\mathcal{S}$ we construct a strategy $\mathcal{S}_t'$ for the EPR test as follows. Let (W,W') be the query received in the EPR test. When prover A receives question W, it generates a uniformly random $\mathcal{G}$ question g_t for the t-th prover, and plays according to the special prover P_t's strategy on question (W, g_t). For prover B we combine the strategies of the six provers that make the composite prover (including the post-processing involved in computing the composite answer

$\overline{A'}$). Prover B simulates the measurements of the six provers on (W', g_i) where g_i is a random $\mathcal{G}_H$ question for the i-th prover, for $i = \{1, \ldots, 7\} \setminus \{t\}$.

The resulting two-prover strategy succeeds in the EPR test with success probability $1 - 7\varepsilon$. Applying the soundness analysis of the EPR test given in Theorem 5.6 it follows that $\mathcal{S}'_t$ is poly$(S;\varepsilon)$-isometric to an honest S-qubit EPR strategy. In particular, there is an isometry V_t for the special prover, such that the special prover's measurement operator associated with the answer A_t to the EPR question W, which is

$$\mathop{\mathbb{E}}_{g_t} \sum_{a_t} M_t((W, g_t), (A_t, a_t)) ,$$

is poly$(S;\varepsilon)$-close to the honest Pauli measurement operator σ_W^A, under V_t, on the S qubits identified by the isometry.

Applying this analysis for each $t \in \{1, \ldots, 7\}$, we obtain an isometry V_t for each prover under which their (marginalized) measurement operators are poly$(S;\varepsilon)$-close to the corresponding honest Pauli measurement operator. Let E_{ij} denote the register that holds the j-th qubit of the i-th prover under the isometry.

It remains to show that the shared state $|\psi\rangle$ (after application of the isometries $\{V_t\}$) is poly$(S;\varepsilon)$-close to the codespace of the Steane code. Let Π denote the projector onto the 7 qubit codespace of the Steane code. Observe that

$$\Pi = \mathop{\mathbb{E}}_{h} h , \tag{11}$$

where the expectation is over a uniformly random stabilizer element h of the Steane code. Using that the stabilizer elements of the Steane code (or any CSS code) are Hermitian and form a group, it is immediate to verify that the expectation in Equation (11) define a projection; by definition the codespace is the eigenvalue-1 eigenspace of the projection. For $j \in \{1, \ldots, S\}$ let Π_j (resp. h_j) denote projector onto the codespace (resp. the stabilizer h) of the Steane code that acts on registers $\mathsf{E}_{1j} \cdots \mathsf{E}_{7j}$.

Let $|\psi'\rangle = \bigotimes_t V_t |\psi\rangle$. Succeeding with probability at least $1 - \varepsilon$ in the Stabilizer Check test implies that for all $j \in \{1, \ldots, S\}$, we have that $|\psi'\rangle$ is *approximately stabilized* by the stabilizers of the Steane code:

$$\mathop{\mathbb{E}}_{h_j} \left\| h_j |\psi'\rangle - |\psi'\rangle \right\| \leq \text{poly}(S;\varepsilon) ,$$

from which it follows that

$$\left\| \Pi_j |\psi'\rangle - |\psi'\rangle \right\| = \left\| \mathop{\mathbb{E}}_{h_j} h_j |\psi'\rangle - |\psi'\rangle \right\| \leq \text{poly}(S;\varepsilon) .$$

By a hybrid argument, this implies that

$$\left\| \bigotimes_j \Pi_j |\psi'\rangle - |\psi'\rangle \right\| \leq \text{poly}(S;\varepsilon) ,$$

which completes the proof. □

Theorem 5.10. *Let $k \geq 7$ be an integer. Let $\mathcal{G}_H$ be a $(k+1)$-prover S-qubit Honest Pauli Prover game that satisfies the following properties:*

1. *The distribution over queries $(g_P, g_1, \ldots, g_k)$ is such that for any $(g_P, g_1, \ldots, g_k)$ in the support, there is at most one $i^* \in \{1, \ldots, k\}$ such that $g_{i^*} \neq \perp$.*

2. *For any query $(g_P, g_1, \dots, g_k)$ the accept or reject decision of $\mathcal{G}_H$ does not depend on the answer of prover PP_i, for all i such that $g_i = \perp$.*

3. *The distribution of g_P is supported on sets of Pauli observables that can be embedded in MS-compatible triples (see Definition 5.4).*

Let $\mathcal{G}_S$ be the Simulated Pauli Prover game described in Figure 11. Then the following hold.

- *(Completeness) For all Honest Pauli Prover strategies $\mathcal{S}_H$ in $\mathcal{G}_H$ there exists a k-prover strategy $\mathcal{S}$ in $\mathcal{G}_S$ that succeeds with probability $\omega^*_{\mathcal{S}_H}(\mathcal{G}_H)$.*
- *(Soundness) For any k-prover honest Stabilizer Check strategy that succeeds in $\mathcal{G}_S$ with probability at least $1 - \varepsilon$, there is a $(k+1)$-prover Honest Pauli prover strategy that is accepted with probability at least $1 - 2\varepsilon$ in $\mathcal{G}_H$.*

Proof. The completeness part of the theorem is straightforward.

We show soundness. Fix an honest Stabilizer Check strategy $\mathcal{S} = (|\psi\rangle, \{M_i\})$ for the k provers in $\mathcal{G}_S$ that has success probability at least $1 - \varepsilon$, for some $\varepsilon \geq 0$. In the game $\mathcal{G}_H$, the provers are labeled $PV, PP_1, \dots, PP_k$. The honest Pauli prover is PV. Using the strategy $\mathcal{S}$, we define an Honest Pauli strategy $\mathcal{S}^H = (|\psi\rangle^H, \{M_i^H\})$ for the provers in $\mathcal{G}_H$ as follows:

- ρ^H is on registers $\mathsf{C}, \mathsf{P}_V^H, \mathsf{P}_1^H, \dots, \mathsf{P}_k^H$, and R, where the honest Pauli prover PV gets P_V^H, and prover PP_i gets P_i^H for $i \in \{1, \dots, k\}$. The register P_V^H is isomorphic to the union of $\mathsf{E}_1, \dots, \mathsf{E}_7$ (i.e. it is $7S$ qubits). The register P_i^H is isomorphic to $\mathsf{F}_i\mathsf{A}_i$. The reduced density ρ^H of the state $|\psi\rangle^H$ on all registers except R is equal to the state $\rho \otimes \sigma$, where ρ is the reduced density of $|\psi\rangle$ on all registers but R, and σ is the maximally mixed state on an ancilla register $\mathsf{F} = \mathsf{F}_1 \cdots \mathsf{F}_7$ that is isomorphic to $\mathsf{E} = \mathsf{E}_1 \cdots \mathsf{E}_7$. The registers have been relabeled according to the scheme described in Figure 12.

Register in $\rho \otimes \sigma$	**Register in ρ^H**
C	C
$\mathsf{E}_1 \cdots \mathsf{E}_7$	P_V^H
$\mathsf{A}_i\mathsf{F}_i$	P_i^H

Figure 12: Relabeling the registers of $\rho \otimes \sigma$ to get ρ^H.

In other words, the honest Pauli prover is given the S code qubits held by each of the 7 provers that constitute the simulated Pauli prover in $\mathcal{G}_S$. The prover PP_i in $\mathcal{G}_H$ gets all the other qubits of prover P_i in $\mathcal{G}_S$, as well as the maximally mixed state in place of the S qubits.

- On reception of a question g_P in $\mathcal{G}_H$ (which is a collection of up to three commuting Pauli observables), the honest Pauli prover PV samples a random EPR question $W = (W^{(1)}, W^{(2)}, W^{(3)})$ that contains g_P. The prover PV measures the three logical observables $W^{(1)}, W^{(2)}, W^{(3)}$ on the $7S$-qubit encoded state to obtain $(A_1^{(j)}, \dots, A_7^{(j)})$ for $j = 1, 2, 3$. Let $(A^{(1)}, A^{(2)}, A^{(3)})$ be the decoded measurement outcomes. For example, PV could apply the logical operator which has weight only on the last 3 qubits and set $A^{(j)} = A_5^{(j)} + A_6^{(j)} + A_7^{(j)}$. The prover PV returns $A|_{g_P}$.

- Suppose prover PP_i in $\mathcal{G}_H$ receives the question g_i. If $g_i = \perp$, then PP_i returns 0. The prover PP_i samples a random EPR question W_i that contains g_P. The prover PP_i performs the same measurement that prover P_i would in game $\mathcal{G}_S$ on question (W_i, g_i). It obtains answer (A_i, a_i) and returns a_i.

The following claim establishes that the answer distribution of the honest Pauli strategy $\mathcal{S}^H$, when restricted to the "relevant" provers (i.e. the provers who receive questions that are not $\perp$), is essentially the same as in the strategy $\mathcal{S}$.

Claim 5.11. *Fix a query $Q = (g_P, g_1, \ldots, g_k)$ in $\mathcal{G}_H$.*

1. *If for all $i \in \{1, \ldots, k\}$ it holds that $g_i = \perp$, then the distribution of $A|_{g_P}$ that is produced by strategy $\mathcal{S}$ in the "$\mathcal{G}_H$ Simulation" part of $\mathcal{G}_S$ when query Q is sampled is the same as the distribution of a_P that is produced by prover PV in the strategy $\mathcal{S}^H$ when it receives the question g_P.*

2. *If there exists an $i^* \in \{1, \ldots, k\}$ such that $g_{i^*} \neq \perp$, then the distribution of $(A|_{g_P}, a_{i^*})$ that is produced by strategy $\mathcal{S}$ in the "$\mathcal{G}_H$ Simulation" part of $\mathcal{G}_S$ is the same as the distribution of (a_P, a_{i^*}) that is produced by prover PV and PP_{i^*} in the strategy $\mathcal{S}^H$ when they receive questions g_P and g_{i^*} respectively.*

We defer the proof of the claim to Section 5.4 and proceed with the proof of Theorem 5.10. Since the strategy $\mathcal{S}$ succeeds with probability at least $1 - \varepsilon$ in $\mathcal{G}_S$, it succeeds with probability at least $1 - 2\varepsilon$ in the $\mathcal{G}_H$ Simulation part of $\mathcal{G}_S$.

From our assumption on the game $\mathcal{G}_H$, for a fixed $\mathcal{G}_H$ question $Q = (g_P, g_1, \ldots, g_k)$ that is sampled in the $\mathcal{G}_H$ Simulation part of $\mathcal{V}_{sim}$, the accept or reject decision of $\mathcal{G}_H$ does not depend on a_i if $g_i = \perp$. Combined with the fact that at most one index i^* is such that $g_{i^*} \neq \perp$, Lemma 5.11 implies that the distribution of "relevant" answers to $\mathcal{G}_H$ are the same in the following two scenarios when Q is fixed: the strategy $\mathcal{S}^H$ in $\mathcal{G}_H$, and the strategy $\mathcal{S}$ in the $\mathcal{G}_H$ Simulation part of $\mathcal{G}_S$.

Thus for a fixed Q, the probability that the "relevant" answers are accepted by $\mathcal{G}_H$ are the same in both scenarios. Since the distribution of Q is the same in both scenarios, this implies that $\mathcal{S}^H$ passes $\mathcal{G}_H$ with probability at least $1 - 2\varepsilon$. □

5.4 Proof of Lemma 5.11

Part 1 of the claim follows directly from the fact that $\mathcal{S}$ is an honest Stabilizer Check strategy, in which the provers $P_1, \ldots, P_7$ measure the honest Pauli observables corresponding to a random EPR question W that contains g_P, which is identical to PV's action in the strategy $\mathcal{S}^H$.

We now argue Part 2. For an EPR question $W = (W^{(1)}, W^{(2)}, W^{(3)})$, we write σ_W for the product $\sigma_{W^{(1)}}\sigma_{W^{(2)}}\sigma_{W^{(3)}}$. For a three-bit vector $A = (A^{(1)}, A^{(2)}, A^{(3)})$, we write σ_W^A for the projector $\prod_{j=1}^{3} \frac{\mathbb{1}+(-1)^{A^{(j)}}}{2}$. This is a projector because the Pauli observables $\sigma_{W^{(j)}}$ all commute.

Assume without loss of generality that $i^* = 1$, and the string $v \in \{0,1\}^7$ chosen by the verifier in $\mathcal{G}_S$ is $v = 0000111$. Let W be a fixed EPR question that contains g_P. For $j \in \{1, 2, 3\}$ let

$$\mathcal{L}_{W^{(j)}} = \sigma_{W^{(j)}}(v)$$

denote the logical operator corresponding to $W^{(j)}$ which is a tensor product of two logical operators (since $W^{(j)}$ is the label for a two-qubit Pauli observable).

For notational clarity we write $g = g_{i^*}$ and $a = a_{i^*}$. Let $A_i = (A_i^{(1)}, A_i^{(2)}, A_i^{(3)})$ denote the three bits returned by prover P_i for its EPR question, and let $A^{(j)} = A_5^{(j)} + A_6^{(j)} + A_7^{(j)}$ denote the j-th bit of the answer vector A, as computed by the verifier.

Let $M_g^a = \mathbb{E}_{W_1} \sum_A M_1((W_1, g), (A, a))$ denote P_1's measurement on question g, where we have marginalized the EPR question (which was chosen independently of W) and the associated answers.

We compute the probability of the answer pair (A, a) in $\mathcal{S}$ when prover P_1 gets the question (W_1, g) for a uniformly random EPR question W_1, provers P_5, P_6, P_7 get the EPR question W, and each prover gets an independently chosen random $\mathcal{G}_H$ question. Since $\mathcal{S}$ is an honest Stabilizer Check strategy, the measurement operator each prover applies (when marginalizing over the prover's answer to its $\mathcal{G}_H$ question) is given by (10). By our choice of v, the outcome (A, a) occurs with probability

$$\sum_{A_5+A_6+A_7=A} \operatorname{Tr}_\rho \left(M_g^a \otimes \sigma_W^{A_5} \otimes \sigma_W^{A_6} \otimes \sigma_W^{A_7} \right) \tag{12}$$

$$= \operatorname{Tr}_\rho \left(M_g^a \otimes \prod_{j=1}^{3} \left(\frac{\mathbb{1} + (-1)^{A^{(j)}} \mathcal{L}_{W^{(j)}}}{2} \right) \right) . \tag{13}$$

Expanding the product, we obtain eight terms of the form

$$\pm \frac{1}{8} \operatorname{Tr}_\rho \left(M_g^a \otimes \mathcal{L}_D \right) ,$$

where $\mathcal{L}_D$ is a product of up to three logical operators $\{\mathcal{L}_{W^{(j)}}\}$. The label D indicates a collection of up to six Pauli observables (for example, $\mathcal{L}_D = \mathcal{L}_{W^{(1)}} \mathcal{L}_{W^{(2)}} \mathcal{L}_{W^{(3)}}$ where each $W^{(j)}$ is a label for a two-qubit Pauli observable).

Fix one of the possible labels D. Let U be the unitary given by Lemma 5.1. Since $\mathcal{S}$ is an honest Stabilizer Check strategy, $\rho_{\mathsf{E}_{1j} \cdots \mathsf{E}_{7j}}$ is in the code space for all $j \in \{1, \ldots, S\}$. Let $\mathsf{F}_1, \mathsf{F}_1'$ be registers isomorphic to E_1, and let X be an ancilla register that is sufficiently large. Applying part 1. of Lemma 5.1 we get

$$U^{\otimes S} \rho \otimes |0\rangle\langle 0|_{\mathsf{F}_1 \mathsf{F}_1' \mathsf{X}} (U^{\otimes S})^\dagger = \rho_{\mathsf{F}_1 \mathsf{E}_2 \cdots \mathsf{E}_7} \otimes |\tau_S\rangle\langle \tau_S|_{\mathsf{E}_1 \mathsf{F}_1' \mathsf{X}} , \tag{14}$$

where $|\tau_S\rangle$ is the S-fold tensor product of the state $|\tau\rangle$ given by Lemma 5.1. Here, the j-th tensor factor of $U^{\otimes S}$ acts on registers $\mathsf{E}_{2j} \cdots \mathsf{E}_{7j} \mathsf{F}_{1j} \mathsf{F}_{1j}' \mathsf{X}_j$. Then

$$\begin{aligned}
\operatorname{Tr}_\rho \left(M_g^a \otimes \mathcal{L}_D \right) &= \operatorname{Tr}\left((M_g^a \otimes \mathcal{L}_D)(\rho \otimes |0\rangle\langle 0|_{\mathsf{F}_1 \mathsf{F}_1' \mathsf{X}})(U^{\otimes S})^\dagger (U^{\otimes S}) \right) \\
&= \operatorname{Tr}\left(M_g^a U^{\otimes S} \ \mathcal{L}_D (\rho \otimes |0\rangle\langle 0|_{\mathsf{F}_1 \mathsf{F}_1' \mathsf{X}})(U^{\otimes S})^\dagger \right) \\
&= \operatorname{Tr}\left((M_g^a \otimes \mathcal{L}_D) \ U^{\otimes S} (\rho \otimes |0\rangle\langle 0|_{\mathsf{F}_1 \mathsf{F}_1' \mathsf{X}})(U^{\otimes S})^\dagger \right) \\
&= \operatorname{Tr}\left((M_g^a \otimes \mathcal{L}_D) \left(\rho_{\mathsf{A}\mathsf{F}_1 \mathsf{E}_2 \cdots \mathsf{E}_7} \otimes |\tau_S\rangle\langle \tau_S|_{\mathsf{E}_1 \mathsf{F}_1' \mathsf{X}} \right) \right) \\
&= \operatorname{Tr}\left((M_g^a \otimes \mathcal{L}_D) \left(\rho_{\mathsf{A}\mathsf{F}_1 \mathsf{E}_2 \cdots \mathsf{E}_7} \otimes \sigma_{\mathsf{E}_1} \right) \right) ,
\end{aligned}$$

where σ_{E_1} is the maximally mixed state on E_1. The second equality follows from the cyclicity of the trace and the fact that U and M_g^a act on different registers. The third equality follows from part 2 of Lemma 5.1. The fourth equality follows from (14). The last equality follows from the fact that the reduced density matrix of $|\tau_S\rangle$ on E_1 is the maximally mixed state.

Thus the probability of obtaining outcome (A, a) expressed in (12) is the same as

$$\operatorname{Tr}\left(\left(\rho_{\mathsf{A}\mathsf{F}_1 \mathsf{E}_2 \cdots \mathsf{E}_7} \otimes \sigma_{\mathsf{E}_1} \right) \left(M_g^a \otimes \prod_j \frac{\mathbb{1} + (-1)^{A^{(j)}} \mathcal{L}_{W^{(j)}}}{2} \right) \right) . \tag{15}$$

Here the operator M_g^a acts on $\mathsf{A}_1\mathsf{E}_1$. Observe that the state $\rho^H = \rho_{\mathsf{AE}_1\mathsf{E}_2\cdots\mathsf{E}_7} \otimes \sigma_{\mathsf{F}_1}$ and therefore (15) is equal to

$$\mathrm{Tr}_{\rho^H}\left(M_g^a \otimes \prod_j \frac{\mathbb{1} + (-1)^{A^{(j)}} \mathcal{L}_{W^{(j)}}}{2} \right)$$

where now we treat the operator M_g^a as acting on registers $\mathsf{A}_1\mathsf{F}_1$. This quantity is precisely the probability that (A, a) is obtained by provers PV and PP_r in the strategy $\mathcal{S}^H$ when given input $g_P = W$ and g_{i^*}, respectively: the prover PV measures the registers $\mathsf{E}_5, \mathsf{E}_6, \mathsf{E}_7$ using the observables $\mathcal{L}_{W^{(1)}}, \mathcal{L}_{W^{(2)}}, \mathcal{L}_{W^{(3)}}$ and the prover PP_{i^*} measures the registers $\mathsf{A}_{i^*}\mathsf{F}_{i^*}$ with the POVM $\{M_g^a\}$. This establishes Part 2 of the claim.

6 The Compression Theorem

In this section we present the proof of our compression result, informally stated as Theorem 1.3 in the introduction, and formally re-stated here.

Theorem 6.1 (Compression Theorem). *Let $k \geq 7$ be an integer, and let G be a GTM for a family of k-prover ENL games $\{\mathcal{G}_n\}$. Let $p(n)$ denote the size of* CKT(G, n)*, the n-th protocol circuit specified by G. There exists a family of k-prover ENL games $\{\mathcal{G}_n^\sharp\}$ such that the following holds, for all integer n:*

1. *The verifier of $\mathcal{G}_n^\sharp$, denoted by $\mathcal{V}_n^\sharp$, is uniformly generated from $(1^n, G)$.*
2. *Each prover's answer in $\mathcal{G}_n^\sharp$ is 4 bits long.*
3. *There are universal constants $\alpha \geq 1, \beta > 0$ such that for $N = 2^n$,*
$$1 - \frac{1 - \omega^*(\mathcal{G}_N)}{p(N) + 1} \leq \omega^*(\mathcal{G}_n^\sharp) \leq 1 - \left(\frac{1 - \omega^*(\mathcal{G}_N)}{\beta\, p(N)} \right)^\alpha . \tag{16}$$
4. *There exists universal constants $\mu \geq 1, \nu > 0, C > 0$ such that any strategy $\mathcal{S}$ for $\mathcal{G}_n^\sharp$ that satisfies $\omega_{\mathcal{S}}^*(\mathcal{G}_n^\sharp) \geq 1 - \varepsilon$ for some $\varepsilon \geq 0$ requires an entangled state such that the local dimension of registers associated with at least 7 of the provers is at least $(1 - C\, p(N)^\mu\, \varepsilon^\nu) 2^{p(N)}$.*

To make the dependence of the games $\{\mathcal{G}_n^\sharp\}$ on the GTM G more explicit, in subsequent sections we use the notation $\mathcal{G}_{G,n}^\sharp$ and $\mathcal{V}_{G,n}^\sharp$ to denote the game and verifier associated with G in Theorem 6.1.

Proof. The proof combines the results of the Section 4 and Section 5. Let $S = p(N)$ and $\mathcal{G}_{H,n}^\sharp$ the S-qubit $(k+1)$-prover Honest Pauli Prover game obtained from G as described in Figure 3. Observe that $\mathcal{G}_{H,n}^\sharp$ satisfies the properties required by Theorem 5.10. Let $\mathcal{G}_n^\sharp$ denote the S-qubit Simulated Pauli Prover game obtained from $\mathcal{G}_{H,n}^\sharp$ as described in Figure 11. Let $\mathcal{V}_{H,n}^\sharp$ and $\mathcal{V}_n^\sharp$ denote the verifiers of $\mathcal{G}_{H,n}^\sharp$ and $\mathcal{G}_n^\sharp$, respectively. The verifiers $\mathcal{V}_{H,n}^\sharp$ and $\mathcal{V}_n^\sharp$ depend on the GTM G, but we leave the dependence implicit.

By inspecting each of the subprotocols of the Honest Pauli Prover game presented in Section 4, it is not hard to verify that the family of verifiers $\left\{\mathcal{V}_{H,n}^\sharp\right\}$ for the games $\left\{\mathcal{G}_{H,n}^\sharp\right\}$ is uniformly generated from $(1^n, G)$. Inspecting the protocols in Section 5, it follows that the family of verifiers $\left\{\mathcal{V}_n^\sharp\right\}$ for the games $\left\{\mathcal{G}_n^\sharp\right\}$ is uniformly generated from $(1^n, G)$ as well. This establishes the first item of the theorem.

The second item follows since answers in $\mathcal{G}_n^\sharp$ consist of 3 bits, to answer the EPR question, and 1 bit, to answer the $\mathcal{G}_H^\sharp$ question.

We show the third item. The completeness statements of Lemma 4.9 and Theorem 5.10 imply that for any $\gamma > 0$ there exists a strategy $\mathcal{S}$ in $\mathcal{G}_n^\sharp$ that succeeds with probability at least $1 - \frac{1-\omega^*(\mathcal{G}_N)+\gamma}{p(N)+1}$. Using that $\omega^*(\mathcal{G}_n^\sharp)$ is defined as a supremum over strategies, taking the limit $\gamma \to 0$ shows the lower bound in (16).

For the upper bound, consider a k-prover strategy $\mathcal{S}$ for $\mathcal{G}_n^\sharp$ that succeeds with probability $1-\varepsilon$, for some $\varepsilon \geq 0$. Then $\mathcal{S}$ passes the Stabilizer Check subroutine of $\mathcal{G}_n^\sharp$ (see Figure 11) with probability at least $1-2\varepsilon$. By Lemma 5.9, $\mathcal{S}$ is $\mathrm{poly}(S;\varepsilon)$-isometric to an honest Stabilizer Check strategy $\mathcal{S}'$. Applying Lemma 3.8, it follows that the strategy $\mathcal{S}'$ succeeds in $\mathcal{G}_n^\sharp$ with probability at least $1-\mathrm{poly}(S;\varepsilon)$.

Observe that $\mathcal{G}_{H,n}^\sharp$ is a Honest Pauli Prover game that satisfies the properties required for the application of Theorem 5.10, and that by definition $\mathcal{G}_n^\sharp$ is the simulated game associated with $\mathcal{G}_{H,n}^\sharp$. It follows from the soundness part of the theorem that there exists a $(k+1)$-prover Honest Pauli strategy $\mathcal{S}''$ such that

$$\omega^*_{\mathcal{S}''}(\mathcal{G}_{H,n}^\sharp) \geq 1 - \mathrm{poly}(S;\varepsilon)\ . \tag{17}$$

Moreover, using that $\mathcal{S}''$ is a Honest Pauli strategy, from Lemma 4.9 we get

$$\omega^*_{\mathcal{S}''}(\mathcal{G}_{H,n}^\sharp) \leq 1 - \left(\frac{1-\omega^*(\mathcal{G}_N)}{\beta'\, p(N)}\right)^{\alpha'}\ , \tag{18}$$

for universal constants $\alpha' \geq 1, \beta' > 0$. Combining (17) and (18), since $\varepsilon = 1-\omega^*(\mathcal{G}_n^\sharp)$ and $S = p(N)$, it follows that

$$\omega^*(\mathcal{G}_n^\sharp) \leq 1 - \left(\frac{1-\omega^*(\mathcal{G}_N)}{\beta\, p(N)}\right)^{\alpha}\ ,$$

for some universal constants $\alpha > 1, \beta > 0$.

Finally we show the fourth item in the theorem.As shown in the course of the proof of the third item, any strategy $\mathcal{S}$ for $\mathcal{G}_n^\sharp$ that is accepted with probability at least $1-\varepsilon$, for some $\varepsilon \geq 0$, is δ-isometric to an honest Stabilizer Check strategy $\mathcal{S}'$, for some $\delta = \mathrm{poly}(S;\varepsilon)$. By definition the provers in an honest Stabilizer Check strategy share a state $|\psi\rangle$ such that for any $i \in \{1,\ldots,S\}$ the reduced density of $|\psi\rangle$ on registers $\mathsf{E}_{i1},\ldots,\mathsf{E}_{i7}$, held by provers $P_1,\ldots,P_7$ respectively, is a 7-qubit state supported on the codespace. Applying item 1. from Lemma 5.1 independently to each of the S reduced densities, it follows that for any $t \in \{1,\ldots,7\}$ the reduced density of $|\psi\rangle$ on register $\mathsf{E}_t = \mathsf{E}_{1t}\ldots\mathsf{E}_{St}$ is the totally mixed state on S qubits. Using the definition of δ-isometric strategies, it follows that for every $t \in \{1,\ldots,7\}$ there exists an isometry V_t mapping register E_t to registers AA', and an isometry V_t' mapping registers $\{\mathsf{E}_j\}_{j\neq t}$ to registers BB', such that

$$V_t \otimes V_t'|\psi\rangle_{\mathsf{E}_1\cdots\mathsf{E}_7\mathsf{R}} \approx_\delta |\Phi\rangle_{\mathsf{AB}} \otimes |\psi'\rangle_{\mathsf{A'B'R}}\ ,$$

where $|\Phi\rangle_{\mathsf{AB}}$ is an S-qubit maximally entangled state between A and B, and the state $|\psi'\rangle$ is arbitrary. Here, the notation $\approx_\delta$ indicates closeness in trace distance. Using that for any two pure states $|\phi\rangle, |\theta\rangle$ it holds that $1 - \big\||\phi\rangle\langle\phi| - |\theta\rangle\langle\theta|\big\|_1 \leq |\langle\phi|\theta\rangle|^2$, we obtain

$$\Big|\big(\langle\Phi|_{\mathsf{AB}} \otimes \langle\psi'|_{\mathsf{A'B'R}}\big)\big(V_t \otimes V_t'|\psi\rangle_{\mathsf{P}_1\cdots\mathsf{P}_7\mathsf{R}}\big)\Big|^2 \geq 1-\delta\ . \tag{19}$$

If $|\theta\rangle_{\mathsf{AA'BB'R}}$ is an arbitrary pure state with Schmidt rank at most r along the cut that separates the registers $\mathsf{AA'}$ and $\mathsf{BB'R}$, then using that all Schmidt coefficients of $|\Phi\rangle_{\mathsf{AB}} \otimes |\psi'\rangle_{\mathsf{A'B'R}}$ along the same cut are at most $2^{-S/2}$ it follows that

$$\left|\left(\langle\Phi|_{\mathsf{AB}} \otimes \langle\psi'|_{\mathsf{A'B'R}}\right)\left(|\theta\rangle_{\mathsf{AA'BB'R}}\right)\right|^2 \leq r2^{-S} \,. \tag{20}$$

Inequalities (19) and (20) imply that the Schmidt rank of $V_t \otimes V_t'|\psi\rangle_{\mathsf{P}_1\cdots\mathsf{P}_7\mathsf{R}}$ between prover t and the other provers is at least $(1-\delta)2^{p(N)}$. Since the isometries V_t and V_t' cannot increase the Schmidt rank between prover t and the other provers as well as the reference system R, the same lower bound holds for the Schmidt rank of $|\psi\rangle$ between register P_t and $\{\mathsf{P}_j\}_{j\neq t}\mathsf{R}$. Finally, since this lower bound holds for all $t = 1,\ldots,7$, this concludes the proof of item 4.

□

7 Recursive compression of quantum interactive proofs

In this section we show how to apply the compression theorem, Theorem 6.1 in Section 6, recursively to prove Theorem 1.1 and Theorem 1.2 stated in the introduction. Before doing so we introduce several definitions.

A function $t : \mathbb{N} \to \mathbb{N}$ is *time-constructible* if there exists an integer $m \geq 0$ and a deterministic Turing machine T such that for all $n \geq m$, the Turing machine halts on input 1^n after exactly $t(n)$ steps. Examples of time-constructible functions include $n, n^2, 2^n, 2^{2^n}$, and so on. Recall the iterated exponential function $\Lambda_R(n)$, defined inductively by $\Lambda_0(n) = n$ for all integer $n \geq 0$, and for integer $R \geq 0$, $\Lambda_{R+1}(n) = 2^{\Lambda_R(n)}$ for all integer $n \geq 0$. We call the parameter R the "height" of $\Lambda_R(n)$.

Definition 7.1. *A time-constructible function $t(n)$ is* hyper-exponential *if there exists a function $R(n)$ such that $t(n) = \Lambda_{R(n)}(n)$.*

Note that with this definition, any hyper-exponential function t satisfies $t(n) \geq n$ for all $n \geq 0$.

Definition 7.2. *Let $t : \mathbb{N} \to \mathbb{N}$ be a time-constructible function. The language $\mathcal{L}[t]$ consists of all pairs $(1^n, M)$ such that M is a nondeterministic Turing machine that halts on input 0 within $t(n)$ steps.*

For any time-constructible t, the language $\mathcal{L}[t]$ is complete for NTIME[t] under polynomial-time Karp reductions. The following result from [NV17b] will be used as the base case for our construction. It shows that for $t(n) = 2^n$ languages in $\mathcal{L}[t]$ can be decided by a polynomial-size verifier in a two-prover nonlocal game.

Theorem 7.3 (The Natarajan-Vidick verifier [NV17b])**.** *There is a universal constant $\delta > 0$ and a family of verifiers $\{\mathcal{V}_{NV}(M,n)\}$ that is uniformly generated from $(1^n, M)$ such that for any integer n and nondeterministic Turing machine M the following hold. The game $\mathcal{G}_{NV}(M,n)$ associated with $\mathcal{V}_{NV}(M,n)$ is a two-prover nonlocal game such that $\omega^*(\mathcal{G}_{NV}(M,n)) = 1$ if $(1^n, M) \in \mathcal{L}[2^n]$ and $\omega^*(\mathcal{G}_{NV}(M,n)) \leq 1-\delta$ otherwise.*

7.1 The main recursive compression result

The main result we prove in this section is the following.

Proposition 7.4. *Let $t : \mathbb{N} \to \mathbb{N}$ be a hyper-exponential function. Let T be a deterministic Turing machine that halts in exactly $t(n)$ steps on input 1^n. Let M be a nondeterministic Turing machine. There exists a family of 7-prover ENL games $\left\{\mathcal{G}_{n,M,T}\right\}$ that is uniformly generated from $(1^n, M, T)$ and such that*

1. *The answer length of the provers is $O(1)$ bits.*
2. *There exists universal constants $c, C > 0$ such that for all integer n,*

$$\begin{aligned} &\omega^*(\mathcal{G}_{n,M,T}) = 1 && \text{if } (1^n, M) \in \mathcal{L}[2^t] \\ &\omega^*(\mathcal{G}_{n,M,T}) \leq 1 - Ct(n)^{-c} && \text{if } (1^n, M) \notin \mathcal{L}[2^t]. \end{aligned}$$

Before proving Proposition 7.4 we show that it implies Theorem 1.1, which we reformulate for convenience.

Theorem 7.5. *There exists universal constants $c', C' > 0$ such that for any hyper-exponential function $t : \mathbb{N} \to \mathbb{N}$,*

$$\mathrm{NTIME}[2^{t(n)}] \subseteq \mathrm{MIP}^*_{1,1-C't^{-c'}}(15, 1) .$$

Proof. Let T be a deterministic Turing machine that halts in exactly $t(n)$ steps on input 1^n. Fix an instance $(1^n, M)$ of $\mathcal{L}[2^t]$. Applying Proposition 7.4 gives a 7-prover game $\mathcal{G}_{n,M,T}$ of size poly(n) such that $\omega^*(\mathcal{G}_{n,M,T}) = 1$ if $(1^n, M) \in \mathcal{L}[2^t]$, and otherwise $\omega^*(\mathcal{G}_{n,M,T}) \leq 1 - Ct(n)^{-c}$ for some universal constants $c, C > 0$.

To convert the game to an MIP* protocol, i.e. remove the provers' initial quantum message in the ENL game, we use the compression result of [Ji17] as a black box. This result provides an efficient method to transform any ENL game $\mathcal{G}$ involving k provers into a nonlocal game $\mathcal{G}'$ of size (as measured by the verifier circuit) poly$(|\mathcal{G}|)$, involving $k + 8$ provers, with the following properties. If $\omega^*(\mathcal{G}) = 1$, then $\omega^*(\mathcal{G}') = 1$. Otherwise,

$$\omega^*(\mathcal{G}') \leq 1 - \left(\frac{1 - \omega^*(\mathcal{G})}{\mathrm{poly}(n)}\right)^d \leq 1 - C'\, t(n)^{-c'} ,$$

for some universal constants $d, c', C' > 0$.Here the second inequality uses that $t(n) = \Omega(n)$ for any hyper-exponential function t. Combining the two reductions gives a polynomial-time reduction from $\mathcal{L}[2^t]$ to 15-prover nonlocal game $\mathcal{G}'_{n,M,T}$. □

To prove Proposition 7.4, we present and analyze a family of verifiers $\{\mathcal{V}_{RC}(n, n_0, M, T, G)\}$, specified in Figure 13. The verifiers are parametrized by two integers $n \geq n_0 > 0$, a nondeterministic Turing machine M, a deterministic Turing machine T, and a GTM G that takes input (n, t, λ). Here, think of n_0 as the input size, and n as a parameter that indicates the size of $\mathcal{V}_{RC}$. For the actual verifier used to define the game, $n = n_0$, but we may also consider the case where n eventually grows very large. Roughly speaking, if $n \geq t(n_0)$, the verifier $\mathcal{V}_{RC}(n, n_0, M, T, G)$ simulates the Natarajan-Vidick protocol from Theorem 7.3 to determine whether $(1^n, M) \in \mathcal{L}[2^t]$. Otherwise, if n is smaller than $t(n_0)$, then $\mathcal{V}_{RC}$ is "too small" to perform the simulation directly. In this case, $\mathcal{V}_{RC}$ instead executes the compressed protocol associated with $\mathcal{V}_{RC}(2^n, n_0, M, T, G)$, i.e. an exponentially bigger version of itself.

Verifier name: $\mathcal{V}_{RC}(n, n_0, M, T, G)$
Description of parameters: $n \geq n_0 > 0$ are integers, M is a nondeterministic Turing machine, T is a deterministic Turing machine, and G is a GTM that takes input (n, t, λ).

1. Run T on input 1^{n_0} for n steps.
2. If T halts in that time, then execute the verifier $\mathcal{V}_{NV}(M, n)$ from Theorem 7.3.
3. Otherwise, execute the verifier $\mathcal{V}^{\sharp}_{G_\lambda,n}$ from Theorem 6.1, where $\lambda = (n_0, M, T, G)$ and $G_\lambda(n, t) = G(n, t, \lambda)$.

Figure 13: The recursive compression verifier

It follows from Theorem 6.1 and Theorem 7.3 that the family of verifiers $\{\mathcal{V}_{RC}(n, n_0, M, T, G)\}$ can be uniformly generated from $(1^n, \lambda)$, where $\lambda = (n_0, M, T, G)$, by a Turing machine R. By Lemma 3.11, there exists a GTM G_R that takes input (n, t, λ) and returns the t-th gate of the protocol circuit corresponding to the verifier $\mathcal{V}_{RC}(n, n_0, M, T, G)$.[8] For the remainder of the section we consider M and T as implicitly fixed, and write $\mathcal{V}_{RC}(n, n_0)$ for $\mathcal{V}_{RC}(n, n_0, M, T, G_R)$. Let $\mathcal{G}_{n,n_0}$ denote the 7-prover game specified by $\mathcal{V}_{RC}(n, n_0)$, and let ω^*_{n,n_0} denote $\omega^*(\mathcal{G}_{n,n_0})$. Let $\mathcal{G}_n = \mathcal{G}_{n,n}$.

Due to its recursive nature the verifier $\mathcal{V}_{RC}$ may be hard to comprehend at first. For concreteness, we go through an execution of the protocol specified by the verifier for the choice of the time-constructible function $t(n) = 2^n$. Thus, T is a Turing machine that on input 1^n iterates for 2^n steps exactly, and then halts. M is an arbitrary nondeterministic Turing machine, and n_0 a positive integer. The verifier $\mathcal{V}_{RC}(n_0, n_0)$ specifies the actions of a verifier in a 7-prover ENL game $\mathcal{G}_0$ that has size poly(n_0). Following the description in Figure 13, the verifier in $\mathcal{G}_0$ performs the following actions. It first executes T on input 1^{n_0} for n_0 steps. By definition of T, since $n_0 < t(n_0) = 2^{n_0}$, the Turing machine has not yet halted. Thus the verifier proceeds to the second step in Figure 13: it executes another verifier, $\mathcal{V}^{\sharp}_{G_\lambda,n_0}$ from Theorem 6.1. The verifier can compute the description of $\mathcal{V}^{\sharp}_{G_\lambda,n_0}$ in polynomial time given 1^{n_0} and the description of G_λ.

By construction (see the proof of Theorem 6.1) the verifier $\mathcal{V}^{\sharp}_{G_\lambda,n_0}$ specifies a 7-prover ENL game $\mathcal{G}^{\sharp}_{G_\lambda,n_0}$, which checks that the provers hold (an encoding of) the history state of the protocol circuit CKT$(G_\lambda, 2^{n_0})$. Let $n_1 = 2^{n_0}$. The protocol circuit CKT(G_λ, n_1) defines a verifier $\mathcal{V}_{RC}(n_1, n_0)$ and a game $\mathcal{G}_1 = \mathcal{G}_{n_1,n_0}$. Notice that $\mathcal{G}_1$ is just as $\mathcal{G}_0$, except that the first input is exponentially larger, from n_0 to n_1.

Theorem 6.1 relates the value of $\mathcal{G}_0$ to the value of $\mathcal{G}_1$. So it suffices to analyze the value of $\mathcal{G}_1$, which means analyzing $\mathcal{V}_{RC}(n_1, n_0, M, T, G_R)$. Since $n_1 \geq 2^{n_0}$, $\mathcal{G}_1$ reduces to the game $\mathcal{G}_{NV}$ specified by the Natarajan-Vidick verifier $\mathcal{V}_{NV}(M, n_1)$. By Theorem 7.3, if $(1^{n_1}, M) \in \mathcal{L}[2^{n_1}]$, then the value of $\mathcal{G}_{NV}(M, n_1)$ is 1, which implies that $\omega^*(\mathcal{G}_1) = 1$, which in turns implies that $\omega^*(\mathcal{G}_0) = 1$. Otherwise if $(1^{n_1}, M) \notin \mathcal{L}[2^{n_1}]$, $\omega^*(\mathcal{G}_1) = \omega^*(\mathcal{G}_{NV}(M, n_1)) \leq 1 - \delta$, which implies that $\omega^*(\mathcal{G}_0) \leq 1 - \frac{\delta^\alpha}{\text{poly}(n_1)} \leq 1 - C2^{-cn_0}$ for some constants $c, C > 0$.

Observe now that $(1^{n_1}, M) \in \mathcal{L}[2^{n_1}]$ if and only if $(1^{n_0}, M) \in \mathcal{L}[2^{2^{n_0}}]$. This establishes Proposi-

[8] Strictly speaking, the protocol circuit corresponds to an *equivalent* verifier to $\mathcal{V}_{RC}$, but for clarity of exposition we will not distinguish between the verifier specified by G_R and $\mathcal{V}_{RC}$ itself.

tion 7.4 for the special case $t(n) = 2^n$. We now give the proof for the general case.

Proof of Proposition 7.4. Since the answer sizes are constant in both the Natarajan-Vidick protocol, as well as the games produced by Theorem 6.1, this establishes item 1. of the proposition. We now show item 2.

Fix n, M, T. Since $t(n)$ is a hyper-exponential function, there exists a smallest integer $R \geq 0$ such that $\Lambda_R(n) = t(n)$ (note that R generally depends on n).

We show by downwards induction on $0 \leq r \leq R$ that there exists a constant $\beta \geq 1$ (depending only on G_λ) such that the following holds. If $(1^n, M) \in \mathcal{L}[2^t]$, then $\omega^*_{\Lambda_r(n),n} = 1$. Otherwise,

$$\omega^*(\mathcal{G}_n) \leq 1 - \frac{\delta^{\alpha^{R-r}}}{\Lambda_R(n)^{\beta\alpha^{R-r}} \cdots \Lambda_{r+1}(n)^{\beta\alpha}} . \tag{21}$$

Note that the case $r = 0$ implies item 2. of the proposition. First, the completeness statement shows that if $(1^n, M) \in \mathcal{L}[2^t]$, then $\omega^*(\mathcal{G}_n) = \omega^*_{\Lambda_0(n),n} = 1$. Second, the soundness statement (21) implies that there exists universal constants $c, C > 0$ depending only on α, β, δ such that $\omega^*_{n,n} \leq 1 - C\Lambda_R(n)^{-c} = 1 - Ct(n)^{-c}$.

For the base case $r = R$, note that on input 1^n the Turing machine T halts in $t(n) \leq \Lambda_R(n)$ steps. Thus the game $\mathcal{G}_{\Lambda_R(n),n}$ is the game associated with the Natarajan-Vidick verifier $\mathcal{V}_{NV}(M, \Lambda_{R(n)})$ (Theorem 7.3). Suppose that $(1^n, M) \in \mathcal{L}[2^t]$. This implies that $(1^{\Lambda_R(n)}, M) \in \mathcal{L}[2^n]$.[9] By Theorem 7.3, $\omega^*_{\Lambda_R(n),n} = 1$. Otherwise, if $(1^n, M) \notin \mathcal{L}[2^t]$, then we have that $\omega^*_{\Lambda_R(n),n} < 1 - \delta$.

Now suppose $r < R$. Then the Turing machine T does not halt on input 1^n in $\Lambda_r(n)$ steps. Therefore, $\mathcal{V}_{RC}(\Lambda_r(n), n)$ executes the verifier $\mathcal{V}^\sharp_{G_\lambda,\Lambda_r(n)}$, where G_λ is the GTM specified in Figure 13, with $\lambda = (n, M, T, G_R)$. In turn, the protocol circuit $\mathrm{CKT}(G_\lambda, 2^{\Lambda_r(n)}) = \mathrm{CKT}(G_\lambda, \Lambda_{r+1}(n))$ corresponds to the game $\mathcal{G}_{\Lambda_{r+1}(n),n}$. Thus it follows from Theorem 6.1 that

$$1 - \frac{1 - \omega^*_{\Lambda_{r+1}(n),n}}{\mathrm{poly}(\Lambda_{r+1}(n))} \leq \omega^*_{\Lambda_r(n),n} \leq 1 - \left(\frac{1 - \omega^*_{\Lambda_{r+1}(n),n}}{\mathrm{poly}(\Lambda_{r+1}(n))}\right)^\alpha ,$$

for some polynomial $\mathrm{poly}(\cdot)$ that depends only on G_λ and not r or n. Using the induction hypothesis (21), this completes the induction step. □

7.2 An alternate proof of the undecidability of nonlocal games

In this section we give an alternate proof that the problem of distinguishing between the cases when a nonlocal game has value equal to 1, or when it has value strictly less than 1, is undecidable [Slo16, Slo17]. This result was stated as Theorem 1.2 in the introduction. Let M be an arbitrary Turing machine, and G a GTM. Consider the family of verifiers $\{\mathcal{V}_{Halt}(n, M, G)\}$ described in Figure 14.

[9] Note: the "2^n" inside $\mathcal{L}[\cdot]$ is a variable that is different from the n used to specify the instance $(1^{\Lambda_R(n)}, M)$.

Verifier name: $\mathcal{V}_{Halt}(n, M, G)$:
Description of input: M is a deterministic Turing machine, and G is a GTM that takes input (n, t, M).

1. Run M on input 0 for n steps. If it halts in this time, then reject.

2. Otherwise, execute the verifier $\mathcal{V}^{\sharp}_{G_M,n}$ from Theorem 6.1 where $G_M(n,t) = G(n,t,M)$.

Figure 14: The verifier $\mathcal{V}_{Halt}$

It follows from the definition and Theorem 6.1 that the verifiers $\{\mathcal{V}_{Halt}(n, M, G)\}$ can be uniformly generated from $(1^n, M, G)$ by a Turing machine H. By Lemma 3.11, there exists a GTM G_H that takes input (n, t, M, G) and outputs the t-th gate of the protocol circuit corresponding to the verifier $\mathcal{V}_{Halt}(n, M, G)$. Define the verifier $\mathcal{V}_{Halt}(n, M) = \mathcal{V}_{Halt}(n, M, G_H)$.

Theorem 7.6. *There exists universal constants $c, C > 0$ such that for any deterministic Turing machine M there exists a 15-prover nonlocal game $\mathcal{G}_M$, that can be computed from the description of M, such that the following hold.*

1. *Suppose that M halts on input 0 in time T, for some $T \geq 0$. Let R be the largest integer such that $T > \Lambda_R(1)$. Then $\omega^*(\mathcal{G}_M) \leq 1 - C\Lambda_R(1)^{-c}$.*

2. *Suppose that M does not halt on input 0. Then $\omega^*(\mathcal{G}_M) = 1$. Furthermore, there is a universal constant $\eta > 0$ such that any strategy $\mathcal{S}$ for $\mathcal{G}_M$ such that $\omega^*_{\mathcal{S}}(\mathcal{G}_M) \geq 1 - \varepsilon$ for some $\varepsilon \geq 0$ requires local dimension at least $2^{\Omega(\varepsilon^{-\eta})}$.*

Theorem 7.6 implies that if there were a Turing machine A that when given a description of a nonlocal game $\mathcal{G}$, decides if $\omega^*(\mathcal{G}) = 1$, then A could be used to solve the Halting Problem. Thus there is no such Turing machine A.

Proof. Fix a deterministic Turing machine M. For any integer $n \geq 1$ let $\mathcal{G}_n$ denote the 7-prover game specified by $\mathcal{V}_{Halt}(n, M)$, and let $\omega^*_n = \omega^*(\mathcal{G}_n)$. It follows from Theorem 6.1 that

$$1 - \frac{1 - \omega^*_{2^n}}{p(2^n) + 1} \leq \omega^*_n \leq 1 - \left(\frac{1 - \omega^*_{2^n}}{\beta\, p(2^n)}\right)^{\alpha}, \tag{22}$$

for some universal constants $\alpha \geq 1, \beta > 0$ and some polynomial p that depends only on G_M.

We first show the completeness statement, item 2. in the theorem. Suppose that M does not halt on input 0. By an immediate induction it follows from the first inequality in (22) that for any $r \geq 0$,

$$1 - \omega^*_1 \leq \frac{1 - \omega^*_{\Lambda_r(1)}}{\big(p(\Lambda_1(1)) + 1\big) \cdots \big(p(\Lambda_r(1)) + 1\big)},$$

from which it follows, by taking the limit $r \to \infty$, that necessarily $\omega^*_1 = 1$.

Next we show the soundness statement, item 1. in the theorem. Suppose that M halts in time T, and let R be the largest integer such that $\Lambda_R(1) < T$. Then $\omega^*_{\Lambda_{R+1}(1)} = 0$. By downwards induction it follows from the second inequality in (22) that there exists constants $c', C' > 0$ that depend on G_M such that

$$1 - \omega_1^* \geq \frac{1}{p(\Lambda_R(1))^{\alpha^R} p(\Lambda_{R-1}(1))^{\alpha^{R-1}} \cdots p(\Lambda_1(1))^{\alpha}} \geq C' \Lambda_R(1)^{-c'} .$$

To conclude, as in the proof of Theorem 7.5 we apply the compression result from [Ji17] to $\mathcal{G}_1$ to obtain a 15-prover nonlocal game $\mathcal{G}_M$ such that $\omega^*(\mathcal{G}_M) = 1$ if M does not halt, and otherwise $\omega^*(\mathcal{G}_M) \leq 1 - \Omega((1-\omega_1^*)^\alpha) < 1 - C\Lambda_R(1)^{-c}$ for universal constants $c, C > 0$. Note that the game $\mathcal{G}_M$ is "constant sized" (there is no asymptotic parameter here).

The "furthermore" part of the theorem follows from the fact that any strategy $\mathcal{S}$ for $\mathcal{G}_M$ that is accepted with probability at least $1-\varepsilon$ is δ-isometric to a strategy $\mathcal{S}'$ such that the provers' shared state is a history state of a strategy $\mathcal{S}_1$ in $\mathcal{G}_1$ that succeeds with probability $1-\delta$ for $\delta = \varepsilon^c$. (This follows from the analysis of the compression result of [Ji17]; details omitted.) By part 4 of Theorem 6.1, the strategy $\mathcal{S}_1$ must have local dimension at least $2^{\Omega(\delta^{-\eta'})}$ for some universal constant $\eta' > 0$. Thus $\mathcal{S}$ must have local dimension $2^{\Omega(\varepsilon^{-\eta})}$ for some universal constant $\eta > 0$. □

8 Improving the Compression Theorem?

We explore the question of whether our compression theorem, Theorem 6.1, is optimal in terms of the trade-off that it provides between "compression in game size" versus "compression of the game value towards 1". Recall that, given a GTM G for a family of games $\{\mathcal{G}_N\}$, the theorem yields a family of games $\{\mathcal{G}_n^\sharp\}$ such that for all n and $N = 2^n$ we have that if $\omega^*(\mathcal{G}_N) = 1$, then $\omega^*(\mathcal{G}_n^\sharp) = 1$, but otherwise $\omega^*(\mathcal{G}_n^\sharp) \leq 1 - \left(\frac{1-\omega^*(\mathcal{G}_N)}{\mathrm{poly}(N)}\right)^\alpha$. The compression of the game size is exponential, from N to $\mathrm{poly}(\log N)$, and the value of $\mathcal{G}_n^\sharp$ is closer to 1 by a factor $\mathrm{poly}(N)$. But suppose that there was a *Hypothetical Compression Theorem (HCT)* with a better trade-off.

Conjecture 8.1 (Hypothetical Compression Theorem). *Given a GTM G for a family of games $\{\mathcal{G}_N\}$, there exists a family of verifiers $\{\mathcal{V}_n^\diamond\}$ that is uniformly generated from $(1^n, G)$, and a monotonically increasing function $g(n) = 2^{o(n)}$, such that the following hold. For any integer $n \geq 0$, the game $\{\mathcal{G}_n^\diamond\}$ associated with $\mathcal{V}_n^\diamond$ has constant answer size, and for $N = 2^n$ we have that if $\omega^*(\mathcal{G}_N) = 1$, then $\omega^*(\mathcal{G}_n^\diamond) = 1$, and in all cases,*

$$\omega^*(\mathcal{G}_n^\diamond) \leq 1 - \left(\frac{1-\omega^*(\mathcal{G}_N)}{g(n)}\right)^\alpha . \tag{23}$$

(Note that when $g(n) = 2^{O(n)}$ (23) recovers the upper bound of Theorem 6.1.) We show that Conjecture 8.1 implies "constant-gap analogues" of Theorem 7.5 and Theorem 7.6: first, MIP* would contain all computable languages. Second, MIP* would contain undecidable languages. The undecidability of MIP*, in turn, implies a negative answer to a multipartite generalization of Tsirelson's problem, which is a open question about the relation between the commuting operator and tensor product models for quantum correlations.

The main tool we need to derive these consequences is a *hardness amplification procedure* for k-prover ENL games. This is a general transformation on ENL games that preserves the game value if the original game has value 1, but otherwise decreases it.

We call an ENL game and its associated verifier *nonadaptive* if the questions to the provers are chosen before the measurement of the provers' first message. The ENL games and verifiers obtained from Theorem 6.1 are nonadaptive. The following hardness amplification procedure is established in [BVY17].

Theorem 8.2 (Hardness amplification via anchoring [BVY17])**.** *Fix an integer $k \geq 2$. For every integer r there exists a transformation $\mathcal{A}_r$ on verifiers such that for any k-prover nonadaptive verifier $\mathcal{V}$ for an ENL game $\mathcal{G}$ the following holds:*

1. *$\mathcal{A}_r(\mathcal{V})$ is a k-prover verifier for a nonadaptive ENL game $\mathcal{G}'$ such that*

$$(\omega^*(\mathcal{G}))^r \leq \omega^*(\mathcal{G}') \leq (1-(1-\omega^*(\mathcal{G}))^c)^{\nu_{\mathcal{G}} r} ,$$

   where $\nu_{\mathcal{G}}$ is a positive real that depends on the number of provers k and the length of answers in $\mathcal{G}$, and $c \geq 1$ is a universal constant.

2. *The size of $\mathcal{A}_r(\mathcal{V})$ is $O(r)$ times the size of $\mathcal{V}$.*

Furthermore, if $\{\mathcal{V}_{n,\lambda}\}$ is a family of verifiers uniformly generated from $(1^n, \lambda)$, the family of verifiers $\{\mathcal{A}_r(\mathcal{V}_{n,\lambda})\}$ can be uniformly generated from $(1^n, 1^r, \lambda)$.

Strictly speaking, the hardness amplification result of [BVY17] is stated for nonlocal games, in which the verifier is completely classical. However, the results extend to nonadaptive ENL games because the verifier's initial measurement can be modeled as the action of an "honest" prover.[10]

8.1 Consequence 1: MIP* contains all computable languages

A language L is *computable* if there exists a Turing machine M that, for all inputs $x \in \{0,1\}^*$, accepts if $x \in L$ and otherwise rejects. In particular, M halts on all inputs.

We introduce a verifier $\widehat{\mathcal{V}}_{RC}$, described in Figure 15, and analyze it in a manner similar to the verifier $\mathcal{V}_{RC}$ considered in Section 7.1. In this section, we use c and ν to denote the constants from Theorem 8.2 that correspond to games with at most 7 provers and the answer length provided by Conjecture 8.1. We also let α and $g(n)$ be the constant and subexponential function $g(n)$ from Conjecture 8.1. We let $\mathcal{V}_n^\diamond$ denote the verifier of the game $\mathcal{G}_n^\diamond$.

VTM name: $\widehat{\mathcal{V}}_{RC}(n, M, G)$:
Description of input: $n > 0$ is an integer, M is a deterministic Turing machine, and G is a GTM that takes input (n, t, λ).

1. Run M on input 0 for n steps. If M accepts in that time, accept. If M rejects in that time, reject.
2. Otherwise, if M does not halt in n steps, perform the following. Let $\lambda = (M, G)$, and $G_\lambda(n,t) = G(n,t,\lambda)$. Let n' be the largest integer less than n such that $(2g(n'))^{\alpha c} \cdot q(n')/\nu \leq n$, where $q(n')$ is the size of $\mathcal{V}^\diamond_{G_\lambda, n'}$. Let $r = (2g(n'))^{\alpha c}/\nu$. Execute the verifier $\mathcal{A}_r(\mathcal{V}^\diamond_{G_\lambda, n'})$.

Figure 15: The verifier $\widehat{\mathcal{V}}_{RC}$

[10] We believe that the nonadaptive condition can be omitted from the statement of Theorem 8.2, but we leave this for future work.

It follows from the definition that the family of verifiers $\{\widehat{\mathcal{V}}_{RC}(n, M, G)\}$ can be uniformly generated by some Turing machine R. (This is the reason for the choice of the parameter n', which guarantees that the size of the verifier $\mathcal{A}_r(\mathcal{V}^{\diamond}_{G_\lambda, n'})$ is at most n.)

Let R be a Turing machine that on input $(1^n, M, G)$ generates the verifier $\widehat{\mathcal{V}}_{RC}(n, M, G)$ in polynomial time. By Lemma 3.11, there exists a GTM G_R that takes input (n, t, M, G) and outputs the t-th gate of the protocol circuit corresponding to the verifier $\widehat{\mathcal{V}}_{RC}(n, M, G)$.

Proposition 8.3. *Suppose Conjecture 8.1 is true. Let M be a deterministic Turing machine that halts on input 0. Then the family of verifiers $\{\widehat{\mathcal{V}}_{RC}(n, M, G_R)\}$ can be uniformly generated from $(1^n, M, G_R)$. Furthermore, the 7-prover ENL game $\mathcal{G}_m$ associated with $\widehat{\mathcal{V}}_{RC}(m, M, G_R)$, where m is the smallest integer larger than $(2g(1))^{\alpha c} q(1)/\nu$,*[11] *satisfies*

$$\begin{aligned} \omega^*(\mathcal{G}_m) &= 1 && \text{if } M \text{ accepts on input } 0, \\ \omega^*(\mathcal{G}_m) &\leq 1/2 && \text{if } M \text{ rejects on input } 0. \end{aligned}$$

Proof. Let M and m be as in the theorem statement. For any integer $n \geq 1$, define the verifier $\widehat{\mathcal{V}}_{RC}(n) = \widehat{\mathcal{V}}_{RC}(n, M, G_R)$. Let $\mathcal{G}_n$ denote the 7-prover ENL game specified by $\widehat{\mathcal{V}}_{RC}(n)$, and let $\omega^*_n = \omega^*(\mathcal{G}_n)$. Let R be the smallest integer such that $\Lambda_R(m)$ is greater than the running time of M (which is well-defined since M halts on input 0).

If M accepts on input 0, then $\omega^*(\mathcal{G}_m) = 1$; this follows by induction on R, using similar reasoning as in the proof of Proposition 7.4. The remaining case is that M does not accept on input 0. By definition, for all $N \geq \Lambda_R(m)$, we have that $\omega^*_N = 0$. We show by downwards induction that $\omega^*_N \leq 1/2$ for all integers $N \geq m$. Assume the inductive hypothesis holds for all $N \geq N_0 + 1$ for some $N_0 < \Lambda_R(m)$. Since $N_0 < \Lambda_R(m)$, M does not halt on input 0 in N_0 steps. Therefore, the verifier in the game $\mathcal{G}_{N_0}$ executes $\mathcal{A}_r(\mathcal{V}^{\diamond}_{G_\lambda, N'_0})$ where λ, G_λ, N'_0, and r are defined in Figure 15. Let $N_1 = 2^{N'_0}$. Since g is a monotonically increasing but subexponential function, we have $N'_0 = \omega(\log N_0)$ and therefore $N_1 > N_0$. Therefore by the induction hypothesis it follows that $\omega^*_{N_1} \leq 1/2$. Using Conjecture 8.1 and Theorem 8.2 together,

$$\omega^*_{N_0} \leq \left(1 - \left(\frac{1 - \omega^*_{N_1}}{g(N'_0)}\right)^{\alpha c}\right)^{\nu r}.$$

Using that $\omega^*_{N_1} \leq 1/2$ and the choice of r made in Figure 15, we get that $\omega^*_{N_0} \leq 1/e \leq 1/2$. This completes the induction and shows that $\omega^*_m \leq 1/2$, as desired. □

Corollary 8.4. *Suppose Conjecture 8.1 is true. Then* MIP* *with constant completeness-soundness gap contains all computable languages. In other words, we have* R ⊆ MIP* *where* R *is the set of all recursive languages.*

Proof. Let L denote a computable language. This means that there exists a deterministic Turing machine M such that for all inputs $x \in \{0,1\}^*$, $M(x)$ accepts if $x \in L$, otherwise $M(x)$ rejects. Let M_x denote the Turing machine M with input x hardwired and otherwise ignores its input tape. Observe that M_x halts in finite time.

There exists a polynomial time deterministic Turing machine A that on input x performs the following. First, A computes a description of the 7-player ENL game $\mathcal{G}_{m,M_x}$ given by Proposition 8.3, with m chosen as in the proposition statement. Let $n = |x|$. This game has the property that if

[11] The justification for this choice of m is to ensure that for all $n \geq m$, the integer n' chosen in step 2. of the definition of $\widehat{\mathcal{V}}_{RC}(n, M, G)$ (Figure 15) is well-defined and at least 1.

M_x accepts, then $\omega^*(\mathcal{G}_{m,M_x}) = 1$, otherwise $\omega^*(\mathcal{G}_{m,M_x}) \leq 1/2$. Furthermore the size of the verifier of $\mathcal{G}_{m,M_x}$ is $\operatorname{poly}(n, |M|)$. Next, the ENL game $\mathcal{G}_{m,M_x}$ is converted to a nonlocal game by using the compression result of [Ji17]; this result gives an efficient reduction from the description of the verifier of $\mathcal{G}_{m,M_x}$ to the verifier of a 15-player nonlocal game $\mathcal{G}'_{m,M_x}$ whose value satisfies

$$\omega^*(\mathcal{G}_{m,M_x}) \leq \omega^*(\mathcal{G}'_{m,M_x}) \leq 1 - \left(\frac{1-\omega^*(\mathcal{G}_{m,M_x})}{\operatorname{poly}(n)}\right)^{\alpha} .$$

Finally, A computes a description of the game $\mathcal{G}''_{m,M_x}$, in which the hardness amplification procedure $\mathcal{A}_s$ of Theorem 8.2 is applied to the verifier of $\mathcal{G}'_{m,M_x}$ for some $s = \operatorname{poly}(n)$. The verifier of $\mathcal{G}''_{m,M_x}$ still has $\operatorname{poly}(n)$ size, but now if $\omega^*(\mathcal{G}'_{m,M_x}) \leq 1 - 1/\operatorname{poly}(n)$, then $\omega^*(\mathcal{G}''_{m,M_x}) \leq 1/2$ (provided that s is a large enough polynomial).

Thus on input x the Turing machine A returns the description of a nonlocal game with a $\operatorname{poly}(n)$-sized verifier, such that if x is accepted by M, the value of the game is 1; otherwise, the value is at most $1/2$. This shows that L has a one-round MIP* proof system with 15 provers and constant completeness-soundness gap. □

8.2 Consequence 2: MIP* contains undecidable languages

In this section we show that Conjecture 8.1 implies that MIP* contains undecidable languages. We show this directly: instead of reducing the halting problem to the problem of approximating the value of a nonlocal game, we show that there is no Turing machine that can approximate the value of a nonlocal game to within constant additive error. Thus MIP* contains undecidable languages: namely, the (promise) language $L_{c,s}$ whose YES instances consist of all nonlocal games whose value is at least c, and whose NO instances consists of all nonlocal games whose value is at most s, for some constants $0 \leq s < c \leq 1$.

In Figure 16 we define a VTM $\widehat{\mathcal{V}}_{undec}$ that is differs slightly from the VTM $\mathcal{V}_{Halt}$ analyzed in Section 7.2. Whereas the games $\{\mathcal{G}_{n,M}\}$ specified by $\mathcal{V}_{Halt}$ have value 1 or less than $1/2$ depending on whether M halts or not, the games $\{\mathcal{G}_{n,M}\}$ specified by $\widehat{\mathcal{V}}_{undec}$ have value 1 or less than $1/2$ depending on whether M accepts or rejects (when given its own description as input). There is no guarantee on the value of the game $\mathcal{G}_{n,M}$ if M does not halt.

In Figure 16, c, ν and α are the constants introduced in Section 8.1.

VTM name: $\widehat{\mathcal{V}}_{undec}(n, M, G)$:
Description of input: M is a deterministic Turing machine.

1. Run M on input M (i.e. the input to M is the description of M itself) for n steps. If M halts and accepts, then accept. If M halts and rejects, then reject.
2. If M does not halt within n steps, then perform the following. Let $\lambda = (M, G)$ and $G_\lambda(n, t) = G(n, t, \lambda)$. Let n' be the largest integer such that $(2g(n'))^{\alpha c} q(n')/\nu \leq n$, where $q(n')$ is the size of $\mathcal{V}^{\diamond}_{G_\lambda, n'}$. Let $r = (2g(n'))^{\alpha c}/\nu$. Execute $\mathcal{A}_r(\mathcal{V}^{\diamond}_{G_\lambda, n'})$.

Figure 16: The verifier $\widehat{\mathcal{V}}_{undec}$

It follows from the definition that the family of verifiers $\{\widehat{\mathcal{V}}_{undec}(n, M, G)\}$ can be uniformly generated by a Turing machine H. By Lemma 3.11, there exists a GTM G_H that takes input (n, t, M, G) and returns the t-th gate of the protocol circuit corresponding to the verifier $\widehat{\mathcal{V}}_{undec}(n, M, G)$. Define the verifier $\widehat{\mathcal{V}}_{undec}(n, M) = \widehat{\mathcal{V}}_{undec}(n, M, G_R)$. Let $\mathcal{G}_{n,M}$ denote the 7-prover ENL game executed by $\widehat{\mathcal{V}}_{undec}(n, M)$.

Proposition 8.5. *Suppose Conjecture 8.1 is true. Let M be a deterministic Turing machine. Then for all n,*

$$\begin{aligned} \omega^*(\mathcal{G}_{n,M}) &= 1 && \text{if } M \text{ accepts on input } M, \\ \omega^*(\mathcal{G}_{n,M}) &\leq 1/2 && \text{if } M \text{ rejects on input } M. \end{aligned}$$

Note that Proposition 8.5 does not specify the value of $\mathcal{G}_{n,M}$ in the case that M does not halt on input M. An ideal version of Proposition 8.5 would state that $\omega^*(\mathcal{G}_{n,M}) = 1$ if M does not halt, and $\omega^*(\mathcal{G}_{n,M}) \leq 1/2$ if M halts, similarly to the conclusion of Theorem 7.6. We are able to obtain a guarantee on the value of $\mathcal{G}_{n,M}$ when M does not halt in Theorem 7.6 because of special properties of the games specified by $\mathcal{V}^\sharp$ (namely, when the size N of the verifier increases, the value of the game goes to 1, no matter what game is being compressed). However, the games specified by Conjecture 8.1 may not satisfy this property; the only guarantee is that $\omega^*(\mathcal{G}^\diamond_{G_\lambda,n}) = 1$ if $\omega^*(\mathcal{G}_{2^n,M}) = 1$, and otherwise $\omega^*(\mathcal{G}^\diamond_{G_\lambda,n})$ is upper-bounded by some function of $\omega^*(\mathcal{G}_{2^n,M})$.

The proof of Proposition 8.5 is essentially the same as the proof of Proposition 8.3, and we omit it. We state a corollary showing that it is possible to construct a family of nonlocal games with similar properties as the ENL games from Proposition 8.5.

Corollary 8.6. *Suppose Conjecture 8.1 is true. Let M be a deterministic Turing machine. There exists a 15-prover nonlocal game $\mathcal{G}_M$ such that*

$$\begin{aligned} \omega^*(\mathcal{G}_{M}) &= 1 && \text{if } M \text{ accepts on input } M, \\ \omega^*(\mathcal{G}_{M}) &\leq 1/2 && \text{if } M \text{ rejects on input } M. \end{aligned}$$

Furthermore, the description of the verifier of $\mathcal{G}_M$ is computable from M.

Proof. The proof is essentially the same as the proof of Corollary 8.4. The only additional step is to apply the hardness amplification procedure $\mathcal{A}_r$ from Theorem 8.2 to the game returned by the Turing machine A, for $r = \mathrm{poly}(|M|)$, to amplify the gap from 1 vs. $1 - 1/\mathrm{poly}(|M|)$ to 1 vs. $1/2$. □

Theorem 8.7. *Suppose Conjecture 8.1 is true. Then there is no deterministic Turing machine A that, given as input the description of the verifier circuits of a nonlocal game $\mathcal{G}$, decides whether $\mathcal{G}$ has value at least $2/3$ or less than $1/3$, promised that one is the case.*

Proof. Suppose for contradiction that there exists such a Turing machine A. Consider the following deterministic Turing machine M. M expects as input an X, which is the description of a deterministic Turing machine. The Turing machine M first computes the descriptions of verifier circuits for two nonlocal games $\mathcal{G}_X$ and $\mathcal{G}^r_X$. The first game, $\mathcal{G}_X$, is the game given by Corollary 8.6. The second game, $\mathcal{G}^r_X$, is the nonlocal game that results from applying the hardness amplification procedure $\mathcal{A}_r$ from Theorem 8.2 to $\mathcal{G}_X$, where r is an integer such that $(1 - (1/3)^c)^{\nu r} \leq 1/3$. Here, c and ν are the constants given by Theorem 8.2. Thus

$$\omega^*(\mathcal{G}^r_X) \leq (1 - (1 - \omega^*(\mathcal{G}_X))^c)^{\nu r} . \tag{24}$$

Furtherore, if $\mathcal{G}_X$ has value 1, then $\mathcal{G}^r_X$ has value 1.

Having computed the descriptions of the games $\mathcal{G}_X$ and $\mathcal{G}_X^r$, the Turing machine M executes two instances of A in parallel (for example, by interleaving the executions of A), where one instance is executed on the description of $\mathcal{G}_X$, and the other on $\mathcal{G}_X^r$. If one of the instances halts first with output bit a, then M rejects if $a = 1$ and accepts if $a = 0$. However, M may not halt (if both instances of A don't halt).

Observe that at most one of the games $\mathcal{G}_X$, $\mathcal{G}_X^r$ has value that is greater than 1/3 and less than 2/3. Indeed, suppose the value of both games were in that range. In particular, we have $1/3 < \omega^*(\mathcal{G}_X) < 2/3$. However, by (24) and our choice of r, this implies that $\omega^*(\mathcal{G}_X^r) \leq 1/3$, a contradiction.

Thus at least one instance of A halts, because by definition A correctly decides whether a given input game $\mathcal{G}$ has value at least 2/3 or at most 1/3. Therefore, M always halts, on all inputs X.

Now we analyze M, when given input M. By definition of $\mathcal{G}_M$, if M accepts input M, then $\omega^*(\mathcal{G}_M) = \omega^*(\mathcal{G}_M^r) = 1$. In this case, both instances of A accept, in which case M rejects, which is a contradiction.

On the other hand, if M rejects input M, then both $\omega^*(\mathcal{G}_M)$ and $\omega^*(\mathcal{G}_M^r)$ have value at most 1/3, in which case both instances of A reject, in which case M accepts, which is a contradiction.[12]

Therefore such a Turing machine A does not exist. □

Thus Theorem 8.7 implies that the language $L_{c,s}$ for $c = 2/3$ and $s = 1/3$ is undecidable, which implies that MIP* contains undecidable languages. We end by formulating the following corollary, that relates Conjecture 8.1 to a famous problem in quantum information, *Tsirelson's problem*. To state the corollary, we introduce the notion of a *k-partite, n-input, m-output correlation*, which is a k-tensor C of complex numbers, with size $nm \times \cdots \times nm = (nm)^k$, where k, n, m are arbitrary integers. We say that a correlation C is *achievable in the tensor product model* if there exists finite-dimensional Hilbert spaces $\mathcal{H}_1, \ldots, \mathcal{H}_k$, a state $|\psi\rangle \in \mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_k$, and for every $\ell \in \{1, \ldots, k\}$ and $i \in \{1, \ldots, n\}$ a POVM $\{A^a_{\ell,i}\}_{a \in \{1,\ldots,m\}}$ acting on $\mathcal{H}_\ell$, such that for all $i_1, \ldots, i_k \in \{1, \ldots, n\}$ and $a_1, \ldots, a_k \in \{1, \ldots, m\}$, we have

$$C(i_1, a_1, \ldots, i_k, a_k) = \langle \psi | A^{a_1}_{1,i_1} \otimes \cdots \otimes A^{a_k}_{k,i_k} | \psi \rangle \ .$$

Similarly, we say that C is *achievable in the commuting operator model* if there exists a (possibly infinite-dimensional) Hilbert space $\mathcal{H}$, a state $|\psi\rangle \in \mathcal{H}$, and for every $\ell \in \{1, \ldots, k\}$ and $i \in \{1, \ldots, n\}$ a POVM $\{A^a_{\ell,i}\}_{a \in \{1,\ldots,m\}}$ acting on $\mathcal{H}$ satisfying the commutativity condition $[A^a_{\ell,i}, A^{a'}_{\ell',i'}] = 0$ for all $\ell \neq \ell'$ and i, i', a, a', such that for all $i_1, \ldots, i_k \in \{1, \ldots, n\}$ and $a_1, \ldots, a_k \in \{1, \ldots, m\}$, we have

$$C(i_1, a_1, \ldots, i_k, a_k) = \langle \psi | A^{a_1}_{1,i_1} \cdots A^{a_k}_{k,i_k} | \psi \rangle \ .$$

We also measure the *distance* between two correlations C, C' as the sum of the absolute differences of their entries:

$$|C - C'| = \sum_{\substack{i_1,\ldots,i_k \\ a_1,\ldots,a_k}} \left| C(i_1, a_1, \ldots, i_k, a_k) - C'(i_1, a_1, \ldots, i_k, a_k) \right| .$$

Tsirelson's problem (more precisely, the multipartite version of it) asks whether for every k, n, and m, for every k-partite, n-input, m-output correlation C achievable in the commuting operator model, for every $\varepsilon > 0$, there exists a k-partite, n-input, m-output correlation C' achievable in the tensor product model such that $|C - C'| \leq \varepsilon$. In other words, a positive answer to Tsirelson's problem

[12] The reader would be justified in asking why we needed to consider two games in the first place. If we only considered $\mathcal{G}_M$, then we wouldn't be able to conclude that $\mathcal{G}_M$ has value either greater than 2/3 or at most 1/3, and thus M could in principle run forever. By defining M in this way we force the resulting game $\mathcal{G}_M$ to satisfy the promise of A.

would establish that correlations in the commuting operator model can be approximated arbitrarily well by correlations in the tensor product model. The next corollary shows that Conjecture 8.1 would yield a negative resolution of Tsirelson's problem.

Corollary 8.8. *Suppose Conjecture 8.1 is true. Then there exists $\varepsilon > 0$, integers $n, m > 0$, and a 15-partite, n-input, m-output correlation C that is achievable in the commuting operator model that has distance at least ε from any correlation C' achievable in the tensor product model.*

Proof. Suppose not, i.e. any 15-partite correlation C achievable in the commuting operator model, for all $\varepsilon > 0$ there exists a correlation C' achievable in the tensor product model such that $|C - C'| \leq \varepsilon$. This implies that for any 15-prover nonlocal game $\mathcal{G}$, the entangled value of $\mathcal{G}$ in the tensor product (denoted by $\omega^*_{tp}(\mathcal{G})$) and commuting operator models (denoted by $\omega^*_c(\mathcal{G})$) are equal: for every $\delta > 0$, let $\mathcal{S}_c$ be a commuting operator strategy in a 15-prover nonlocal game $\mathcal{G}$ such that $\omega^*_c(\mathcal{G}) \leq \omega^*_{\mathcal{S}_c}(\mathcal{G}) + \delta$. Then by our assumption, for all $\varepsilon > 0$ there is a tensor product model strategy $\mathcal{S}_{tp}$ such that

$$\left|\omega^*_{\mathcal{S}_{tp}}(\mathcal{G}) - \omega^*_{\mathcal{S}_c}(\mathcal{G})\right| \leq \varepsilon.$$

By taking $\varepsilon = \delta$, for every $\delta > 0$ we have that there is a strategy $\mathcal{S}_{tp}$ in the tensor product model such that $\left|\omega^*_{\mathcal{S}_{tp}}(\mathcal{G}) - \omega^*_c(\mathcal{G})\right| \leq 2\delta$. Since the entangled value in the tensor product model is defined as the supremum over tensor product model strategies, by taking $\delta \to 0$ we get that $\omega^*_c(\mathcal{G}) = \omega^*_{tp}(\mathcal{G})$.

We provide an algorithm that decides if the value of a 15-prover nonlocal game is larger than 2/3, or at most 1/3, promised that one is the case. The algorithm interleaves two procedures. The first procedure exhaustively searches for strategies in the tensor product model of increasing dimension, and with increasing accuracy. If this procedure returns a value that is larger than 1/2, the algorithm halts and returns YES. A second procedure computes a non-increasing sequence of upper bounds by solving semidefinite programs obtained at increasing levels of the hierarchy introduced in [DLTW08, NPA08]. If this procedure returns a value that is smaller than 1/2, the algorithm halts and returns NO.

We show that this algorithm always halts, and always returns the correct decision. It is clear that the first procedure provides a non-decreasing sequence that converges to the value of the game in the tensor product model from below. Conversely, it is known that the second procedure provides a non-increasing sequence that converges to the value of the game in the commuting operator model from above. Since the values in both models coincide, this implies that the algorithm described in the previous paragraph always halts with the correct decision.

However, Conjecture 8.1 and Theorem 8.7 implies that there is no such algorithm, a contradiction. Thus there is a correlation C achievable in the commuting operator model that cannot be approximated arbitrarily well by correlations in the tensor product model. □

References

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[AVDK⁺08] Dorit Aharonov, Wim Van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787, 2008.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[Boo58] William W. Boone. The word problem. *Proceedings of the National Academy of Sciences*, 44(10):1061–1065, 1958.

[BVY17] Mohammad Bavarian, Thomas Vidick, and Henry Yuen. Hardness amplification for entangled games via anchoring. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 303–316. ACM, 2017.

[CHTW04] Richard Cleve, Peter Hoyer, Benjamin Toner, and John Watrous. Consequences and limits of nonlocal strategies. In *Computational Complexity, 2004. Proceedings. 19th IEEE Annual Conference on*, pages 236–249. IEEE, 2004.

[Con76] Alain Connes. Classification of injective factors cases II_1, II_∞, III_λ, $\lambda \neq 1$. *Annals of Mathematics*, pages 73–115, 1976.

[CPGW15] Toby S Cubitt, David Perez-Garcia, and Michael M Wolf. Undecidability of the spectral gap. *Nature*, 528(7581):207, 2015.

[CRSV16] Rui Chao, Ben W Reichardt, Chris Sutherland, and Thomas Vidick. Test for a large amount of entanglement, using few measurements. *arXiv preprint arXiv:1610.00771*, 2016.

[CS96] Robert Calderbank and Peter W Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098, 1996.

[CS17] Andrea Coladangelo and Jalex Stark. Robust self-testing for linear constraint system games. *arXiv preprint arXiv:1709.09267*, 2017.

[CS18] Matthew Coudron and William Slofstra. Complexity lower bounds for approximating entangled games to high precision. 2018.

[DLTW08] Andrew C Doherty, Yeong-Cherng Liang, Ben Toner, and Stephanie Wehner. The quantum moment problem and bounds on entangled multi-prover games. In *Computational Complexity, 2008. CCC'08. 23rd Annual IEEE Conference on*, pages 199–210. IEEE, 2008.

[Fri12] Tobias Fritz. Tsirelson's problem and Kirchberg's conjecture. *Reviews in Mathematical Physics*, 24(05):1250012, 2012.

[FV15] Joseph Fitzsimons and Thomas Vidick. A multiprover interactive proof system for the local Hamiltonian problem. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 103–112. ACM, 2015.

[IKW12] Tsuyoshi Ito, Hirotada Kobayashi, and John Watrous. Quantum interactive proofs with weak error bounds. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 266–275. ACM, 2012.

[IV12] Tsuyoshi Ito and Thomas Vidick. A multi-prover interactive proof for NEXP sound against entangled provers. *Proc. 53rd FOCS*, pages 243–252, 2012.

[Ji16] Zhengfeng Ji. Classical verification of quantum proofs. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 885–898. ACM, 2016.

[Ji17] Zhengfeng Ji. Compression of quantum multi-prover interactive proofs. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 289–302. ACM, 2017.

[JJUW10] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. QIP = PSPACE. *Communications of the ACM*, 53(12):102–109, 2010.

[JLV18] Zhengfeng Ji, Debbie Leung, and Thomas Vidick. A three-player coherent state embezzlement game. *arXiv preprint arXiv:1802.04926*, 2018.

[JMVW16] N. Johnston, R. Mittal, Russo V., and J. Watrous. Extended nonlocal games and monogamy-of-entanglement games. *Proceedings of the Royal Society A*, 472:20160003, 2016.

[JNP$^+$11] Marius Junge, Miguel Navascues, Carlos Palazuelos, D Perez-Garcia, Volkher B Scholz, and Reinhard F Werner. Connes' embedding problem and Tsirelson's problem. *Journal of Mathematical Physics*, 52(1):012102, 2011.

[Kar82] OG Karlampovič. A finitely presented solvable group with unsolvable word problem. *Mathematics of the USSR-Izvestiya*, 19(1):151, 1982.

[KSV02] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47. American Mathematical Soc., 2002.

[Nov55] P. S. Novikov. On the algorithmic unsolvability of the word problem in group theory. *Trudy Mat. Inst. Steklov.*, 44:3–143, 1955.

[NPA08] Miguel Navascués, Stefano Pironio, and Antonio Acín. A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations. *New Journal of Physics*, 10(7):073013, 2008.

[NV17a] Anand Natarajan and Thomas Vidick. A quantum linearity test for robustly verifying entanglement. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 1003–1015, New York, NY, USA, 2017. ACM.

[NV17b] Anand Natarajan and Thomas Vidick. Two-player entangled games are NP-hard. 2017.

[NV18] Anand Natarajan and Thomas Vidick. Low-degree testing for quantum states. 2018.

[ON02] Tomohiro Ogawa and Hiroshi Nagaoka. A new proof of the channel coding theorem via hypothesis testing in quantum information theory. In *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, page 73. IEEE, 2002.

[Oza13] Narutaka Ozawa. About the Connes embedding conjecture. *Japanese Journal of Mathematics*, 8(1):147–183, 2013.

[PF79] Nicholas Pippenger and Michael J Fischer. Relations among complexity measures. *Journal of the ACM (JACM)*, 26(2):361–381, 1979.

[Shi02] Yaoyun Shi. Both Toffoli and controlled-NOT need little help to do universal quantum computation. *arXiv preprint quant-ph/0205115*, 2002.

[Slo16] William Slofstra. Tsirelson's problem and an embedding theorem for groups arising from non-local games. *arXiv preprint arXiv:1606.03140*, 2016.

[Slo17] William Slofstra. The set of quantum correlations is not closed. *arXiv preprint arXiv:1703.08618*, 2017.

[Ste96a] Andrew Steane. Multiple-particle interference and quantum error correction. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 452, pages 2551–2577. The Royal Society, 1996.

[Ste96b] Andrew M Steane. Error correcting codes in quantum theory. *Physical Review Letters*, 77(5):793, 1996.

[Vid13] Thomas Vidick. Three-player entangled XOR games are NP-hard to approximate. In *Proc. 54th FOCS*, 2013.

[Wat09] John Watrous. Quantum computational complexity. In *Encyclopedia of complexity and systems science*, pages 7174–7201. Springer, 2009.

A Succinct representation of uniform circuit families

In this appendix we show that any uniformly generated family of circuits has a succinct description, in the sense of Section 3.4. First we introduce a generic method for constructing a circuit that implements the same computation as a Turing machine. Then, we show that any such circuit can be written in a regular form, that has a succinct description. Finally, we apply these two steps for the case of a Turing machine that specifies a family of circuits.

A.1 Simulation of a Turing machine with a quantum circuit

A universal Turing machine simulator circuit is a quantum circuit TMSIM that, given as input the description of a Turing machine M, a positive integer time T, and a designated output tape for M, computes the contents of the output tape after M has been executed for T steps.

Lemma A.1. *For any integer $k \geq 1$ there exists a family of quantum circuits $\{\mathrm{TMSIM}_k(T)\}_{T\in\mathbb{N}}$ of size $\mathrm{poly}(T)$ such that the following hold for all $T \geq 1$.*

1. $\mathrm{TMSIM}_k(T)$ *acts on registers* S *(the* Turing machine state register*),* M *(the* Turing machine specification register*), and* $\mathsf{A}_1, \dots \mathsf{A}_k$*(the* Turing machine tape registers*).*

2. *Let M be the classical description of a k-tape Turing machine and $a = (a_1, \dots, a_k)$ be a k-tuple of strings of symbols for the k tapes of M, such that each a_i has length at most the size of A_i. Let $a' = (a'_1, \dots, a'_k)$ be the contents of M's tapes after it has been executed for T steps, starting from the tape values specified by a. Then after the circuit $\mathrm{TMSIM}_k(T)$ has been executed on input $|0\rangle_{\mathsf{S}} \otimes |M\rangle_{\mathsf{M}} \otimes |a\rangle_{\mathsf{A}_1 \cdots \mathsf{A}_k}$, the registers $\mathsf{A}_1, \dots, \mathsf{A}_k$ are in state $|a'\rangle_{\mathsf{A}_1 \cdots \mathsf{A}_k}$.*

Furthermore, there exists a deterministic Turing machine $\mathrm{TMSIM\text{-}DESC}_k$ *that on input T and an integer t in binary runs in polynomial time and returns a description of the t-th gate of* $\mathrm{TMSIM}_k(T)$ *when it exists, and a special failure symbol when it does not.*

Proof. Fix an integer $k \geq 1$ and let U be a universal $(k+1)$-tape Turing machine. When provided as input the description of a k-tape Turing machine M and a number of steps 1^T on its first tape, and some values a on the remaining k tapes, U performs the computation of M on input a for T steps. Furthermore, U runs in polynomial time, and we assume without loss of generality [PF79] that U is *oblivious*: the movements of the head of U are independent of its input. Without loss of generality, each tape head of U alternates between weeping left for T steps and then right for T steps, and the heads move in sequence (i.e., the first tape's head moves first, then the second tape's head moves, and so on).

The circuit $\mathrm{TMSIM}_k(T)$ is defined as follows. The register S stores the state of the universal Turing machine U. The register M stores the description of the k-tape Turing machine M. The registers $\{\mathsf{A}_j\}$ store the contents of the work tapes of M. Each movement of the heads of U is implemented by a layer in the circuit. The computation of the head transition function is computed in register S, which is connected via two-qubit gates to the corresponding locations in the registers A_j. (Due to the assumption that U is oblivious, these locations only depend on the index of the layer in the circuit.)

The number of gates of $\mathrm{TMSIM}_k(T)$ is clearly polynomial, establishing item 1. in the lemma. Furthermore, item 2. holds by construction.

For the "Furthermore" part of the lemma, note that the structure of each layer is identical, with the only difference being that the gates that cross between S and the registers $\{\mathsf{A}_j\} \cup \{\mathsf{M}\}$ are different depending on which cells of the tapes are supposed to be read/written to at that layer. Using that U is oblivious, the location of the t-th gate of $\mathrm{TMSIM}_k(T)$ can be computed in time polynomial in t. □

A.2 Simulating regular circuits

Analogously to the circuit TMSIM that simulates a Turing Machine, we introduce the notion of a universal circuit CKTSIM that simulates an arbitrary quantum circuit. For purposes of efficient description it is convenient to consider *regular circuits*, which are defined as follows.

Definition A.2. *An n-qubit* regular circuit *of size s is specified by a sequence of gates $g_1, \ldots, g_s$ where each $g_i \in \{H, T\}$, and the set of qubits that the gate g_i acts on only depends on the triple (i, n, s), and can be computed in polynomial time from the triple (i, n, s) specified in binary. (For consistency, the Hadamard gate is interpreted as a 3-qubit gate $I \otimes H \otimes I$.)*

We record the easy observation that every n-qubit circuit of size s has an equivalent regular circuit of size $\mathrm{poly}(n, s)$ as the following lemma.

Lemma A.3. *There exists a deterministic polynomial-time Turing machine that takes as input the description of a quantum circuit C and outputs a regular quantum circuit C' that implements the same unitary transformation as C does.*

The next lemma establishes the existence of a simulation procedure for circuits analogous to the one shown for Turing machines in Lemma A.1.

Lemma A.4. *There is a family of quantum circuits $\{\mathrm{CKTSIM}_{n,s}\}_{n,s\geq 1}$ of size $\mathrm{poly}(n,s)$ such that the following hold. For any $n, s \geq 1$ the circuit $\mathrm{CKTSIM}_{n,s}$ acts on two registers A (the* circuit specification register*) and B (the* target register*), where B has n qubits. For any $C \in \{0,1\}^s$ and state $|\theta\rangle_{\mathsf{B}}$*

$$\mathrm{CKTSIM}_{n,s}\Big(|C\rangle_{\mathsf{A}} \otimes |\theta\rangle_{\mathsf{B}}\Big) = |C\rangle_{\mathsf{A}} \otimes C|\theta\rangle_{\mathsf{B}} ,$$

where C is interpreted as the description of a regular n-qubit quantum circuit of size s.
Furthermore, there exists a deterministic Turing machine CKTSIM-DESC *that on input (n, s, t) runs in*

polynomial time and returns a description of the t-th gate of $\mathrm{CKTSIM}_{n,s}$ *when it exists, and a special failure symbol when it does not.*

Proof. For $n, s \geq 1$ the circuit $\mathrm{CKTSIM}_{n,s}$ has s layers, where the i-th layer applies either a Hadamard or a Toffoli gate, depending on g_i, on the appropriate qubits. The indices of those qubits can be computed in $\mathrm{poly}(\log n, \log s)$ time. □

A.3 Succinct representation of uniform families of circuits

Lemma A.5. *Let* $\{C_n\}_{n\geq 1}$ *be family of circuits that is uniformly generated by the Turing machine M. Then there exists a deterministic Turing machine G, that is computable from M, such that on input* (n, t)*, where both n and t are integer written in binary, G runs in polynomial time and returns a description of the t-th gate of a regular circuit* C'_n *that implements the same unitary transformation as* C_n *(but uses additional ancilla registers).*

Proof. Without loss of generality assume the number of tapes used by M is $k = 3$, with an input tape, a work tape and an output tape. Let p_M be a polynomial that bounds the running time of M. Let $n \geq 1$. We describe the circuit C'_n. The circuit first initializes ancilla registers for TMSIM (see Lemma A.1) as follows. The register S contains the initial state of M. The register M contains a description of M. The registers $\mathsf{A}_1, \mathsf{A}_2, \mathsf{A}_3$ are empty, except that the register A_1 associated with the input tape contains the input 1^n. The next step in the circuit C'_n is to execute the circuit $\mathrm{TMSIM}_k(p_M(n))$ on these registers to obtain a description of C_n. Using Lemma A.3 we may without loss of generality assume that C_n is regular. Finally, the last step in the circuit C'_n it to execute the circuit CKTSIM on the register A_3 associated with the output tape of M, that contains the description of C_n and plays the role of the circuit specification register, and the target register, that is identified with the register containing the input state to C_n.

It is clear that C'_n implements the same transformation as C_n. The existence of the Turing machine G follows directly from the description of C'_n and the existence of the Turing machines TMSIM-DESC and CKTSIM-DESC from Lemma A.1 and Lemma A.4 respectively. Specifically, from its input (n, t), G may efficiently determine which of its three phases (input preparation, TMSIM, CKTSIM) the t-th gate of C'_n is associated with, and then compute the gate itself using the appropriate succinct description Turing machine. □