On the Complexity of Zero Gap MIP*

Hamoon Mousavi
Department of Computer Science, University of Toronto, Canada
hmousavi@cs.toronto.edu

Seyed Sajjad Nezhadi
Department of Computer Science, University of Toronto, Canada
sajjad.nezhadi@mail.utoronto.ca

Henry Yuen
Department of Computer Science and Department of Mathematics, University of Toronto, Canada
hyuen@cs.toronto.edu

Abstract

The class MIP^* is the set of languages decidable by multiprover interactive proofs with quantum entangled provers. It was recently shown by Ji, Natarajan, Vidick, Wright and Yuen that MIP^* is equal to RE, the set of recursively enumerable languages. In particular this shows that the complexity of approximating the quantum value of a non-local game G is equivalent to the complexity of the Halting problem.

In this paper we investigate the complexity of deciding whether the quantum value of a non-local game G is *exactly* 1. This problem corresponds to a complexity class that we call *zero gap* MIP^*, denoted by MIP_0^*, where there is no promise gap between the verifier's acceptance probabilities in the YES and NO cases. We prove that MIP_0^* extends beyond the first level of the arithmetical hierarchy (which includes RE and its complement coRE), and in fact is equal to Π_2^0, the class of languages that can be decided by quantified formulas of the form $\forall y\, \exists z\, R(x,y,z)$.

Combined with the previously known result that MIP_0^{co} (the *commuting operator* variant of MIP_0^*) is equal to coRE, our result further highlights the fascinating connection between various models of quantum multiprover interactive proofs and different classes in computability theory.

2012 ACM Subject Classification Theory of computation → Interactive proof systems; Theory of computation → Computability; Theory of computation → Quantum complexity theory

Keywords and phrases Quantum Complexity, Multiprover Interactive Proofs, Computability Theory

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.87

Category Track A: Algorithms, Complexity and Games

Related Version Arxiv version available at `https://arxiv.org/abs/2002.10490`.

Funding *Hamoon Mousavi*: was supported by the Ontario Graduate Scholarship (OGS).
Henry Yuen: was supported by NSERC Discovery Grant 2019-06636.

Acknowledgements We thank Matt Coudron, Thomas Vidick, and especially William Slofstra for numerous helpful discussions. We also thank the reviewers of ICALP 2020 for suggestions to improve the presentation.

1 Introduction

A two-player *non-local game* is played between a verifier and two cooperating players named Alice and Bob who cannot communicate with each other once the game starts. During the game, the verifier samples a pair of questions (x,y) from a joint distribution μ, sends x to Alice and y to Bob, who respond with answers a and b respectively. The verifier accepts if and only if $D(x,y,a,b) = 1$ for some predicate D. The *quantum value* of a non-local game G, denoted by $\omega_q(G)$, is defined to be the supremum of the verifier's acceptance probability over all possible finite dimensional quantum strategies of Alice and Bob for the game G.

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 87; pp. 87:1–87:12
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

What is the complexity of computing the quantum value of non-local games? In [16], Slofstra proved that the problem of determining whether a given game G has $\omega_q(G) = 1$ is *undecidable.* Recently, it was shown that *approximating* $\omega_q(G)$ up to any additive constant is also an uncomputable problem [11]. In particular, there is a computable reduction from Turing machines M to non-local games G_M such that if M halts (when run on an empty input), then $\omega_q(G_M) = 1$, and otherwise $\omega_q(G_M) \leq \frac{1}{2}$. Since determining whether a given Turing machine halts (i.e. the Halting problem) is undecidable, so is the problem of determining whether the quantum value of a non-local game is 1 or at most $\frac{1}{2}$.

Conversely, one can reduce the problem of approximating the quantum value of non-local games to the Halting problem; there is an algorithm that for every non-local game G exhaustively searches over finite-dimensional strategies of increasing dimension to find one that succeeds with probability close to 1 (above 0.99, say). If $\omega_q(G) = 1$ then the algorithm is guaranteed to find such a strategy; otherwise if $\omega_q(G) \leq 1/2$ the algorithm will run forever. In complexity-theoretic terms, this shows that the class MIP^*, the set of languages decidable by multiprover interactive proofs with quantum provers, is equal to RE, the set of recursively enumerable languages (i.e. the class for which the Halting problem is complete).

In this paper, we return to the problem originally investigated by Slofstra [16]: what is the complexity of deciding if $\omega_q(G)$ is *exactly* equal to 1 for nonlocal games G? This corresponds to the complexity class that we call *zero gap* MIP^*, denoted by MIP^*_0. In this model of interactive proofs, in the YES case (i.e. $x \in L$), there is a sequence of finite-dimensional prover strategies that cause the verifier to accept with probability approaching 1. In the NO case (i.e. $x \notin L$), all finite-dimensional prover strategies are rejected with positive probability – but could be arbitrarily close to 0. It is easy to see that $\mathsf{MIP}^* \subseteq \mathsf{MIP}^*_0$ and thus MIP^*_0 contains undecidable languages. Furthermore, we know that MIP^*_0 *cannot* be equal to MIP^*; the results of [16, 5] imply that coRE, the complement of RE, is also contained in MIP^*_0. Since $\mathsf{RE} \neq \mathsf{coRE}$, this implies that MIP^*_0 strictly contains $\mathsf{MIP}^* = \mathsf{RE}$.

What problems can be reduced to the task of exactly computing the quantum value of non-local games, rather than "just" approximating it? We characterize the class MIP^*_0 by showing that it is equal to Π^0_2, a class that belongs to the *arithmetical hierarchy* from computability theory. The arithmetical hierarchy is defined by classes of languages decidable via formulas with alternating quantifiers. For example, the class RE is equal to the class Σ^0_1, which is the set of languages L of the form $\{x : \exists y.\, R(x,y) = 1\}$ for some decidable predicate R. The class coRE is equal to Π^0_1, the set of languages of the form $\{x : \forall y.\, R(x,y) = 1\}$. The class Π^0_2 is the set of languages L of the form $\{x : \forall y.\, \exists z.\, R(x,y,z) = 1\}$.

An equivalent definition of the class Π^0_2 is that it is the set of languages L such that there is a Turing machine A that has *oracle access* to the Halting problem, and $x \notin L$ if and only if $A(x) = 1$. It is known that Π^0_2 strictly contains $\Sigma^0_1 = \mathsf{RE}$. This shows that MIP^*_0 contains problems that are *harder* (in a computability sense) than the Halting problem.

We specifically show that there exists a computable reduction from Π^0_2 languages to the problem of deciding whether a *three-player* non-local game G has quantum value 1. It is likely that a similar reduction holds for two-player non-local games but we leave this for future work. We also show that the problem of deciding if a non-local game has quantum value 1 can be reduced to a Π^0_2 language, thus establishing the equality $\mathsf{MIP}^*_0 = \Pi^0_2$.

This paper, combined with the results of [11] and [16], paints a fascinating landscape about the complexity of quantum multiprover interactive proofs, in which there are four different complexity classes to consider. The first two are MIP^* and MIP^*_0, which we defined already. The second two are $\mathsf{MIP}^{\mathsf{co}}$ and its zero-gap variant $\mathsf{MIP}^{\mathsf{co}}_0$. The class $\mathsf{MIP}^{\mathsf{co}}$ stands for languages that are decidable by quantum multiprover interactive proofs in the *commuting* *operator* model: here, the provers are allowed to use infinite-dimensional quantum strategies, and the measurement operators of Alice only need to commute with those of Bob (rather than be in tensor product).

$$\begin{tikzcd}
\Sigma_2^0 & & \boldsymbol{\Pi_2^0} = \mathsf{MIP}_{\boldsymbol{0}}^* \\
& \Delta_2^0 \arrow[ul] \arrow[ur] & \\
\mathsf{MIP}^* = \Sigma_1^0 \arrow[uu] \arrow[ur] & & \Pi_1^0 = \mathsf{MIP}_0^{\mathsf{co}} \overset{?}{=} \mathsf{MIP}^{\mathsf{co}} \arrow[uu] \arrow[ul] \\
& \Delta_1^0 \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Figure 1 The computability landscape of quantum multiprover interactive proofs. Arrows denote inclusion. The set Δ_1^0 denotes the set of all decidable languages. The set Σ_1^0 denotes the recursively enumerable languages, and Π_1^0 denotes the set of co-recursively enumerable languages. It is known that $\mathsf{MIP}^{\mathsf{co}} \subseteq \mathsf{MIP}_0^{\mathsf{co}}$, but unknown whether they are equal.

One of the consequences of the fact that $\mathsf{MIP}^* = \mathsf{RE}$ is that $\mathsf{MIP}^{\mathsf{co}} \neq \mathsf{MIP}^*$. This is because $\mathsf{MIP}^{\mathsf{co}} \subseteq \mathsf{coRE}$, due to the fact that the commuting operator value of a non-local game can be upper-bounded using a convergent sequence of semidefinite programs [14, 4]. It is also the case that $\mathsf{MIP}_0^{\mathsf{co}} \subseteq \mathsf{coRE}$, and in fact equality holds due to [16, 3]. It remains an open question to determine if $\mathsf{MIP}^{\mathsf{co}} = \mathsf{MIP}_0^{\mathsf{co}} = \mathsf{coRE}$.

There are a number of curious and counter-intuitive aspects about this landscape of complexity for non-local games. First, if $\mathsf{MIP}^{\mathsf{co}} = \mathsf{coRE}$, then there would be a pleasing symmetry in that $\mathsf{MIP}^* = \mathsf{RE}$ and $\mathsf{MIP}^{\mathsf{co}} = \mathsf{coRE}$ (even though the "co" refer to different things on each side of the equation!). On the other hand, we have that $\mathsf{MIP}_0^* = \Pi_2^0$ and $\mathsf{MIP}_0^{\mathsf{co}} = \mathsf{coRE}$, meaning that – in the zero gap setting – there are *more* languages that can be verified with provers using (a limit of) finite-dimensional strategies than can be decided with provers using infinite-dimensional commuting operator strategies! Of course, in the setting of interactive proofs, giving provers access to more resources can change the complexity of the interactive proof model in unexpected ways.

1.1 Proof overview

We prove the lower bound $\Pi_2^0 \subseteq \mathsf{MIP}_0^*$ by combining two components: first we leverage the result of [11] that $\mathsf{MIP}^* = \mathsf{RE}$ as a black box, which implies that there is a quantum multiprover interactive proof for the Halting problem. Next, we use a *compression theorem* for quantum multiprover interactive proofs that was proved in [5]. A compression theorem, roughly speaking, states that given a verifier V for a quantum multiprover interactive protocol (which can be modeled as a Turing machine with tapes to receive/send messages to the provers), one can compute a much more time-efficient verifier V' whose quantum value is related in some predictable way to the quantum value of V. Several recent results about the complexity of non-local games crucially rely on proving compression theorems with various properties [10, 5, 13, 11].

In more detail, the compression theorem of [5] (which in turn is a refinement of the compression theorem of [10]) states that given a description of a verifier V, one can compute a description of a three-player[1] non-local game G_V (which is a multiprover protocol with only one round of interaction) whose properties are as follows:

1. The time complexity of the verifier in G_V is *polylogarithmic* in the time complexity of V.
2. The quantum value of the protocol executed by V is related to the quantum value of G_V in the following manner:

$$\omega_q(G_V) \geq \frac{1}{2} + \frac{1}{2}\omega_q(V)$$

and furthermore if $\omega_q(V) < 1$ then $\omega_q(G_V) < 1$.

The utilization of the compression theorem of [5] is the reason why the main result of this paper holds for three-player non-local games, rather than two.

We call this compression theorem a "zero gap" compression theorem, because it does not preserve any promise gap on the value of the input verifier V: if the value of V is promised to be either 1 or 1/2, then G_V is only guaranteed to have value either 1 or 3/4. If we iterate this compression procedure, then we get a promise gap that goes to zero. In contrast, the compression theorem used to prove $\mathsf{MIP}^* = \mathsf{RE}$ *is* gap-preserving.

The zero gap compression theorem was used to prove that $\mathsf{coRE} \subseteq \mathsf{MIP}_0^*$ in [5]. At a high level, this is shown by constructing a verifier that recursively calls the zero gap compression procedure on itself. In this paper, we follow this approach, except we also embed an MIP^* protocol for RE inside the verifier that is recursively calling the zero gap compression procedure; this composition of protocols allows the verifier to verify languages in Π_2^0.

1.2 Further remarks

$\mathsf{MIP}^* = \mathsf{RE}$ is equivalent to gap-preserving compression

As mentioned, the key to proving $\mathsf{MIP}^* = \mathsf{RE}$ [11] was establishing a gap-preserving compression theorem for non-local games, albeit for a special case of non-local games satisfying a so-called "normal form" property. In Section 4, we present a relatively simple – but in our opinion quite interesting – observation that $\mathsf{MIP}^* = \mathsf{RE}$ is in some sense, *equivalent* to a gap-preserving compression theorem.

A proof of $\mathsf{MIP}_0^* = \Pi_2^0$ under weaker assumptions?

One might wonder if there might be an elementary way of proving that $\mathsf{MIP}_0^* = \Pi_2^0$, *without* relying on the statement that $\mathsf{MIP}^* = \mathsf{RE}$. For example, the results of [16, 5] show that $\mathsf{coRE} \subseteq \mathsf{MIP}_0^*$ and furthermore [16] shows that $\mathsf{coRE} = \mathsf{MIP}_0^{co}$. These previous "zero-gap results" do not appear to have the same mathematical consequences as $\mathsf{MIP}^* = \mathsf{RE}$ (e.g. yielding a negative answer to Connes' embedding problem if $\mathsf{RE} \subseteq \mathsf{MIP}^*(2)$, the two-player variant of MIP^*), which suggests the intuition that characterizing the complexity of *exactly* computing the quantum (or commuting operator) value of nonlocal games may be fundamentally easier than characterizing the complexity of *approximating* it.

This intuition is not entirely correct: the "zero-gap" statement $\mathsf{MIP}_0^* = \Pi_2^0$ is already enough to yield a negative answer to Tsirelson's problem: there exists a k where k-partite commuting operator correlations cannot be approximated by finite dimensional correlations.

[1] The results of [5] are stated for games with 15 players, but can be improved to hold for 3-player games by using a different error correcting code in the construction.

Put another way, if Tsirelson's problem has a positive answer, then the commuting operator and quantum values of games are always equal, and then $\mathsf{MIP}_0^* = \mathsf{MIP}_0^{co} = \mathsf{coRE}$. However, Π_2^0 strictly contains coRE – thus Tsirelson's problem has a negative answer. Furthermore, Tsirelson's problem for $k = 2$ is known to be equivalent to Connes' embedding problem [6, 12, 15].

This suggests that our characterization of the class MIP_0^* must necessarily involve a nontrivial tool such as $\mathsf{MIP}^* = \mathsf{RE}$.

1.3 Open problems

We list some open problems.

1. Just as the complexity statement $\mathsf{MIP}^* = \mathsf{RE}$ has consequences for questions in pure mathematics (such as the Connes' embedding problem), does the equality $\mathsf{MIP}_0^* = \Pi_2^0$ have any implications for operator algebras? We believe there may be a connection to model-theoretic approaches to the Connes' embedding problem (see, e.g., [8, 7]).
2. What is the complexity of $\mathsf{MIP}^{\mathsf{co}}$? Is it equal to coRE?
3. Can the reduction from Π_2^0 languages to the problem of deciding whether $\omega_q(G) = 1$ be improved to hold for two-player games G?
4. We showed that, essentially, $\mathsf{MIP}^* = \mathsf{RE}$ implies a gap-preserving compression theorem. Can one show that it also implies in a black-box fashion, a zero gap compression theorem, of the same kind as proved in [5]? This then proves that $\mathsf{MIP}^* = \mathsf{RE}$ directly implies $\mathsf{MIP}_0^* = \Pi_2^0$.
5. Does $\mathsf{MIP}_0^* = \Pi_2^0$ imply $\mathsf{MIP}^* = \mathsf{RE}$ in a "black-box" fashion?

2 Preliminaries

We write $\mathbb{N}$ to denote the natural numbers $\{1, 2, 3, \ldots\}$. All logarithms are base 2. For a string $x \in \{0,1\}^*$ let $|x|$ denote the length of x. We let

$$\log^*(n) = \begin{cases} 0, & n \leq 1 \\ 1 + \log^*(\log(n)), & n > 1 \end{cases}$$

denote the iterated logarithm function.

2.1 Turing machines and the arithmetical hierarchy

A total Turing machine is one that halts on every input. Fix a string encoding of Turing machines, and for a Turing machine M, let $|M|$ denote the length of the encoding of M.

▶ **Proposition 1** (Universal Turing machine). *There exists a universal constant $C > 0$ and a universal Turing machine $\mathfrak{U}$ that, given an input pair (M, x) where M is an encoding of a Turing machine, computes $M(x)$ in time $C \max(|M|, \mathsf{TIME}(M,x))^2$, where $\mathsf{TIME}(M,x)$ is the number of steps taken by M on input x before it halts.*

▶ **Definition 2.** *The i-th level of the* arithmetical hierarchy *contains 3 classes Σ_i^0, Π_i^0, and Δ_i^0. The class Σ_i^0 is the set of languages defined as*

$$L = \{x \in \{0,1\}^* : \exists y_1 \forall y_2 \exists y_3 \cdots Q\, y_i\, R(x, y_1, \cdots, y_i) = 1\}$$

for some total Turing machine R, where Q is the $\forall$ quantifier when i is even and otherwise is the $\exists$ quantifier. The class Π_i^0 is the complement of Σ_i^0, and $\Delta_i^0 = \Sigma_i^0 \cap \Pi_i^0$.

In particular the first level of the arithmetical hierarchy corresponds to the classes $\Sigma_1^0 = \mathsf{RE}$, $\Pi_1^0 = \mathsf{coRE}$, and Δ_1^0 the set of decidable languages $\mathsf{RE} \cap \mathsf{coRE}$.

2.2 Interactive verifiers

In this section, we model multiprover interactive protocols, which is specified by a *verifier V*, as a randomized algorithm. In the protocol, the verifier V interacts with multiple provers, and at the end of the protocol the verifier outputs a bit indicating whether to accept or reject. A verifier can be identified with the interactive protocol it executes, and vice versa.

In more detail, define a *k-input, r-prover verifier V* to be a randomized interactive Turing machine that has k designated input tapes, r communication tapes, a single workspace tape, and a single output tape. An interaction with r provers is executed in the following way: the Turing machine V alternates between computation and communication phases; in the computation phase, the Turing machine behaves like a normal Turing machine with $k+r+2$ tapes, and it may halt and indicate accept or reject on the output tape. It can also pause its computation and go into a communication phase, in which case the contents of each of i-th communication tape is read by the i-th prover, who then edits the i-th communication tape with its answer. After all the provers have finished with their responses, the next computation phase resumes. This is the standard way of modeling interactive Turing machines [1]. In this formulation, a non-local game is simply specified by a 0-input, 2-prover verifier V that has only one communication phase.

Given a k-input, r-prover verifier V, define its *time complexity* with respect to a k-tuple of inputs $(x_1, \ldots, x_k)$ to be the maximum number of time steps taken by the verifier V when it is initialized with $(x_1, \ldots, x_k)$ on its k input tapes, over all possible responses of the r-provers, before it halts. We denote this by $\mathsf{TIME}(V(x_1, \ldots, x_k))$.

We now define, in a somewhat informal level, *finite-dimensional prover strategies* (or simply a *strategy*) $\mathcal{S}$ for the interaction specified by a k-input, r-prover verifier V. This is a specification of the following data:

1. Local dimension $d \in \mathbb{N}$,
2. A state $|\psi\rangle \in (\mathbb{C}^d)^{\otimes r}$, and
3. For every prover i, for every round $t \in \mathbb{N}$, for every string $\pi \in \{0,1\}^*$, a POVM $\{M^a_{i,t,\pi}\}_a$ acting on $\mathbb{C}^d$.

Given a verifier V, a k-tuple $(x_1, \ldots, x_k)$, and a prover strategy $\mathcal{S}$ for V, the interaction proceeds as follows: at the beginning of the protocol, the provers share the state $|\psi\rangle$, and the verifier's input tapes are initialized to $(x_1, \ldots, x_k)$. At round t, the i-th prover performs the measurement $\{M^a_{i,t,\pi}\}_a$ on its local space to obtain an outcome a, where π is the *history* of all the messages seen by prover i in all previous rounds (including the message from the verifier in the t-th round). It then writes outcome a on the i-th communication tape of the verifier. Thus at each round the shared state between the provers depend on the outcomes of their measurements, and evolves probabilistically over time. The *value of strategy $\mathcal{S}$ in the interaction with verifier V on input* $(x_1, \ldots, x_k)$ is defined to be the probability that the verifier halts and accepts. We denote this by $\omega_q(V(x_1, \ldots, x_k), \mathcal{S})$. The *quantum value of verifier V on input* $(x_1, \ldots, x_k)$ is defined to be the supremum of $\omega_q(V(x_1, \ldots, x_k), \mathcal{S})$ over all finite-dimensional strategies $\mathcal{S}$, which we denote by $\omega_q(V(x_1, \ldots, x_k))$.

▶ **Definition 3.** *Let $m, r \in \mathbb{N}$ and let $0 \le s \le c \le 1$. The class $\mathsf{MIP}^*[m, r, c, s]$ is defined to be the set of languages L for which there exists a verifier V and a polynomial $p(n)$ with the following properties:*

1. *V is a 1-input, r-prover verifier that halts after m communication phases.*
2. *For all x, $\mathsf{TIME}(V(x)) \le p(|x|)$.*
3. *If $x \in L$, then $\omega_q(V(x)) \ge c$.*
4. *If $x \notin L$, then $\omega_q(V(x)) < s$.*

We define the class MIP^* to be the union of $\mathsf{MIP}^*[m, r, c, s]$ for all $m, r \in \mathbb{N}$ and $c > s$. We define the class MIP^*_0 to be the union of $\mathsf{MIP}^*[m, r, 1, 1]$ over all $m, r \in \mathbb{N}$. In other words, in the YES case (i.e., $x \in L$), there is a sequence of finite-dimensional prover strategies that are accepted with probability approaching 1. In the NO case (i.e., $x \notin L$), there exists a positive $\varepsilon > 0$ (that generally depends on x) such that all finite dimensional strategies are rejected with probability at least ε.

2.3 Compression of quantum multiprover interactive protocols

In this section we formally present the two main ingredients used in our proof: the zero gap compression procedure of [5], and the reduction from the Halting problem to the problem of approximating the quantum value of a quantum multiprover interactive protocol.

First we introduce the definition of λ-boundedness, which specifies how both the description and time complexity of a verifier is bounded by a polynomial with exponent λ.

▶ **Definition 4.** *Let $\lambda \in \mathbb{N}$. A $(k+1)$-input r-prover verifier V is λ-bounded if for all integers $n \in \mathbb{N}$, $x_1, \dots, x_k \in \{0,1\}^*$, we have $\mathsf{TIME}(V(n, x_1, ..., x_k)) \leq \lambda(n \cdot |x_1| \cdots |x_k|)^\lambda$.*

Here, we assume that the first input to a verifier V is an integer $n \in \mathbb{N}$ which intuitively specifies a "complexity" parameter.

▶ **Theorem 5** (Zero-gap compression [5, Theorem 6.1]). *Let $r \geq 3$ be an integer. There exists a universal constant $C_{comp} \in \mathbb{N}$ such that for every $\lambda \in \mathbb{N}$, there exists a Turing machine $\mathrm{COMPRESS}_\lambda$ with the following properties. Given as input a $(k+1)$-input r-prover verifier V, the Turing machine $\mathrm{COMPRESS}_\lambda$ outputs a $(k+1)$-input r-prover verifier $V^\#$ in time $C_{comp}(|V| \cdot \lambda)^{C_{comp}}$ with the following properties: for all $x_1, \dots, x_k \in \{0,1\}^*$, we have*

1. *if V is λ-bounded, then $\omega_q(V^\#(n, x_1, ...x_k)) \geq \frac{1}{2} + \frac{1}{2}\omega_q(V(2^n, x_1, ...x_k))$,*
2. *if V is λ-bounded and $\omega_q(V(2^n, x_1, ...x_k)) < 1$, then $\omega_q(V^\#(n, x_1, ...x_k)) < 1$,*
3. *for all integers $n \in \mathbb{N}$, $x_1, \dots, x_k \in \{0,1\}^*$, we have $\mathsf{TIME}(V^\#(n, x_1, ..., x_k)) \leq C_{comp}(\lambda \cdot n \cdot |x_1| \cdots |x_k|)^{C_{comp}}$.*

The zero-gap compression theorem, as presented here, differs from the one presented in [5, Theorem 6.1]. For example, verifiers in [5] are described using so-called "Gate Turing Machines" (GTMs). However, using the same oblivious Turing machine simulation techniques as discussed in the appendix of [5], from a verifier V (as defined in this paper), we can obtain a GTM that specifies the same interactive protocol. Another difference, as remarked in the introduction, is that here the compression result applies to protocols with three or more players, whereas it is stated for protocols with 15 or more players in [5]. However, the results of [5] can be adapted to the case of three players by using a $[[3, 1, 2]]_3$ error detecting code with qutrits (instead of using the 7-qubit Steane code with qubits) [2].

Next we present the main result of [11], which presents a computable reduction from the Halting problem to the problem of approximating the quantum value of a non-local game.

▶ **Theorem 6** ($\mathsf{MIP}^* = \mathsf{RE}$ [11]). *There exists a Turing machine H and a universal constant $C_{\mathrm{HALT}} \in \mathbb{N}$ with the following properties. Given as input a Turing machine M, it runs in time $C_{\mathrm{HALT}}|M|^{C_{\mathrm{HALT}}}$ and outputs a 0-input 2-prover verifier $V_{\mathrm{HALT},M}$ such that*

1. *If M halts on empty tape then $\omega_q(V_{\mathrm{HALT},M}) = 1$, and otherwise $\omega_q(V_{\mathrm{HALT},M}) \leq \frac{1}{2}$.*
2. $\mathsf{TIME}(V_{\mathrm{HALT},M}) \leq C_{\mathrm{HALT}}|M|^{C_{\mathrm{HALT}}}$.

3 $\mathsf{MIP}^*_0 = \Pi^0_2$

We start this section by showing the upper bound $\mathsf{MIP}^*_0 \subseteq \Pi^0_2$.

▶ **Theorem 7.** $\mathsf{MIP}^*_0 \subseteq \Pi^0_2$

Proof. Let $L \in \mathsf{MIP}^*_0$. There exists a 1-input r-prover verifier V such that $x \in L$ iff $\omega_q(V(x)) = 1$ for all $x \in \{0,1\}^*$. Let $\mathbb{S}_{\varepsilon,d}$ be an ε-net for the space of strategies of dimension d; in particular, for every dimension-d strategy $\mathcal{S}$ there exists a strategy $\mathcal{S}' \in \mathbb{S}_{\varepsilon,d}$ such that for all verifiers V we have that $|\omega_q(V,\mathcal{S}) - \omega_q(V,\mathcal{S}')| \le \varepsilon$ (in other words, the winning probability of the strategies differ by at most ε). Because the set of strategies over a finite dimensional Hilbert space of a fixed dimension is a compact set [9], we can take $\mathbb{S}_{\varepsilon,d}$ to be a finite set. Let $\mathbb{S}_\varepsilon = \bigcup_{d\in\mathbb{N}} \mathbb{S}_{\varepsilon,d}$, and let $\{\mathcal{S}_\varepsilon(1), \mathcal{S}_\varepsilon(2), \ldots\}$ be an enumeration of strategies in $\mathbb{S}_\varepsilon$.

Consider the following total Turing machine T: On input triple (x,m,n) where $x \in \{0,1\}^*, m,n \in \mathbb{N}$. It outputs 1 if and only $\omega_q(V(x), \mathcal{S}_{1/2m}(n)) \ge 1 - 1/m$. Now it is easy to verify that

$$L = \{x : \forall m.\ \exists n. T(x,m,n) = 1\},$$

and therefore L is a Π^0_2 language.

To see this, let $x \in L$. Then $\omega_q(V(x)) = 1$, and for any gap (i.e. $\frac{1}{m}$) there exists a strategy S such that $\omega_q(V(x), S) \ge 1 - \frac{1}{2m}$. Choosing $\varepsilon = 1/2m$, then there must also exist a strategy $S' \in \mathbb{S}_{1/2m}$ such that $\omega_q(V(x), S') \ge \omega_q(V(x), S) - \frac{1}{2m} \ge 1 - \frac{1}{m}$. Therefore $\forall m.\ \exists n. T(x,m,n) = 1$.

Likewise, if $x \notin L$ then there exists $m \in \mathbb{N}$ for which $\omega_q(V(x)) < 1 - \frac{1}{m}$ and so no strategy can win with probability greater or equal to $1 - \frac{1}{m}$. Therefore $\exists m.\ \forall n. T(x,m,n) = 0$. ◀

Now we prove the reverse inclusion. Fix an $L \in \Pi^0_2$ and let R be a total Turing machine such that $L = \{x \in \{0,1\}^* : \forall m. \exists n. R(x,m,n) = 1\}$. To prove $L \in \mathsf{MIP}^*_0$, we construct a 2-input 3-prover verifier V that takes as input $m \in \mathbb{N}$ and $x \in \{0,1\}^*$, and has the key property that $\omega^*(V(m,x)) = 1$ if and only if $\forall m' \ge \log^*(m). \exists n. R(x,m',n) = 1$. Therefore $\omega_q(V(1,x)) = 1$ if and only if $x \in L$.

We first give the explicit description of a 3-input 3-prover verifier V' below. We then use that to construct V. In the description of V', we refer to the Turing machine $R_{x,m}$. For every $x \in \{0,1\}^*$ and $m \in \mathbb{N}$, $R_{x,m}$ is the Turing machine that on the empty tape enumerates over $n \in \mathbb{N}$ and accepts if $R(x,m,n) = 1$, otherwise it loops forever.

Now let V be the 2-input 3-prover verifier that on the input (m,x) runs $V'(m,x,V')$. Informally, $V(m,x)$ first decides $\exists n. R(x, \log^*(m), n) = 1$ by simulating the verifier in $V_{\mathrm{HALT},R_{x,\log^*(m)}}$ from Theorem 6. Recall that the existence of the MIP^* protocol $V_{\mathrm{HALT},R_{x,\log^*(m)}}$ is due to $\mathsf{MIP}^* = \mathsf{RE}$ and the fact that $\exists n. R(x,\log^*(m),n) = 1$ is an RE predicate. Now if $R(x,\log^*(m),n) = 0$ for all n, then $R_{x,\log^*(m)}$ never halts. This in turn implies that V rejects with probability at least 1/2. Otherwise, if $\exists n. R(x,\log^*(m),n) = 1$, V proceeds to run the compression algorithm to obtain $V'^{\#} = \mathrm{COMPRESS}_\lambda(V')$. It then executes $V'^{\#}(m,x,V')$. Informally speaking, due to the compression theorem, the execution of $V'^{\#}(m,x,V')$ has the same effect as recursively executing $V(2^m,x)$. Now the first duty of the verifier $V(2^m,x)$ is to decide $\exists n. R(x, 1+\log^*(m), n) = 1$. So we can apply the above reasoning this time on $V(2^m,x)$ instead of $V(m,x)$. Following this reasoning ad infinitum, we establish that $\omega_q(V(m,x)) = 1$ if and only if $\forall m' \ge \log^*(m). \exists n. R(x,m',n) = 1$. This is made precise in the proof of Theorem 9.

Input: (m, x, W) where $m \in \mathbb{N}$, $x \in \{0,1\}^*$, W is a 3-input 3-prover verifier.
Perform the following steps:

1. Compute $V_{\text{HALT},R_{x,\log^*(m)}} = H(R_{x,\log^*(m)})$ (where H is from Theorem 6).
2. Execute the interactive protocol specified by the verifier $V_{\text{HALT},R_{x,\log^*(m)}}$. If the verifier $V_{\text{HALT},R_{x,\log^*(m)}}$ rejects then reject, otherwise continue.
3. Compute $W^{\#} = \text{COMPRESS}_{\lambda}(W)$ (where $\text{COMPRESS}_{\lambda}$ is from Theorem 5).
4. Execute the interactive protocol specified by the verifier $W^{\#}(m, x, W)$ and accept if and only if the verifier $W^{\#}(m, x, W)$ accepts.

Figure 2 Specification of the 3-input 3-prover verifier V'.

Note that Theorem 5 relates $V^{\#}(m, x)$ to $V(2^m, x)$. That is the reason $\log^*(m)$ (as opposed to m) is appearing in Figure 2. Note that as m increases, $\log^*(m)$ ranges over all nonnegative integers.

In order to apply Theorem 5 to compress V in step 3, we must ensure that the verifier is λ-bounded for some $\lambda \in \mathbb{N}$.

▷ Claim 8. There exists a $\lambda \in \mathbb{N}$ such that V is λ-bounded.

Proof. We bound the running time of V by bounding the running time of each of the steps in its specification. The time to generate $R_{x,\log^*(m)}$, in step 1, is $C((|R| \cdot |x| \cdot m))$ for some constant C. The time to generate the encoding of $V_{\text{HALT},R_{x,\log^*(m)}}$ is $C_{\text{HALT}}(|R| \cdot |x| \cdot m)^{C_{\text{HALT}}}$. This also bounds the running time of $V_{\text{HALT},R_{x,\log^*(m)}}$. Therefore the time to simulate $V_{\text{HALT},R_{x,\log^*(m)}}$ is bounded by $C^2_{\text{HALT}}(|R| \cdot |x| \cdot m)^{2C_{\text{HALT}}}$. The time to simulate $\text{COMPRESS}_{\lambda}(V)$ is $C^2_{comp}(|V| \cdot \lambda)^{2C_{comp}}$. The time to simulate $V^{\#}(m, x)$ is bounded by $C^2_{comp}(\lambda \cdot m \cdot |x|)^{2C_{comp}}$. Therefore the running time of $V(m, x)$ is bounded above by

$$2C^2_{\text{HALT}}(|R| \cdot |x| \cdot m)^{2C_{\text{HALT}}} + C(|R| \cdot |x| \cdot m) + C^2_{comp}(|V| \cdot \lambda)^{2C_{comp}} + C^2_{comp}(\lambda \cdot m \cdot |x|)^{2C_{comp}}.$$

The values $C_{comp}, C_{\text{HALT}}, C$, and $|R|$ are all constants so we can choose $\lambda \in \mathbb{N}$ sufficiently large so that $\lambda(m \cdot |x|)^{\lambda}$ is larger then the quantity above and therefore V is λ-bounded. ◁

Now that we established that V is λ-bounded, we can apply Theorem 5 to get the main theorem of this paper.

▶ **Theorem 9.** *$x \in L$ if and only if $\omega_q(V(1, x)) = 1$*

Proof. First suppose $x \in L$. Then $\forall m. \exists n. R(x, m, n) = 1$. Since the Turing machine $R_{x,m}$ halts for every $m \in \mathbb{N}$, by Theorem 6, $\omega_q(V_{\text{HALT},R_{x,m}}) = 1$. Therefore $\omega_q(V(p, x)) = \omega_q(V^{\#}(p, x))$, for any $p \in \mathbb{N}$, by construction (step 4). Now, from Theorem 5, we have

$$\omega_q(V(p, x)) \geq \frac{1}{2} + \frac{\omega_q(V(2^p, x))}{2},$$

and by k applications of the theorem, we obtain

$$\omega_q(V(p, x)) \geq \frac{\omega_q(V(\overbrace{2^{2^{\cdots^{2^p}}}}^{k}, x))}{2^k} + \sum_{i=1}^{k} \frac{1}{2^i}.$$

for every k. Taking the limit $k \to \infty$, we have $\omega_q(V(p, x)) = 1$ for all $p \in \mathbb{N}$. In particular $\omega_q(V(1, x)) = 1$.

Now suppose $x \notin L$. Then $\exists m. \forall n. R(x,m,n) = 0$. We prove that $\omega(V(1,x)) < 1$. Let p be the smallest integer for which $R(x, \log^*(p), n) = 0$ for every n. In other words, the Turing machine $R_{x,\log^*(p)}$ does not halt. Therefore by Theorem 6 we have that $\omega_q(V(p,x)) \leq \omega_q(V_{\text{HALT},R_{x,\log^*(p)}}) \leq \frac{1}{2}$.

If $p = 1$, we are done. Suppose $p > 1$. For all $k < p$, the game $V_{\text{HALT},R_{x,\log^*(k)}}$ never rejects since the Turing machine $R_{x,\log^*(k)}$ halts, by the minimality of p. Therefore $\omega_q(V(k,x)) = \omega_q(V^\#(k,x))$. So by recursively applying Theorem 5, we have that

$$\omega_q(V(p,x)) < 1 \implies \omega_q(V(1,x)) < 1.$$

Since $\omega_q(V(p,x)) \leq \omega_q(V_{\text{HALT},R_{x,\log^*(p)}}) \leq \frac{1}{2}$ then $\omega_q(V(1,x)) < 1$. ◀

▶ **Corollary 10.** $\Pi_2^0 \subseteq \mathsf{MIP}_0^*$.

Proof. Let $L \in \Pi_2^0$ then $L = \{x \in \{0,1\}^* : \forall m. \exists n. R(x,m,n) = 1\}$. Let U be the 1-input 3-prover verifier, that on input x executes the verifier $V(1,x)$ where $x \in \{0,1\}^*$. By Claim 8, $\mathsf{TIME}(U(x)) = \mathsf{TIME}(V(1,x)) \leq \lambda(1+|x|)^\lambda$ and by Theorem 9, $x \in L$ iff $\omega^*(U(x)) = 1$. Thus U is an MIP_0^* protocol for the language L, and $L \in \mathsf{MIP}_0^*$. ◀

This concludes the proof of the main result of this paper.

4 MIP* = RE implies gap-preserving compression

As mentioned in the introduction, the key to proving $\mathsf{MIP}^* = \mathsf{RE}$ in [11] was establishing a gap-preserving compression theorem for non-local games. Here we observe that the reverse holds: $\mathsf{MIP}^* = \mathsf{RE}$ implies a gap-preserving compression theorem.

▶ **Theorem 11.** *If* $\mathsf{MIP}^* = \mathsf{RE}$*, then there exists a Turing machine* COMPRESS*, with the following properties. Given as input a k-input r-prover verifier V,* COMPRESS *outputs a k-input 2-prover verifier $V^\#$ in time polynomial in the description length of V, with the following properties:*

1. *if $\omega_q(V(x_1,...,x_k)) = 1$ then $\omega_q(V^\#(x_1,...,x_k)) = 1$*
2. *if $\omega_q(V(x_1,...,x_k)) \leq \frac{1}{2}$ then $\omega_q(V^\#(x_1,...,x_k)) \leq \frac{1}{2}$*
3. *The runtime of the verifier $V^\#$ is polynomial in the description length of V and its input.*

Proof. COMPRESS is the Turing machine that, when given input a verifier V, it returns the description of the verifier $V^\#$ from Figure 3.

In the description of $V^\#$, we refer to the Turing machine $T_{V,(x_1,...,x_k)}$. For every k-input r-prover verifier V and $x_1, \ldots, x_k \in \{0,1\}^*$, $T_{V,(x_1,...,x_k)}$ is the Turing machine that on empty tape enumerates over finite-dimensional quantum strategies for $V(x_1,...,x_k)$ and only accepts if it finds a strategy that wins the game with probability greater than $\frac{1}{2}$. It does this via enumerating over ε-nets (for $\varepsilon = \frac{1}{4}$) for strategies of dimension d for all $d \in \mathbb{N}$, as with the proof of Theorem 7.

By Theorem 6, if the Turing machine $T_{V,(x_1,...,x_k)}$ halts then

$$\omega_q(V_{\text{HALT},T_{V,(x_1,...,x_k)}}) = 1,$$

otherwise $\omega_q(V_{\text{HALT},T_{V,(x_1,...,x_k)}}) \leq \frac{1}{2}$. Also the runtime of $V_{\text{HALT},T_{V,(x_1,...,x_k)}}$ is $p(|V| + |x_1| + ... + |x_n|)$, for some polynomial p.

Then if $\omega_q(V(x_1,...,x_k)) = 1$ the Turing machine $T_{V,(x_1,...,x_k)}$ finds a strategy that wins with probability greater than $\frac{3}{4}$ and halts. Therefore

$$\omega_q(V^\#(x_1,...,x_k)) = \omega_q(V_{\text{HALT},T_{V,(x_1,...,x_k)}}) = 1.$$

Input: $(x_1, ..., x_k)$, where $x_1, \ldots, x_k \in \{0,1\}^*$
Perform the following steps:

1. Compute $V_{\mathrm{HALT}, T_{V,(x_1,\ldots,x_k)}} = H(T_{V,(x_1,\ldots,x_k)})$ (where H is from Theorem 6).
2. Execute the interactive protocol specified by the verifier $V_{\mathrm{HALT}, T_{V,(x_1,\ldots,x_k)}}$ and accept if and only if the verifier accepts.

Figure 3 Specification of the compressed verifier $V^{\#}$.

Otherwise, if $\omega_q(V(x_1, ..., x_k)) \leq \frac{1}{2}$ then there is no strategy that wins the game with probability $\frac{1}{2}$ and the Turing machine $T_{V,(x_1,\ldots,x_k)}$ never halts. Therefore

$$\omega_q(V^{\#}(x_1, ..., x_k)) = \omega_q(V_{\mathrm{HALT}, T_{V,(x_1,\ldots,x_k)}}) \leq \frac{1}{2}. \qquad \blacktriangleleft$$

Note that in this gap-preserving compression theorem, the time complexity of the verifier $V^{\#}$ is polynomial in the *description length* of V and its input – rather than the *time complexity* of V.

References

1 Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 113–131, 1988.

2 Richard Cleve, Daniel Gottesman, and Hoi-Kwong Lo. How to share a quantum secret. *Physical Review Letters*, 83(3):648, 1999.

3 Matthew Coudron and William Slofstra. Complexity lower bounds for computing the approximately-commuting operator value of non-local games to high precision. *arXiv preprint*, 2019. `arXiv:1905.11635`.

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

5 Joseph Fitzsimons, Zhengfeng Ji, Thomas Vidick, and Henry Yuen. Quantum proof systems for iterated exponential time, and beyond. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 473–480, New York, NY, USA, 2019. Association for Computing Machinery. `doi:10.1145/3313276.3316343`.

6 Tobias Fritz. Tsirelson's problem and kirchberg's conjecture. *Reviews in Mathematical Physics*, 24(05):1250012, 2012.

7 Isaac Goldbring. Enforceable operator algebras. *Journal of the Institute of Mathematics of Jussieu*, pages 1–33, 2017.

8 Isaac Goldbring and Bradd Hart. A computability-theoretic reformulation of the connes embedding problem. *arXiv preprint*, 2013. `arXiv:1308.2638`.

9 Gus Gutoski and John Watrous. Toward a general theory of quantum games. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, page 565–574, New York, NY, USA, 2007. Association for Computing Machinery. `doi:10.1145/1250790.1250873`.

10 Zhengfeng Ji. Compression of quantum multi-prover interactive proofs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 289–302, 2017.

11 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. $\mathsf{MIP}^* = \mathsf{RE}$. *arXiv preprint*, 2020. `arXiv:2001.04383`.

12 Marius Junge, Miguel Navascues, Carlos Palazuelos, David Perez-Garcia, Volkher B Scholz, and Reinhard F Werner. Connes' embedding problem and tsirelson's problem. *Journal of Mathematical Physics*, 52(1):012102, 2011.

13 Anand Natarajan and John Wright. NEEXP $\subseteq$ MIP*. In *IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 510–518, 2019.

14 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.

15 Narutaka Ozawa. About the connes embedding conjecture, algebraic approaches. *Jpn. J. Math.*, 8:147–183, 2013.

16 William Slofstra. The set of quantum correlations is not closed. In *Forum of Mathematics, Pi*, volume 7. Cambridge University Press, 2019.